

Bull

Concepts de gestion de système
Système d'exploitation et unités

AIX



Bull

Concepts de gestion de système Système d'exploitation et unités

AIX

Logiciel

Juin 2003

**BULL CEDOC
357 AVENUE PATTON
BP 20845
49008 ANGERS CEDEX 01
FRANCE**

REFERENCE
86 F2 28EF 03

L'avis juridique de copyright ci-après place le présent document sous la protection des lois de Copyright des États-Unis d'Amérique et des autres pays qui prohibent, sans s'y limiter, des actions comme la copie, la distribution, la modification et la création de produits dérivés à partir du présent document.

Copyright © Bull S.A. 1992, 2003

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin de ce document.

Pour commander d'autres exemplaires de ce manuel ou d'autres publications techniques Bull, veuillez utiliser le bon de commande également fourni en fin de manuel.

Marques déposées

Toutes les marques déposées sont la propriété de leurs titulaires respectifs.

AIX[®] est une marque déposée d'IBM Corp. et est utilisée sous licence.

UNIX est une marque déposée licenciée exclusivement par Open Group Company Ltd.

Linux est une marque déposée de Linus Torvalds.

Les informations contenues dans le présent document peuvent être modifiées sans préavis. Bull S.A. ne pourra être tenu pour responsable des erreurs qu'il peut contenir ni des dommages accessoires ou indirects que son utilisation peut causer.

A propos de ce manuel

Ce manuel fournit les informations permettant à l'administrateur d'assimiler les tâches quotidiennes de gestion. Il présente également les outils d'administration système. Utilisez ce manuel conjointement au document *AIX 5L Version 5.2 - Guide d'administration : système d'exploitation et unités*.

Cette édition prend en charge la version AIX 5L 5.2 avec le module de maintenance recommandé 5200-01. Les références à ce module de maintenance sont signalées par *AIX 5.2 avec 5200-01*.

Utilisateurs concernés

Ce manuel est destiné aux personnes responsables de la gestion du système sur l'ordinateur et du système d'exploitation. Ces personnes sont censées en connaître les commandes de base.

L'administrateur est supposé familiarisé avec les informations et les concepts des documents suivants :

- *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*
- *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*
- *AIX 5L Version 5.2 System User's Guide: Communications and Networks*
- *AIX 5L Version 5.2 Installation Guide and Reference*.

Comment utiliser ce manuel

Ce manuel contient des informations générales et, si nécessaire, des informations conceptuelles relatives aux principaux éléments du système d'exploitation AIX. Seules les tâches de base sont décrites de manière générale. Pour plus d'informations sur les instructions associées aux tâches, reportez-vous au document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

Depuis l'existence de la bibliothèque de documentation AIX 5.2, les informations de ce manuel associées à la sécurité du système AIX ou à la sécurité en générale se trouvent désormais dans un autre document. Toutes les informations associées à la sécurité se trouvent dans le document *AIX 5L Version 5.2 Security Guide*.

Conventions typographiques

Les conventions typographiques adoptées dans ce manuel sont les suivantes :

Gras	Commandes, sous-routines, mots-clés, fichiers, structures, répertoires et autres éléments dont le nom est prédéfini par le système, ainsi que les objets graphiques (tels que boutons, labels, icônes). Identifie également les objets graphiques (tels que boutons, labels, icônes).
<i>Italique</i>	Paramètres dont la valeur ou le nom est fourni par l'utilisateur.
Espacement fixe	Exemples de valeurs spécifiques, de texte affiché, de code programme, messages système, ou données entrées par l'utilisateur.

Prise en compte de la casse dans AIX

Le système d'exploitation AIX tient compte de la casse, ce qui implique qu'il tient compte des majuscules et des minuscules. Vous pouvez, par exemple, utiliser la commande **ls** pour afficher des listes de fichiers. Si vous tapez **LS**, le système envoie un message indiquant que la commande n'existe pas. De même, **FILEA**, **FileA** et **filea** correspondent chacun à des noms de fichiers différents, même s'ils résident dans le même répertoire. Pour éviter d'exécuter des actions indésirables, veillez à toujours utiliser la casse appropriée.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Documentation connexe

- *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*
- *AIX 5L Version 5.2 Security Guide*
- *AIX 5L Version 5.2 Installation Guide and Reference.*
- *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*
- *AIX 5L Version 5.2 Communications Programming Concepts*
- *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*
- *AIX 5L Version 5.2 Files Reference*
- *Performance Toolbox 1.2 and 2.1 for AIX: User's Guide*
- *AIX 5L Version 5.2 Performance Management Guide*
- *Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide.*

Table des matières

A propos de ce manuel	iii
Utilisateurs concernés	iii
Comment utiliser ce manuel	iii
Conventions typographiques	iii
Prise en compte de la casse dans AIX	iv
ISO 9000	iv
Documentation connexe	iv
Chapitre 1. Gestion du système	1-1
Aspects uniques de la gestion de système dans AIX	1-1
Interfaces de gestion de système disponibles	1-1
Fonctions uniques du système d'exploitation	1-2
LVM (Logical Volume Manager)	1-2
Contrôleur de ressources système SRC (System Resource Controller)	1-2
Object Data Manager (ODM)	1-2
Base des données techniques essentielles (SWVPD)	1-3
Workload Manager (WLM)	1-3
Mises à jour du système d'exploitation	1-3
Identification des éléments installés	1-4
Commande man	1-4
Chapitre 2. Démarrage et arrêt du système	2-1
Démarrage du système	2-2
Amorçage du système	2-2
Création d'images d'amorçage	2-2
Identification et modification du niveau d'exécution du système	2-2
Description du processus d'amorçage	2-3
Description du traitement de l'amorçage	2-4
Phase d'initialisation du noyau ROS	2-4
Phase de configuration de l'unité de base	2-5
Phase d'amorçage du système	2-5
Description du processus d'amorçage de maintenance	2-6
Description du système de fichiers RAM	2-7
Description du processus de fermeture	2-8
Détection du blocage du système	2-9
Détection PHD (Priority Hang Detection)	2-9
Détection des blocages par les E/S perdues	2-10
Chapitre 3. Volumes logiques	3-1
Généralités sur le stockage des volumes logiques	3-1
Concepts de stockage des volumes logiques	3-2
Volumes physiques	3-2
Groupes de volumes	3-3
Partitions physiques	3-4
Volumes logiques	3-4
Partitions logiques	3-5
Systèmes de fichiers	3-5

Limitations de la gestion de la mémoire logique	3-5
LVM (Logical Volume Manager)	3-6
Concepts de quorum	3-7
Procédure de mise en service	3-7
Quorum	3-7
Groupes de volumes sans quorum	3-8
Développement d'une stratégie relative aux groupes de volumes	3-9
Création de groupes de volumes distincts	3-9
Haute disponibilité face aux incidents de disque	3-10
Haute disponibilité face aux incidents de carte ou d'alimentation	3-10
Développement d'une stratégie relative aux volumes logiques	3-11
Analyse des besoins associés à la mise en miroir et à l'entrelacement	3-11
Règles de programmation des écritures miroir sur disque	3-12
MWC (cohérence écrit-miroir) pour un volume logique	3-13
Règles d'affectation inter-disque	3-14
Copie unique du volume logique	3-15
Copies multiples du volume logique	3-16
Règles d'affectation intra-disque	3-17
Affectations combinées	3-18
Affectation affinée avec des fichiers mappe	3-18
Développement d'une stratégie relative à la répartition du volume logique	3-19
Règles de contrôle de l'écriture	3-19
Détermination des stratégies de disques de secours remplaçables à chaud	3-19
Gestion des zones actives dans les volumes logiques	3-20
Mise en œuvre de stratégies de groupe de volumes	3-22
Chapitre 4. Espace de pagination et mémoire virtuelle	4-1
Généralités sur VMM (Virtual Memory Manager)	4-2
Gestion de la mémoire réelle	4-2
Liste libre	4-2
Segments de mémoire persistants et segments de mémoire actifs	4-2
Segments actifs et espace de pagination	4-3
Fonction VMM Memory Load Control Facility	4-3
Généralités sur l'espace de pagination	4-4
Stratégies d'allocation d'espace de pagination	4-4
Comparaison du mode d'allocation d'espace de pagination Late et du mode d'allocation d'espace de pagination Early	4-4
Interface de programmation	4-6
Taille par défaut d'espace de pagination	4-7
Fichier d'espace de pagination, commandes et options	4-7
Chapitre 5 Systèmes de fichiers	5-1
Organisation et contenu de l'arborescence de fichiers	5-2
Description du système de fichiers racine	5-2
Description du système de fichiers /usr	5-5
Lien symbolique vers le répertoire /var.	5-6

Liens symbolique vers les répertoires /usr/share et /usr/lib	5-6
Description du système de fichiers /usr/share	5-7
Description du système de fichiers /var	5-8
Description du répertoire /export	5-9
Tâches de gestion des systèmes de fichiers	5-11
Commandes de système de fichier	5-12
Montage : généralités	5-13
Description des points de montage	5-13
Montage des systèmes de fichiers, des répertoires et des fichiers	5-14
Contrôle des montages automatiques	5-14
Description de la sécurité du montage des stations de travail sans disque	5-15
Montages sans disque	5-16
Sécurisation des montages sans disque	5-16
Types de systèmes de fichiers	5-18
JFS (Journaled File System) ou JFS2 (Enhanced Journaled File System)	5-18
NFS (Network File System)	5-18
CDRFS (CD-ROM File System)	5-18
UDFS (DVD-ROM File System)	5-18
Description de JFS et de JFS2	5-18
JFS2	5-19
Description de la segmentation de l'espace disque	5-19
Description du nombre variable d'i-nodes	5-22
Description des limitations de taille	5-23
Description des fichiers supplémentaires	5-25
Description des fichiers volumineux et JFS	5-26
Description de la compression des données JFS	5-27
Description des sauvegardes en ligne JFS et des clichés JFS2	5-29
Description de la compatibilité et de la migration	5-30
Description de CDRFS	5-31
Description de UDFS	5-32
Chapitre 6. Sauvegarde et restauration	6-1
Généralités sur la sauvegarde	6-2
Méthodes de sauvegarde	6-2
Choix d'une politique de sauvegarde	6-3
Description des supports de sauvegarde	6-4
Restauration des données	6-4
Développement d'une stratégie de sauvegarde	6-5
Structure du système de fichiers	6-5
Données système et données utilisateur	6-5
Sauvegarde	6-6
Reproduction d'un système (clonage)	6-7
Sauvegarde des fichiers des utilisateurs ou des Systèmes de fichiers	6-8
Sauvegarde de l'image système et des groupes de volumes définis par l'utilisateur	6-9
Configuration du système source	6-9
Montage et démontage des systèmes de fichiers	6-10
Remarques sur la sécurité	6-10
Restauration d'une image de sauvegarde	6-10

Chapitre 7. Environnement système	7-1
Profils - généralités	7-2
Fichier /etc/profile	7-2
fichier .profile	7-2
Services de manipulation des données sur l'heure	7-4
Désallocation dynamique de processeur	7-5
Impact potentiel sur les applications	7-5
Désallocation de processeur	7-6
Activation et désactivation de la désallocation de processeur	7-6
Relance d'une désallocation de processeur abandonnée	7-6
Considérations relatives à l'état du processeur	7-7
Exemple	7-7
Entrées du journal des erreurs	7-8
 Chapitre 8. Gestion des processus	 8-1
 Chapitre 9. Gestion de la charge de travail	 9-1
Concepts et généralités sur WLM	9-2
Définitions	9-2
Classes	9-4
Affectation de process aux classes	9-5
Contrôle des ressources	9-5
Droits à ressources	9-6
Modes de fonctionnement WLM	9-7
Contrôle dynamique	9-7
Outils de contrôle	9-7
API WLM	9-8
Comptabilité par classe	9-8
Généralités sur les classes WLM	9-9
Superclasses	9-10
Sous-classes	9-11
Attributs de classe	9-11
Attribut de niveau	9-12
Attribut d'héritage	9-12
Attribut localshm	9-13
Attributs Administrator	9-14
Attributs Authorization	9-14
Attribut Resource Set	9-14
Affectation de process à des classes	9-15
Affectation automatique	9-15
Affectation manuelle	9-18
Exemple 1 : Première affectation de process	9-18
Exemple 2 : Réaffectation et annulation de process	9-18
Mise à jour de WLM	9-19
Sécurité	9-19
Gestion des ressources à l'aide de WLM	9-20
Types de ressources	9-20
Partages cibles	9-21
Limites de ressources	9-23

Compatibilité ascendante	9-25
Partages de ressources	9-25
Nouvelle ressource	9-25
Description de la priorité de classe	9-27
Groupes de ressources	9-28
Registre Rset	9-28
Configuration de WLM	9-29
Identification des besoins des applications	9-29
Définition des niveaux, des partages et des limites	9-30
Ajustement de la configuration	9-31
API de Workload Manager	9-32
Etiquette d'application	9-32
API de gestion de classe	9-33
API de gestion de WLM	9-34
API de statistiques de WLM	9-34
API de classification de WLM	9-34
Compatibilité binaire	9-34
Commandes WLM	9-35
Exemple de classification, de règles et de limites WLM	9-35
Exemples d'affectation de règles	9-35
Exemples de partages et de limites	9-36
Exemple avec des limites d'UC	9-37
Exemple avec des limites de mémoire	9-37
Compatibilité amont	9-38
Chapitre 10. SRC et sous-systèmes	10-1
SRC - généralités	10-2
Composants du sous-système	10-2
Groupe de sous-systèmes	10-2
Sous-serveur	10-3
Structure hiérarchique de SRC	10-3
Commandes d'administration SRC	10-3
Chapitre 11. Comptabilité système	11-1
Comptabilité - généralités	11-2
Collecte et rapport de données système	11-2
Collecte de données comptables	11-3
Durée de connexion	11-3
Processus	11-3
Utilisation disque	11-4
Utilisation de l'imprimante	11-4
Taxation	11-4
Rapport de données comptables	11-4
Durée de connexion	11-5
Processus	11-5
Utilisation disque	11-5
Utilisation de l'imprimante	11-5
Taxation	11-5
Rapport quotidien	11-6
Rapport mensuel	11-6
Commandes comptables	11-6
Commandes exécutées automatiquement	11-7
Commandes disponibles à partir du clavier	11-7
Fichiers comptables	11-8
Fichiers de données	11-9

Fichiers de rapport et de synthèse	11-9
Fichiers de commande runacct	11-9
Fichiers du répertoire /var/adm/acct/nite	11-10
Fichiers du répertoire /var/adm/acct/sum	11-10
Fichiers du répertoire /var/adm/acct/fiscal	11-11
Formats des fichiers comptables	11-11
Chapitre 12 Web-based System Manager	12-1
Chapitre 13. SMIT (System Management Interface Tool)	13-1
Chapitre 14. Unités	14-1
Configuration d'un grand nombre d'unités	14-1
Nœuds d'unité	14-2
Classes d'unité	14-3
Base de configuration d'unités	14-3
Etats des unités	14-4
Gestion des unités	14-4
Codes d'emplacement des unités	14-5
Carte	14-5
Imprimante/traceur	14-6
Unité tty	14-6
Unité SCSI	14-7
Codes d'emplacement des disques connectés directement au bus	
Pour une unité de disque connectée directement,	14-7
Disque série	14-7
Unité de disquette	14-8
Rotateur/clavier LPFK	14-8
Port multiprotocole	14-8
Gestion des connexions à chaud PCI	14-9
Unconfiguring PCI Communications Adapters	14-10
Exemple	14-11
E/S à plusieurs chemins	14-11
Configuration d'une unité MPIO	14-12
Unités à plusieurs chemins prises en charge	14-13
Attributs d'unité MPIO	14-13
Attributs de module PCM (Path Control Module)	14-14
Chapitre 15. Unités de bande	15-1
Attributs des unités de bande	15-2
Présentation générale	15-2
Taille de bloc	15-2
Mémoires tampon	15-2
Marques de fichier étendues	15-2
Tension	15-2
Densité	15-2
Réservation	15-3
Taille de bloc de longueur variable	15-3
Compression de données	15-3
Autochargement	15-3
Délai entre deux tentatives	15-3
Délai de lecture/écriture	15-3
Renvoyer erreur sur changement de bande	15-3

Attributs pour unités de bande 4 mm 2 Go (type 4mm2gb)	15-4
Taille de bloc	15-4
Mémoires tampon	15-4
Attributs à valeur fixe	15-4
Attributs pour unités de bande 4 mm 4 Go (type 4mm4gb)	15-4
Taille de bloc	15-4
Mémoires tampon	15-4
Densité	15-4
Compression de données	15-4
Attributs à valeur fixe	15-4
Attributs pour unités de bande 8 mm 2,3 Go (type 8mm)	15-4
Taille de bloc	15-4
Mémoires tampon	15-4
Marques de fichier étendues	15-4
Attributs à valeur fixe	15-5
Attributs pour unités de bande 8 mm 5 Go (type 8mm5gb)	15-5
Taille de bloc	15-5
Mémoires tampon	15-5
Marques de fichier étendues	15-5
Densité	15-5
Compression de données	15-5
Attributs à valeur fixe	15-5
Attributs pour unités de bande 8 mm 20000 Mo (autoconfiguration)	15-5
Taille de bloc	15-5
Mémoires tampon	15-5
Marques de fichier étendues	15-5
Densité	15-6
Compression de données	15-6
Attributs à valeur fixe	15-6
Attributs pour unités de bande 35 Go (type 35gb)	15-6
Taille de bloc	15-6
Mémoires tampon	15-7
Marques de fichier étendues	15-7
Densité	15-7
Compression de données	15-7
Attributs à valeur fixe	15-7
Attributs pour unités de bande 1/4 pouce 150 Mo (type 150mb)	15-7
Taille de bloc	15-7
Mémoires tampon	15-7
Marques de fichier étendues	15-7
Tension	15-8
Densité	15-8
Attributs à valeur fixe	15-8
Attributs pour unités de bande 1/4 pouce 525 Mo (type 525mb)	15-8
Taille de bloc	15-8
Mémoires tampon	15-8
Marques de fichier étendues	15-8
Tension	15-8
Densité	15-8
Attributs à valeur fixe	15-9
Attributs pour unités de bande 1/4 pouce 1200 Mo (type 1200mb-c)	15-9
Taille de bloc	15-9
Mémoires tampon	15-9
Marques de fichier étendues	15-9
Tension	15-9

Densité	15-9
Attributs à valeur fixe	15-9
Attributs pour unités de bande 4 mm 12 000 Mo (autoconfiguration)	15-10
Taille de bloc	15-10
Mémoires tampon	15-10
Marques de fichier étendues	15-10
Densité	15-10
Compression de données	15-11
Attributs à valeur fixe	15-11
Attributs pour unités de bande 1/4 pouce 13 000 Mo (autoconfiguration)	15-11
Taille de bloc	15-11
Mémoires tampon	15-11
Marques de fichier étendues	15-11
Tension	15-11
Densité	15-11
Attributs à valeur fixe	15-11
Attributs pour unités de bande 9 pistes 1/2 pouce (type 9trk)	15-12
Taille de bloc	15-12
Mémoires tampon	15-12
Densité	15-12
Attributs à valeur fixe	15-12
Attributs pour cartouche 1/2 pouce 3490e (type 3490e)	15-12
Taille de bloc	15-12
Mémoires tampon	15-12
Compression	15-12
Autochargement	15-12
Attributs pour autres bandes SCSI (type ost)	15-12
Taille de bloc	15-12
Mémoires tampon	15-13
Marques de fichier étendues	15-13
Densité	15-13
Réservation	15-13
Taille de bloc de longueur variable	15-13
Délai entre deux tentatives	15-13
Délai de lecture/écriture	15-13
Attributs à valeur fixe	15-13
Fichiers spéciaux pour unités de bande	15-14

Annexe A. Comparaisons pour les gestionnaires de systèmes BSD	A-1
Comparaison entre ce système d'exploitation et BSD pour les administrateurs système	A-2
Introduction à ce système d'exploitation pour administrateurs système BSD	A-3
Principales différences entre BSC 4.3 et ce système d'exploitation	A-4
Stockage des données de configuration	A-4
Gestion de la configuration	A-5
Gestion de disque	A-5
Nouvelles commandes	A-5
Amorçage et lancement	A-6
Autorisation utilisateur	A-6
Impression	A-6
Shells	A-6
Comptabilité pour administrateurs système BSD 4.3	A-7
Sauvegarde pour administrateurs système BSD 4.3	A-9
Support de bande SCSI non IBM	A-9
Amorçage et lancement pour administrateurs système BSD 4.3	A-10
Commandes d'administration pour administrateurs système BSD 4.3	A-11
Cron pour administrateurs système BSD 4.3	A-15
Unités pour administrateurs systèmes BSD 4.3	A-16
Tableau de comparaison de fichiers BSD 4.3, SVR4 et de ce système d'exploitation	A-17
Systèmes de fichiers pour administrateurs système BSD 4.3	A-19
Fichiers /etc/filesystems et /etc/fstab	A-19
Support des systèmes de fichiers sur ce système d'exploitation	A-19
Recherche et examen de fichiers pour administrateurs système BSD 4.3	A-20
Espace de pagination pour administrateurs système BSD 4.3	A-21
Réseau pour administrateurs système BSD 4.3	A-22
Configuration BSD 4.3 : modification du lancement par défaut	A-22
Autres options des commandes ifconfig et netstat	A-22
Autres commandes de gestion de réseau	A-22
Résolution de noms et d'adresses	A-23
Différences entre ce système d'exploitation et BSD 4.3	A-23
Commande tn3270	A-23
Documentation en ligne et commande man pour administrateurs système BSD 4.3	A-24
NFS et NIS (ex"Yellow Pages") pour administrateurs système BSD 4.3	A-25
Mots de passe pour administrateurs système BSD 4.3	A-26
Définition d'un mot de passe utilisateur	A-26
Importation d'un fichier de mots de passe BSD 4.3	A-26
Edition du fichier de mots de passe (Password)	A-26
Mesure des performances et réglage pour les gestionnaires systèmes BSD 4.3 ..	A-29
Imprimantes pour les gestionnaires de systèmes BSD 4.3 Dans AIX 5.1 et les versions supérieures,	A-30
Terminaux pour administrateurs système BSD 4.3	A-32
termcap et terminfo	A-32
UUCP pour administrateurs système BSD 4.3	A-33
Index	X-1

Chapitre 1. Gestion du système

La personne chargée de la gestion du système est l'administrateur système (ainsi appelé dans la littérature UNIX). Malheureusement, seule une faible part des activités de l'administrateur système sont suffisamment simples pour être définies comme étant des tâches d'administration. Ce manuel et la documentation connexe ont pour objectif de les assister dans leurs nombreuses fonctions.

Les sujets traités dans ce chapitre sont les suivants :

- Aspects uniques de la gestion de système dans AIX
- Caractéristiques uniques du système d'exploitation
- Commande man

Aspects uniques de la gestion de système dans AIX

Ce système d'exploitation fournit sa propre version du support de la gestion de système pour faciliter l'utilisation et améliorer la sécurité et l'intégrité. Ce chapitre présente ces caractéristiques uniques :

- Interfaces de gestion de système disponibles, page 1-1
- Fonctions uniques du système d'exploitation, page 1-2
- Commande man, page 1-4

Interfaces de gestion de système disponibles

Ce système d'exploitation fournit en option, en plus de l'administration standard de type ligne de commande, les interfaces suivantes :

- SMIT (System Management Interface Tool), interface utilisateur permettant de créer et d'exécuter des commandes à partir d'options, et d'assurer :

L'interface SMIT vous permet d'assurer :

- l'installation, la mise à jour et la maintenance du logiciel,
- la configuration des unités,
- la configuration des unités de disque pour le stockage en groupes de volumes et en volumes logiques,
- la création et l'extension de systèmes de fichiers et d'espaces de pagination,
- la gestion des utilisateurs et des groupes,
- la configuration des réseaux et des applications de communication,
- l'impression,
- l'identification des incidents,
- l'organisation des travaux,
- la gestion des ressources système et de la charge de travail
- Gestion des environnements de système
- Gestion des données de système en cluster. Voir SMIT (System Management Interface Tool) pour plus d'informations sur la gestion du système avec l'outil SMIT.

- Web-based System Manager, interface utilisateur graphique orientée objet prenant en charge les mêmes tâches de gestion du système que SMIT, mais facilitant les tâches de gestion système grâce :
 - à la réduction des erreurs utilisateur à l'aide du contrôle d'erreurs et de la conception des boîtes de dialogue
 - à des procédures pas à pas permettant d'exécuter des tâches nouvelles ou compliquées
 - à des options avancées pour les administrateurs expérimentés
 - à la visualisation facilitée de données ou de relations complexes entre objets système
 - au contrôle de l'activité système et à l'alerte de l'administrateur lorsque des événements prédéfinis se produisent
 - à la présence d'aide contextuelle, de généralités, de conseils et de liens vers de la documentation en ligne

Web-based System Manager fonctionne sous AIX desktop ou par le biais d'une connexion distante sécurisée vers tout client disposant d'un navigateur prenant en charge Java. Une seule console Web-based System Manager peut gérer plusieurs machines AIX.

Fonctions uniques du système d'exploitation

Ces fonctions sont présentées ci-après de façon succincte.

LVM (Logical Volume Manager)

LVM (Logical Volume Manager) maintient la hiérarchie des structures logiques qui gère le stockage sur disque. Dans cette hiérarchie, les unités de disque sont définies comme des volumes physiques. Chaque volume physique utilisé appartient à un groupe de volumes. Chaque groupe de volumes contient au moins un volume logique. Les données des volumes logiques semblent contiguës pour l'utilisateur, mais peuvent ne pas être contiguës dans le volume physique. Cela permet aux systèmes de fichiers, à l'espace de pagination et aux autres volumes logiques d'être redimensionnés et réalloués, de couvrir plusieurs volumes physiques et de répliquer leur contenu pour améliorer la flexibilité et la disponibilité.

Pour plus d'informations, reportez-vous à la section Généralités sur le stockage des volumes logiques.

Contrôleur de ressources système SRC (System Resource Controller)

Le contrôleur SRC fournit un ensemble de commandes et de sous-routines pour créer et contrôler les sous-systèmes et permet de réduire l'intervention humaine dans le traitement de système. Il fournit un mécanisme de contrôle des processus des sous-systèmes en utilisant une ligne de commande ou une interface C. Vous pouvez ainsi démarrer, arrêter ces processus et collecter des informations d'état sur ces derniers à l'aide de scripts shell, de commandes et de programmes personnalisés.

Object Data Manager (ODM)

ODM est un gestionnaire de données dédié au stockage des données du système. Nombre de fonctions de gestion système font appel à ODM. Les données servant à nombre de fonctions SMIT et de commandes sont stockées et actualisées comme des objets, avec leurs caractéristiques associées. Les données système gérées par ODM comprennent :

- des informations sur la configuration des unités ;
- des menus, des sélecteurs et des dialogues d'écrans SMIT ;

- des VPD (données techniques essentielles) pour les procédures d'installation et de mise à jour ;
- des informations relatives à la configuration des communications ;
- des informations relatives aux ressources système.

Base des données techniques essentielles (SWVPD)

Certaines informations concernant les logiciels et leurs options installables sont actualisées dans la base de données SWVPD (données techniques essentielles du logiciel). Cette base est constituée d'un ensemble de commandes et de classes d'objets ODM, dédiées à la maintenance des données relatives au logiciel. Par le biais de ces commandes, l'utilisateur peut formuler des requêtes (commande **lsipp**) et vérifier (commande **lppchk**) les produits logiciels installés. Les classes d'objets ODM définissent la portée et le format des données actualisées.

La commande **installp** fait appel à ODM pour actualiser dans la base de données SWVPD :

- le nom du logiciel installé ;
- sa version ;
- son niveau d'édition, indiquant les modifications de son interface de programmation externe ;
- son niveau de modification, indiquant les modifications n'affectant pas l'interface de programmation externe ;
- son niveau de correction, indiquant les mises à jour mineures qui feront ultérieurement l'objet d'un nouveau niveau de modification ;
- l'identification de la correction ;
- les noms, les totaux de contrôle, et la taille des fichiers constituant le logiciel ou l'option ;
- Etat d'installation du produit logiciel : application, appliqué, validation, validé, rejet ou brisé.

Workload Manager (WLM)

Workload Manager (WLM) vous permet de créer différentes classes de services pour travaux et d'en spécifier les attributs. Ces attributs spécifient les quantités minimale et maximale de CPU, de mémoire physique et de débit d'E/S disque allouées à une classe. WLM assigne alors automatiquement des travaux à des classes à l'aide des règles d'affectation aux classes fournies par un administrateur système. Ces règles d'affectation sont basées sur les valeurs d'un ensemble d'attributs pour un processus. L'administrateur système ou un utilisateur privilégié peut également attribuer manuellement des travaux à des classes, remplaçant l'affectation automatique. Pour plus d'informations, reportez-vous à Gestion de la charge de travail.

Mises à jour du système d'exploitation

Le système d'exploitation est divisé en ensembles de fichiers qui contiennent chacun un groupe de fichiers client associés logiquement. Chaque ensemble de fichiers peut être installé et mis à jour individuellement.

Les révisions des ensembles de fichiers peuvent être suivies à l'aide des niveaux VRMF (Version, Release, Maintenance and Fix). Par convention, chaque fois qu'une mise à jour d'ensemble de fichiers est appliquée, le niveau du correctif est ajusté. Chaque fois qu'un niveau de maintenance AIX est appliqué, le niveau de modification est ajusté et le niveau du correctif est remis à zéro. L'installation initiale d'une version AIX, par exemple, AIX 5.2, s'appelle une installation de base. Le système d'exploitation fournit les mises à jour de ses fonctions et fonctionnalités qui peuvent être disponibles sous la forme d'un niveau de maintenance, d'un PTF (Program Temporary Fix) ou d'un module de maintenance recommandé.

Niveaux de maintenance

Un niveau de maintenance fournit une nouvelle fonctionnalité destinée à mettre à jour la version. La partie maintenance du niveau VRMF est mise à jour dans un niveau de maintenance. Le premier niveau de maintenance pour AIX 5.2, par exemple, est 5.2.1.0; le deuxième 5.2.2.0, et ainsi de suite.

PTF

Entre les versions, vous pouvez recevoir des PTF qui corrigent ou éliminent un problème particulier. Une installation peut nécessiter tous, certains ou aucun des PTF disponibles.

Modules de maintenance recommandés

Un module de maintenance recommandé est un groupe de PTF entre les niveaux de maintenance, qui ont été soigneusement testés ensemble et qui sont recommandés pour la maintenance préventive.

Identification des éléments installés

Pour identifier le niveau de maintenance installé sur un système, tapez :

```
oslevel
```

Pour identifier les ensembles de fichiers à mettre à jour pour que le système corresponde à un niveau de maintenance donné (4.3.3.0, en l'occurrence), utilisez la commande suivante :

```
oslevel -l 4.3.3.0
```

Pour déterminer si un module de maintenance recommandé est installé (5100-02, en l'occurrence), utilisez la commande suivante :

```
oslevel -r 5100-02
```

Pour identifier les ensembles de fichiers à mettre à jour pour que le système corresponde au niveau 5100-02, utilisez la commande suivante :

```
oslevel -rl 5100-02
```

Pour identifier le niveau d'un ensemble de fichiers (**bos.mp**, en l'occurrence) utilisez la commande suivante :

```
lslpp -L bos.mp
```

Commande man

La commande **man** permet principalement d'accéder aux informations de référence sur les commandes, les sous-routines et les fichiers. Pour afficher les informations relatives à la commande **ls**, par exemple, entrez :

```
>man ls
```

La majorité des informations affichées sont issues principalement de fichiers HTML formatés. La plupart des gestionnaires de systèmes considèrent qu'il est plus pratique d'utiliser la commande **man** que de démarrer un navigateur Web pour rechercher simplement des informations sur une option ou la syntaxe d'une commande.

Pour plus d'informations sur la commande **man**, reportez-vous à *AIX 5L Version 5.2 Commands Reference*. Consultez également la documentation en ligne et la section Commande man pour les gestionnaires de système BSD 4.3, page A-24.

Chapitre 2. Démarrage et arrêt du système

Ce chapitre est consacré aux activités de démarrage du système, notamment l'amorçage, la création d'images ou de fichiers d'amorçage et la définition du niveau d'exécution du système. L'arrêt du système, au moyen des commandes **reboot** et **shutdown**, est également traité.

Les sujets abordés sont les suivants :

- Démarrage du système
- Description du processus d'amorçage
- Description du traitement de l'amorçage du système
- Description du processus d'amorçage de maintenance
- Description du système de fichiers RAM
- Description du processus d'arrêt
- Détection du blocage du système

Démarrage du système

A l'amorçage du système d'exploitation de base, le système lance un ensemble complexe de tâches. En conditions normales d'exploitation, ces tâches s'exécutent automatiquement.

Amorçage du système

Vous demandez l'amorçage du système dans certaines situations : par exemple, pour entériner l'installation d'un nouveau logiciel, pour réinitialiser les unités périphériques, pour exécuter des routines de maintenance (telles que la vérification des systèmes de fichiers), ou pour relancer le système après une panne ou une interruption. Vous trouverez des informations sur ces procédures aux sections :

- Amorçage d'un système non installé.
- Réamorçage d'un système en cours d'exploitation.
- Amorçage à l'issue d'une panne système.

Création d'images d'amorçage

Lors de la première installation, la commande **bosboot** crée une image d'amorçage à partir d'une image du système de fichiers disque en RAM et du noyau du système d'exploitation. L'image d'amorçage est transférée sur un support donné, un disque dur par exemple. Au réamorçage, l'image d'amorçage est chargée en mémoire à partir de son support.

Pour plus d'informations, reportez-vous à la section "Creating Boot Images" du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Identification et modification du niveau d'exécution du système

Le niveau d'exécution spécifie l'état du système et définit les processus à démarrer. Par exemple, avec un niveau 3, tous les processus définis pour opérer à ce niveau sont lancés. Juste avant la fin de phase d'amorçage système du processus d'amorçage, le niveau d'exécution est lu dans l'entrée `initdefault` du fichier **/etc/inittab**. Il peut être modifié par le biais de la commande **init**. Le fichier **/etc/inittab** contient un article par processus, qui définit les niveaux d'exécution. A l'amorçage du système, la commande **init** lit le fichier **/etc/inittab** pour définir les processus à démarrer. Vous trouverez des informations sur ces procédures aux sections :

- Identification des niveaux d'exécution.
- Modification du niveau d'exécution.
- Modification du fichier `/etc/inittab`.

Description du processus d'amorçage

Lors du processus d'amorçage, le système teste le matériel, charge puis exécute le système d'exploitation et configure les unités. L'amorçage du système d'exploitation nécessite les ressources suivantes :

- une *image d'amorçage* chargeable après la mise sous tension ou la réinitialisation de la machine,
- l'accès aux systèmes de fichiers racine et **/usr**.

Il existe trois types d'amorçage :

Amorçage à partir du disque dur	Une machine est démarrée pour des opérations normales. Pour plus d'informations, reportez-vous à la section Description du traitement de l'amorçage du système.
Amorçage réseau sans disque	Une station de travail sans disque et sans données est démarrée à distance sur le réseau. Une machine est démarrée pour des opérations normales. Un ou plusieurs serveurs de fichiers fournissent les fichiers et les programmes dont ont besoin les stations de travail sans disque et sans données pour s'amorcer.
Amorçage de maintenance	Une machine est démarrée à partir d'un disque dur, d'un réseau, d'une bande ou d'un CD-ROM en mode Maintenance. Un administrateur système peut exécuter des tâches telles qu'installer un nouveau logiciel ou une mise à jour d'un logiciel et exécuter des programmes de diagnostics. Pour plus d'informations, reportez-vous à la section Description du processus d'amorçage de maintenance.

Pendant l'amorçage à partir d'un disque, le système recherche l'image d'amorçage sur un disque local défini à l'installation du système d'exploitation. Au cours du processus, le système configure toutes les unités trouvées dans la machine et initialise le logiciel de base nécessaire pour l'exploitation du système, tel que LVM (Logical Volume Manager). En fin de processus, les systèmes de fichiers sont montés et prêts à l'emploi. Pour plus de détails sur le système de fichiers utilisé lors de l'amorçage, reportez-vous à "Description du système de fichiers RAM".

Ce traitement s'applique également, dans les grandes lignes, aux clients de réseau sans disque. Ils requièrent également une image d'amorçage et l'accès à l'arborescence du fichier système d'exploitation. Ce type de client sans disque ne possédant pas de système de fichiers local y accède à distance.

Description du traitement de l'amorçage

Lors du démarrage du système pour une exploitation dans des conditions normales, l'amorçage s'effectue généralement à partir du disque. Le système trouve sur le disque toutes les données nécessaires au processus d'amorçage.

Lorsque le système est démarré par une mise sous tension ("à froid") ou redémarré par la commande **reboot** ou **shutdown** ("à chaud"), un certain nombre d'événements ont lieu avant qu'il ne soit prêt à l'emploi. Ces événements sont répartis en trois phases :

1. Initialisation du noyau ROS (Read Only Storage),
2. Configuration des unités de base,
3. Phase d'amorçage de maintenance

Phase d'initialisation du noyau ROS

Cette phase comprend les étapes suivantes :

1. Le microcode détermine si la carte-mère du système fonctionne correctement. Le contrôle est transmis à ROS qui exécute un autotest à la mise sous tension.
2. L'IPL (Initial Program Load = chargement initial du programme) ROS vérifie la liste d'amorçage utilisateur, qui répertorie les unités d'amorçage disponibles. Vous pouvez adapter cette liste à vos besoins au moyen de la commande **bootlist**. Si la liste d'amorçage utilisateur dans la mémoire vive non volatile (NVRAM) n'est pas correcte ou si aucune unité d'amorçage correcte n'est trouvée, la liste d'amorçage par défaut est vérifiée. Dans les deux cas, la première unité d'amorçage valide rencontrée est utilisée pour le démarrage du système. Les unités sont vérifiées dans l'ordre de la liste (à condition qu'elle soit valide) située, le cas échéant, au niveau de la NVRAM. A défaut de liste, toutes les cartes et unités du bus sont vérifiées. Dans les deux cas, les unités sont vérifiées en boucle (continue) jusqu'à trouver une unité d'amorçage valide.

Remarque : Le système se charge de la maintenance d'une liste d'amorçage par défaut, située en ROS, et d'une liste d'amorçage utilisateur, stockée en NVRAM, pour l'amorçage normal. Font aussi l'objet de maintenance deux listes d'amorçage distinctes (une liste utilisateur et une liste par défaut), pour l'amorçage avec le sélecteur sur maintenance.

3. Lorsqu'une unité d'amorçage valide est trouvée, c'est le premier article d'amorçage ou le premier PSN (numéro de secteur de programme) qui est vérifié ; si cet article est valide, il est lu en mémoire et ajouté au bloc de contrôle de l'IPL en mémoire. Les principales données de l'article d'amorçage comprennent la position de démarrage de l'image d'amorçage sur l'unité d'amorçage, la taille de cette image et les instructions relatives à l'emplacement mémoire sur laquelle charger l'image d'amorçage.
4. L'image d'amorçage est lue de manière séquentielle depuis l'unité d'amorçage dans la mémoire à partir de l'emplacement défini dans la mémoire NVRAM. L'image d'amorçage sur disque est constituée du noyau, d'un système de fichiers RAM et d'informations d'unités personnalisées.
5. Le contrôle est transmis au noyau qui initialise le système.
6. Le noyau exécute **init** qui exécute la phase 1 du script **rc.boot**.

A l'issue de cette phase, la configuration de l'unité de base est lancée.

Phase de configuration de l'unité de base

Le processus **init** lance le script **rc.boot**. La phase 1 de ce script se charge de la configuration de l'unité de base. Elle comprend les différentes étapes suivantes :

1. Le script d'amorçage appelle le programme **restbase** pour créer la base de données ODM (Object Data Manager) dans un système de fichiers RAM à partir des données personnalisées compressées.
2. Le script d'amorçage lance le gestionnaire de configuration, qui accède à la phase 1 des règles de configuration ODM pour configurer les unités de base.
3. Le gestionnaire de configuration démarre le **système, le bus, le disque**, SCSI et LVM (Logical Volume Manager) et les méthodes de configuration du groupe de volumes rootvg.
4. Les méthodes de configuration chargent les gestionnaires d'unités, créent les fichiers spéciaux et mettent à jour les données personnalisées dans la base de données ODM.

Phase d'amorçage du système

Les étapes de la phase d'amorçage du système étaient les suivantes :

1. Le processus **init** lance la phase 2 du script **rc.boot** comprenant les étapes suivantes :
 - a. Appelez le programme **ipl_varyon** pour mettre en service le groupe de volumes rootvg.
 - a. Monter les systèmes de fichiers de disque dur sur leurs points de montage normaux.
 - b. Exécution de **swapon** pour lancer la pagination.
 - c. Copie des données personnalisées de la base ODM du système de fichiers RAM dans la base ODM du système de fichiers disque.
 - d. Sortie du script **rc.boot**.
2. Le processus d'amorçage passe ensuite du système de fichiers RAM au système de fichiers disque.
3. Le processus **init** exécute les processus définis dans les articles du fichier **/etc/inittab**. Selon l'une des instructions de ce fichier, la phase 3 du script **rc.boot** est exécutée ; elle comprend les étapes suivantes :
 - a. Montage du système de fichiers disque **/tmp**
 - b. Lancement du gestionnaire de configuration – phase 2 pour configurer les unités restantes
 - c. Exécution de la commande **savebase** pour sauvegarder les données personnalisées sur le volume logique d'amorçage
 - d. Sortie du script **rc.boot**.

En fin de processus, le système est monté et prêt à l'emploi.

Description du processus d'amorçage de maintenance

Il peut arriver qu'il soit nécessaire de réamorcer le système pour effectuer des tâches spécifiques telles qu'installer un nouveau logiciel ou une mise à jour de logiciel, effectuer des diagnostics ou exécuter des opérations de maintenance. Dans ce cas, le système démarre depuis un support amorçable tel qu'un CD-ROM, une unité de bande, le réseau ou une unité de disque.

La séquence des événements d'amorçage en mode maintenance est similaire à la séquence d'amorçage normale.

1. Le microcode détermine si la carte-mère du système fonctionne correctement.
2. Le contrôle est transmis à ROS qui exécute un autotest à la mise sous tension.
3. ROS vérifie la liste d'amorçage personnalisée. Vous pouvez utiliser la commande **bootlist** pour modifier cette liste en fonction des besoins. Si la liste en mémoire NVRAM n'est pas valide ou que l'unité d'amorçage est introuvable, la liste d'amorçage par défaut est utilisée. Dans les deux cas, la première unité d'amorçage valide détectée dans la liste est utilisée pour démarrer le système.

Remarque : Pour un amorçage normal, le système gère une liste d'amorçage par défaut dans ROS et une liste d'amorçage en mémoire NVRAM. Des listes d'amorçage par défaut et personnalisées sont également gérées pour l'amorçage en mode de maintenance.

4. Lorsqu'une unité d'amorçage valide est détectée, le premier numéro d'enregistrement ou PSN (Program Sector Number) est vérifié. S'il s'agit d'un enregistrement d'amorçage valide, il est lu en mémoire et ajouté au bloc de contrôle IPL (Initial Program Load) en mémoire. Les principales données d'enregistrement d'amorçage contiennent l'emplacement de départ de l'image d'amorçage sur l'unité d'amorçage, la longueur de l'image d'amorçage et le décalage par rapport au point d'entrée du démarrage lorsque l'image d'amorçage est en mémoire.
5. L'image d'amorçage est lue de manière séquentielle depuis l'unité d'amorçage dans la mémoire à partir de l'emplacement défini dans la mémoire NVRAM.
6. Le contrôle est transmis au noyau qui exécute les programmes dans le système de fichiers RAM.
7. Le contenu de la base de données ODM détermine les unités présentes et la commande **cfgmgr** configure dynamiquement toutes les unités détectées, y compris tous les disques qui figurent dans le système de fichiers racine.
8. Si un CD-ROM, une bande ou le réseau est utilisé pour démarrer le système, le groupe de volumes rootvg n'est pas activé, car le groupe de volumes rootvg peut ne pas exister (comme c'est le cas lors de l'installation du système d'exploitation sur un nouveau système). La configuration réseau peut avoir lieu à ce stade. Aucune pagination n'a lieu lors du démarrage en mode maintenance.

A la fin de cette procédure, le système est prêt à être installé et pour exécuter des opérations de maintenance et effectuer des diagnostics.

Remarque : Si le système est démarré depuis le disque dur, rootvg est activé, le fichier racine du disque dur et le système de fichiers utilisateur du disque dur sont montés dans le système de fichiers RAM et un menu s'affiche pour vous permettre d'utiliser divers modes de diagnostics ou un mode mono-utilisateur. L'utilisation du mode mono-utilisateur permet de poursuivre la procédure d'amorçage et d'entrer dans ce mode où le niveau d'exécution **init** est "S". Le système est ensuite prêt pour la maintenance, les mises à jour logicielles ou l'exécution de la commande **bosboot**.

Description du système de fichiers RAM

Le système de fichiers RAM, qui fait partie de l'image d'amorçage, réside en mémoire et contient tous les programmes assurant la poursuite du processus d'amorçage. Les fichiers de ce système déterminent le type d'amorçage.

Il se peut qu'un système de fichiers RAM d'amorçage ne contienne pas les routines de volume logique, car il peut ne pas être nécessaire de mettre en service le groupe de volumes rootvg. Toutefois, lors d'un amorçage depuis le disque dur, il est préférable d'activer le plus rapidement possible le groupe de volumes rootvg et la pagination. Bien que les deux scénarios d'amorçage soient différents, la structure du système de fichiers reste pratiquement identique.

La commande **init** du système de fichiers RAM utilisée lors de l'amorçage est en fait le programme **ssh** (simple shell). Ce programme contrôle le processus d'amorçage en appelant le script **rc.boot** dont la première étape consiste à déterminer à partir de quelle unité la machine a été amorcée. L'unité d'amorçage détermine les unités à configurer dans le système de fichiers RAM. Si la machine est amorcée à partir du réseau, les unités du réseau doivent être configurées pour permettre le montage à distance des systèmes de fichiers client. En cas d'amorçage à partir d'un bande ou d'un CD-ROM, la console affiche les menus d'installation de BOS. Quand **rc.boot** a identifié l'unité d'amorçage, les routines de configuration correspondantes sont appelées dans le système de fichiers RAM. **ssh** appelle **rc.boot** à deux reprises, c'est-à-dire pour les deux phases de configuration de l'amorçage. Un troisième appel de **rc.boot** se produit au moment de l'appel de la commande **init** pendant un amorçage disque ou réseau. Une strophe de **rc.boot** figurant dans le fichier **inittab** se charge de la configuration finale de la machine.

Le système de fichier RAM de chaque unité d'amorçage est également unique du fait des divers types d'unités à configurer. Un fichier prototype est associé à chaque type d'unité d'amorçage. Le fichier prototype est un modèle de fichiers qui constituent le système de fichiers RAM. La commande **mkfs** est utilisée par la commande **bosboot** pour créer le système de fichiers RAM au moyen de différents fichiers de prototype. Pour plus de détails, reportez-vous à la commande **bosboot**.

Description du processus de fermeture

Vous avez la possibilité de fermer le système sous contrôle dans les situations suivantes :

- après l'installation d'un nouveau logiciel ou la modification de la configuration logicielle,
- lors d'un incident matériel,
- en cas d'interruption anormale persistante,
- en cas de performances décroissantes,
- si un système de fichiers est vraisemblablement endommagé.

Détection du blocage du système

AIX peut détecter les blocages et tenter de résoudre ces problèmes en fonction d'actions définies par l'utilisateur. Actuellement, AIX prend en charge deux types de détection de blocage.

- Détection PHD (Priority Hang Detection)
- Détection des blocages par les E/S perdues

Ces types de blocages sont décrits dans les sections suivantes. Pour plus d'informations sur la détection des blocages du système, reportez-vous à la section System Hang Management du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Détection PHD (Priority Hang Detection)

Tous les processus (également appelés routines) ont une priorité d'exécution. Cette priorité est comprise entre 0 et 126, par ordre croissant. Zéro est la priorité la plus élevée et 126 la plus basse. La priorité par défaut pour toutes les routines est 60. Tout utilisateur peut baisser la priorité d'un processus à l'aide de la commande **nice**. Toute personne ayant un privilège d'utilisateur racine peut également augmenter la priorité d'un processus.

L'ordonnanceur du noyau envoie toujours au CPU la routine exécutable ayant la priorité la plus élevée. Il est donc possible, en présence d'un nombre suffisant de routines à haute priorité, de bloquer complètement la machine, de sorte que les routines à faibles priorités ne s'exécutent jamais. Si les routines en cours d'exécution ont une priorité plus élevée que la valeur par défaut, qui est de 60, les connexions et les shells normaux peuvent s'en trouver verrouillés à un point tel que le système semble bloqué.

La fonction (System Hang Detection) offre un mécanisme permettant de détecter cette situation et offrant à l'administrateur système un moyen de la corriger. Cette fonction est mise en oeuvre sous forme de démon (**shdaemon**) s'exécutant avec la priorité la plus élevée. Ce démon interroge le noyau pour connaître la routine à plus faible priorité exécutée sur un intervalle spécifié. Si sa priorité dépasse un seuil configuré, le démon peut exécuter différentes actions. Chacune de ces actions peut être activée indépendamment et configurée pour se déclencher en fonction d'une priorité et d'un intervalle de temps quelconque. Les actions et leurs valeurs par défaut sont les suivantes :

Action	Activée par défaut	Priorité par défaut	Tempor. p. défaut	Périph. par défaut
1) Consigner erreur	non	60	2	
2) Message console	non	60	2	/dev/console
3) Shell de connexion à haute priorité	oui	60	2	/dev/tty0
4) Exécution d'une commande à haute priorité	non	60	2	
5) Panne et redémarrage	non	39	5	

Pour plus d'informations sur la détection de blocage du système, reportez-vous à System Hang Management dans *System Management Guide: Operating System and Devices*.

Détection des blocages par les E/S perdues

Du fait de l'existence d'erreurs d'E/S, le chemin E/S peut se bloquer et les autres E/S du chemin sont affectées. Dans ce cas, il est important que le système d'exploitation puisse avertir l'utilisateur et exécuter les actions définies par ce dernier. Dans le cadre de la détection des E/S perdues et de la notification, le processus **shdaemon**, avec LVM, contrôle les tampons d'E/S pendant un certain délai et détermine si des E/S sont en attente depuis un certain temps. Si le délai d'attente est supérieur au seuil d'attente défini par le fichier **shconf**, une perte d'E/S est détectée et d'autres actions sont exécutées. Les informations relatives à l'E/S perdue sont consignées dans le journal des erreurs. En outre, en fonction des paramètres définis dans le fichier **shconf**, le système peut être redémarré pour résoudre le problème de perte d'E/S.

Pour la détection des pertes d'E/S, vous pouvez définir le délai et activer les actions suivantes :

Action	Activée par défaut	Unité par défaut
Message de console	Non	/dev/console
Blocage et réamorçage	Non	–

Chapitre 3. Volumes logiques

Ce chapitre décrit les concepts de gestion du stockage des volumes logiques et porte sur les sujets suivants :

- Généralités sur le stockage des volumes logiques
- Développement d'une stratégie de groupe de volumes
- Développement d'une stratégie de volume logique
- Mise en œuvre de stratégies de groupe de volumes

Généralités sur le stockage des volumes logiques

Une hiérarchie de structures est utilisée pour gérer la mémoire physique. Chaque unité de disque, appelée *volume physique* (PV), porte un nom tel que `/dev/hdisk0`. Chaque volume physique utilisé appartient à un *groupe de volumes* (VG). Tous les volumes physiques d'un groupe de volumes sont divisés en *partitions physiques* (PP) de même taille. Pour les besoins d'allocation d'espace, chaque volume physique est divisé en cinq régions (`outer_edge`, `inner_edge`, `outer_middle`, `inner_middle` et `center`). Le nombre de partitions physiques dans chaque région varie en fonction de la capacité totale de l'unité de disque. Si le groupe de volumes est créé avec l'option **-B** dans la commande **mkvg**, les limites augmentent jusqu'à 128 volumes physiques et 511 volumes logiques.

Chaque groupe de volumes contient au moins un *volume logique* (LV). Les volumes logiques sont des groupes d'informations situés dans les volumes physiques. Les données des volumes logiques semblent contiguës pour l'utilisateur, mais peuvent ne pas être contiguës dans le volume physique. Cela permet aux systèmes de fichiers, à l'espace de pagination et aux autres volumes logiques d'être redimensionnés et réalloués, de couvrir plusieurs volumes physiques et de répliquer leur contenu pour améliorer la flexibilité et la disponibilité du stockage des données.

Chaque volume logique est constitué d'au moins une *partition logique* (LP). Chaque partition logique correspond à au moins une partition physique. Si la mise en miroir est définie pour le volume logique, d'autres partitions physiques sont allouées pour stocker les copies supplémentaires de chaque partition logique. Bien que les partitions logiques soient numérotées chronologiquement, les partitions physiques sous-jacentes ne sont pas nécessairement consécutives, ni contiguës.

Les volumes logiques peuvent entrer dans divers processus système, tels que la pagination, mais chaque volume logique n'a qu'une seule fonction. Un grand nombre de volumes logiques contient un seul système de fichiers (JFS ou JFS2). Chaque système de fichiers JFS est constitué d'un groupe de blocs de page (4 Ko). Lorsque des données doivent être écrites dans un fichier, un ou plusieurs blocs supplémentaires sont alloués au fichier. Ces blocs peuvent ne pas être contigus les uns par rapport aux autres ou à d'autres blocs déjà alloués au fichier. Un système de fichiers peut être défini avec une taille de fragment inférieur à 4 Ko (512 octets, 1 Ko, 2 Ko).

Après l'installation, le système dispose d'un groupe de volumes (`rootvg`) constitué d'un ensemble de volumes logiques de base, nécessaires au démarrage du système, et d'autres volumes logiques que vous définissez dans le script d'installation. Tous les autres volumes physiques que vous connectez au système peuvent être ajoutés à un groupe de volumes (à l'aide de la commande **extendvg**). Vous pouvez ajouter le volume physique au groupe de volumes `rootvg` ou à un autre groupe de volumes (définir par la commande **mkvg**). Les volumes logiques peuvent être personnalisés à l'aide de commandes, de SMIT (System Management Interface Tool) ou de Web-based System Manager.

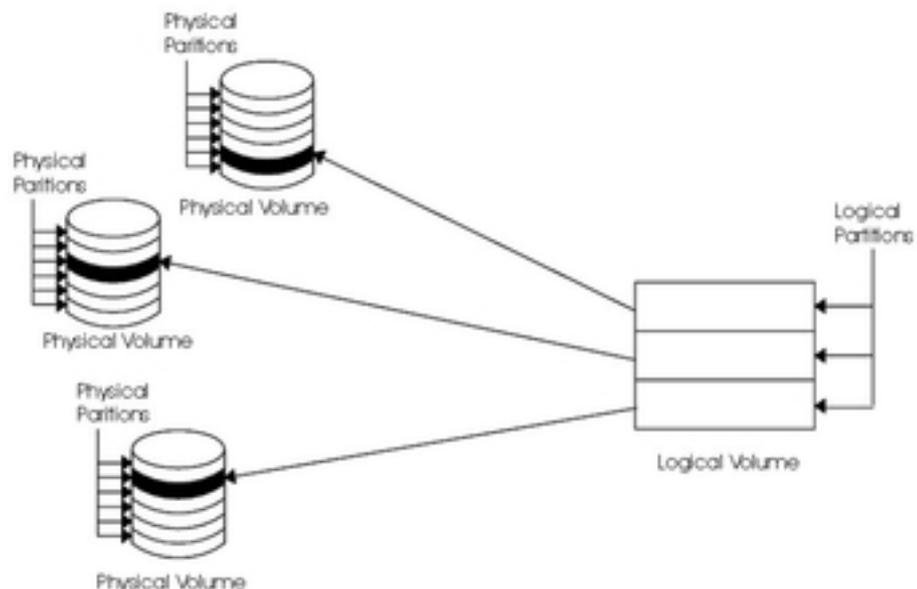
Concepts de stockage des volumes logiques

Les concepts de base du stockage des volumes logiques sont les suivants :

- Volumes physiques
- Groupes de volumes
- Partitions physiques
- Volumes logiques
- Partitions logiques

La figure ci-dessous montre les relations entre ces concepts.

Figure 1. Groupe de volumes Cette illustration montre un groupe de volumes constitué de trois volumes physiques avec une plage maximale. Le volume logique (qui peut être étendu à plusieurs volumes physiques) est constitué de partitions logiques allouées à des partitions physiques.



Volumes physiques

Un disque doit être défini comme volume physique et placé dans un état disponible pour pouvoir être affecté à un groupe de volumes. Un volume physique contient des informations de configuration et d'identification. Ces informations incluent un identificateur de volume physique qui est unique pour le système.

Dans AIX 5.2 et les versions supérieures, LVM peut utiliser l'espace supplémentaire que peut ajouter un module RAID (Redundant Array of Identical Disks) à un LUN (Logical Unit Number), en ajoutant des partitions physiques au volume physique associé au LUN.

Groupes de volumes

Le volume physique doit maintenant faire partie d'un *groupe de volumes*. Un groupe de volumes normal est un ensemble constitué de 1 à 32 volumes physiques de tailles et de types différents. Un gros groupe de volumes peut contenir entre 1 et 128 volumes physiques. Un volume physique peut appartenir à un seul groupe de volumes par système. Il peut exister jusqu'à 255 groupes de volumes actifs.

Lorsqu'un volume physique est affecté à un groupe de volumes, les blocs physiques sur le support de stockage sont organisés en partitions physiques dont la taille correspond à celle que vous définissez lorsque vous créez le groupe de volumes. Pour plus d'informations, reportez-vous à la section Partitions physiques.

Lorsque vous installez le système, un groupe de volumes (le groupe de volume racine appelé *rootvg*) est créé automatiquement. Il contient le jeu de base de volumes logiques nécessaires au démarrage du système ainsi que les autres volumes logiques que vous définissez dans le script d'installation. Le groupe de volumes *rootvg* contient l'espace de pagination, le journal des erreurs, les données d'amorçage, les clichés système, chacun de ces éléments étant enregistrés dans son propre volume logique. Les attributs du groupe de volumes *rootvg* sont différents de ceux des groupes de volumes que vous définissez. Ce volume ne peut pas être importé, ni exporté, par exemple. Lorsque vous exécutez une commande ou une procédure sur le groupe de volumes *rootvg*, vous devez connaître ses caractéristiques spécifiques.

Utilisez la commande **mkvg** pour créer un groupe de volumes. Pour ajouter un volume physique à un groupe de volumes, utilisez la commande **extendvg**. Pour changer la taille d'un volume physique, utilisez la commande **chvg**. Pour supprimer un volume logique d'un groupe de volumes, utilisez la commande **reducevg**. Vous pouvez utiliser également les commandes suivantes avec les groupes de volumes : liste (**lsvg**), retrait (**exportvg**), installation (**importvg**), réorganisation (**reorgvg**), synchronisation (**syncvg**), activation (**varyonvg**) et désactivation (**varyoffvg**).

Les petits systèmes peuvent nécessiter un seul groupe de volumes contenant tous les volumes physiques connectés au système. Vous pouvez créer des groupes de volumes différents, mais pour des raisons de sécurité, car chaque groupe de volumes peut avoir ses propres autorisations de sécurité. Des groupes de volumes différents facilitent également la maintenance, car les groupes autres que celui en cours de maintenance peuvent rester actifs. Du fait que le groupe de volumes *rootvg* doit toujours être en service, il contient uniquement le nombre de volumes physiques nécessaires au fonctionnement du système.

Vous pouvez transférer les données d'un volume physique vers un autre *dans le même groupe de volumes* avec la commande **migratepv**. Cette commande permet de libérer un volume physique pour le supprimer du groupe de volumes. Par exemple, vous pouvez transférer des données d'un volume physique à remplacer.

Un groupe de volumes créé avec des limites de volume physique et de volume logique plus petites peut être agrandi pour contenir un plus grand nombre de volumes physiques (jusqu'à 128) et de volumes logiques (jusqu'à 511). Cette opération nécessite qu'il existe un nombre suffisant de partitions dans chaque volume physique du groupe de volumes dans l'extension de la zone VGDA (Volume Group Descriptor Area). Le nombre de partitions libres nécessaires dépend de la taille de la zone VGDA actuelle et de la taille des partitions physiques. Du fait que la zone VGDA réside au bord du disque et qu'elle nécessite un espace contigu, les partitions libres doivent se trouver au bord du disque. Si ces partitions sont allouées à l'utilisateur, elles sont migrées vers d'autres partitions libres du même disque. Les autres partitions physiques sont renumérotées pour refléter la perte des partitions pour la zone VGDA. Cette renumérotation change l'association des partitions logiques et des partitions physiques dans tous les volumes physiques du groupe de volumes. Si vous avez sauvegardé les associations des volumes logiques pour une opération de reprise potentielle, régénérez les mappes à la fin de la conversion. En outre, si la sauvegarde du groupe de volumes est effectuée avec l'option *map* et que vous envisagez d'effectuer la restauration en utilisant ces mappes, la restauration peut échouer car le nombre de partitions peut ne plus exister (du fait de la réduction). Il est recommandé

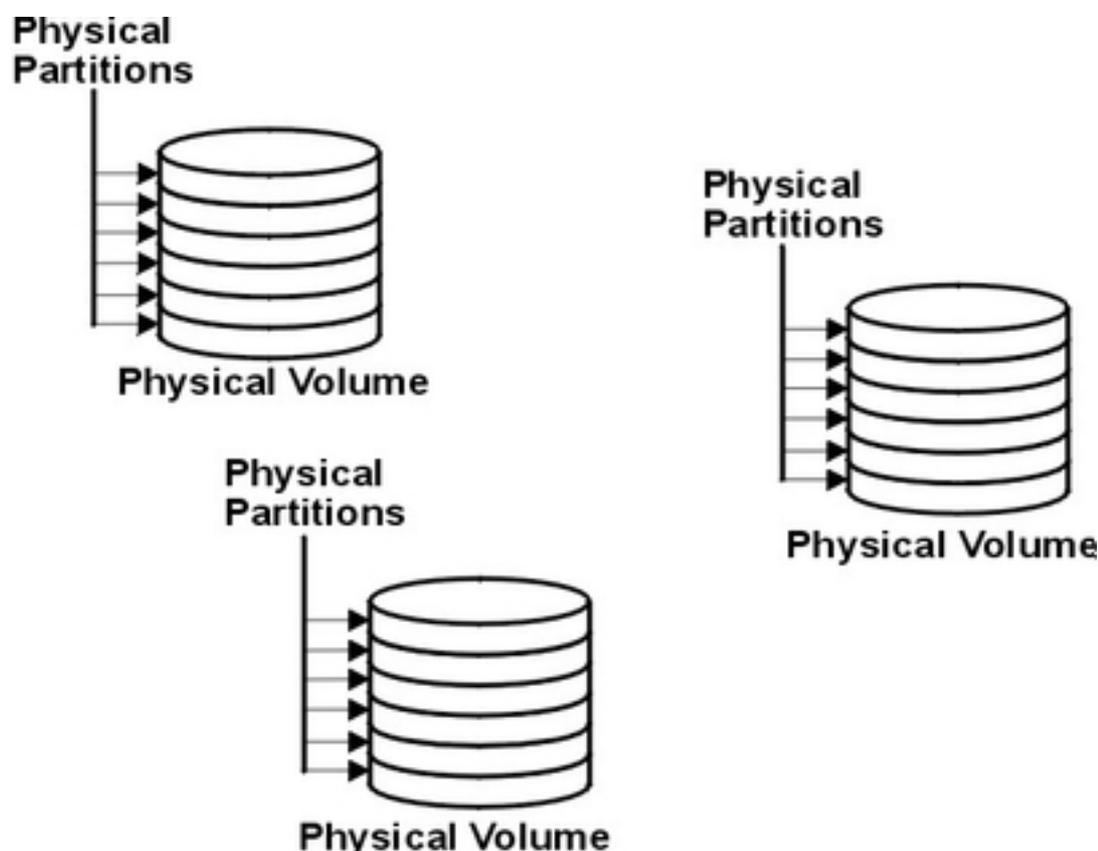
d'effectuer la sauvegarde avant la conversion et après la conversion si l'option `map` est utilisée. Du fait que l'espace VGDA a été augmenté de manière substantielle, chaque mise à jour VGDA (création d'un volume logique, modification d'un volume logique, ajout d'un volume physique, etc.) peut durer beaucoup plus longtemps.

Partitions physiques

Lorsque vous ajoutez un volume physique à un groupe de volumes, le volume physique est partitionné en espace d'unités égales contiguës appelées *partitions physiques*. Une partition physique est la plus petite unité d'allocation d'espace de stockage et correspond à un espace contigu dans un volume physique.

Les volumes physiques héritent de la taille de partition physique du groupe de volumes, que vous définissez uniquement lorsque vous créez le groupe de volumes (par exemple, à l'aide de la commande `mkvg -s`). L'illustration suivante montre la relation entre les partitions physiques des volumes physiques et les groupes de volumes.

Figure 2. Groupe de volumes contenant trois volumes physiques. Cette illustration montre trois volumes physiques contenant chacun six partitions physiques.



Volumes logiques

Après avoir créé un groupe de volumes, vous pouvez créer ses volumes logiques. Un *volume logique*, bien qu'il réside dans des partitions physiques non contiguës ou même sur plusieurs volumes physiques, apparaît pour les utilisateurs et les applications sous la forme d'un volume de disque extensible unique contigu. Vous pouvez créer d'autres volumes logiques avec la commande `mklv`. Cette commande permet de définir le nom du volume logique et ses caractéristiques, notamment le nombre et l'emplacement des partitions logiques à lui associer.

Après avoir créé un volume logique, vous pouvez changer son nom et ses caractéristiques avec la commande **chlv**. Vous pouvez augmenter le nombre de partitions logiques du volume logique à l'aide de la commande **extendlv**. La taille maximale par défaut d'un volume logique est égale à 512 partitions logiques, à moins que vous définissiez une autre valeur. Pour changer cette valeur, utilisez la commande **chlv**.

Remarque : Après avoir créé un volume logique, les caractéristiques LV STATE, que vous pouvez afficher avec la commande **lslv**, sont fermées. Elles sont rouvertes lorsque, par exemple, un système de fichiers est créé dans le volume logique et que le volume logique est monté.

Les volumes logiques peuvent être également copiés avec la commande **cplv**, listés avec la commande **lslv** et supprimés avec la commande **rmlv**. Vous pouvez augmenter ou diminuer le nombre de copies qu'ils gèrent à l'aide des commandes **mklvcopy** et **rmlvcopy**, respectivement. Vous pouvez également déplacer les volumes logiques lorsque le groupe de volumes est réorganisé.

Le système permet de définir jusqu'à 255 volumes logiques par groupe de volumes normal (511 pour un grand groupe de volumes), mais le nombre réel que vous pouvez définir dépend de la quantité totale de mémoire physique définie pour le groupe de volumes et de la taille des volumes logiques que vous définissez.

Partitions logiques

Lorsque vous créez un volume logique, vous définissez le nombre de *partitions logiques* du volume logique. Une partition logique est constituée d'une, de deux ou de trois partitions physiques, selon le nombre d'instances des données à gérer. Si vous définissez une seule instance, cela implique qu'il existe qu'une seule copie du volume logique (valeur par défaut). Dans ce cas, il existe une association directe entre une partition logique et une partition physique. Chaque instance, y compris la première, s'appelle une *copie*. L'emplacement des partitions physiques (à savoir leur emplacement physique par rapport aux autres) est déterminé par des options que vous définissez lorsque vous créez le volume logique.

Systèmes de fichiers

Le volume logique définit l'allocation d'espace disque jusqu'au niveau de la partition physique. Des niveaux de gestion plus fins des données peuvent être atteints à l'aide de composants logiciels de plus haut niveau, tels que Virtual Memory Manager ou le système de fichiers. En conséquence, la dernière étape de l'évolution d'un disque correspond à la création des *systèmes de fichiers*. Vous pouvez créer un système de fichiers par volume logique. Pour créer un système de fichiers, utilisez la commande **crfs**. Pour plus d'informations sur les systèmes de fichiers, reportez-vous à la section Systèmes de fichiers.

Limitations de la gestion de la mémoire logique

Le tableau suivant répertorie les limitations de la gestion de la mémoire logique. Bien que le nombre maximum par défaut de volumes physiques par groupe de volumes soit égal à (128 pour un grand groupe de volumes), vous pouvez définir le nombre maximum de groupes de volumes définis par l'utilisateur avec la commande **mkvg**. Pour le groupe de volumes **rootvg**, toutefois, cette variable est définie automatiquement par le système lors de l'installation.

Limitations de la gestion de la mémoire logique

Catégorie	Limite
Groupes de volumes	255 par système
Volume physique	(MAXPVS/facteur de groupe de volumes) par groupe de volumes. MAXPVS est égal à 32 pour un groupe de volumes normal et à 128 pour un grand groupe de volumes.
Partitions physiques	(1016 x facteur de groupe de volumes) par volume physique jusqu'à 1 024 Mo chacun
Volume logique	MAXLVS par groupe de volume, qui est égal à 255 pour un groupe de volumes normal et à 511 pour un grand groupe de volumes.
Partitions logiques	(MAXPVS * 1016) par volume logique

Si vous avez créé un groupe de volumes avant l'entrée en vigueur de la limite de 1016 partitions physiques par volume physique, les partitions périmées (qui ne contiennent plus des données à jour) dans le groupe de volumes ne peuvent pas être identifiées correctement si vous n'affectez pas un état pris en charge au groupe de volumes. Vous pouvez convertir le groupe de volumes avec la commande **chvg -t**. Une valeur de facteur adaptée est définie par défaut pour prendre en charge le plus grand disque du groupe de volumes.

Si, par exemple, vous avez créé un groupe de volumes avec un disque de 9 Go et des partitions de 4 Mo, le groupe de volumes contiendra 2 250 partitions environ. En utilisant le facteur de conversion 3 ($1016 * 3 = 3048$), vous pouvez suivre 2 250 partitions correctement. La conversion d'un groupe de volumes avec un facteur plus élevé permet d'inclure un plus grand nombre de partitions jusqu'au facteur 1 016*. Vous pouvez également définir un facteur plus élevé lorsque vous créez un groupe de volumes pour prendre en charge un plus grand disque avec des partitions plus petites.

Cette opération réduit le nombre total de disques que vous pouvez ajouter à un groupe de volumes. Le nouveau nombre maximum de disques que vous pouvez ajouter est MAXPVS/facteur. Par exemple, pour un groupe de volumes normal, le facteur 2 ramène à 16 ($32/2$) le nombre de disques maximum dans le groupe de volumes. Pour un groupe de volumes normal, le facteur 2 ramène à 64 ($128/2$) le nombre de disques maximum dans le groupe de volumes.

LVM (Logical Volume Manager)

Les commandes du système d'exploitation, les sous-routines de bibliothèque et les autres outils qui permettent de définir et de contrôler la mémoire logique s'appellent LVM (Logical Volume Manager). LVM contrôle les ressources disque en associant des données entre une vue *logique* plus simple et plus souple et les disques *physiques*. LVM effectue cette opération en utilisant une couche de code de pilotes d'unité qui s'exécute au-dessus des pilotes d'unité de disque traditionnels.

LVM est constitué du pilote LVDD (Logical Volume Device Driver) et de la bibliothèque d'interfaces de sous-routines LVM. Le pilote LVDD (*Logical Volume Device*) est un pseudo-pilote d'unité qui gère et traite toutes les E/S. Il convertit les adresses logiques en adresses physiques et envoie les demandes E/S à des pilotes d'unités spécifiques. La *bibliothèque d'interfaces de sous-routines* contient des routines utilisées par les commandes de gestion du système pour effectuer des tâches de gestion de système pour les volumes logiques et physiques d'un système. L'interface de programmation de la bibliothèque est disponible à quiconque veut étendre la fonction des commandes de gestion de système des volumes logiques.

Pour plus d'informations sur le fonctionnement de LVM, voir *Understanding the Logical Volume Device Driver* dans le document *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts* et *Logical Volume Programming Overview* dans le document *AIX 5L Version 5.2 General Programming Concepts*:

Concepts de quorum

Les sections suivantes décrivent la procédure de mise en service et le quorum qui correspond au mécanisme qu'utilise LVM pour garantir qu'un groupe de volumes est prêt à être utilisé et qu'il contient les données les plus à jour.

Procédure de mise en service

Les commandes **varyonvg** et **varyoffvg** activent et désactivent (rendent disponible et indisponible) un groupe de volumes que vous avez défini dans le système. Le groupe de volumes doit être activé pour que le système puisse y accéder. Au cours de l'activation, LVM lit les données de gestion sur les volumes physiques définis dans le groupe de volumes. Ces données de gestion, qui incluent une zone VGDA (Volume Group Descriptor Area) et une zone VGSA (Volume Group Status Area), sont stockées dans chaque volume physique du groupe de volumes.

La zone VGDA contient des informations qui décrivent l'association des partitions physiques et des partitions logiques de chaque volume logique du groupe de volumes ainsi que des informations importantes telles que la date et l'heure. La zone VGSA contient, par exemple, des informations sur les partitions physiques périmées et les volumes physiques manquants (non disponibles ou actifs) lorsqu'une activation est tentée dans un groupe de volumes.

Si la procédure d'activation ne peut pas accéder à un ou plusieurs volumes physiques définis dans le groupe de volumes, la commande affiche les noms de tous les volumes physiques du groupe de volumes et leur état. Cela vous permet de décider d'activer ou non le groupe de volumes.

Quorum

Un quorum est le rapport entre le nombre de zones VGDA et de zones VGSA actives. Un quorum garantit l'intégrité des données des zones VGDA/VGSA en cas de défaillance du disque. Chaque disque physique d'un groupe de volumes dispose d'au moins une zone VGDA/VGSA. Lorsque vous créez un groupe de volumes sur un disque, le disque contient deux zones VGDA/VGSA. Si un groupe de volumes est constitué de deux disques, un disque contient deux zones VGDA/VGSA et le second disque contient une zone VGDA/VGSA. Lorsqu'un groupe de volumes est constitué d'au moins trois disques, chaque disque contient une seule zone VGDA/VGSA.

Le quorum est perdu lorsqu'un nombre suffisant de disques et leurs VGDA/VGSA ne sont pas accessibles de sorte que 51 % des zones VGDA/VGSA n'existent plus. Dans un groupe de volumes à deux disques, si le disque qui contient une seule zone VGDA/VGSA est défaillant, un quorum existe toujours, car deux des trois zones VGDA/VGSA sont toujours accessibles. Si le disque avec deux zones VGDA/VGSA est défaillant, le quorum n'existe plus. Les risques de perte de quorum en cas de défaillance d'un disque sont inversement proportionnels au nombre de disques dans un groupe de volumes.

Lorsque le quorum est perdu, le groupe de volumes se désactive automatiquement pour que LVM ne puisse plus accéder aux disques. Cela évite de créer d'autres E/S dans le groupe de volumes pour que les données ne soient pas perdues ou supposées avoir été écrites lors de l'incident physique. En outre, suite à la désactivation, l'utilisateur reçoit un message d'erreur dans le journal des erreurs indiquant qu'un incident matériel s'est produit et qu'une opération de maintenance est nécessaire.

Il existe des cas dans lesquels il est préférable de continuer à utiliser le groupe de volumes, même si le quorum n'existe plus. Dans ce cas, vous pouvez désactiver la vérification du quorum du groupe de volumes. Ce type de groupe de volumes s'appelle un *groupe de volumes sans quorum*. En règle générale, un groupe de volumes sans quorum existe lorsque les volumes logiques sont mis en miroir. Lorsqu'un disque est défaillant, les données ne sont pas perdues s'il existe une copie du volume logique sur un disque actif accessible. Toutefois, il peut arriver que, dans des groupes de volumes sans quorum, mis en miroir ou non, les données (y compris les données) résident sur le disque ou les disques qui sont indisponibles. Dans ce cas, les données peuvent ne pas être accessibles, même si le groupe de volumes est toujours actif.

Groupes de volumes sans quorum

LVM (Logical Volume Manager) désactive automatiquement le groupe de volumes lorsque le quorum de zones VGDA ou VGSA n'est pas respecté. Toutefois, vous pouvez choisir une option qui permet au groupe de rester actif dès qu'il existe une zone VGDA/VGSA intacte. Cette option produit un *groupe de volumes sans quorum*. LVM doit pouvoir accéder à tous les disques des groupes de volumes sans quorum pour pouvoir permettre la réactivation et garantir que les zones VGDA et VGSA contiennent des données à jour.

Vous pouvez utiliser cette procédure sur des systèmes dont chaque volume logique dispose d'au moins de deux copies. En cas de défaillance de disque, le groupe de volumes reste actif tant qu'il existe un disque actif.

Remarque : Les groupes de volumes définis par l'utilisateur et le groupe de volumes rootvg peuvent fonctionner sans quorum, mais les méthodes utilisées pour configurer les groupes de volumes utilisateur et le groupe de volumes rootvg comme groupes de volumes sans quorum et pour restaurer les données à la suite d'incidents matériels sont différentes. Veillez à utiliser la méthode adaptée au groupe de volumes.

Développement d'une stratégie relative aux groupes de volumes

L'incident matériel auquel le système de stockage est le plus souvent confronté est la défaillance du disque, suivi de la défaillance des cartes et du système d'alimentation. La configuration des volumes logiques joue un rôle important de prévention des incidents de disque. Pour plus d'informations, reportez-vous à la section Développement d'une stratégie de volume logique. La taille des groupes de volumes peut être également déterminante.

Prévenir les défaillances d'alimentation et de cartes suppose une configuration matérielle particulière pour tout groupe de volumes spécifiques, avec deux cartes et au moins un disque par carte, avec utilisation de disques miroirs par le biais de cartes et une configuration de groupe de volumes "nonquorum". L'investissement en jeu pour de telles configurations n'est pas à la portée de tous les sites ou systèmes. Il est surtout recommandé lorsque la haute disponibilité est l'exigence prioritaire du système. Selon la configuration, la haute disponibilité est apte à couvrir les incidents matériels se produisant entre la dernière sauvegarde et l'entrée des données en cours; Elle ne couvre pas les fichiers supprimés par accident.

Création de groupes de volumes distincts

Voici différentes raisons de structurer des volumes physiques en groupes de volumes indépendants de rootvg :

- Pour sécuriser et faciliter les opérations de maintenance.
 - Les mises à jour du système d'exploitation, les réinstallations et les reprises sur panne système sont plus fiables avec des systèmes de fichiers utilisateur séparés du système d'exploitation, préservant ainsi les fichiers utilisateur.
 - Les opérations de maintenance sont facilitées : vous pouvez actualiser le système d'exploitation ou le réinstaller sans restaurer les données utilisateur. Par exemple, avant de lancer une mise à jour, vous pouvez retirer du système un groupe de volumes utilisateur en démontant ses systèmes de fichiers, en le désactivant (avec la commande **varyoffvg**), puis en exportant le groupe (avec la commande **exportvg**). Ensuite, après la mise à jour, il suffit de réintégrer le groupe de volumes utilisateur (avec la commande **importvg**), puis de remonter ses systèmes de fichiers.
- Pour définir des tailles de partitions physiques hétérogènes. Tous les volumes physiques d'un même groupe de volumes doivent avoir la même taille de partition physique. Pour que des volumes physiques aient des tailles de partition physique différentes, placez chaque taille dans un groupe de volumes distinct.
- Pour définir des caractéristiques de quorum hétérogènes. Vous pouvez prévoir un groupe de volumes distinct pour un système de fichiers de type "nonquorum", tous les autres systèmes de fichiers faisant partie du groupe de volumes soumis au quorum.
- Pour la sécurité. Par exemple, vous pouvez être amené à retirer de nuit un groupe de volumes.
- Pour commuter des volumes physiques entre systèmes. La commutation de volumes physiques entre systèmes est possible en créant un groupe de volumes par système connecté à une carte accessible aux systèmes, ceci sans interrompre les opérations en cours (voir les commandes **varyoffvg**, **exportvg**, **importvg** et **varyonvg**).

Haute disponibilité face aux incidents de disque

Les principales mesures de prévention des incidents de disque comprennent les définitions de configuration de volume logique, telles que l'écriture miroir. Bien que secondaires, les informations ci-après ont des répercussions économiques significatives, impliquant le nombre de volumes physiques par groupe de volumes :

- La configuration avec quorum (valeur par défaut) maintient le groupe de volumes actif (en service) tant que le quorum des disques (51%) existe. Pour plus d'informations sur les conditions, reportez-vous à la section Processus de mise en service. Dans la plupart des cas, vous devez utiliser au moins trois disques en plaçant leur image miroir dans le groupe de volumes pour la protection contre les incidents de disque.
- La configuration sans quorum maintient le groupe de volumes actif (en service) tant qu'une zone VGDA est disponible sur le disque (voir "Changing a Volume Group to Nonquorum Status" dans le document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*). Dans cette configuration, vous n'utilisez que deux disques en plaçant leur image miroir dans le groupe de volumes pour la protection contre les incidents de disque.

Lors de la détermination du nombre de disques associés à chaque groupe de volumes, vous devez également prendre en compte l'espace nécessaire à la mise en miroir des données. Notez que vous ne pouvez mettre en miroir les données et déplacer les données qu'entre les disques qui appartiennent à un même groupe de volumes. Si le site utilise des systèmes de fichiers volumineux, la recherche d'espace disque pour la mise en miroir peut poser des problèmes ultérieurement. Tenez compte des conséquences sur la disponibilité des paramètres inter disques pour les copies de volumes logiques (voir Paramètres inter disques pour les copies de volumes logiques) et l'allocation intra-disque (voir Détermination d'une stratégie d'allocation intra-disque pour chaque volume logique) pour chaque volume logique.

Haute disponibilité face aux incidents de carte ou d'alimentation

Pour prévenir les incidents de carte ou d'alimentation, en fonction de la rigueur de vos besoins :

- Utilisez deux cartes, installées ou non dans la même armoire. Si les cartes ne sont pas situées dans la même armoire, elles seront toutes deux protégées en cas de problème d'alimentation dans une armoire.
- Utilisez deux cartes et connectez un ou plusieurs disques sur chacune : ceci pour prévenir tout incident de carte (ou d'alimentation lorsque les cartes sont installées dans des armoires séparées) en maintenant le quorum dans le groupe de volumes, compte tenu de l'écriture miroir croisée (pour une partition logique, les copies ne pouvant partager le même volume physique) entre les volumes logiques du disque A (carte A) et les volumes logiques du disque B (carte B). Ceci signifie que vous copiez les volumes logiques des disques connectés à la carte A sur les disques connectés à la carte B, et inversement.
- Configurez tous les disques à partir des deux cartes dans le même groupe de volumes. Ainsi, au moins une copie de volume logique reste intacte si une carte est défectueuse, ou, en cas d'armoires séparées, si l'alimentation est défaillante.
- Passez le groupe de volumes à l'état "nonquorum" Ainsi, le groupe reste actif tant qu'une zone VGDA est accessible sur un disque quelconque du groupe. Reportez-vous à "Passage d'un groupe de volumes à l'état nonquorum" dans le manuel *AIX - Guide d'administration : système d'exploitation et unités*.
- Si le groupe de volumes possède deux disques, configurez l'écriture miroir croisée entre les cartes. Si plusieurs disques sont connectés à chaque carte, configurez l'écriture miroir double en créant une copie miroir sur un disque utilisant la même carte et une autre sur un disque utilisant une autre carte.

Développement d'une stratégie relative aux volumes logiques

Les règles ci-après vous aideront à élaborer une stratégie en matière d'exploitation des volumes logiques, orientée à la fois sur la disponibilité, les performances et le coût.

La *disponibilité* correspond à la possibilité d'accéder aux données, même si le disque associé est défaillant ou inaccessible. Les données peuvent rester accessibles en les copiant et en les gérant sur des disques et des cartes différents lorsque le système fonctionne normalement. La mise en miroir et l'utilisation de disques de secours, par exemple, permettent de garantir la disponibilité des données.

Les *performances* correspondent au délai d'accès moyen aux données.

L'écriture–vérification et la mise en miroir, par exemple, améliorent la disponibilité, mais augmentent la charge de travail du système et affectent donc les performances. La mise en miroir double ou triple la taille du volume logique. En règle générale, l'amélioration de la disponibilité affecte les performances. L'entrelacement des disques peut améliorer les performances. Depuis la version AIX 4.3.3, l'entrelacement peut être utilisé avec la mise en miroir. Depuis la version AIX 5.1, vous pouvez détecter et résoudre les problèmes qui se produisent lorsque des partitions logiques du disque sont soumises à un nombre excessif d'opérations E/S qui affectent sensiblement les performances.

En contrôlant l'allocation des données sur le disque et entre les disques, vous pouvez régler le système de stockage pour optimiser les performances. Pour plus d'informations sur la manière d'optimiser les performances du système de stockage, reportez-vous au document *AIX 5L Version 5.2 Performance Management Guide*.

Utilisez les sections suivantes pour évaluer les compromis entre les performances, la disponibilité et les coûts. Notez que l'amélioration de la disponibilité dégrade généralement les performances et vice versa. Toutefois, la mise en miroir améliore les performances si LVM choisit de copier les données sur le disque le moins sollicité pour la lecture.

Remarque : La mise en miroir n'offre pas de protection contre la perte de fichiers individuels supprimés ou perdus de manière accidentelle à la suite de problèmes logiciels. Ces fichiers ne peuvent être restaurés que depuis une bande classique ou les sauvegardes des disques.

Analyse des besoins associés à la mise en miroir et à l'entrelacement

Déterminez si les données stockées dans le volume logique sont suffisamment importantes pour justifier le traitement et les coûts d'espace disque qu'impliquent la mise en miroir. Si vous utilisez un système de fichiers à accès séquentiel volumineux dont l'efficacité dépend des performances, envisagez d'utiliser l'entrelacement.

Performances et mise en miroir ne sont pas forcément antagonistes. Si les copies des partitions logiques se trouvent sur des volumes physiques différents, connectés de préférence à des cartes différentes, LVM peut améliorer les opérations de lecture en lisant la copie sur le disque le moins utilisé. Les performances des opérations d'écriture, si les disques ne sont pas connectés à des cartes différentes, ont le même coût, car vous devez mettre à jour toutes les copies. Il est uniquement nécessaire de lire une seule copie pour une opération de lecture.

LVM AIX prend en charge les options RAID suivantes :

Tableau 1. Prise en charge LVM pour RAID

RAID 0	Entrelacement
RAID 1	Mise en miroir
RAID 10 ou 0+1	Mise en miroir et entrelacement

Bien que la mise en miroir améliore la disponibilité du système de stockage, elle ne remplace pas les sauvegardes sur bande classique.

Vous pouvez mettre en miroir le groupe de volumes rootvg, mais dans ce cas, créez un volume logique cliché différent. La création d'un cliché dans un volume logique en miroir peut générer un cliché incohérent. En outre, du fait que l'unité de cliché par défaut correspond au volume logique de pagination principal, créez un volume logique cliché différent si vous mettez en miroir les volumes logiques de pagination.

Normalement, lorsque les données d'une partition logique sont mises à jour, toutes les partitions physiques qui contiennent la partition logique sont mises à jour automatiquement. Toutefois, les partitions physiques peuvent devenir *obsolètes* (elles ne contiennent pas les toutes dernières données) suite à des dysfonctionnements du système ou du fait de la non-disponibilité du volume physique lors d'une mise à jour. LVM peut actualiser les partitions obsolètes avec les informations appropriées en copiant les données en cours depuis une partition physique à jour. Ce processus s'appelle la *synchronisation de la mise en miroir*. L'actualisation peut avoir lieu lors du démarrage du système, lorsque le volume physique est remis en ligne ou lorsque vous exécutez la commande **syncvg**.

Toute modification qui affecte la partition physique d'un volume logique d'amorçage nécessite d'exécuter la commande **bosboot** à la suite de la modification. Cela implique que des actions telles que la modification de la mise en miroir d'un volume logique d'amorçage nécessite d'exécuter la commande **bosboot**.

Règles de programmation des écritures miroir sur disque

Pour des données qui n'ont qu'une copie physique, LVDD traduit l'adresse des demandes de lecture ou d'écriture logique en adresse physique, et appelle le gestionnaire d'unités physiques approprié pour traiter la demande. Cette politique de copie unique ou d'écriture non miroir gère la réaffectation des blocs défectueux pour les demandes d'écriture et renvoie toutes les erreurs de lecture au processus appelant.

Si vous utilisez des volumes logiques en miroir, vous pouvez définir quatre politiques de programmation différentes pour l'écriture sur disque dans le cadre d'un volume logique avec copies multiples : le traitement *séquentiel*, le traitement *parallèle*, l'écriture parallèle avec lecture séquentielle et l'écriture parallèle avec lecture "round robin".

Politique de traitement séquentiel

L'écriture sur plusieurs copies ou l'écriture miroir est traitée en séquence. Plusieurs partitions physiques représentant les copies miroir d'une seule partition logique sont qualifiées de primaires, secondaires et tertiaires. Les partitions physiques sont écrites séquentiellement (l'une après l'autre). Le système attend que l'opération d'écriture sur une partition physique soit terminée avant de commencer l'opération d'écriture sur la partition suivante. Une fois toutes les opérations d'écriture terminées sur tous les miroirs, l'opération d'écriture globale est terminée.

Politique de traitement parallèle

En traitement parallèle, l'écriture de toutes les partitions physiques d'une partition logique démarre en même temps. Elle prend fin après l'écriture de la partition physique la plus longue. La définition de volumes logiques mis en miroir et d'une stratégie de planification parallèle peut améliorer les performances des opérations de lecture E/S, car plusieurs copies permettent au système d'envoyer les opérations de lecture vers le disque le moins utilisé du volume logique.

Politique d'écriture parallèle avec lecture séquentielle

L'écriture de toutes les partitions physiques d'une partition logique démarre en même temps. La lecture s'effectue toujours d'abord sur la copie primaire. Si elle n'aboutit pas, elle est effectuée sur la copie suivante. A la seconde tentative de lecture sur la copie suivante, la copie primaire dont la lecture a échoué est corrigée par LVM, avec une réaffectation matérielle. Ceci permet de corriger le bloc défectueux pour un accès ultérieur.

Politique d'écriture parallèle avec politique de lecture "round robin"

L'écriture de toutes les partitions physiques d'une partition logique démarre en même temps. Les lectures s'effectuent alternativement sur les différentes copies en miroir.

MWC (cohérence écrit-miroir) pour un volume logique

Lorsqu'il est activé, MWC identifie les partitions logiques susceptibles d'être incohérentes si le système ou le groupe de volumes n'est pas correctement fermé. Lorsque le groupe de volumes est à nouveau mis en fonction, ces informations sont utilisées pour rendre cohérentes les partitions logiques. On nomme cette opération MWC actif.

Lorsqu'un volume logique utilise le MWC actif, les requêtes portant sur ce volume logique sont retenues dans la couche de programmation jusqu'à ce que les blocs de la mémoire cache MWC puissent être mis à jour sur les volumes physiques cible. Une fois les blocs de mémoire cache MWC mis à jour, la requête passe aux opérations d'écriture des données physiques.

Les blocs de mémoire cache MWC ne doivent être écrits que sur les disques sur lesquels les données résident effectivement avant que l'opération d'écriture puisse se poursuivre.

Lorsque le MWC actif est utilisé, les performances système peuvent être diminuées. Ce fait est dû à la surcharge de la journalisation et de la consignation par suite de l'activité de la requête d'écriture dans un LTG (Logical Track Group). Les tailles de LTG autorisées pour un groupe de volumes sont 128 Ko, 256 Ko, 512 Ko et 1024 Ko, la taille par défaut étant 128 Ko.

Remarque : Pour que la taille de LTG soit supérieure à 128 Ko, les disques faisant partie du groupe de volumes doivent prendre en charge les demandes d'E/S de cette taille à partir des routines de stratégie du disque. Le LTG est un bloc contigu contenu dans le volume logique, dont l'alignement s'effectue en fonction de sa taille. Les opérations d'E/S ne doivent pas dépasser la taille d'un LTG, et doivent être contenues un seul LTG. Cette surcharge ne s'applique qu'à la mise en miroir des opérations d'écriture.

Il est nécessaire d'assurer la cohérence des données entre les miroirs uniquement en cas de panne du système ou du groupe de volumes avant la fin de l'écriture vers tous les miroirs. Tous les volumes logiques d'un groupe de volumes partagent le journal MWC. Ce dernier est tenu à jour sur le bord externe de chaque disque. Placez les volumes logiques tirant parti de MWC actif sur le bord externe du disque, de façon qu'ils se trouvent à proximité du journal MWC sur disque.

Lorsque MWC est à l'état passif, le groupe de volume consigne le fait que le volume logique a été ouvert. Lorsque le groupe de volumes est mis en fonction après une panne, une synchronisation forcée automatique du volume logique est lancée. La cohérence est assurée pendant que la synchronisation forcée est en cours en faisant appel à une copie de la politique de récupération de lecture qui propage les blocs en cours de lecture sur les autres miroirs du volume logique. Cette règle n'est prise en charge que sur le type de groupe de volumes BIG.

Lorsque MWC est **désactivé**, les miroirs d'un volume logique en miroir peuvent être laissés dans un état incohérent au cas où le système ou le groupe de volumes tomberait en panne. Il n'y a pas de protection automatique de la cohérence du miroir. Les écritures restant à faire au moment de la panne peuvent laisser les miroirs dans un état incohérent lors de la prochaine mise en fonction du groupe de volumes. Après une panne, un volume logique en miroir avec MWC désactivé doit effectuer une synchronisation forcée avant l'utilisation des données du volume. Par exemple :

```
syncvg -f -l LTV nom
```

Il y a exception à la synchronisation forcée dans le cas de volumes logiques dont le contenu n'est valide que pendant l'ouverture du volume logique, par exemple les espaces de pagination.

Un volume logique mis en miroir est identique à un volume logique non mis en miroir en terme d'opération de lecture. Une fois LVM a fini de traiter une demande d'écriture, les données sont écrites sur toutes les unités sous LVM. Le résultat de l'écriture est inconnu jusqu'à ce que LVM exécute un processus **iodone** sur l'opération d'écriture. Une fois cette opération terminée, aucune restauration n'est nécessaire après un blocage. Tout bloc écrit non terminé (**iodone**) lors d'un blocage de la machine doit être vérifié et réécrit, quel que soit le paramètre MWC ou qu'il soit mis en miroir ou non.

Du fait qu'un volume logique mis en miroir est identique à un volume logique non mis en miroir, il ne peut y avoir de problème d'actualisation de données de ce type. Toutes les applications qui doivent tenir compte de la validité des données doivent déterminer la validité des données des opérations d'écriture en attente ou en cours lors de l'incident sur le groupe de volume ou la défaillance du système que le volume logique soit mis en miroir ou non.

Le MWC actif et passif ne rendent les miroirs cohérents que lorsque le groupe de volumes est remis en fonction après une panne, en choisissant un miroir et en propageant ces données sur les autres miroirs. Les règles MWC ne conservent pas la trace des dernières données. Le MWC actif n'assure que le suivi des LTG en cours d'écriture. Il ne garantit donc pas la propagation des dernières données à tous les miroirs. Le MWC passif assure la cohérence des miroirs en passant dans un mode de propagation sur lecture après une panne. C'est l'application au-dessus de LVM qui détermine la validité des données après une panne. D'après la perspective LVM, si l'application réémet toujours toutes les demandes d'écriture en suspens au moment de la panne, les miroirs éventuellement incohérents deviendront cohérents à la fin des opérations d'écriture (du moment que les blocs écrits sont les mêmes que ceux en suspens au moment de la panne).

Remarque : Les volumes logiques en miroir contenant des journaux JFS ou des systèmes de fichiers doivent être synchronisés après une panne, soit par synchronisation forcée avant utilisation, en activant MWC, soit en activant le MWC passif.

Règles d'affectation inter-disque

Cette procédure permet de spécifier le nombre de disques sur lesquels sont situées les partitions physiques du volume logique. Celles-ci peuvent être placées sur un seul disque ou réparties sur l'ensemble des disques. Les options suivantes sont utilisées avec les commandes **mklv** et **chlv** pour déterminer la stratégie inter disque :

- *Range* définit le nombre de disques utilisés pour une copie physique unique du volume logique.
- *Strict* détermine si **mklv** peut aboutir lorsque plusieurs copies doivent occuper le même volume physique.
- L'option *Super Strict* définit que les partitions allouées à un miroir ne peuvent pas partager un volume physique avec les partitions d'un autre miroir.
- Si les volumes logiques sont répartis sur plusieurs disques, les seules valeurs admises pour *Range* et **Strict** sont **maximum** et **yes**.

Copie unique du volume logique

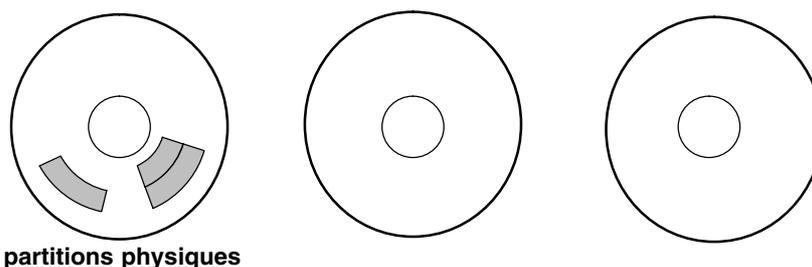
Si vous optez pour l'affectation inter- (*Range = minimum*), les partitions physiques affectées au volume logique seront situées sur un seul disque, afin d'accroître la disponibilité. Avec l'affectation inter-disque optimale (*Range = maximum*), les partitions physiques sont situées sur plusieurs disques pour améliorer la performance. L'affectation de copies miroir aux partitions d'origine est décrite dans la section suivante.

Pour les volumes logiques de type "non miroir", prenez la valeur **minimum** qui confère au système une disponibilité optimale (quant à l'accès aux données en cas de d'incident matériel). Le paramètre **minimum** indique qu'un volume physique doit contenir, si possible, toutes les partitions physiques d'origine de ce volume. Si le programme d'affectation a besoin de plusieurs volumes physiques, il utilise le nombre minimal, tout en restant cohérent par rapport aux autres paramètres.

En utilisant le nombre minimal de volumes physiques, vous réduisez le risque de perdre des données en cas de défaillance d'un disque. Tout volume physique supplémentaire utilisé pour une copie physique unique accroît ce risque. En cas de défaillance d'un volume physique, le risque de perdre des données pour un volume logique de type "non miroir" réparti sur quatre volumes physiques est quadruplé par rapport à un volume logique logé dans un seul volume physique.

La figure ci-après illustre une affectation inter-disque minimale.

Figure 3. Stratégie d'allocation inter disque minimum Cette illustration contient trois disques. Un disque contient trois partitions physiques et les deux autres disques ne contiennent pas de partitions physiques.

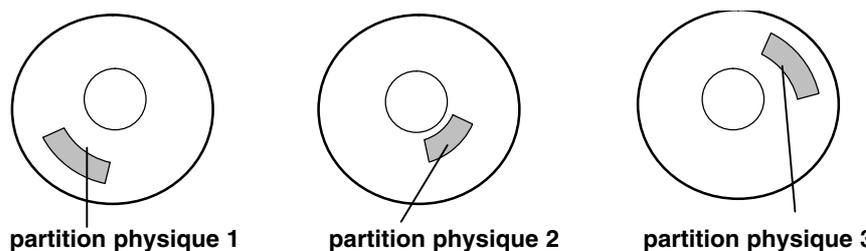


partitions physiques

Le paramètre **maximum**, compte tenu d'autres contraintes, répartit de façon aussi égale que possible, les partitions physiques du volume logique sur le plus grand nombre de volumes physiques possible. Cette option vise la performance, en tendant à réduire le temps d'accès du volume logique. Pour améliorer la disponibilité, elle ne doit être utilisée qu'avec des volumes logiques avec traitement miroir.

La figure ci-après illustre une affectation inter-disque optimale.

Figure 4. Stratégie d'allocation inter disque maximum Cette illustration contient trois disques contenant chacun une partition physique.



partition physique 1

partition physique 2

partition physique 3

Ces paramètres sont également applicables à l'extension et à la copie d'un volume logique existant. L'affectation de nouvelles partitions physiques est déterminée en fonction de la politique en cours, ce à l'emplacement où sont situées les partitions physiques existantes.

Copies multiples du volume logique

L'affectation d'une seule copie d'un volume logique est une procédure relativement simple. Toutefois, lors de la création de copies miroir, l'affectation résultante est quelque peu complexe. Les figures suivantes illustrent l'emploi des paramètres **minimum** et **maximum** (*Range*) pour la première instance d'un volume logique, et les définitions possibles de *strict* pour les copies miroir de ce volume.

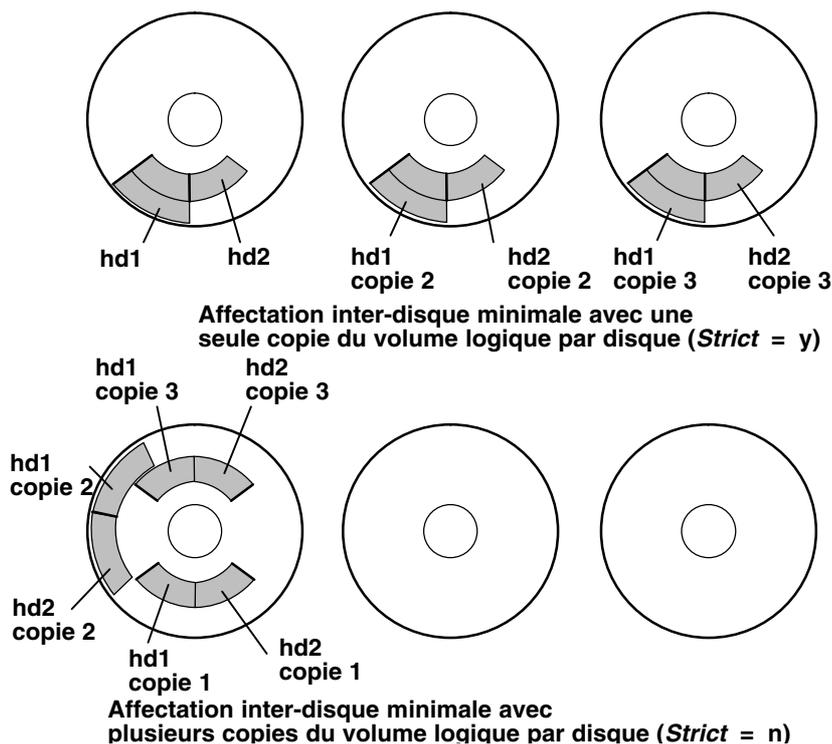
Par exemple, avec des copies miroir du volume logique, **minimum** permet d'affecter, si possible, les partitions physiques contenant la première instance du volume logique sur un seul volume physique. Puis, en fonction de la définition de *Strict*, la ou les copies supplémentaires sont affectées sur le même volume physique ou sur des volumes distincts. Autrement dit, l'algorithme utilise le plus petit nombre possible de volumes physiques, en respectant les contraintes imposées par les autres paramètres tels que *Strict*, pour maintenir toutes les partitions physiques.

Strict = y signifie que chaque copie de la partition logique est placée sur un volume physique différent. *Strict = n* signifie que les copies ne sont pas tenues d'être situées sur des volumes distincts. En comparaison, l'option *Super Strict* ne permet pas à une partition physique d'un miroir de se trouver sur le même disque qu'une partition physique d'un autre miroir du même volume logique.

Remarque : Lorsque le nombre de volumes physiques du groupe est inférieur au nombre de copies par partition logique sélectionnée, affectez la valeur **n** à *Strict*. Si *Strict = y*, un message d'erreur est renvoyé quand vous tentez de créer le volume logique.

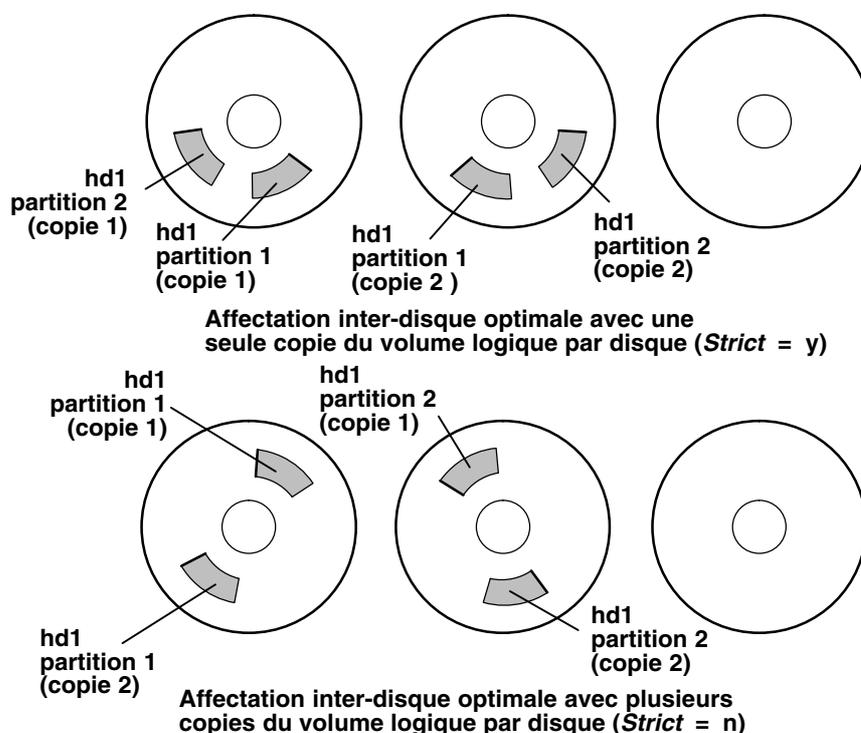
La figure Affectation inter-disque minimale/ *Strict* illustre une règle d'affectation inter-disque minimale avec différents paramètres *Strict* :

Figure 5. Stratégie inter disque minimum/Option *Strict* Cette figure montre que si l'option *Strict* est affectée de la valeur *Yes*, chaque copie de la partition logique se trouve sur un volume physique différent. Si l'option *Strict* est affectée de la valeur *No*, toutes les copies des partitions logiques se trouvent dans le même volume physique.



La figure Affectation inter-disque optimale/*Strict* illustre une règle d'affectation inter-disque optimale avec différents paramètres *Strict* :

Figure 6. Stratégie inter disque maximum/Option Strict Cette figure montre que si l'option *Strict* est affectée de la valeur *Yes*, chaque copie d'une partition se trouve sur un volume physique différent. Si l'option *Strict* est affectée de la valeur *No*, toutes les copies des partitions logiques se trouvent dans le même volume physique.



Règles d'affectation intra-disque

Plus une partition physique est proche du centre d'un volume physique, plus le temps de recherche moyen est rapide, la distance de recherche moyenne étant la plus réduite par rapport aux autres zones du disque.

Le journal du système de fichiers est un bon candidat pour l'allocation au centre d'un volume physique du fait qu'il est très fréquemment utilisé par le système d'exploitation. A l'autre extrémité, le volume logique d'amorçage est utilisé peu fréquemment et il se trouve donc au bord ou au milieu du volume physique.

En règle générale, plus les E/S (en valeur absolue ou pendant l'exploitation d'une application importante) sont nombreuses, plus les partitions physiques du volume logique doivent être affectées près du centre des volumes physiques. Deux exceptions à cette règle :

Il existe une exception importante à cette règle : Les volumes logiques mis en miroir avec l'option MWC (Mirror Write Consistency) affectée de la valeur *On* se trouvent sur le bord extérieur, car il s'agit de l'emplacement dans lequel le système écrit les données MWC. Si la mise en miroir n'est pas active, MWC ne s'applique pas et n'affecte pas les performances.

Les choix de stratégies d'allocation intra-disque reposent sur les cinq régions d'un disque dans lesquelles les partitions physiques peuvent être allouées. Les cinq régions sont les suivantes :

1. outer edge
2. inner edge
3. outer middle
4. inner middle
5. center

La politique d'affectation intra-disque est fondée sur les cinq régions de disque où placer les partitions physiques. Ces cinq régions sont : *outer edge* (bord extérieur), *inner edge* (bord intérieur), *outer middle* (milieu extérieur), *inner middle* (milieu intérieur), et *center* (centre). Au bord, le temps de recherche moyen est le plus lent, d'où des temps de réponse plus longs pour les applications. Les partitions du centre ont le temps de recherche moyen optimal, donc les meilleurs temps de réponse. Toutefois, le centre contient moins de partitions sur un volume physique que les autres régions du disque.

Affectations combinées

Le choix de politiques d'affectation inter-disque et intra-disque incompatibles peut générer des résultats imprévisibles. Le système donnera la priorité à un type d'affectation. Par exemple, si vous optez pour une affectation intra-disque de type "center" au centre et inter-disque de type minimum, l'affectation inter-disque sera prioritaire. Le système placera, si possible, toutes les partitions du volume logique sur un seul disque, même si elles ne tiennent pas toutes au centre du disque. Avant de combiner différentes politiques, assurez-vous d'en avoir assimilé les interactions.

Affectation affinée avec des fichiers mappe

Si les options par défaut des politiques d'affectation inter-disque et intradisque ne sont pas adaptées à vos besoins, vous pouvez créer des fichiers mappe pour spécifier l'ordre et l'emplacement exacts des partitions physiques d'un volume logique.

Vous pouvez utiliser WSM (Web-based System Manager), SMIT ou la commande **mkiv -m** pour créer des fichiers de mappes.

Par exemple, pour créer un volume logique de dix partitions nommé lv06 dans rootvg en partitions 1 à 3, 41 à 45 et 50 à 60 sur le disque hdisk1, procédez comme suit :

1. Pour vérifier si les partitions physiques que vous envisagez d'utiliser peuvent être allouées, tapez :

```
lspv -p hdisk1
```

2. Créez un fichier, tel que /tmp/mymap1, contenant :

```
hdisk1:1-3
hdisk1:41-45
hdisk1:50-60
```

La commande **mkiv** affectera les partitions physiques dans l'ordre où elles figurent dans le fichier mappe. Assurez-vous que ce fichier contient un nombre suffisant de partitions à affecter à l'ensemble du volume logique spécifié par la commande **mkiv**. (Au besoin, vous pouvez en afficher la liste.)

3. Exécutez la commande :

```
mkiv -t jfs -y lv06 -m /tmp/mymap1 rootvg 10
```

Développement d'une stratégie relative à la répartition du volume logique

La répartition des volumes logiques concerne les systèmes de fichiers séquentiels volumineux, sensibles aux performances, et à forte fréquence d'accès. Ce procédé vise à améliorer les performances.

Remarque : un espace de cliché ou un volume logique d'amorçage ne peut être entrelacé. Le volume logique d'amorçage doit correspondre à des partitions physiques contiguës.

Pour créer le volume logique entrelacé à 12 partitions 1v07 dans NomVG avec une taille de segment de 16 Ko sur hdisk1, hdisk2 et hdisk3, tapez :

```
mklv -y 1v07 -S 16K NomVG 12 hdisk1 hdisk2 hdisk3
```

Pour créer un volume logique de 12 partitions nommé 1v08 dans NomVG réparti sur trois disques quelconques, avec une taille de 8 Ko pour la répartition, entrez :

```
mklv -y 1v08 -S 8K -u 3 NomVG 12
```

Pour plus de détails sur l'amélioration des performances avec cette stratégie, reportez-vous à *AIX 5L Version 5.2 - Guide d'optimisation*.

Règles de contrôle de l'écriture

L'option de contrôle de l'écriture lit en temps réel toutes les opérations d'écriture pour vérifier qu'elles sont lisibles. Un message d'erreur s'affiche lorsqu'elles sont incorrectes. Cette option améliore la disponibilité, ceci au détriment de la performance, en raison du temps de lecture nécessaire. Vous pouvez définir cette option sur un volume logique lors de sa création (**mklv**) ou, ultérieurement, en le modifiant (**chlv**).

Détermination des stratégies de disques de secours remplaçables à chaud

Depuis la version AIX 5.1, vous pouvez définir des disques comme disques de secours remplaçables à chaud pour un groupe de volumes avec des volumes logiques mis en miroir. Lorsque vous définissez les disques de secours remplaçables à chaud, vous pouvez définir une stratégie en cas de défaillance d'un ou de plusieurs disques, ainsi que les caractéristiques de synchronisation. Ce support complète mais ne remplace pas le support de création de disques de secours disponible avec les disques SSA (Serial Storage Architecture). Vous pouvez également utiliser des disques de secours remplaçables à chaud lorsque vous en ajoutez un au groupe de volumes.

Si vous ajoutez un volume physique à un groupe de volumes (pour le définir comme disque de secours remplaçable à chaud), le disque doit avoir au minimum la même capacité que le plus petit disque du groupe de volumes. Lorsque cette fonction est mise en œuvre, les données sont envoyées vers un disque de secours remplaçable à chaud lorsque les échecs d'écriture MWC (Mirror Write Consistency) marquent un volume physique manquant.

Les commandes de prise en charge des disques de secours remplaçables à chaud, **chvg** et **chpv**, fournissent diverses options de mise en œuvre de la fonction sur votre site, comme l'indique la syntaxe suivante :

```
chvg -h stratégie_replacement_à_chaud -s stratégie_synchronisation  
Groupe_volumes
```

où *stratégie_replacement_à_chaud* définit l'une des stratégies suivantes à utiliser en cas de défaillance d'un disque :

- y Transfère automatiquement les partitions du disque défaillant vers un disque de secours. Dans le groupe de disques de secours remplaçables à chaud, le plus petit disque ayant la capacité suffisante pour remplacer le disque défaillant est utilisé.
- Y Transfère automatiquement les partitions du disque défaillant, mais peut utiliser l'ensemble des disques de secours remplaçables à chaud.

- n Ne transfère pas automatiquement les partitions (valeur par défaut).
- r Supprime tous les disques du groupe de disques de secours remplaçables à chaud du groupe de volumes.

L'argument *stratégie_synchronisation* indique si vous voulez synchroniser automatiquement les partitions contenant des données obsolètes :

- y Tente automatiquement de synchroniser les partitions obsolètes.
- n Ne synchronise pas automatiquement les partitions obsolètes. (il s'agit de l'option par défaut).

L'argument *Groupe_volumes* définit le nom du groupe de volumes associé mis en miroir.

Gestion des zones actives dans les volumes logiques

Depuis la version AIX 5.1, vous pouvez identifier les *zones actives* des volumes logiques et résoudre les problèmes sans arrêter le système. Un problème de zone active se produit lorsque le nombre d'E/S traitées par des partitions logiques du disque affectent de manière significative les performances.

La première étape pour résoudre le problème consiste à l'identifier. Par défaut, le système ne collecte pas de statistiques sur l'utilisation du volume logique. Lorsque vous activez la collecte des statistiques et que vous entrez la commande **lvmstat** pour la première fois, le système affiche des valeurs de compteurs depuis le dernier réamorçage. Ensuite, chaque fois que vous entrez la commande **lvmstat**, le système affiche la différence par rapport à la commande **lvmstat** précédente.

En interprétant le résultat de la commande **lvmstat**, vous pouvez identifier les partitions logiques ayant le trafic le plus élevé. Si des partitions logiques d'un disque sont très utilisées et que vous voulez équilibrer les partitions sur les disques disponibles, vous pouvez utiliser la commande **migratelp** pour transférer ces partitions logiques vers d'autres disques physiques.

Dans l'exemple suivant, la collecte des statistiques est active et la commande **lvmstat** est utilisée de manière répétitive pour collecter des statistiques de base :

```
# lvmstat -v rootvg -e
# lvmstat -v rootvg -C
# lvmstat -v rootvg
```

Le résultat se présente comme suit :

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd8	4	0	16	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00
hd3	0	0	0	0.00
hd9var	0	0	0	0.00
hd2	0	0	0	0.00
hd4	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

La sortie précédente montre que tous les compteurs sont remis à zéro. Dans l'exemple suivant, les données sont d'abord copiées du répertoire **/unix** vers le répertoire **/tmp**. La sortie de la commande **lvmstat** reflète l'activité du groupe de volumes rootvg :

```
# cp -p /unix /tmp
# lvmstat -v rootvg
```

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd3	296	0	6916	0.04
hd8	47	0	188	0.00
hd4	29	0	128	0.00
hd2	16	0	72	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00
hd9var	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

La sortie indique l'activité du volume logique hd3 qui est monté dans le répertoire **/tmp**, sur hd8 qui correspond au volume logique du journal JFS, sur hd4 qui correspond à **/** (racine), sur hd2 qui correspond au répertoire **/usr** et sur hd9var qui correspond au répertoire **/var**. La sortie suivante fournit des informations sur hd3 et hd2 :

```
# lvmstat -l hd3
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
1	1	299	0	6896	0.04
3	1	4	0	52	0.00
2	1	0	0	0	0.00
4	1	0	0	0	0.00

```
# lvmstat -l hd2
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
2	1	9	0	52	0.00
3	1	9	0	36	0.00
7	1	9	0	36	0.00
4	1	4	0	16	0.00
9	1	1	0	4	0.00
14	1	1	0	4	0.00
1	1	0	0	0	0.00

La sortie d'un groupe de volumes fournit un résumé de toute l'activité E/S d'un volume logique. La sortie indique le nombre de demandes E/S (iocnt), le nombre de kilo-octets lus et écrits (Kb_read et Kb_wrtn) et le débit du transfert des données en Kbps. Si vous demandez les informations relatives à un volume logique, vous recevez les mêmes informations pour chaque partition logique. S'il existe des volumes logiques mis en miroir, vous affichez les statistiques de chaque volume mis en miroir. Dans l'exemple de sortie précédent, plusieurs lignes de partitions logiques sans activité ont été omises. La sortie est toujours triée en ordre décroissant dans la colonne iocnt.

La commande **migratelp** utilise comme paramètres le nom du volume logique, le numéro de partition logique (tel qu'il est affiché par la commande **lvmstat**) et un numéro facultatif de copie miroir. Si des informations sont omises, la première copie miroir est utilisée. Vous devez définir le volume physique cible du transfert. En outre, vous pouvez spécifier un numéro de partition physique cible. Si l'opération aboutit, la sortie se présente comme suit :

```
# migratelp hd3/1 hdisk1/109
migratelp: Mirror copy 1 of logical partition 1 of logical volume
hd3 migrated to physical partition 109 of hdisk1.
```

Lorsque vous activez la fonction de zone active sur un volume logique ou un groupe de volumes, vous pouvez définir les rapports et les statistiques, afficher les statistiques, sélectionner les partitions logiques à migrer, définir la partition physique de destination et vérifier les informations avant de valider les modifications. Web-based System Manager vous aide à définir les rapports de zone active et gérer le résultat. Pour plus d'informations sur les instructions d'activation des rapports de zone active, reportez-vous au document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

Mise en œuvre de stratégies de groupe de volumes

Après avoir déterminé les stratégies de groupes de volumes à utiliser, analysez la configuration actuelle en tapant la commande **lspv** sur la ligne de commande. La configuration standard fournit un groupe de volumes unique qui contient plusieurs volumes physiques connectés à la même carte de disque et d'autres matériels de stockage. Dans une configuration standard, plus le nombre de disques constituant un groupe de volumes à quorum est élevé, plus il est aisé de respecter le quorum en cas de défaillance de disque. Dans un groupe sans quorum, le groupe de volumes doit être constitué d'au moins deux disques. Pour mettre en œuvre la modification de votre stratégie de groupe de volumes, procédez comme suit :

1. Utilisez la commande **lspv** pour vérifier les volumes alloués et les volumes physiques libres.
2. Définissez un quorum en ajoutant un ou plusieurs volumes physiques. Pour plus d'informations, reportez-vous à section Ajout de disques lorsque le système reste disponible, Ajout d'un disque fixe sans données à un groupe de volumes existant ou Ajout d'un disque fixe sans données à un nouveau groupe de volumes.
3. Pour convertir un groupe de volumes en groupe de volumes sans quorum, reportez-vous à la section Conversion d'un groupe de volumes en groupe de volumes sans quorum.
4. Reconfigurez le matériel uniquement pour garantir une disponibilité optimale. Pour plus d'informations, reportez-vous au guide de maintenance de votre système.

Chapitre 4. Espace de pagination et mémoire virtuelle

AIX utilise la mémoire virtuelle pour accéder à une plus grande quantité de mémoire que n'en dispose physiquement le système. La gestion des pages de mémoire en RAM ou sur disque est effectuée par VMM (Virtual Memory Manager). Les segments de mémoire virtuelle sont partitionnés en unités appelées *pages*. Un *espace de pagination* est un type de volume logique associé un espace disque qui contient des informations qui résident en mémoire virtuelle, mais qui ne sont actuellement utilisées. Ce volume logique dispose d'un type d'attribut de pagination et s'appelle généralement espace de pagination ou *espace d'échange*. Lorsque la quantité de mémoire RAM libre du système est faible, les programmes ou les données qui n'ont pas été utilisés récemment sont transférés de la mémoire vers l'espace de pagination pour libérer la mémoire.

Les sections suivantes fournissent des informations supplémentaires sur VMM et l'espace de pagination. Pour plus d'informations sur les performance associées aux espaces de pagination, reportez-vous à la section relative aux considérations de performance dans le document *AIX 5L Version 5.2 Performance Management Guide*. Pour plus d'informations sur la gestion des espaces de pagination, reportez-vous à la section Paging Space and Virtual Memory dans le document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Généralités sur VMM (Virtual Memory Manager)

VMM (Virtual Memory Manager) gère les demandes de mémoire du système et de ses applications. Des segments de mémoire virtuelle sont partitionnés en unités appelées *pages*, chaque page étant située dans la mémoire physique réelle (RAM) ou sur disque tant que cela est nécessaire. AIX utilise la mémoire virtuelle pour accéder à une plus grande quantité de mémoire que n'en dispose physiquement le système. La gestion des pages de mémoire en RAM ou sur disque est effectuée par VMM.

Gestion de la mémoire réelle

Dans AIX, les segments de mémoire virtuelle sont partitionnés en unités de 4 096 octets appelés pages. La mémoire réelle est divisée en cadres de page de 4 096 octets. VMM remplit deux fonctions principales :

- Il gère l'allocation des cadres de page.
- Il résout les références aux pages de mémoire virtuelle qui se trouvent en mémoire RAM (stockées dans l'espace de pagination) ou qui n'existent pas encore.

Pour exécuter ces fonctions, VMM gère une *liste libre* de cadres de page disponibles. VMM utilise également un algorithme de remplacement de page pour déterminer les pages de mémoire virtuelle dans la mémoire RAM dont les cadres de page doivent être réaffectés à la liste libre. Cet algorithme tient compte de l'existence des segments persistants par rapport aux segments actifs, de la repagination et des seuils VMM.

Liste libre

VMM gère une liste de cadres de page libres (non alloués) qui est utilisée pour résoudre les erreurs de pages. AIX tente d'utiliser en permanence toute la mémoire RAM, mais conserve une petite quantité de mémoire dans la liste libre. Pour gérer cette petite quantité de pages non allouées, VMM utilise *page outs* et *page steals* pour libérer de l'espace et réaffecter les pages à la liste libre. Les pages de mémoire virtuelle dont les cadres de page ne sont pas réaffectés sont sélectionnés en utilisant l'algorithme de remplacement de page de VMM.

Segments de mémoire persistants et segments de mémoire actifs

AIX utilise deux types de segments de mémoire. Pour comprendre VMM, il est important de comprendre la différence existant entre les segments persistants et les segments actifs. Un *segment persistant* dispose d'un emplacement de stockage permanent sur le disque. Les fichiers qui contiennent des données ou des programmes exécutables sont associés à des segments persistants. Lorsqu'un fichier JFS ou JFS2 est ouvert ou que le système y accède, les données du fichier sont copiés vers la mémoire RAM. Les paramètres VMM contrôlent le moment où les cadres de mémoire physique alloués aux pages persistantes doivent être remplacés et utilisés pour stocker d'autres données.

Les *segments actifs* sont temporaires et existent uniquement lorsqu'ils sont utilisés par un processus. Ces segments n'ont pas d'emplacement de stockage sur disque. La pile des processus et les régions de données sont associées à des segments actifs et à des segments texte partagés dans la bibliothèque. Les pages de segments actifs doivent également se trouver dans des emplacements de stockage sur disque lorsqu'elles ne peuvent pas être conservées dans la mémoire réelle. L'espace de pagination du disque est utilisé à cet effet. Lorsqu'un programme se termine, toutes les pages actives sont replacées immédiatement dans la liste libre.

Segments actifs et espace de pagination

Les pages actives dans la mémoire RAM qui peuvent être modifiées et sorties de la mémoire sont affectées d'une fenêtre d'espace de pagination. L'espace de pagination alloué est utilisé uniquement si la page doit être sortie de la mémoire. Toutefois, une page allouée dans l'espace de pagination ne peut pas être utilisée par une autre page. Elle reste réservée à une page particulière tant que la page existe dans la mémoire virtuelle. Du fait que les pages persistantes sont placées dans le même emplacement d'origine sur le disque, il n'est pas nécessaire d'allouer un espace de pagination aux pages persistantes en mémoire RAM.

VMM utilise deux modes d'allocation d'espace de pagination : *Early* et *Late*. Chaque stratégie d'allocation *Early* réserve un espace de pagination lorsqu'une demande de mémoire pour une page active est effectuée. La stratégie d'allocation *Late* affecte uniquement un espace de pagination lorsque la page active est sortie de la mémoire, ce qui réduit sensiblement les besoins d'espace de pagination du système.

Fonction VMM Memory Load Control Facility

Lorsqu'un processus fait référence à une page de mémoire virtuelle sur disque, car il est sortie de la mémoire ou n'a jamais été lu, la page référencée doit être remise en mémoire, ce qui peut amener une ou plusieurs pages à être sorties de la mémoire si le nombre de cadres de page (libres) est faible. VMM tente de "voler" des cadres de page qui n'ont pas été référencés dernièrement et qui ne sont donc pas susceptibles de l'être dans un avenir proche, en utilisant un algorithme de remplacement de page.

Un remplacement de page qui aboutit conserve les pages de mémoire de tous les processus actifs en cours dans la mémoire RAM alors que les pages de mémoire des processus inactifs sont sorties de la mémoire. Toutefois, lorsque la mémoire RAM est surestimée, il devient difficile de déterminer les pages à sortir de la mémoire, car elles seront vraisemblablement référencées dans un avenir proche par les processus actifs en cours. En conséquence, les pages susceptibles d'être référencées dans un avenir proche sont toujours sorties de la mémoire, puis replacées dans la mémoire lorsqu'elles sont référencées. Lorsque la mémoire RAM est surestimée, la pagination continue en mémoire et hors de la mémoire, appelée *thrashing*, se produit. Lorsqu'un système connaît le *thrashing*, le système passe la majorité de son temps à sortir et entrer les données en mémoire au lieu d'exécuter des instructions utiles, et aucun processus actif ne progresse. VMM utilise un algorithme de contrôle du chargement en mémoire qui détecte quand système est soumis au *thrashing* et qui permet de résoudre le problème.

Généralités sur l'espace de pagination

Un *espace de pagination* est un type de volume logique associé un espace disque qui contient des informations qui résident en mémoire virtuelle, mais qui ne sont pas actuellement utilisées. Ce volume logique dispose d'un type d'attribut de pagination et s'appelle généralement espace de pagination ou *espace d'échange*. Lorsque la quantité de mémoire RAM libre du système est faible, les programmes ou les données qui n'ont pas été utilisées récemment sont transférés de la mémoire vers l'espace de pagination pour libérer la mémoire.

Un autre type d'espace de pagination est accessible via une unité qui utilise un serveur NFS pour le stockage dans l'espace de pagination. Pour qu'un client NFS puisse accéder à cet espace de pagination, il est nécessaire de créer un fichier pour le serveur NFS et de l'exporter vers le client. La taille du fichier représente la taille de l'espace de pagination du client.

La quantité d'espace de pagination nécessaire dépend du type des activités exécutées sur le système. Si l'espace de pagination devient insuffisant, des processus sont perdus et si l'espace de pagination est saturé, le système panique. Lorsque l'espace de pagination est insuffisant, définissez un espace de pagination supplémentaire. Pour plus d'informations, reportez-vous la section Paging Space Troubleshooting du document *AIX 5L Version 5.2 System Management Guide: Système d'exploitation et unités*

L'espace de pagination des volumes logiques est défini en créant un nouveau volume d'espace de pagination ou en augmentant la taille des volumes logiques d'espace de pagination existants. Pour augmenter la taille d'un espace de pagination NFS, vous devez augmenter la taille du fichier sur le serveur en exécutant les actions appropriées sur ce dernier.

L'espace de pagination total du système correspond à la somme des tailles de tous les volumes logiques d'espace de pagination.

Les sections suivantes contiennent des informations sur les espace de pagination :

- Stratégies d'allocation d'espace de pagination
- Taille par défaut d'espace de pagination
- Fichier d'espace de pagination, commandes et options

Stratégies d'allocation d'espace de pagination

AIX utilise deux modes d'allocation d'espace de pagination. La variable d'environnement **PSALLOC** détermine l'algorithme d'allocation d'espace de pagination à utiliser : Late ou Early. Le mode par défaut est Late. Vous pouvez activer le mode Early en changeant la variable d'environnement **PSALLOC**, mais il existe plusieurs facteurs à prendre en compte avant d'effectuer cette modification. Lorsque vous utilisez l'algorithme d'allocation Early, dans le pire des cas, vous pouvez bloquer le système en utilisant tout l'espace de pagination disponible.

Comparaison du mode d'allocation d'espace de pagination Late et du mode d'allocation d'espace de pagination Early

Le système d'exploitation utilise la variable d'environnement **PSALLOC** pour déterminer le mécanisme utilisé pour l'allocation de mémoire et l'espace de pagination. Si la variable d'environnement **PSALLOC** n'est pas définie ou est affectée de la valeur `null` ou d'une valeur autre que `early`, le système utilise l'algorithme d'allocation par défaut `late`.

L'algorithme d'allocation `late` aide à utiliser efficacement les ressources disque et prend en charge les applications qui préfèrent un algorithme d'allocation sparse pour gérer les ressources. Cet algorithme ne réserve pas d'espace de pagination lorsqu'une demande de mémoire est effectuée. Il améliore la demande et affecte un espace de pagination lorsque les pages sont utilisées. Certains programmes allouent de grande quantité de mémoire virtuelle et utilisent uniquement une portion de la mémoire. Les applications techniques, par exemple, utilisent des vecteurs ou des matrices sparse comme structures de données. L'algorithme d'allocation `Late` est également plus efficace pour un noyau de demande de pages temps réel, tels que celui du système d'exploitation.

Toutefois, cet espace de pagination peut ne jamais être utilisé, notamment sur les systèmes qui disposent d'une grande quantité de mémoire réelle où la pagination est rare. En conséquence, l'algorithme `Late` retarde davantage l'allocation d'espace de pagination jusqu'à ce qu'il soit nécessaire de lire la page, ce qui permet de ne pas perdre d'espace de pagination. Toutefois, cela ne se traduit par une surestimation de l'espace de pagination.

Il est possible de surestimer les ressources avec l'algorithme `Late` pour l'allocation d'espace de pagination. Dans ce cas, lorsqu'un processus obtient la ressource avant une autre, un incident se produit. Le système d'exploitation tente d'éviter l'échec du système en arrêtant les processus affectés par la surestimation de ressources. Le signal **SIGDANGER** est envoyé pour indiquer aux processus que la quantité d'espace de pagination libre est faible. Si l'espace de pagination atteint un niveau encore plus critique, les processus sélectionnés qui ne reçoivent pas le signal **SIGDANGER** reçoivent le signal **SIGKILL**.

Utilisez la variable d'environnement **PSALLOC** pour utiliser l'algorithme d'allocation `Early` qui alloue l'espace de pagination au processus en cours lorsque de la mémoire est demandée. Si l'espace de pagination est insuffisant lors de la demande, l'allocation `Early` ne répond pas à la demande de mémoire.

Si vous affectez à la variable d'environnement **PSALLOC** la valeur `early`, chaque programme démarré dans cet environnement, mais qui n'inclue pas de processus en cours, est exécuté dans l'environnement d'allocation `Early`. Dans l'environnement d'allocation `Early`, les interfaces, telles que la sous-routine **malloc** et la sous-routine **brk**, échouent si un espace de pagination suffisant ne peut pas être réservé lorsque la demande est effectuée.

Les processus exécutés dans l'environnement d'allocation `Early` ne reçoivent pas le signal **SIGKILL** si l'espace de pagination est insuffisant.

Vous pouvez affecter à la variable d'environnement **PSALLOC** la valeur `early` de différentes manières en fonction de l'étendue de la modification. (Voir *Setting the PSALLOC Environment Variable for Early Allocation Mode* dans le document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.)

Les sous-routines d'interface d'allocation de mémoire suivantes sont affectées par le passage à l'environnement d'allocation `Early` :

malloc

free

calloc

realloc

brk

sbrk

shmget

shmctl

Considérations relatives au mode d'allocation Early

L'algorithme d'allocation Early permet de fournir l'espace de pagination nécessaire à une demande d'allocation de mémoire. Ainsi, l'allocation de l'espace de pagination approprié sur le disque système est important pour garantir l'efficacité des opérations. Lorsqu'un espace de pagination tombe en dessous d'un seuil, de nouveaux processus ne peuvent pas être démarrés et les processus en cours peuvent ne pas recevoir de mémoire supplémentaire. Tout processus exécuté dans le mode d'allocation par défaut Late devient très vulnérable au signal **SIGKILL**. En outre, du fait que le noyau du système d'exploitation nécessite parfois une allocation de mémoire, il est possible de bloquer le système en utilisant tout l'espace de pagination disponible.

Avant d'utiliser le mode d'allocation Early sur l'ensemble du système, il est important de définir un espace de pagination approprié pour le système. L'espace de pagination nécessaire au mode d'allocation Early est presque plus grand que l'espace de pagination nécessaire au mode d'allocation par défaut Late. La quantité d'espace de pagination à définir dépend de la manière dont le système est utilisé et des programmes exécutés. Un bon point de départ pour déterminer cet élément pour le système consiste à définir un espace de pagination quatre fois plus grand que la mémoire physique.

Certaines applications peuvent utiliser de grandes quantités d'espace de pagination si elles sont exécutées en mode d'allocation Early. Le serveur AIXwindows nécessite actuellement plus de 250 Mo d'espace de pagination lorsque l'application est exécutée en mode d'allocation Early. L'espace de pagination nécessaire par une application dépend de la manière dont l'application est écrite et de son exécution.

Toutes les commandes et sous-routines qui utilisent un espace de pagination et de la mémoire incluent l'espace de pagination alloué en mode d'allocation Early. La commande **lsps** utilise l'option **-s** pour afficher l'allocation d'espace de pagination total, y compris l'espace de pagination alloué en mode d'allocation Early.

Interface de programmation

L'interface de programmation qui contrôle le mode d'allocation d'espace de pagination utilise la variable d'environnement **PSALLOC**. Pour garantir qu'une application s'exécute toujours dans le mode approprié (avec ou sans allocation d'espace de pagination Early), procédez comme suit :

1. Utilisez la sous-routine **getenv** pour identifier l'état en cours de la variable d'environnement **PSALLOC**.
2. Si la variable d'environnement **PSALLOC** ne correspond pas à la valeur nécessaire à l'application, utilisez la sous-routine **setenv** pour changer sa valeur. Du fait que seule la sous-routine **execve** examine l'état de la variable d'environnement **PSALLOC**, appelez la sous-routine **execve** avec les mêmes paramètres et l'environnement reçu par l'application. Lorsque l'application réexamine l'état de la variable d'environnement **PSALLOC** et trouve la valeur correcte, l'exécution de l'application se poursuit normalement.
3. Si la sous-routine **getenv** détecte que l'état en cours de la variable d'environnement **PSALLOC** est correcte, aucune modification n'est nécessaire. L'exécution de l'application se poursuit normalement.

Taille par défaut d'espace de pagination

La taille d'espace de pagination par défaut est déterminée lors de la personnalisation de l'installation AIX en fonction des standards suivants :

- L'espace de pagination ne peut utiliser moins de 16 Mo, sauf pour hd6 qui ne peut utiliser moins de 64 Mo dans AIX 4.3 et les versions supérieures.
- L'espace de pagination ne peut utiliser plus de 20 % de l'espace disque total.
- Si la mémoire réelle est inférieure à 256 Mo, l'espace de pagination est égale à deux fois la mémoire réelle.
- Si la mémoire réelle est supérieure ou égale à 256 Mo, l'espace de pagination est de 512 Mo.

Fichier d'espace de pagination, commandes et options

Le fichier `/etc/swapspaces` définit les unités d'espace de pagination activées par la commande **swapon -a**. Un espace de pagination est ajouté au fichier lorsqu'il est créé par la commande **mkps -a**, supprimé du fichier lorsqu'il est supprimé par la commande **rmps** et ajouté ou supprimé par la commande **chps -a**. Si l'espace de pagination est trop grand, vous pouvez supprimer des partitions logiques de l'espace de pagination sans avoir à réamorcer le système, en utilisant la commande **chps -d**.

Les commandes suivantes permettent de gérer l'espace de pagination :

chps	Change les attributs de l'espace de pagination.
lsps	Change les caractéristiques de l'espace de pagination.
mkps	Ajoute un espace de pagination. La commande mkps utilise la commande mklv avec des options spécifiques lors de la création d'un volume logique d'espace de pagination. Pour créer des espaces de pagination NFS, la commande mkps utilise la commande mkdev avec des options différentes. Pour les espaces de pagination NFS, la commande mkps doit utiliser le nom d'hôte du serveur NFS et le chemin du fichier exporté du serveur.
rmps	Supprime un espace de pagination inactif.
swapoff	Désactive un ou plusieurs espaces de pagination sans réamorcer le système. Les informations dans l'espace de pagination sont transférées vers d'autres zones d'espace de pagination. L'espace de pagination désactivé peut être ensuite supprimé avec la commande rmps .
swapon	Active un espace de pagination. La commande swapon est utilisée lors de l'initialisation anticipée du système pour activer l'unité d'espace de pagination initiale. Lors d'une phase d'initialisation ultérieure, lorsque d'autres commandes deviennent disponibles, la commande swapon est utilisée pour activer d'autres espaces de pagination pour que l'activité de pagination ait lieu sur plusieurs unités.

Certaines des options suivantes sont nécessaires à tous les espaces de pagination de volume logique:

- Paging type
- No bad-block relocation
- No mirroring

Les options suivantes permettent d'optimiser les performances de la pagination avec un volume logique :

- Allocate in the middle of the disk to reduce disk arm travel
- Use multiple paging spaces, each allocated from a separate physical volume.

Chapitre 5 Systèmes de fichiers

Un *système de fichiers* est une structure hiérarchique (arborescence) de fichiers et de répertoires. Ce type de structure ressemble à une arborescence inversé avec les racines en haut de l'arbre et les branches vers le bas. Cette arborescence utilise des répertoires pour organiser les données et les programmes dans des groupes, ce qui permet de gérer plusieurs répertoires et fichiers simultanément. Pour plus d'informations sur la structure du système de fichiers, reportez-vous à Organisation et contenu de l'arborescence de fichiers.

Un système de fichiers réside dans un seul volume logique. Chaque fichier et chaque répertoire appartiennent à un système de fichiers dans un volume logique. Du fait de sa structure, certaines tâches sont exécutées plus efficacement sur un système de fichiers que sur chaque répertoire d'un système de fichiers. Vous pouvez, par exemple, sauvegarder, déplacer ou sécuriser l'ensemble d'un système de fichiers. Vous pouvez créer une image dans le temps d'un système de fichiers JFS ou d'un système de fichiers JFS2, appelée *cliché* (AIX 5.2 et versions supérieures). La commande **mkfs** (make file system) ou System Management Interface Tool (commande **smit**) crée un système de fichiers dans un volume logique. Pour plus d'informations sur la gestion des systèmes de fichiers, voir Tâches de gestion des systèmes de fichiers.

Pour être accessible, un système de fichiers doit être monté sur un point de montage de répertoire. Lorsque plusieurs systèmes de fichiers sont montés, une structure de répertoires, qui représente l'image d'un système de fichiers, est créée. Il s'agit d'une structure hiérarchique avec une seule racine. Cette structure inclut les systèmes de fichiers de base et les systèmes de fichiers que vous créez. Vous pouvez accéder aux systèmes de fichiers locaux et distants à l'aide de la commande **mount**. La commande vous permet d'accéder au système de fichiers en lecture et en écriture depuis votre système. Le montage ou le démontage d'un système de fichiers nécessite généralement d'appartenir au groupe système. Les systèmes de fichiers peuvent être montés automatiquement s'ils sont définis dans le fichier **/etc/filesystems**. Vous pouvez démonter un système de fichiers local ou distant à l'aide de la commande **umount** si aucun utilisateur ou processus n'utilise le système de fichiers. Pour plus d'informations sur le montage d'un système de fichiers, voir Généralités sur le montage.

Plusieurs types de systèmes de fichiers sont pris en charge par AIX 5.2, notamment JFS (Journaled File System) et JFS2 (Enhanced Journaled File System). Pour plus d'informations sur les types de fichiers et les caractéristiques de chaque type, voir Types de systèmes de fichiers.

Organisation et contenu de l'arborescence de fichiers

L'arborescence de fichiers organise les fichiers dans des répertoires qui contiennent des informations similaires. Cette organisation facilite le montage à distance des répertoires et des fichiers. Les administrateurs système peuvent utiliser ces répertoires comme composants pour créer une arborescence de fichiers unique pour chaque client qui monte des répertoires individuels depuis un ou plusieurs serveurs. Monter les fichiers et les répertoires à distance au lieu de stocker localement les informations offre les avantages suivants :

- Economise d'espace disque
- Facilite l'administration centralisée du système.
- Fournit un environnement plus sécurisé.

L'arborescence de fichiers a les caractéristiques suivantes :

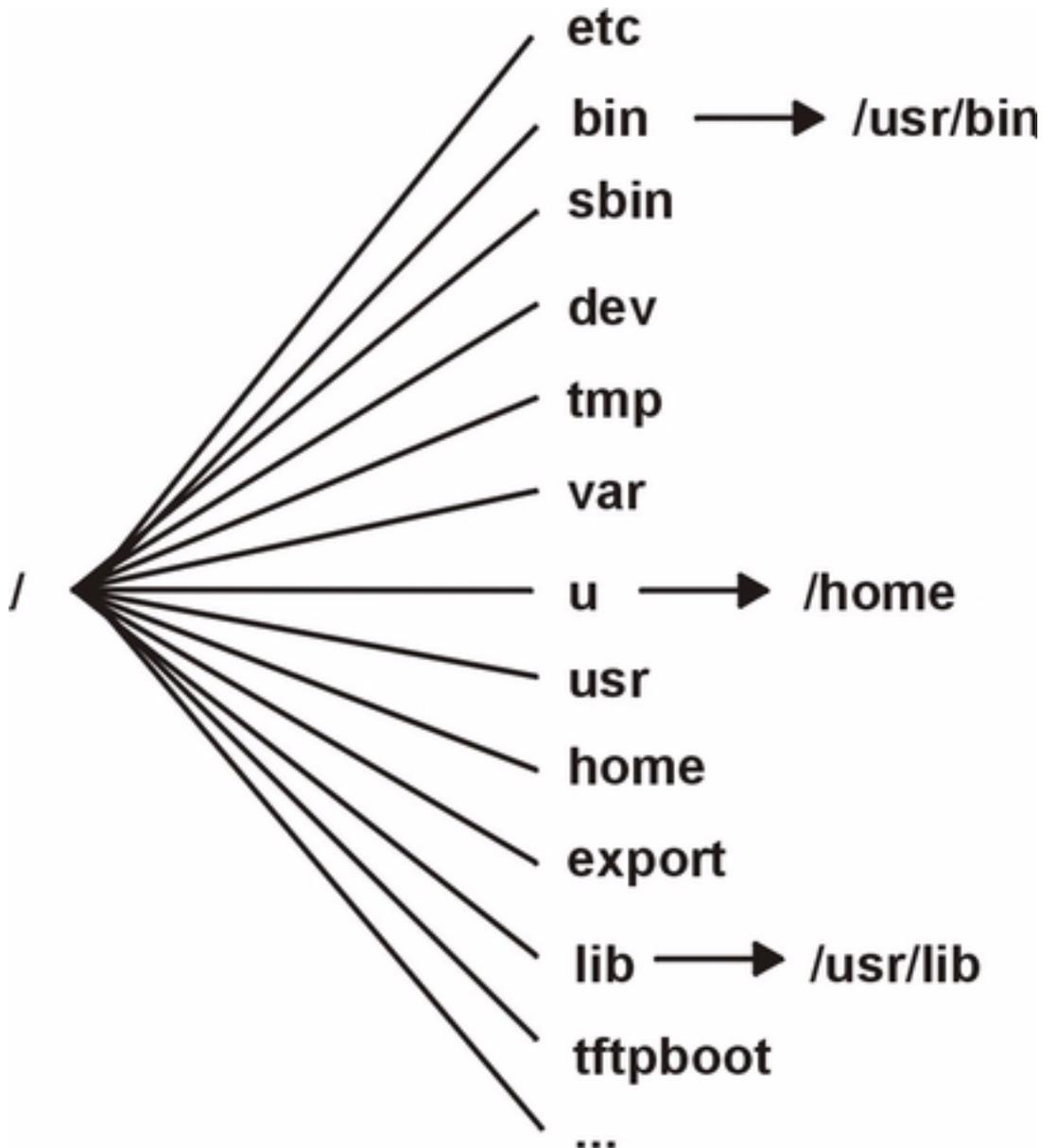
- Les fichiers qui peuvent être partagés par les machines ayant une architecture matérielle identique se trouvent dans le système de fichiers **/usr**.
- Les fichiers variables par client, tels que les fichiers spool et les fichiers de messagerie, se trouvent dans le système de fichiers **/var**.
- Les fichiers texte partageables indépendants de l'architecture, tels que des pages de manuel, se trouvent dans le répertoire **/usr/share**.
- Le système de fichiers **/** (racine) contient les fichiers et les répertoires importants du système d'exploitation. Par exemple, il contient un répertoire d'unités, les programmes utilisés pour démarrer le système et les points de montage sur lesquels les systèmes de fichiers peuvent être montés sur le système de fichiers racine.
- Le système de fichiers **/home** est le point de montage des répertoires locaux de l'utilisateur.
- Pour les serveurs, le répertoire **/export** contient des fichiers d'espace de pagination, des systèmes de fichiers (non partagés) par client, un cliché, un répertoire local, des répertoires **/usr/share** pour les clients sans disque et des répertoires exportés **/usr**.

Description du système de fichiers racine

Le système de fichiers racine est le premier système de fichiers de l'arborescence de fichiers hiérarchique. Il contient les fichiers et les répertoires importants du système d'exploitation, notamment le répertoire des unités et les programmes de démarrage du système. Le système de fichiers racine contient aussi des points de montage de systèmes de fichiers pour les connecter à la hiérarchie des systèmes de fichiers racine.

L'illustration ci-dessous montre les nombreux sous-répertoire du système de fichiers racine.

Figure 7. Système de fichiers racine Cette illustration montre le système de fichiers racine et ses sous-répertoires. Les points de sous-répertoires **/bin** du répertoire **/usr/bin**. Les points de sous-répertoires **/lib** du répertoire **/usr/lib**. Les points de sous-répertoires **/u** du répertoire **/home**.



La liste suivante fournit des informations sur le contenu des sous-répertoires du système de fichiers / (racine).

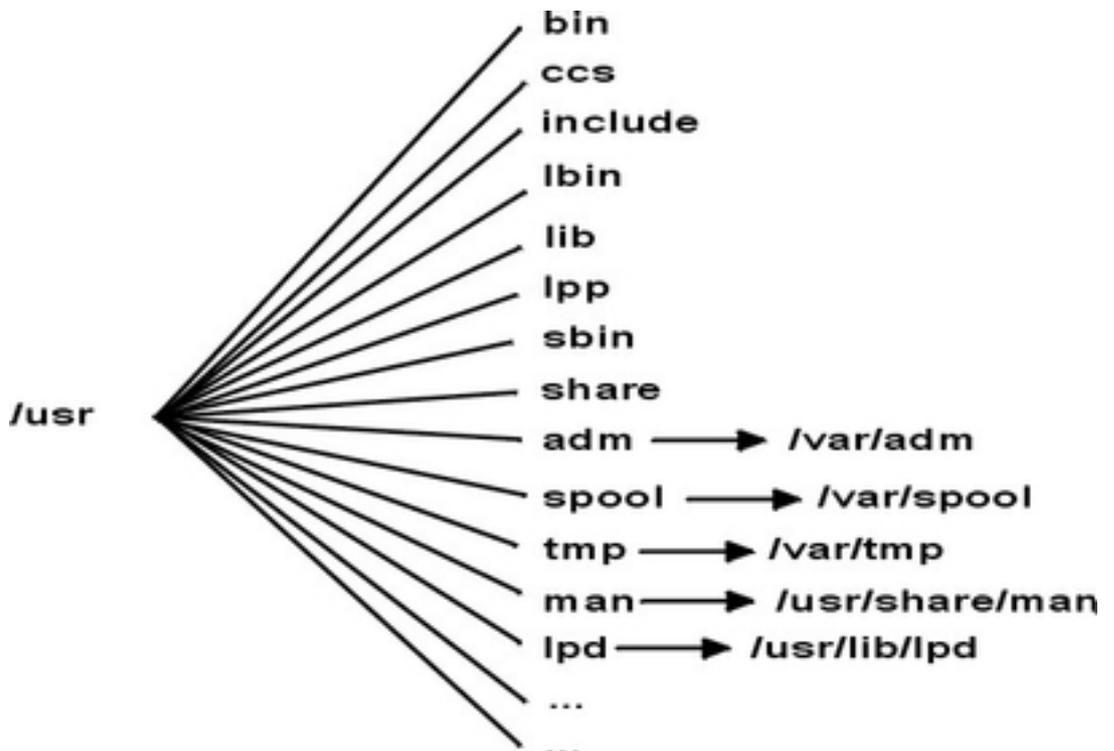
/etc	<p>Contient les fichiers de configuration qui varient en fonction de la machine. Exemples :</p> <ul style="list-style-type: none"> • /etc/hosts • /etc/passwd <p>Le répertoire /etc contient les fichiers généralement utilisés pour l'administration de système. La plupart des commande qui se trouvaient auparavant dans le répertoire /etc figurent désormais dans le répertoire /usr/sbin. Toutefois, pour des raisons de compatibilité, le répertoire /usr/sbin contient des liens symboliques vers des emplacements de fichiers exécutables. Exemples :</p> <ul style="list-style-type: none"> • /etc/chown est un lien symbolique vers /usr/bin/chown. • /etc/exportvg est un lien symbolique vers /usr/sbin/exportvg.
/bin	<p>Lien symbolique vers le répertoire /usr/bin . Dans les systèmes de fichiers UNIX antérieurs, la répertoire /bin contenait des commandes utilisateur qui résident maintenant dans le répertoire /usr/bin.</p>
/sbin	<p>Contient les fichiers nécessaires au démarrage de la machine et au montage du système de fichiers /usr. La plupart des commandes utilisées pendant l'amorçage proviennent du système de fichiers du disque RAM de l'image d'amorçage et en conséquence un nombre limité de commandes se trouve ans le répertoire /sbin.</p>
/dev	<p>Contient les nœuds d'unités des fichiers spéciaux des unités locales. Le répertoire /dev contient des fichiers spéciaux pour les unités de bande, les imprimantes, les partitions de disque et les terminaux.</p>
/tmp	<p>Sert de point de montage d'un système de fichiers qui contient des fichiers temporaires générés par le système. Le système de fichiers /tmp est un répertoire vide.</p>
/var	<p>Sert de point de montage de fichiers qui varient fonction de la machine. Le système de fichiers /var est configuré comme système de fichiers étant donné que la taille des fichiers qu'il contient tend à augmenter. Voir Description du système de fichiers /var , page 5-8 pour plus d'informations.</p>
/u	<p>Lien symbolique vers le répertoire /home.</p>
/usr	<p>Contient des fichiers qui ne changent pas (exécutables et documentation ASCII, par exemple) et qui peuvent être partagés par des machines.</p> <p>Les machines autonomes montent la racine d'un système de fichiers local séparé sur le répertoire /usr. Les machines sans disque et les machines dont les ressources disque sont limitées montent un répertoire depuis un serveur distant sur le système de fichiers /usr. Pour plus d'informations sur l'arborescence de fichiers montée sur le répertoire /usr, reportez-vous à la section Description du système de fichiers /usr , page 5-5.</p>
/home	<p>Sert de point de montage pour un système de fichiers qui contient les répertoires locaux de l'utilisateur. Le système de fichiers /home contient des fichiers par utilisateur et des répertoires.</p> <p>Sur une machine autonome, le répertoire /home se trouve dans un système de fichiers différent dont la racine est montée sur le système de fichiers racine du répertoire /home. Dans un réseau, un serveur peut contenir des fichiers utilisateur accessibles depuis plusieurs machines. Dans ce cas, la copie serveur du répertoire /home est montée à distance sur un système de fichiers local /home.</p>

/export	Contient les répertoires et les fichiers, destinés à des clients distants, sur un serveur. Pour plus d'informations sur l'arborescence de fichiers qui réside dans le répertoire /usr , reportez-vous à la section Description du système du répertoire /export , page 5-9.
/lib	Lien symbolique vers le répertoire /usr/lib . Voir Description du système de fichiers /usr , page 5-5 pour plus d'informations.
/tftpboot	Contient les images et les informations d'amorçage des clients sans disque.

Description du système de fichiers **/usr**

Le système de fichiers **/usr** contient des fichiers exécutables qui peuvent être partagés entre les machines. Les principaux sous-répertoires du répertoire **/usr** figurent dans l'illustration ci-dessous.

Figure 8. Système de fichiers **/usr** Cette illustration montre les principaux sous-répertoires du répertoire **/usr** qui contient : les sous-répertoires **/bin**, **/ccs**, **/lib**, **/lpp**, **/adm** et **/var/adm** et les sous répertoires **/man** et **/usr/share/man**.



Sur une machine autonome, le système de fichiers **/usr** est un système de fichiers distinct (dans le volume logique **/dev/hd2**). Sur une machine sans disque ou une machine dont les ressources disque sont limitées, un répertoire d'un serveur distant est monté avec l'autorisation Lecture seule sur le système de fichiers **/usr**. Le système de fichiers **/usr** contient des commandes, des bibliothèques et des données accessibles en lecture seule.

Sauf pour le contenu du répertoire **/usr/share**, les fichiers et les répertoires du système de fichiers **/usr** peuvent être partagés par toutes les machines ayant la même architecture matérielle.

Le système de fichiers **/usr** contient les répertoires suivants :

/usr/bin	Contient des commandes ordinaires et des scripts shell. Par exemple, le répertoire /usr/bin contient les commandes ls , cat et mkdir .
/usr/ccs	Contient des fichiers binaires de développement de module non modulaires.
/usr/include	Contient include ou des fichiers d'en-tête.
/usr/lbin	Contient des fichiers exécutables qui sont nécessaires à des commandes.
/usr/lib	Contient des bibliothèques dépendantes de l'architecture dont les noms ont le format lib*.a . Le répertoire /lib du répertoire / (racine) est un lien symbolique vers le répertoire /usr/lib . En conséquence, tous les fichiers qui se trouvaient dans le répertoire /lib figurent désormais dans le répertoire /usr/lib . Ces fichiers incluent des fichiers ne correspondant pas à des bibliothèques, à des fins de compatibilité.
/usr/lpp	Contient des produits installés optionnels.
/usr/sbin	Contient des utilitaires utilisés pour l'administration de système, tels que les commandes SMIT (System Management Interface Tool). La plupart des commandes qui se trouvaient auparavant dans le répertoire /etc figurent désormais dans le répertoire /usr/sbin .
/usr/share	Contient des fichiers qui peuvent être partagés par des machines ayant une architecture différente. Voir description du répertoire /usr/share pour plus d'informations.

Lien symbolique vers le répertoire **/var**.

/usr/adm	Lien symbolique vers le répertoire /usr/adm .
/usr/mail	Lien symbolique vers le répertoire /var/spool/main .
/usr/news	Lien symbolique vers le répertoire /var/news .
/usr/preserve	Lien symbolique vers le répertoire /var/preserve .
/usr/spool	Lien symbolique vers le répertoire /var/spool .
/usr/tmp	Lien symbolique vers le répertoire /var/tmp , car le répertoire /usr est potentiellement partagé par un grand nombre de nœuds et qu'il est accessible en lecture seule.

Liens symbolique vers les répertoires **/usr/share** et **/usr/lib**

/usr/dict	Lien symbolique vers le répertoire /usr/share/dict .
/usr/man	Lien symbolique vers le répertoire /usr/share/man .
/usr/lpd	Lien symbolique vers le répertoire /usr/lib/lpd .

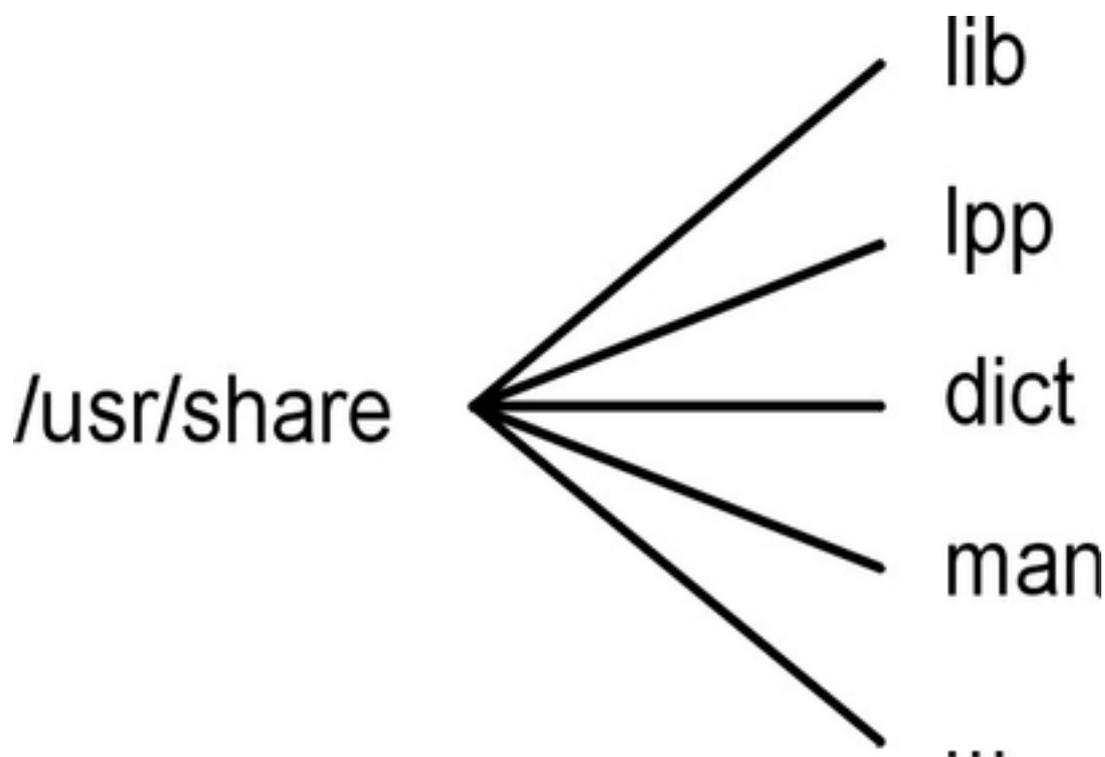
Description du système de fichiers /usr/share

Le répertoire **/usr/share** contient des fichiers texte partageables indépendants de l'architecture. Le contenu de ce répertoire peut être partagé par toutes les machines, quelle que soit l'architecture matérielle.

Dans un environnement architectural mixte, le client sans disque type monte un répertoire serveur sur son propre répertoire **/usr** et monte un répertoire différent sur le répertoire **/usr/share**. Les fichiers sous le répertoire **/usr/share** se trouvent dans un ou plusieurs modules installables séparés. Ainsi, les autres parties du répertoire **/usr** dont dépend un nœud localement peuvent être installées sur le nœud tout en utilisant un serveur pour fournir le répertoire **/usr/share**.

Certains fichiers du répertoire **/usr/share** fournissent les répertoires et les fichiers indiqués dans l'illustration ci-dessous.

Figure 9. Répertoire /usr/share L'illustration montre divers répertoires du répertoire **/usr/share**, notamment **/lib**, **/lpp**, **/dict** et **/man**.



Le répertoire **/usr/share** contient :

/usr/share/man	Contient les pages du manuel si elles ont été chargées.
/usr/share/dict	Contient le dictionnaire orthographique et ses index.
/usr/share/lib	Contient des fichiers de données dépendants de l'architecture, notamment terminfo , learn , tmac , me et macros
/usr/share/lpp	Contient des données et des informations sur les produits installables optionnels du système.

Description du système de fichiers /var

Attention: La taille du système de fichiers /var tend à augmenter, car il contient des sous-répertoires et des fichiers de données utilisés par les applications courantes telles que comptabilité, messagerie et spouleur d'impression. Si des applications sur système utilisent très souvent le système de fichiers /var, exécutez régulièrement la commande `skulker` ou définissez une taille de système de fichiers /var supérieure à la valeur par défaut 4 Mo.

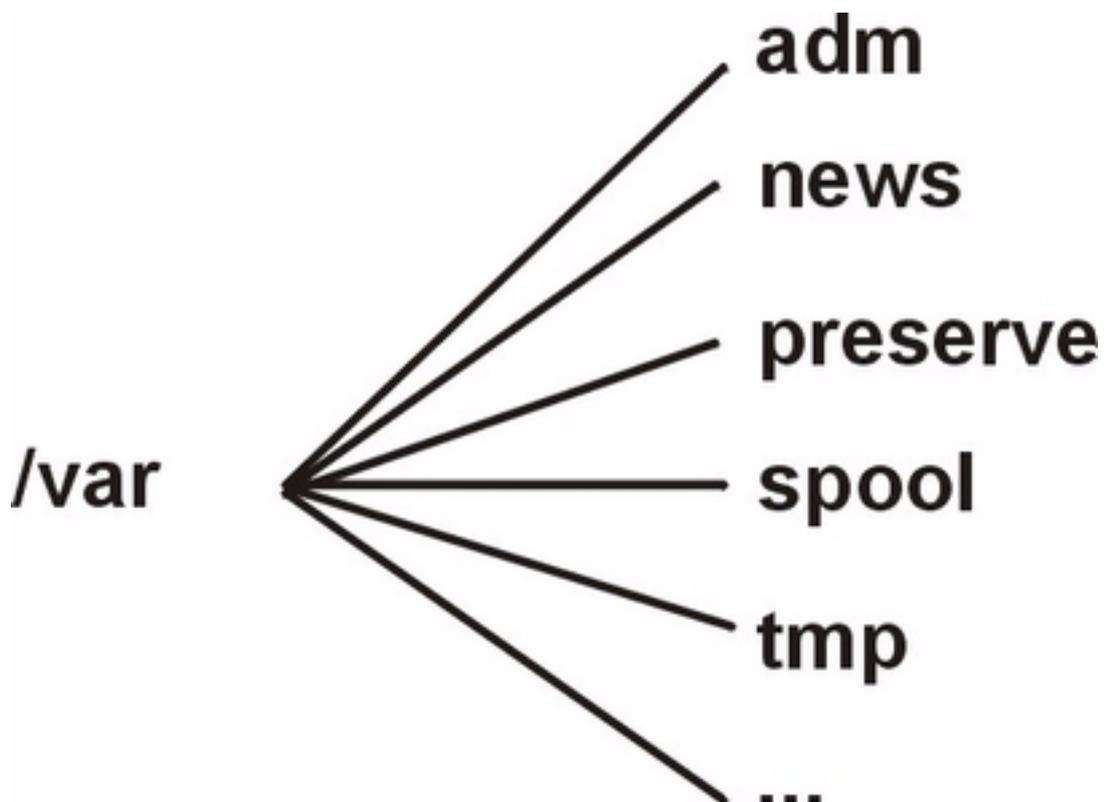
Les fichiers spécifiques /var qui garantissent un contrôle régulier sont `/var/adm/wtmp` et `/var/adm/ras/errlog`.

Autres fichiers /var à contrôler :

<code>/var/adm/ras/trcfile</code>	Si la fonction de trace est active
<code>/var/tmp/snmpd.log</code>	Si la commande <code>snmpd</code> est exécutée sur le système

Le graphique du répertoire /var montre des répertoires du système de fichiers /var.

Figure 10. Répertoire /var L'illustration montre les principaux répertoires du répertoire /var, notamment /adm, /news, /preserve, /spool et /tmp.



<code>/var/adm</code>	Contient les fichiers de journalisation et de comptabilité.
<code>/var/news</code>	Contient des informations récentes sur le système.
<code>/var/preserve</code>	Contient les données protégées des sessions d'édition interrompues. Similaires au répertoire <code>/usr/preserve</code> des versions précédentes.

/var/spool	Contient les fichiers traités par des programmes tels que de messagerie électronique. Similaire au répertoire /usr/spool des versions précédentes.
/var/tmp	Contient des fichiers temporaires. Similaire au répertoire /usr/tmp des versions précédentes. Le répertoire /usr/tmp est désormais un lien symbolique vers /var/tmp .

Description du répertoire **/export**

Le répertoire **/export** contient des fichiers de serveur exportés vers les clients, tels que des machines sans disque, des machines sans données ou des machines limitées en ressource disque. Un serveur peut exporter plusieurs types d'espaces disque, notamment des modules de programmes exécutables, un espace de pagination pour les clients sans disque et des systèmes de fichiers racines pour les clients sans disques ou dont les ressources en disques sont limitées. L'emplacement standard d'un tel espace disque dans l'arborescence de fichiers est le répertoire **/export**. Certains sous-répertoires du répertoire **/export** figurent dans la liste ci-dessous :

/exec	Contient des répertoires sur lesquels les clients sans disque montent leurs systèmes de fichiers /usr .
/swap	Contient les fichiers de pagination à distance des clients sans disque.
/share	Contient des répertoires sur lesquels les clients sans disque montent leur répertoire /usr .
/root	Contient des répertoires sur lesquels les clients sans disque montent leur système de fichiers / (racine) .
/dump	Contient les répertoires des fichiers clichés distants des clients sans disque.
/home	Contient des répertoires sur lesquels les clients sans disque montent leur système de fichiers / home .

Le répertoire **/export** est l'emplacement par défaut des ressources client pour les commandes sans disque. Le répertoire **/export** est le seul emplacement des ressources client sur le serveur. Du fait que les clients montent ces ressources dans leur propre arborescence de fichiers, ces ressources apparaissent pour les client dans des emplacement normaux dans une arborescence de fichiers. Les principaux sous-répertoires du répertoire **/export** et leurs points de montage correspondant dans une arborescence de fichiers incluent :

/export/root	Ce répertoire est monté sur le système de fichiers racine (/) du client. Les répertoires racines des clients se trouvent par défaut dans le répertoire /export/root et portent le nom d'hôte du client.
/export/exec	Appelé également répertoire SPOT (Shared Product Object Tree). Ce répertoire est monté sur le système de fichiers client /usr . Les répertoires SPOT sont des versions du système de fichiers /usr stocké dans le répertoire /export/exec et leurs noms reflètent leur niveau de version. Le nom par défaut est RISCAIX .
/export/share	Ce répertoire est monté sur le répertoire client /usr/share . Ce répertoire contient des données qui peuvent être partagées par un grand nombre d'architectures. L'emplacement par défaut est /export/share/AIX/usr/share .
/export/home	Ce répertoire est monté sur le système de fichiers racine /home (/) du client. Il contient des répertoires utilisateur groupés par nom d'hôte de client. L'emplacement par défaut des répertoires racines des clients est /export/home .

- `/export/swap` Appelé également répertoire de pagination. Dans un système autonome ou sans données, la pagination fournit un disque local pour les clients sans disque. Ce service est fourni par un fichier sur un serveur. Le nom de ce fichier provient du nom d'hôte du client. Le fichier se trouve par défaut dans le répertoire **`/export/swap`**.
- `/export/dump` Les systèmes autonomes utilisent un disque local comme unité de cliché. Les clients sans disque utilisent un fichier sur un serveur. Ce fichier réside dans un répertoire dont le nom provient du nom d'hôte du client. Le fichier se trouve par défaut dans le répertoire **`/export/dump`**.
- `microcode` Ce répertoire contient le microcode des unités physiques. L'emplacement par défaut est **`/export/exec/RISCAIX/usr/lib/microcode`**.

Tâches de gestion des systèmes de fichiers

Un système de fichiers est une structure de répertoires complète qui est constituée d'un répertoire racine et de répertoires et de fichiers stockés sous le répertoire racine. Les systèmes de fichiers se trouvent dans un seul volume logique. Certaines des tâches de gestion les plus importantes concernent les systèmes de fichiers, notamment :

- Allocation d'espace aux systèmes de fichiers dans les volume logiques
- Création de systèmes de fichiers
- Rendre l'espace de système de fichiers disponibles aux utilisateurs du système
- Contrôle de l'utilisation de l'espace de système de fichiers
- Sauvegarde des systèmes de fichiers pour la protection contre la perte de données en cas d'incident
- Création d'un cliché pour capturer une image cohérente de bloc d'un système de fichiers à un moment donné
- Maintien des systèmes de fichiers dans un état cohérent.

La liste suivante contient les commandes de gestion de systèmes de fichiers :

backup	Exécute une sauvegarde complète ou incrémentielle d'un système de fichiers.
chfs -a splitcopy	Crée une sauvegarde en ligne d'un système de fichiers JFS monté.
dd	Copie les données directement d'une unité vers une autre pour créer des sauvegardes de systèmes de fichiers.
df	Indique la quantité d'espace utilisé et l'espace libre dans un système de fichiers
fsck	Vérifie les systèmes de fichiers et répare les incohérences.
mkfs	Crée un système de fichiers d'une taille donnée dans un volume logique défini.
mount	Connecte un système de fichiers à une structure de nom de système pour pouvoir accéder aux fichiers et aux répertoires du système de fichiers.
restore	Restaure des fichiers depuis une sauvegarde.
snapshot	Crée un cliché d'un système de fichiers JFS2.
umount	Supprime un système de fichiers d'une structure de nom de système pour rendre les fichiers et les répertoires du système de fichiers inaccessibles.

Commandes de système de fichier

Il existe des commandes associées à tous les types de systèmes de fichiers. Le fichier **/etc/filesystems** contrôle la liste des systèmes de fichiers que peuvent manipuler les commandes suivantes :

chfs	Change les caractéristiques d'un système de fichiers.
crfs	Ajoute un système de fichiers.
lsfs	Affiche les caractéristiques d'un système de fichiers.
rmfs	Supprime un système de fichiers.
mount	Rend accessible un système de fichiers

Quatre commandes sont disponibles pour les types de systèmes de fichiers virtuels. Le fichier **/etc/vfs** contient les informations sur les types de systèmes de fichiers que les commandes suivantes manipulent :

chvfs	Change les caractéristiques d'un type de système de fichiers.
crvfs	Ajoute un nouveau type de système de fichiers.
lsvfs	Indique les caractéristiques d'un type de système de fichiers.
rmvfs	Supprime un type de système de fichiers.

Montage : généralités

Le *montage* met les systèmes de fichiers, les fichiers, les répertoires, les unités et les fichiers spéciaux à disposition de l'utilisateur à un emplacement spécifique. Il représente l'unique moyen de donner l'accès à un système de fichiers. Par le biais de la commande **mount**, le système d'exploitation reçoit l'instruction d'associer un système de fichiers au répertoire spécifié.

Pour monter un fichier ou un répertoire, vous devez posséder les droits d'accès à ce fichier ou à ce répertoire, et avoir l'autorisation d'écriture au niveau du point de montage. Les membres du groupe système également peuvent procéder à des montages d'unités (dans lesquels les systèmes de fichiers ou les unités sont montés sur les répertoires), ainsi qu'aux montages décrits dans le fichier **/etc/filesystems** les montages décrits plus loin. L'utilisateur racine peut aussi monter un système de fichiers arbitrairement, en nommant l'unité et le répertoire sur la ligne de commande. Dans **/etc/filesystems**, il est possible de définir les montages qui seront automatiquement effectués à l'initialisation du système. La commande **mount** sert au montage une fois le système démarré.

Description des points de montage

Un *point de montage* est un répertoire ou un fichier au niveau duquel un nouveau système de fichiers, un répertoire ou un fichier est accessible. Le point de montage d'un fichier est obligatoirement un fichier, et celui d'un répertoire ou d'un système de fichiers, obligatoirement un répertoire.

Un système de fichiers, un répertoire ou un fichier est généralement monté au niveau d'un point de montage vide, mais ceci n'est pas obligatoire. Si le répertoire ou le fichier servant de point de montage contient des données, celles-ci ne sont plus accessibles. En effet, elles sont recouvertes par les données du répertoire ou du fichier monté qui se trouvait préalablement dans ce répertoire. Elles redeviennent accessibles après le démontage du répertoire ou du fichier monté.

Quand un système de fichiers est monté sur un répertoire, les droits associés au répertoire racine du système de fichier monté ont priorité sur ceux du point de montage. Une exception, toutefois, concernant le répertoire parent `..` (point point) du répertoire de montage. Ses informations doivent être disponibles pour permettre au système d'exploitation d'accéder au nouveau système de fichiers.

Par exemple, si le répertoire courant est **/home/frank**, la commande **cd ..** passe au répertoire **/home**. Si **/home/frank** est le répertoire racine d'un système de fichiers monté, le système d'exploitation doit avoir accès aux informations du répertoire parent dans `/home/frank` pour faire aboutir la commande **cd ..**

Pour toute commande dont l'exécution requiert des informations du répertoire parent, l'utilisateur doit être autorisé à rechercher ces informations dans le répertoire de montage. En cas d'échec du répertoire de montage à accorder cette autorisation, le résultat est imprévisible, d'autant que ce type d'autorisation n'est pas visible. L'échec de la commande **pwd** est courant. En l'absence d'autorisation, **pwd** renvoie le message :

```
pwd: Permission denied
```

Affecter la valeur 111 aux autorisations du répertoire de montage permet d'éviter ce problème.

Montage des systèmes de fichiers, des répertoires et des fichiers

Il existe deux types de montages : local et à distance. Le *montage à distance* s'effectue sur un système distant par transfert des données sur une ligne de télécommunication. Les systèmes de fichiers distants, tels que NFS, doivent être exportés avant d'être montés. Le *montage local* est effectué sur le système local.

Chaque système de fichiers est associé à une unité distincte (volume logique). Un système de fichiers, pour être exploité, doit être connecté au préalable à la structure de répertoire existante (soit le système de fichiers racine soit un autre système de fichiers déjà connecté). C'est la commande **mount** qui se charge de cette connexion.

Plusieurs chemins permettent d'accéder au même système de fichiers, répertoire ou fichier. Par exemple, pour l'accès multi-utilisateur à une base de données, il est préférable d'avoir plusieurs points de montage. Chaque montage doit avoir ses propres noms et mot de passe, pour des raisons de suivi et de séparation des travaux. Ainsi, le même système de fichiers peut être monté sur différents points de montage. Par exemple, à partir de `/home/server/database`, vous pouvez monter au niveau du point de montage `/home/user1`, `/home/user2` et `/home/user3` :

```
/home/server/database      /home/user1
/home/server/database      /home/user2
/home/server/database      /home/user3
```

Un système de fichiers, un répertoire ou un fichier peut être mis à disposition de l'utilisateur par le biais de liens symboliques. Ces liens symboliques sont créés par le biais de la commande **ln -s**. Les liens entre plusieurs utilisateurs et un fichier central permettent de refléter les modifications du fichier à chaque accès utilisateur.

Contrôle des montages automatiques

Les montages peuvent être configurés pour s'effectuer automatiquement à l'initialisation du système. Il existe deux types de montages automatiques : les montages requis pour amorcer et exploiter le système, et les montages utilisateur. En ce qui concerne les premiers, les systèmes de fichiers sont automatiquement montés par le processus d'amorçage. Leurs strophes, dans le fichier **/etc/filesystems**, sont assorties de **mount = automatic**. Le second type de montage est contrôlé par l'utilisateur. Les systèmes de fichiers sont montés automatiquement par le script **/etc/rc** à l'exécution de la commande **mount all**. Les strophes de ces systèmes sont assorties de **mount = true** dans **/etc/filesystems**.

C'est le fichier **/etc/filesystems** qui contrôle les montages automatiques, effectués un à un, selon la hiérarchie. L'ordre spécifié dans ce fichier peut être modifié et restructuré.

/etc/filesystems est structuré en strophes, une par montage. La strophe décrit les attributs du système de fichiers correspondant et le procédé de montage. Les systèmes de fichiers sont montés dans l'ordre où ils figurent dans **/etc/filesystems**. Voici un extrait de **/etc/filesystems** avec un exemple de strophes :

```

/:
dev=/dev/hd4
vol="root"
mount=automatic
check=false
free=true
vfs=jfs
log=/dev/hd8
type=bootfs

/home:
dev=/dev/hd1
vfs=jfs
log=/dev/hd8
mount=true
check=true
vol="/home"
free=false

/usr:
dev=/dev/hd2
vfs=jfs
log=/dev/hd8
mount=automatic
check=false
type=bootfs
vol="/usr"
free=false

```

Pour vérifier l'ordre de montage, vous pouvez afficher le fichier **/etc/filesystems**. Quand un montage n'aboutit pas, le processus de montage continue. Par exemple, si le montage du système de fichiers **/home** échoue, le système de fichiers suivant, **/usr**, est monté. Un montage peut échouer en raison d'une erreur de typographie, d'une dépendance ou d'un incident système.

Description de la sécurité du montage des stations de travail sans disque

Les stations de travail sans disque doivent pouvoir créer des fichiers spéciaux d'unité et y accéder sur des machines distantes pour pouvoir monter leurs répertoires **/dev** depuis un serveur. Du fait que les serveurs ne peuvent pas distinguer les fichiers spéciaux d'unité d'un client de ceux d'un serveur, l'utilisateur sur le serveur peut être capable d'accéder aux unités physiques du serveur en utilisant les fichiers spéciaux de l'unité du client.

Par exemple, la propriété d'un terminal **tty** est automatiquement affectée à l'utilisateur qui utilise le terminal **tty**. Si les ID utilisateur ne sont pas identiques sur le client et le serveur, un utilisateur non privilégié sur le serveur peut accéder à un terminal **tty** utilisé par un autre utilisateur sur le serveur.

Un utilisateur privilégié sur un client peut créer des fichiers spéciaux d'unités pour qu'ils correspondent aux unités physiques du serveur et ne pas avoir à disposer de privilège pour y accéder. L'utilisateur peut ensuite utiliser un compte privilégié sur le serveur pour accéder aux unités protégées normalement en utilisant les nouveaux fichiers spéciaux d'unités.

Un problème de sécurité similaire est associé aux programmes **setuid** et **setgid** sur le client et le serveur. Les clients sans disque doivent pouvoir créer et exécuter les programmes **setuid** et **setgid** sur le serveur dans des conditions d'exploitation normales. Là encore, le serveur ne peut pas distinguer les programmes du serveur de ceux du client.

En outre, les ID d'utilisateur et les ID de groupe peuvent ne pas être identiques sur le serveur et le client. En conséquence, les utilisateurs du serveur peuvent être capables d'exécuter les programmes en accédant à des fonctions auxquelles ils ne devraient pas accéder.

Ce problème existe par le fait que les programmes **setuid** et **setgid** et les fichiers spéciaux d'unités ne devraient pouvoir être utilisés que sur la machine sur laquelle ils ont été créés.

La solution consiste à utiliser des options de sécurité avec la commande **mount**, qui restreignent l'utilisation de ces objets. Ces options peuvent être également utilisées dans les strophes du fichier **/etc/filesystems**.

L'option `nosuid` de la commande **mount** empêche d'exécuter les programmes **setuid** et **setgid** accessibles via le système de fichiers monté résultant. Cette option est utilisée avec n'importe quel système de fichiers monté sur un hôte pour être utilisé uniquement par un autre hôte (par exemple, exporté pour des clients sans disque).

L'option `nodev` de la commande **mount** empêche d'ouvrir les unités en utilisant des fichiers spéciaux d'unités accessibles via le système de fichiers monté résultant. Cette option est utilisée avec n'importe quel système de fichiers monté pour être utilisé uniquement par un autre hôte (par exemple, exporté pour des clients sans disque).

Montages sans disque

Bien que le système de fichiers d'une station de travail soit monté depuis le répertoire **/exports** d'un serveur, pour la machine sans disque, le système de fichiers correspond au système de fichiers d'une machine autonome.

Les informations suivantes montrent la relation entre le répertoire exports du serveur et les points de montage de la station de travail sans disque :

Exports serveur	Imports sans disque
/export/root/ <i>Nom_hôte</i>	/ (root)
/export/exec/ <i>Nom_SPOT</i>	/usr
/export/home/ <i>Nom_hôte</i>	/home
/export/share	/usr/share
/export/dump	Utilisé par le client sans disque comme espace de cliché
/export/swap	Utilisé par le client sans disque comme espace de cliché distant

Pour plus d'informations sur le répertoire **/export**, reportez-vous à la section Description du répertoire **/export**, page 5-9.

Sécurisation des montages sans disque

En règle générale, les utilisateurs d'un serveur ne peuvent pas accéder au répertoire **/export**.

Exportation du répertoire **/export/root**

Le répertoire **/export/root** doit être exporté avec les autorisations Lecture/Ecriture et l'utilisateur **root** sur le serveur doit y avoir accès. Toutefois, vous pouvez monter ce répertoire avec les options suivantes de la commande **mount** :

nosuid	Empêche un utilisateur du serveur d'exécuter les programmes setuid du client.
nodev	Empêche un utilisateur d'accéder aux unités du serveur en utilisant un fichier spécial d'unité du client.

Une alternative au montage du répertoire **/export/root** avec ces options consiste à empêcher les utilisateurs du serveur d'accéder au répertoire **/export/root**.

Exportation du répertoire **/export/exec**

Le répertoire **/export/exec** est exporté avec l'autorisation Lecture seule et doit fournir un accès root. Toutefois, vous pouvez monter ce répertoire avec les options suivantes de la commande **mount** :

nosuid	Empêche un utilisateur du serveur d'exécuter les programmes setuid du client. Si vous exportez le répertoire /usr du serveur, vous ne pouvez pas utiliser l'option nousid .
nodev	Empêche un utilisateur d'accéder aux unités du serveur en utilisant un fichier spécial d'unité du client.

Exportation du répertoire **/export/share**

Le répertoire **/export/share** est exporté avec l'autorisation Lecture seule et doit fournir un accès root. Du fait que ce répertoire ne contient généralement que des données (pas d'exécutables, ni des unités), il n'est pas nécessaire d'utiliser les options de sécurité de la commande **mount**.

Exportation du répertoire **/export/home**

Vous pouvez monter le répertoire utilisateur **/home** de différentes manières :

- Vous pouvez monter le répertoire **/export/home/ Nom_hôte_client** sur le répertoire **/home** du client. Dans ce cas, le client dispose des autorisations Lecture/Ecriture et l'utilisateur dispose d'un accès. Pour protéger le système, monter le répertoire **/export/home** avec les options suivantes de la commande **mount** :

nosuid	Empêche un utilisateur du serveur d'exécuter les programmes setuid du client.
nodev	Empêche un utilisateur d'accéder aux unités du serveur en utilisant un fichier spécial d'unité du client.

- Vous pouvez monter le répertoire **/home** sur le serveur sur le répertoire **/home** du client. Dans ce cas, le répertoire **/home** est exporté avec les autorisations Lecture/Ecriture et sans accès root. Pour protéger le système, montez le répertoire **/home** sur le serveur et le client avec les options **nosuid** et **nodev** de la commande **mount**.
- Vous pouvez également monter le client sur chaque répertoire **/home/ Nom_Utilisateur** du serveur sur le répertoire **/home/ Nom_utilisateur** du client pour que les utilisateurs puissent se connecter à des machines différentes et toujours accéder à leurs répertoires home. Dans ce cas, les répertoires **/home/ Nom_utilisateur** du serveur et des clients sont montés avec les options **nousid** et **nodev** de la commande **mount**.

Exportation du répertoire **/export/dump**

Exportez le répertoire **/export/dump/ Nom_hôte_client** avec les autorisations Lecture/Ecriture et l'accès root. Les utilisateurs du serveur n'ont pas accès aux fichiers **/export/dump/ Nom_hôte_client**.

Exportation du répertoire **/export/swap**

Exportez le répertoire **/export/swap/ Nom_hôte_client** avec les autorisations Lecture/Ecriture et l'accès root. Aucune mesure de sécurité n'est nécessaire. Les utilisateurs du serveur n'ont pas accès aux fichiers **/export/swap/ Nom_hôte_client**.

Types de systèmes de fichiers

AIX prend en charge plusieurs types de systèmes de fichiers, à savoir :

- JFS (Journaled File System) ou JFS2 (Enhanced Journaled File System)
- NFS (Network File System)
- CDRFS (CD-ROM File System)
- UDFS (DVD-ROM File System)

JFS (Journaled File System) ou JFS2 (Enhanced Journaled File System)

Il prend en charge l'ensemble de la sémantique de système de fichiers. Ces systèmes de fichiers utilisent des techniques de journalisation de base de données pour maintenir la cohérence structurelle. Cela permet d'éviter d'endommager le système de fichiers lorsque le système s'arrête de manière anormale.

Chaque système de fichiers JFS ou JFS2 réside dans un volume logique distinct. Le système d'exploitation monte le système de fichiers lors de l'initialisation. Cette configuration à plusieurs systèmes de fichiers s'avère utile pour les fonctions de gestion de système telles que les sauvegardes, les restaurations et les réparations, car elle isole une partie de l'arborescence de fichiers pour vous permettre de travailler dessus.

JFS2 se distingue de JFS dans la mesure où ce système de fichiers prend en charge des fichiers et des systèmes de fichiers volumineux. Ces types de systèmes de fichiers sont décrits plus en détail dans la section Description de JFS et de JFS2, page 5-18.

NFS (Network File System)

Il s'agit d'un système de fichiers distribué qui permet aux utilisateurs d'accéder aux fichiers et aux répertoires situés sur des ordinateurs distants et qui les utilise comme s'il s'agissait de fichiers et de répertoires locaux. Les utilisateurs, par exemple, peuvent utiliser les commandes du système d'exploitation pour créer, supprimer, lire, écrire des fichiers et définir des attributs sur les fichiers et les répertoires distants. NFS est décrit plus en détail dans le document *AIX 5L Version 5.2 Security Guide*.

CDRFS (CD-ROM File System)

Ce système de fichiers permet d'accéder au contenu d'un CD-ROM via des interfaces de système de fichiers normales. CDRFS est décrit plus en détail dans la section Description de CDRFS, page 5-31.

UDFS (DVD-ROM File System)

Ce système de fichiers permet d'accéder au contenu d'un DVD via des interfaces de système de fichiers normales. UDFS est décrit plus en détail dans la section Description de UDFS.

Description de JFS et de JFS2

Les systèmes de fichiers JFS (Journaled File System) et JFS2 (Enhanced Journaled File System) sont intégrés au système d'exploitation de base. Les deux types de systèmes de fichiers lient leurs données de fichiers et de répertoires à la structure utilisée par LVM (Logical Volume Manager) AIX pour le stockage et l'extraction des données. JFS2 diffère de JFS dans la mesure où ce système de fichiers prend en charge un noyau de 64 bits et des fichiers plus volumineux.

Ces types de fichiers sont décrits dans les sections suivantes. Sauf indication contraire, les sections suivantes s'appliquent aux systèmes de fichiers JFS et JFS2.

JFS2

JFS2 (Enhanced Journaled File System) est un système de fichiers introduit dans AIX 5L for POWER Version 5.1 qui permet de stocker des fichiers beaucoup plus volumineux que le système de fichiers JFS (Journaled File System). Les clients peuvent décider de mettre en œuvre JFS, le système de fichiers recommandé pour les environnements 32 bits, ou JFS2 qui offre des fonctionnalités 64 bits.

Remarque : Contrairement au système de fichiers JFS, le système de fichiers JFS2 ne permet pas d'utiliser l'API **link()** sur des fichiers de type répertoire. Cette limitation peut empêcher certaines applications qui fonctionnent correctement avec un système de fichiers JFS, de fonctionner correctement avec le système de fichiers JFS2.

Le tableau ci-dessous répertorie les fonctions JFS et JFS2 :

Fonctions	JFS2	JFS
Fragments/taille de bloc	Tailles de blocs 512–4096	Fragments 512–4096
Taille maximale de système de fichiers	16 téraoctets	1 téraoctet
Taille maximale de fichier	16 téraoctets	64 Go
Nombre d'i-nodes	Dynamique, limitée par l'espace disque	Fixe, défini lors de la création d'un système de fichiers
Organisation des répertoires	Arborescence B	Linéaire
Compression	Non	Oui
Quotas	Non	Oui

Remarque : La taille de fichier maximale et la taille de système de fichiers maximale sont limitées à 1 téraoctet avec le noyau 32 bits.

Description de la segmentation de l'espace disque

La plupart des systèmes de fichiers UNIX allouent uniquement un espace disque contigu en unités dont la taille est égale à celle des blocs logiques utilisés pour la division logique des fichiers et des répertoires. Ces unités d'allocation s'appellent généralement des *blocs de disque* et un seul bloc de disque est utilisé exclusivement pour stocker des données contenues dans un seul bloc logique de fichier ou de répertoire.

L'utilisation d'une taille de bloc logique relativement grande (4096 octets par exemple) et la gestion des allocations de blocs de disque dont la taille est égale à celle du bloc logique permettent de réduire le nombre d'opérations E/S exécutées par une seule opération de système d'exploitation. Les données d'un fichier ou d'un répertoire sont stockées sur disque dans un plus petit nombre de blocs de plus grande taille au lieu d'être stockées dans un grand nombre de petits blocs de disque. Par exemple, un fichier de 4096 octets ou moins est alloué à un bloc de disque de 4096 octets si la taille de bloc logique est égale à 4096 octets. En conséquence, une opération de lecture ou d'écriture n'effectue qu'une seule opération E/S sur disque pour accéder aux données du disque. Si la taille de bloc logique est plus petite et nécessite plusieurs allocations pour la même quantité de données, plusieurs opérations E/S sur disque sont nécessaires pour accéder aux données. Un grand bloc logique et une taille de bloc de disque égale permettent également de réduire le nombre d'allocations d'espace disque qui doivent être exécutées lorsque des données sont ajoutées aux fichiers et aux répertoires, car les grands blocs de données contiennent plus de données.

La limitation de l'unité d'allocation d'espace disque à la taille de bloc logique peut toutefois générer une perte d'espace disque dans un système de fichiers qui contient de nombreux petits fichiers et répertoires. Une perte d'espace disque se produit lorsque la valeur d'un bloc logique d'espace disque est allouée à un bloc logique partiel d'un fichier ou d'un répertoire. Du fait que les blocs logiques partiels contiennent toujours moins d'un bloc logique de données, un bloc logique partiel ne consomme qu'une partie de l'espace disque qui lui est alloué. La partie restante est inutilisée, car aucun autre fichier ou répertoire ne peut écrire son contenu dans l'espace de disque qui a déjà été alloué. La quantité totale d'espace disque perdue peut être importante pour les systèmes de fichiers qui contiennent un grand nombre de petits fichiers et répertoires.

Le système de fichiers (Journaled File System) divise l'espace disque en unités d'allocation appelées *fragments*. Le système de fichiers JFS2 (Enhanced Journaled File System) segmente l'espace disque en *blocs*. L'objectif est le même : stocker efficacement les données.

Les fragments JFS sont plus petits que la taille d'allocation de disque par défaut de 4096 octets. Les fragments réduisent les pertes d'espace disque en stockant beaucoup plus efficacement les données dans des blocs logiques partiels de fichier ou de répertoire. Le comportement fonctionnel du support des fragments JFS repose sur celui fourni par le support des fragments BSD (Berkeley Software Distribution).

JFS2 prend en charge plusieurs tailles de blocs de système de fichiers, à savoir 512, 1 024, 2 048 et 4 096. Des blocs de plus petite taille réduisent les pertes d'espace disque en stockant plus efficacement les données dans des blocs logiques partiels de fichier ou de répertoire. Les blocs de petite taille augmentent la charge de travail. La taille des blocs d'un système de fichiers JFS2 est définie lors de sa création. Des systèmes de fichiers différents peuvent avoir des tailles de blocs différentes, mais une seule taille de bloc peut être utilisée dans un même système de fichiers.

Fragments JFS

Dans JFS, l'unité d'allocation d'espace disque s'appelle un *fragment* dont la taille peut être inférieure à celle d'un bloc logique de 4096 octets. En utilisant des fragments de moins de 4 096 octets, les données contenues dans un bloc logique partiel peuvent être stockées plus efficacement en utilisant uniquement le nombre de fragments nécessaire pour stocker les données. Par exemple, un bloc logique partiel de 500 octets peut être alloué à un fragment de 512 octets (en supposant que la taille de fragment est de 512 octets), ce qui permet de réduire considérablement les pertes d'espace disque. Si les besoins en stockage d'un bloc logique partiel augmentent, un ou plusieurs fragments supplémentaires sont alloués.

La taille de fragment d'un système de fichiers est définie lors de sa création. Des fragments de 512, 1024, 2048 et 4096 octets peuvent être alloués à un système de fichiers JFS. Des systèmes de fichiers différents peuvent avoir des tailles de fragments différentes, mais une seule taille de fragment peut être utilisée dans un même système de fichiers. Différentes tailles de fragments peuvent également coexister sur un même système (machine) pour que les utilisateurs puissent sélectionner la taille la mieux adaptée à chaque système de fichiers.

Le support de fragment JFS fournit une vue du système de fichiers sous la forme d'une série de fragments contigus et non sous la forme d'une série de blocs de disque contigus. Pour garantir l'efficacité des opérations sur disque, l'espace disque est souvent alloué en unités de 4096 octets pour que les blocs de disque ou les unités d'allocation aient la même taille que les blocs logiques. Dans ce cas, une allocation de bloc de disque correspond à une allocation de 4096 octets de fragments contigus.

L'augmentation de la charge de travail (recherches sur disque, transferts de données et activité d'allocation supplémentaires) et la meilleure utilisation de l'espace disque sont inversement proportionnelles à la taille de fragment d'un système de fichiers. Pour garantir un équilibre optimum entre l'augmentation de la charge de travail et la diminution de l'espace disque, les facteurs suivants s'appliquent au support de fragment JFS :

- Dans la mesure du possible, les allocations de fragments de 4096 octets sont prises en charge pour un fichier ou les blocs logiques d'un répertoire.
- Des fragments de moins de 4096 octets peuvent être alloués uniquement à des blocs logiques partiels de fichier ou de répertoire de moins de 32 Ko

Du fait que la taille des fichiers et des répertoires d'un système de fichiers dépasse 32 Ko, l'avantage de gérer des allocations d'espace disque de moins de 4096 octets pour les blocs logiques partiels diminue. L'économie d'espace disque en tant que pourcentage de l'espace de système de fichiers total diminue alors que les performances de gestion de petites allocations d'espace disque restent constantes. Du fait que les allocations d'espace disque de moins de 4096 octets permettent d'utiliser l'espace disque plus efficacement avec des petits fichiers et répertoires, les blocs logiques des fichiers et des répertoires dont la taille est égale ou supérieure à 32 Ko se voient toujours alloués 4096 octets de fragments. 4096 octets de fragment sont alloués à tout bloc logique partiel associé à un fichier ou un répertoire volumineux.

Blocs JFS2

Le système de fichiers JFS2 (Enhanced Journaled File System) segmente l'espace disque en *blocs*. JFS2 prend en charge différentes tailles de blocs de système de fichiers, à savoir 512, 1024, 2048 et 4096. Différents systèmes de fichiers peuvent avoir différentes tailles de blocs, mais une seule taille de bloc peut être utilisée dans un même système de fichiers.

Les petits blocs réduisent les pertes d'espace disque en stockant beaucoup plus efficacement les données dans des blocs logiques partiels de fichier ou de répertoire. Les blocs de petite taille augmentent également la charge de travail. En outre, les pilotes d'unités doivent fournir une adressabilité de bloc de disque égale ou inférieure à la taille de bloc du système de fichiers.

Du fait que l'espace disque est alloué en petites unités pour un système de fichiers avec une taille de bloc autre que 4096 octets, les activités d'allocation sont plus fréquentes lorsque la taille des fichiers ou des répertoires augmentent régulièrement. Par exemple, une opération d'écriture qui augmente la taille d'un fichier vide de 512 octets génère une allocation d'un bloc pour le fichier, en supposant une taille de bloc de 512 octets. Si la taille du fichier augmente de nouveau de 512 octets en écriture, un bloc supplémentaire doit être alloué au fichier. Si l'on applique cet exemple à un système de fichiers qui utilise des blocs de 4096 octets, l'allocation d'espace disque se produit une seule fois dans le cadre de la première opération d'écriture. Aucune autre activité d'allocation supplémentaire n'est exécutée lors de la seconde opération d'écriture du fait que l'allocation initiale de 4096 octets est suffisamment importante pour stocker les données ajoutées par la seconde opération d'écriture.

La taille de bloc d'un système de fichiers est définie lors de la création du système de fichiers avec Web-based System Manager, SMIT (System Management Interface Tool) ou les commandes **crfs** et **mkfs**. La taille de bloc de système de fichiers à utiliser doit être déterminée en fonction de la taille prévisible des fichiers du système de fichiers.

La taille de bloc d'un système de fichiers peut être identifiée via Web-based System Manager, SMIT (System Management Interface Tool) ou la commande **lsfs**. Pour les applications, utilisez la sous-routine **statfs** pour identifier la taille des blocs du système de fichiers.

Les blocs font office d'unités de base pour l'allocation d'espace disque et l'état de l'allocation de chaque bloc dans un système de fichiers est enregistré dans la mappes d'allocation de blocs du système de fichiers. Une plus grande quantité de mémoire virtuelle et d'espace disque de système de fichiers peuvent être nécessaire pour stocker les mappes d'allocation de blocs des systèmes de fichiers dont la taille de bloc est inférieure à 4096 octets.

Description du nombre variable d'i-nodes

L'utilisation de segments d'espace disque de moins de 4 096 octets permet d'optimiser l'utilisation de l'espace disque, mais elle augmente le nombre de petits fichiers et de répertoires qui peuvent être stockés dans un système de fichiers. Toutefois, l'espace disque correspond uniquement à l'une des ressources de système de fichiers nécessaires aux fichiers et aux répertoires : chaque fichier ou répertoire nécessite également un i-node de disque.

JFS et i-nodes

JFS permet de définir le nombre d'i-nodes de disque créés dans un système de fichier lorsqu'un nombre d'i-nodes de disque inférieur ou supérieur au nombre d'i-nodes de disque par défaut est nécessaire. Le nombre d'i-nodes de disque lors de la création du système de fichiers est défini par une valeur appelée *nombre d'octets par i-node* ou *NBPI*. Par exemple, une valeur NBPI de 1024 génère la création d'i-nodes de disque pour chaque 1024 octets d'espace disque du système de fichiers. Il convient également de considérer qu'une petite valeur NBPI (512, par exemple) génère un grand nombre d'i-nodes alors qu'une grande valeur NBPI (telle que 16384) génère un plus petit nombre d'i-nodes.

Pour un système de fichiers JFS, un i-node est créé pour chaque octet NBPI d'espace de groupe d'allocation alloué au système de fichiers. Le nombre total d'i-nodes dans un système de fichiers limite le nombre total de fichiers et la taille totale du système de fichiers. Un groupe d'allocation peut être alloué partiellement bien que le nombre total d'i-nodes par groupe d'allocation soit toujours alloué. La valeur NBPI est inversement proportionnelle au nombre total d'i-nodes dans un système de fichiers.

JFS limite tous les systèmes de fichiers à 16 Mo (2^{24}) d'i-nodes.

Le groupe de valeurs NBPI pouvant être alloué varie en fonction de la taille du groupe d'allocation (*agsize*). La valeur par défaut est 8 Mo. Les valeurs NBPI pouvant être allouées sont 512, 1024, 2048, 4096, 8192 et 16384 avec une valeur *agsize* de 8 Mo. Une plus grande taille peut être utilisée. Les valeurs *agsize* peuvent être égales à 8, 16, 32 et 64. La plage de valeurs NBPI augmente lorsque la valeur *agsize* augmente. Si la valeur *agsize* est le double de 16 Mo, les valeurs de la plage de valeurs NBPI double également : 1024, 2048, 4096, 8193, 16384 et 32768.

La taille de fragment et la valeur NBPI sont définies lors de la création du système de fichiers avec Web-based System Manager, SMIT (System Management Interface Tool) ou les commandes **crfs** et **mkfs**. La détermination de la taille de fragment et du nombre d'i-nodes à créer pour le système de fichiers dépendent du nombre de fichiers et de la taille des fichiers prévisibles du système de fichiers.

La taille de fragment et la valeur NBPI peuvent être identifiées via Web-based System Manager, SMIT (System Management Interface Tool) ou la commandes **lsfs**. Pour les applications, utilisez la sous-routine **statfs** pour identifier la taille des fragments du système de fichiers.

JFS2 et i-nodes

JFS2 alloue des i-nodes en fonction des besoins. Si le système de fichiers contient un espace suffisant pour créer d'autres i-nodes, des i-nodes sont alloués automatiquement. En conséquence, le nombre d'i-nodes disponible est limité par la taille du système de fichiers.

Description des limitations de taille

Vous définissez la taille maximale d'un système de fichiers JFS lors de sa création. La taille du système de fichiers JFS dépend de plusieurs éléments importants. La section suivante décrit les principales considérations.

La taille maximale recommandée pour un système de fichiers JFS2 est de 16 téraoctets. Les principales considérations associées aux systèmes de fichiers JFS2 volumineux sont décrites dans la section Taille de journal, page 5-23 et Limitations de taille JFS2 , page 5-25.

Bien que les systèmes de fichiers qui utilisent une unité d'allocation inférieure à 4096 octets nécessitent substantiellement moins d'espace disque que ceux dont l'unité d'allocation par défaut est de 4096 octets, l'utilisation de fragments plus petits peut avoir un impact sur les performances.

L'état d'allocation de chaque fragment (JFS) ou bloc (JFS2) dans un système de fichiers est enregistré dans la mappe d'allocation du système de fichiers. Une plus grande quantité de mémoire virtuelle et d'espace disque de système de fichiers peuvent être nécessaires pour stocker les mappes d'allocation des systèmes de fichiers dont la taille de fragment ou de bloc est inférieure à 4096 octets.

Du fait que l'espace disque est alloué en petites unités pour un système de fichiers avec une taille de fragment autre que 4096 octets, les activités d'allocation sont plus fréquentes lorsque la taille des fichiers ou des répertoires augmente régulièrement. Par exemple, une opération d'écriture qui augmente la taille d'un fichier vide de 512 octets génère une allocation de fragment ou de bloc de 512 octets pour le fichier, en fonction du type de système de fichiers. Si la taille du fichier augmente de nouveau de 512 octets en écriture, un fragment ou un bloc supplémentaire doit être alloué au fichier. Si l'on applique cet exemple à un système de fichiers qui utilise des fragments ou des blocs de 4096 octets, l'allocation d'espace disque se produit une seule fois dans le cadre de la première opération d'écriture. Aucune autre activité d'allocation supplémentaire n'est exécutée lors de la seconde opération d'écriture du fait que l'allocation initiale de 4 096 octets est suffisamment importante pour stocker les données ajoutées par la seconde opération d'écriture. L'activité d'allocation peut être réduite si la taille des fichiers augmente de 4 096 octets à la fois.

Taille de journal

L'un des problèmes de taille est associé à la taille du journal du système de fichiers.

Pour JFS, dans la majorité des cas, plusieurs systèmes de fichiers utilisent un journal commun dont la taille est de 4 Mo. Par exemple, après l'installation initiale, tous les systèmes de fichiers dans le groupe de volume racine utilisent le volume logique hd8 comme journal JFS. La taille de partition par défaut du volume logique est de 4 Mo et la taille par défaut du journal étant égale à une partition, le groupe de volume racine contient normalement un journal JFS de 4 Mo. Lorsque la taille des systèmes de fichiers par défaut est supérieure à 2 Go ou que l'espace total du système de fichiers utilisant un seul journal est supérieur à 2 go, la taille de journal par défaut peut s'avérer insuffisante. Dans les deux cas, les tailles de journaux augmentent lorsque la taille des systèmes de fichiers augmente. Lorsque la taille du volume logique du journal change, la commande **logform** doit être exécutée pour réinitialiser le journal et pouvoir utiliser le nouvel espace. La taille du journal JFS est limité à 256 Mo.

Un journal JFS peut prendre en charge une certaine taille cumulée de systèmes de fichiers. Comme référence, une capacité totale de systèmes de fichiers d'un trois millions d'octets est la limite recommandée pour un journal JFS. Lorsque cette limite est dépassée ou est sur le point d'être dépassée ou qu'un manque de mémoire existe lors de l'exécution de la commande **logredo** (exécutée par la commande **fsck**), ajoutez un journal JFS et partagez la charge entre les journaux JFS.

Pour JFS2, dans la plupart des cas, plusieurs systèmes de fichiers utilisent un journal commun. Lorsque la taille des systèmes de fichiers par défaut est supérieure à 2 Go ou que l'espace total des systèmes de fichiers utilisant un seul journal est supérieur à 2 Go, la taille de journal par défaut peut s'avérer insuffisante. Dans les deux cas, vous pouvez augmenter la taille des journaux à mesure que celle du système de fichier augmente ou vous pouvez ajouter un journal JFS2 et partager la charge entre les deux journaux JFS2.

Limites de taille JFS

La taille maximale JFS est définie lors de la création du système de fichiers. La valeur NBPI, la taille de fragment et la taille de groupe d'allocation déterminent la taille maximale. La limitation de la taille du système de fichiers correspond aux valeurs minimales suivantes :

$$NBPI * 2^{24}$$

OU

$$Taille_fragment * 2^{28}$$

Par exemple, si vous sélectionnez un taux NBPI de 512, la taille du système de fichiers est limitée à 8 Go ($512 * 2^{24} = 8 \text{ Go}$). JFS prend en charge les valeurs NBPI 512, 1 024, 2 048, 4 096, 8 192, 16 384, 32 768, 65 536 et 131 072.

JFS limite tous les systèmes de fichiers à 16 Mo (2^{24}) d'i-nodes.

Un i-node est créé pour chaque octet NBPI d'espace de groupe d'allocation alloué au système de fichiers. Un groupe d'allocation peut être alloué partiellement bien que le nombre total d'i-nodes par groupe d'allocation soit toujours alloué. La valeur NBPI est inversement proportionnelle au nombre total d'i-nodes dans un système de fichiers.

JFS divise l'espace de système de fichiers en groupes d'i-nodes et de blocs de disque pour les données utilisateur. Ces groupes s'appellent des groupes d'allocation. La taille de groupe d'allocation peut être définie lors de la création du système de fichiers. Les tailles de groupe d'allocation sont 8 Mo, 16 Mo, 32 Mo et 64 Mo. Chaque taille de groupe d'allocation est associée à une plage NBPI. Les plages sont définies en fonction du tableau suivant :

Groupe d'allocation Taille en mégaoctets	Valeurs NBPI allouables
8	512, 1024, 2048, 4096, 8192, 16384
16	1024, 2048, 4096, 8192, 16384, 32768
32	2048, 4096, 8192, 16384, 32768, 65536
64 131072	4096, 8192, 16384, 32768, 65536,

JFS supporte quatre tailles de segments, à savoir des unités de 512, 1 024, 2 048 et 4 096 octets d'espace disque contigus. JFS gère les adresses des segments dans les i-nodes et les blocs indirects sous forme de nombre de 28 bits. Chaque segment doit être adressable par un nombre compris entre 0 et (2^{28}).

Description des limitations de taille JFS2

Les tests montrent que des systèmes de fichiers JFS2 très volumineux sont plus faciles à gérer que les systèmes de fichiers qui contiennent un grand nombre de petits fichiers. Lorsqu'un système de fichiers volumineux contient un grand nombre de petits fichiers, l'exécution de la commande **fsck** et des autres tâches de maintenance des systèmes de fichiers est plus longue. Les limitations de tailles suivantes sont recommandées :

Taille maximale de système de fichiers JFS2 :	16 To
Taille maximale de fichier JFS2 :	16 To

Fragmentation de l'espace libre JFS

Pour les systèmes de fichiers JFS, l'utilisation de fragments inférieurs à 4 096 octets peut provoquer une plus fragmentation plus importante de l'espace disque libre. Par exemple, considérez une zone du disque qui est divisée en huit fragments de 512 octets chacun. Supposons que différents fichiers, nécessitant chacun 512 octets, sont écrits dans le premier, le quatrième, le cinquième et le septième segments dans cette zone de disque et que le deuxième, le troisième, le sixième et le huitième segments sont libres. Bien que quatre fragments représentant 2048 octets d'espace disque soient libres, aucun bloc logique partiel nécessitant quatre fragments (ou 2048 octets) n'est alloué pour ces fragments libres du fait que les fragments d'une allocation doivent être contigus.

Du fait que les fragments pour un fichier ou des blocs logiques de répertoire doivent être contigus, la fragmentation de l'espace libre peut provoquer l'échec d'un système de fichiers qui demande un nouvel espace disque, même si la quantité d'espace libre disponible est suffisamment grande pour répondre aux besoins de l'opération. Pour exemple, une opération d'écriture qui augmente la taille d'un fichier vide d'un bloc logique nécessite l'allocation de 4096 octets d'espace disque contigu. Si l'espace libre de système fichiers est fragmenté et qu'il est constitué de 32 fragments de 512 octets non contigus ou d'un total de 16 Ko d'espace libre, l'opération d'écriture échoue car huit fragments contigus (ou 4096 octets d'espace disque contigu) ne sont pas disponibles pour exécuter l'opération.

Un système de fichiers JFS avec une quantité d'espace libre fragmenté non gérable peut être défragmenté avec la commande **defrags**. La commande **defrags** a un impact positif sur les performances.

Description des fichiers supplémentaires

Un fichier est une séquence des blocs indexés. Les blocs sont associés depuis l'i-node au décalage logique du fichier qu'ils représentent.

Un fichier qui dispose d'un ou de plusieurs index qui ne sont pas associés à un bloc de données est considéré être un *fichier alloué sporadiquement* ou *fichier sporadique*. Un fichier sporadique a une taille donnée, mais il ne contient pas tous les blocs alloués pour répondre aux exigences de taille. Pour déterminer si un fichier est alloué sporadiquement, utilisez la commande **fileplace**. La commande indique tous les blocs du fichier qui ne sont pas alloués.

Remarque : Dans la plupart des cas, la commande **du** peut être également utilisée pour déterminer si le nombre de blocs de données alloués à un fichier ne correspondent pas à ceux nécessaires au stockage d'un fichier de cette taille. Un système de fichiers compressé a le même comportement que les fichiers qui ne sont pas alloués sporadiquement.

Un fichier supplémentaire est créé lorsqu'une application étend un fichier en effectuant une recherche dans un emplacement externe aux index alloués, tandis que les données qui sont écrites n'occupent pas l'ensemble des nouveaux index affectés. La nouvelle taille de fichier reflète l'opération d'écriture à la dernière extrémité dans le fichier.

Une opération de lecture dans une section d'un fichier qui contient des blocs de données non alloués génère un tampon de zéros qui sont retournés. Lorsqu'une opération d'écriture a lieu dans un fichier qui contient des blocs de données non alloués, les blocs de données nécessaires sont alloués et les données sont écrites.

Ce comportement peut affecter la manipulation du fichier ou les commandes d'archivage. Par exemple, les commandes suivantes ne préservent pas l'allocation ponctuelle d'un fichier :

- **cp**
- **mv**
- **tar**
- **cpio**

Remarque : Dans le cas de la commande **mv**, cela s'applique uniquement au transfert d'un fichier vers un autre système de fichiers. Si le fichier est transféré dans le même système de fichiers, il demeure un fichier sporadique.

Lorsqu'un fichier est copié ou restauré depuis les commandes précédentes, chaque bloc de données est alloué et le fichier ne dispose pas de caractéristiques supplémentaires. Toutefois, les commandes d'archivage préservent les caractéristiques sporadiques ou utilisent activement un fichier sporadique :

- **backup**
- **restore**
- **pax**

Du fait qu'il est possible d'omettre des ressources pour un systèmes de fichiers avec des fichiers sporadiques, vous devez utiliser et gérer ce type de fichier avec précaution.

Description des fichiers volumineux et JFS

Cette section explique comment créer des fichiers volumineux et l'allocation de fichiers avec le système de fichiers JFS. Tous les systèmes de fichiers JFS2 prennent en charge des fichiers volumineux.

Les systèmes de fichiers compatibles avec les fichiers volumineux peuvent être créés avec les commandes **crfs** et **mkfs**. Les deux commandes disposent d'une option (`bf=true`) qui permet de définir les systèmes de fichiers compatibles avec les fichiers volumineux. Vous pouvez également utiliser SMIT ou Web-based System Manager pour créer des systèmes de fichiers.

Dans les systèmes de fichiers compatibles avec les fichiers volumineux, les données des fichiers sont stockées avant que les 4 Mo des fichiers soient alloués dans des blocs de 4 096 octets. Les données des fichiers stockées au-delà de la limite de 4 Mo sont affectées de grand blocs de disque de 128 Ko. En fait, les grands blocs de disque correspondent à 32 blocs de 4096 octets contigus.

Par exemple, dans un système de fichiers normal, le fichier de 132 Mo nécessite 33 blocs de disque de 4 Ko (33 blocs indirects contenant chacun 1024 adresses de disque de 4 Ko). Un fichier de 132 Mo dans un système de fichiers compatible avec des fichiers volumineux contient 1024 blocs de disque de 4 Ko et 1024 blocs de disque de 128 Ko. La géométrie de fichier volumineux nécessite uniquement des blocs indirects simples pour un fichier de 132 Mo. Les types de fichiers volumineux et normaux nécessitent un double bloc indirect.

Les blocs de disque volumineux nécessitent 32 blocs de 4 Ko contigus. Si vous écrivez dans des fichiers volumineux au-delà de 4 Mo, le décalage de fichier échoue avec ENOSPC si le système de fichiers ne contient pas 32 blocs de 4 Ko contigus non utilisés.

Remarque : Le système de fichiers peut contenir des milliers de blocs libres, mais si 32 d'entre eux ne sont pas contigus, l'allocation échoue.

La commande **defragfs** identifie les blocs de disque pour fournir des zones de blocs libres contiguës.

Le système de fichiers JFS nécessite d'initialiser toutes les nouvelles allocations de disque. JFS démarre la procédure kproc du noyau pour mettre à zéro l'allocation de fichier initiale lors du montage du premier système de fichiers compatible avec les fichiers volumineux du système. La procédure kproc reste active une fois que le système de fichiers compatible avec les fichiers volumineux a été démonté.

Description de la compression des données JFS

Le système de fichiers JFS prend en charge les systèmes de fichiers fragmentés et compressés qui permettent d'économiser l'espace disque en permettant à un bloc logique d'être stocké sur le disque sous forme d'unités ou de "fragments" d'une taille inférieure à la taille totale de bloc de 4096 octets. JFS2 ne prend pas en charge la compression des données.

Dans un système de fichiers fragmenté, seul le dernier bloc logique des fichiers inférieurs à 32 Ko est stocké de cette manière. Le support de fragment n'est ainsi que bénéfique aux systèmes de fichiers qui contiennent de nombreux petits fichiers. Toutefois, la compression des données permet de stocker tous les blocs logiques d'un fichier de n'importe quelle taille sous la forme d'un ou de plusieurs fragments contigus. En moyenne, la compression des données permet d'économiser la moitié de l'espace disque.

L'utilisation de fragments et de la compression de données, augmentent toutefois la fragmentation potentielle de l'espace disque libre. Les fragments alloués à un bloc logique doivent être contigus sur le disque. Un système de fichiers soumis à une fragmentation d'espace peut avoir des difficultés à rechercher suffisamment de fragments contigus pour une allocation de bloc logique, même si le nombre total de fragments libres est supérieur aux besoins de bloc logique. Le système de fichiers JFS réduit la fragmentation de l'espace en fournissant le programme **defragfs** qui "défragmente" un système de fichiers en augmentant la quantité d'espace libre contiguë. Cet utilitaire peut être utilisé pour fragmenter et compresser des systèmes de fichiers. L'économie d'espace disque issue des fragments et de la compression des données peut être significative tandis que la fragmentation de l'espace libre reste gérable.

La compression des données dans le système de fichiers JFS en cours est compatible avec les versions précédentes du système d'exploitation. L'API constituée de tous les appels système reste la même dans les deux versions du système de fichiers JFS.

Pour plus d'informations sur la prise en charge des fragments, l'utilisation du disque, la fragmentation de l'espace libre et les coûts en performances associés aux fragments, reportez-vous à la section Description de la segmentation de l'espace libre.

Mise en œuvre de la compression des données

Attention: Le système de fichiers racine (/) ne doit pas être compressé. La compression du système de fichiers /usr n'est pas recommandée, car installp doit pouvoir calculer précisément sa taille pour les mises à jour et les nouvelles installations. Pour plus d'informations sur la taille et les calculs, reportez-vous à la section Comportement implicite.

La compression des données est un attribut d'un système de fichiers, qui est défini lors de la création du système de fichiers avec la commande **crfs** ou **mkfs**. Vous pouvez également utiliser Web-based System Manager ou SMIT pour définir la compression des données. La compression ne s'applique qu'aux fichiers normaux et aux liens symboliques longs des systèmes de fichiers de ce type. Le support de la fragmentation continue de s'appliquer aux répertoires et aux métadonnées qui ne sont pas comprimés. Chaque bloc logique d'un fichier s'auto-comprime avant d'être écrit sur le disque. De cette manière, la compression facilite les recherches aléatoires et les mises à jour tout en ne perdant qu'une petite quantité d'espace disque libre par rapport à la compression des données dans des unités plus grandes.

Après la compression, un bloc logique nécessite en général moins de 4096 octets d'espace libre. Le bloc logique compressé est écrit sur le disque et reçoit uniquement le nombre de fragments contigus nécessaire à son stockage. S'il est impossible de compresser un bloc logique, le bloc est écrit sur le disque non compressé et il reçoit 4096 octets de fragments contigus.

La commande **lsfs -q** affiche la valeur en cours de la compression. Vous pouvez également utiliser Web-based System Manager ou SMIT pour identifier la compression des données.

Comportement implicite

Du fait qu'un programme qui écrit un fichier ne s'attend pas à manquer d'espace (ENOSPC) après la réussite de l'écriture (ou le stockage des fichiers mappés), il est nécessaire de garantir la disponibilité de l'espace lorsque les blocs logiques sont écrits sur le disque. Cette opération est effectuée en allouant 4096 octets à un bloc logique lorsqu'il est modifié pour la première fois pour qu'il existe un espace disque disponible si le bloc ne peut pas être compressé. Si une allocation de 4096 octets n'est pas disponible, le système retourne une erreur ENOSPC ou EDQUOT, même s'il existe un espace disque suffisant pour contenir le bloc logique compressé. La signalisation du manque d'espace anticipé se produit en général lorsque les limites de quotas de disque sont presque atteintes ou lorsque le système de fichiers est presque plein.

Les systèmes de fichiers compressés peuvent avoir également les comportements suivants :

- Du fait que 4096 octets sont initialement alloués à un bloc logique, certains appels système peuvent recevoir une erreur ENOSPC ou EDQUOT. Par exemple, un ancien fichier peut être mappé à l'aide de l'appel système **mmap** et une opération dans un emplacement déjà écrit peut provoquer une erreur ENOSPC.
- Avec la compression des données, un bloc de disque plein reste alloué à un bloc modifié jusqu'à ce qu'il soit écrit sur le disque. Si le bloc a eu précédemment une allocation validée d'une taille inférieure à un bloc complet, la quantité d'espace disque occupée par le bloc correspond à la somme des deux, l'allocation précédente n'étant pas libre tant que le fichier (i-node) n'est pas validé. C'est le cas pour les fragments normaux. Le nombre de blocs logiques dans un fichier dont les allocations précédentes peuvent être validées correspond, au plus, au nombre de fragments normaux, mais peut correspondre au nombre de blocs dans un fichier avec compression.
- Aucune des ressources déjà validées d'un bloc logique n'est libérée tant que l'appel **fsync** ou **sync** n'est pas exécuté par l'application.
- L'appel système **stat** indique le nombre de fragments alloués à un fichier. Le nombre indiqué est basé sur les 4096 octets alloués aux blocs modifiés mais non écrits et à la taille compressée des blocs non modifiés. Les ressources déjà validées ne sont pas prises en compte par l'appel système **stat**. L'appel système **stat** indique le nombre correct de fragments alloués après une validation d'i-node si aucun des blocs modifiés n'est compressé. De même, les quotas de disque sont pris en compte pour l'allocation en cours. Lors de l'écriture des blocs logiques d'un fichier sur le disque, le nombre de fragments qui leur sont associés diminue s'ils sont compressés et ils changent donc les quotas de disque et le résultat de **stat**.

Algorithme de compression

L'algorithme de compression correspond à une version LZ IBM. En général, les algorithmes LZ compressent les données en représentant la deuxième et la dernière occurrence d'une chaîne donnée avec un pointeur qui identifie l'emplacement de la première occurrence de la chaîne et sa longueur. Au début de la compression, aucune chaîne n'est identifiée et en conséquence au moins le premier octet de données doit être représenté sous la forme d'un caractère "brut" nécessitant 9 bits (0, octet). Une fois qu'une certaine quantité de données est compressée, par exemple *N* octets, le compresseur recherche la plus longue chaîne dans les octets *N* qui correspond à la chaîne de début de l'octet suivant non compressé. Si la longueur la plus grande correspondante est égale à 0 ou 1, l'octet suivant est codé comme caractère "brut". Sinon, la chaîne est représentée sous la forme d'une paire

(pointeur, longueur) et le nombre d'octets traité est incrémenté de la longueur. D'un point de vue architectural, LZ supporte des valeurs N de 512, 1024 et 2048. LZ définit le codage des paires (pointeur, longueur) et les caractères bruts. Le pointer est un champ de longueur fixe de taille $\log_2 N$, alors que la longueur est codée sous la forme d'un champ de longueur variable.

Coûts de performances

Du fait que la compression des données est une extension de la fragmentation, les performances associées aux fragments s'appliquent à la compression des données. Les systèmes de fichiers compressés affectent également les performances des manières suivantes :

- La compression et la décompression des données peuvent nécessiter beaucoup de temps pour que l'utilisation d'un système de fichiers compressé puisse être limitée pour certains environnements utilisateur.
- La plupart des fichiers normaux UNIX sont écrits une seule fois, mais certains sont mis à jour sur place. Pour ces derniers, la compression des données ajoute une surcharge d'allocation de 4 096 octets d'espace disque lorsqu'un bloc logique est modifié, puis réalloue l'espace disque après l'écriture du bloc logique sur le disque. Cette activité d'allocation supplémentaire n'est pas nécessaire aux fichiers normaux d'un système de fichiers non compressé.
- La compression des données augmente le nombre de cycles de processeur. Pour le compresseur logiciel, le nombre de cycles de compression est d'environ 50 par octet et de 10 par octet pour la décompression.

Description des sauvegardes en ligne JFS et des clichés JFS2

Vous pouvez créer un cliché d'un système de fichiers JFS ou JFS2 (AIX 5.2 et versions supérieures) que vous pouvez utiliser ensuite pour les sauvegardes. Toutefois, il existe des différences dans les besoins et le comportement de cette image pour chaque type de système de fichiers.

Pour un système de fichiers JFS, vous pouvez diviser une copie statique en lecture seule d'une copie miroir du système de fichiers. Généralement, une copie miroir est mise à jour chaque fois que le système de fichiers est mis à jour, mais ce cliché ne change pas. Il s'agit d'une image stable au moment où la copie a été effectuée. Lorsque cette image est utilisée pour la sauvegarde, toute modification après le début de la procédure de création de l'image peut ne pas figurer dans la copie de sauvegarde. En conséquence, il est recommandé d'effectuer la division lorsque l'activité du système de fichiers est minimale. Toute modification effectuée après la division ne figure pas dans la copie de sauvegarde.

Pour un système de fichiers JFS2, l'image cliché s'appelle *cliché*. Le cliché reste statique et il conserve les autorisations de sécurité que le système de fichiers d'origine (appelé *snappedFS*) avait avant le cliché. En outre, vous pouvez créer un cliché JFS2 sans monter ou arrêter le système de fichiers. Vous pouvez utiliser un cliché JFS2 comme sauvegarde en ligne du système de fichiers, pour accéder aux fichiers ou aux répertoires tels qu'ils existaient lors du cliché ou pour sauvegarder sur un support amovible. Notez les éléments suivants sur les clichés JFS2.

- Un cliché du système de fichiers racine ($/$) ou du système de fichiers $/usr$ est remplacé lorsque le système est réamorcé. Les clichés des autres systèmes de fichiers peuvent être préservés en démontant les systèmes de fichiers avant de réamorcer le système. Les clichés créés dans AIX 5.2 avec 5200-01 sont récupérables. Lorsque **fsck** ou **logredo** est exécuté sur un système de fichiers JFS2 avec un cliché créé sur AIX 5.2 avec 5200-01, le cliché est préservé. Un système de fichiers proprement démonté avec un cliché AIX 5.2 est également récupérable une fois monté sur un système AIX 5.2 avec 5200-01.

- Il est recommandé de ne pas exécuter la commande **defragfs** sur un système de fichiers avec des clichés. Chaque bloc transféré lors de la défragmentation doit être également copié vers le cliché, ce qui prend du temps et génère une perte d'espace dans le volume logique du cliché.
- Si un cliché manque d'espace, tous les clichés de ce système de fichiers cliché sont supprimés. Cette erreur est consignée dans le journal des erreurs.
- Si l'écriture d'un cliché échoue, tous les clichés de ce système de fichiersz cliché sont supprimés. Cette erreur est consignée dans le journal des erreurs.
- Un cliché créé ou auquel vous accédez sur un système AIX 5.2 avec 5200–01 n'est pas accessible sur un système AIX 5.2. Ces clichés doivent être supprimés pour pouvoir monter le système de fichiers.

Description de la compatibilité et de la migration

Les systèmes de fichiers JFS sont totalement compatibles avec AIX 5.1 et AIX 5.2. Les versions précédentes prises en charge du système d'exploitation sont compatibles avec le système de fichiers actuel, bien que les systèmes de fichiers sans taille de fragment par défaut, valeur NBPI ou taille de groupe d'allocation puissent nécessiter une certaine attention s'ils sont migrés vers une version précédente.

Les systèmes de fichiers JFS2, à l'exception des clichés, sont compatibles avec AIX 5.1 et AIX 5.2, mais pas avec les versions précédentes du système d'exploitation. Les systèmes de fichiers JFS2 avec des clichés ne sont pas compatibles avec AIX 5.1. Veuillez à toujours démonter proprement tous les systèmes de fichiers JFS2 avant de revenir à une version précédente de AIX car la commande **logredo** ne s'exécute pas nécessairement sur un système de fichiers créé pour une version antérieure.

Les sections suivantes décrivent les éléments qui peuvent générer des problèmes avec les systèmes de fichiers créés avec des versions précédentes du système d'exploitation.

Images de système de fichiers JFS

Toute image de système de fichiers JFS créée avec la taille de fragment par défaut, la valeur NBPI 4 096 octets et la taille de groupe d'allocation par défaut (agsize) 8 peut être interchangeée avec une image de système de fichiers JFS créée sous AIX 4.3 et les versions antérieures de ce système d'exploitation sans activités de migration spéciale.

Remarque : Clichés JFS2 :

Les clichés JFS2 créés ou auxquels vous accédez dans AIX 5.2 avec 5200–01 ne sont pas accessibles avec les versions antérieures. Ces clichés doivent être supprimés pour pouvoir monter le système de fichiers.

Sauvegarde et restauration entre systèmes de fichiers

Systèmes de fichiers
Les séquences de sauvegarde et de restauration peuvent être effectuées entre des systèmes de fichiers ayant des tailles de blocs différentes. Toutefois, du fait de l'augmentation de l'utilisation du disque, les restaurations peuvent échouer du fait d'un manque de blocs libres si la taille de bloc du système de fichiers source est inférieure à celle du système de fichiers cible. Cela est particulièrement intéressant pour la sauvegarde et la restauration totale d'un système de fichiers et cette situation peut même se produire lorsque la taille totale du système de fichiers cible est supérieure à celle du fichier source.

Bien que les sauvegardes et les restaurations puissent être effectuées depuis des systèmes de fichiers compressés vers des systèmes de fichiers non compressés ou entre des systèmes de fichiers compressés ayant des tailles de différentes, les restaurations peuvent échouer du fait de l'utilisation étendue du disque des systèmes de fichiers compressés et du manque d'espace disque. Cela est particulièrement intéressant pour la sauvegarde et la restauration totale d'un système de fichiers et cette situation peut même se produire lorsque la taille totale du système de fichiers cible est supérieure à celle du fichier source.

Limitations des pilotes d'unités JFS et JFS2

Un pilote d'unité doit fournir une adressabilité de bloc de disque égale ou inférieure à la taille de fragment du système de fichiers JFS ou à la taille de bloc JFS2. Si, par exemple, un système de fichiers JFS a été créé sur un pilote d'unité de disque RAM fourni par l'utilisateur, le pilote doit permettre à des blocs de 512 octets de contenir un système de fichiers qui avait des segments de 512 octets. Si le pilote ne fournit qu'une adressabilité au niveau de la page, un système de fichiers JFS avec une taille de fragment de 4 096 octets peut être utilisé uniquement.

Description de CDRFS

Un système de fichiers CDRFS est un système de fichiers local en lecture seule sous la couche FLS (Logical File System). Il est stocké sur un support CD-ROM. Depuis la version AIX 5.2, les CD sont montés automatiquement par défaut, mais cette fonction peut être désactivée. Si la fonction est inactive, utilisez la commande **cdmount** pour monter le système de fichiers CDRFS.

Pour AIX 5.1 et les versions supérieures, montez le CD-ROM et son système de fichiers à l'aide de la commande **mount** et protégez le CD contre l'écriture. Par exemple :

```
mount -r -v cdrfs /dev/cd0 /mnt
```

Vous devez définir les éléments suivants lors du montage d'un CD-ROM (AIX 5.1 et versions supérieures) :

Nom d'unité Définit le nom de l'unité contenant le support.

Point de montage
 Définit le répertoire de montage du système de fichiers.

Montage automatique
 Indique si le système de fichiers sera monté automatiquement au démarrage du système.

AIX prend en charge le volume CDRFS et les formats de structure suivants :

Type	Description
Norme ISO 9660:1988(E)	CDRFS prend en charge l'interchange ISO 9660 niveau 3 et la mise en œuvre niveau 1.
High Sierra Group Specification	Précède la norme ISO 9660 et fournit une compatibilité amont avec les CD-ROM précédents.
Rock Ridge Group Protocol	Spécifie les extensions ISO 9660 totalement compatibles avec la norme ISO 9660 et qui fournissent une sémantique de système de fichiers complète POSIX basée sur SUSP (System Use Sharing Protocol) et RRIP (Rock Ridge Interchange Protocol) pour permettre le montage/l'accès CD-ROM comme avec n'importe quel autre système de fichiers UNIX.
Format de fichier CD-ROM eXtended Architecture (format de secteur Mode 2 Form 1 uniquement)	Le format de fichier CD-ROM eXtended Architecture (XA) spécifie les extensions de la norme ISO 9660 qui sont utilisées dans les applications multimédias CD-ROM telles que Photo CD.

Les restrictions suivantes s'appliquent à tous les formats de volumes et de structures de fichiers :

- Groupe de volumes à un seul volume uniquement
- Pas de fichiers entrelacés uniquement

CDRFS dépend du pilote d'unité CD-ROM sous-jacent pour fournir la transparence du format de secteur physiques (CD-ROM Mode 1 et CD-ROM XA Mode 2 Form 1) et le format multisession des disques (mappage du groupe de descripteurs de volumes depuis la zone de reconnaissance de volume de la dernière session).

Remarque : CDRFS doit être démonté depuis le système pour pouvoir retirer le support CD-ROM.

Description de UDFS

La version AIX 5.2 a introduit le type de système de fichiers UDFS qui est un système de fichiers en lecture seule stocké sur un support DVD-ROM. Les DVD sont montés automatiquement. UDFS doit être démonté depuis le système pour pouvoir retirer le support CD-ROM. Le système d'exploitation prend en charge le format UDFS des versions 1.50, 2.00 et 2.01.

Lorsqu'un DVD est monté automatiquement, le système de fichiers UDFS est accessible en lecture seule par défaut. Pour que la fonction automount ou la commande `cdmount` monte automatiquement un UDFS en écriture/lecture; modifiez le fichier `cdromd.conf`. Vous pouvez également monter manuellement un UDFS en lecture/écriture avec la commande **mount**.

Chapitre 6. Sauvegarde et restauration

Ce chapitre donne des informations relatives aux méthodes de sauvegarde et de restauration des données.

Les sujets traités sont les suivants :

- Généralités sur la sauvegarde
- Développement d'une stratégie de sauvegarde
- Sauvegarde des fichiers des utilisateurs ou des Systèmes de fichiers
- "Sauvegarde du système".

Généralités sur la sauvegarde

La sauvegarde des systèmes de fichiers, des répertoires et des fichiers prend du temps et implique de nombreuses tâches. D'autre part, tous les fichiers peuvent être modifiés ou supprimés par inadvertance ou par accident. Lors de la défaillance d'un disque dur, les informations stockées sur le disque sont perdues. La seule manière de restaurer les données détruites consiste à les extraire de la copie de sauvegarde. Si vous sauvegardez soigneusement et méthodiquement les systèmes de fichiers, vous pouvez toujours restaurer aisément les dernières versions des fichiers et des systèmes de fichiers.

Méthodes de sauvegarde

Il existe plusieurs méthodes de sauvegarde des données. La méthode la plus fréquente s'appelle la *sauvegarde par nom* ou l'*archivage par nom de fichier*. Cette méthode de sauvegarde est appliquée lorsque l'option *i* est définie et elle est utilisée pour créer une copie de sauvegarde de fichiers et de répertoires individuels. Cette méthode est communément utilisée par des utilisateurs individuels pour sauvegarder leurs comptes.

Une autre méthode fréquemment utilisée s'appelle la *sauvegarde par i-node* ou *archive de système de fichiers*. Cette méthode est appliquée lorsque l'option *i* n'est pas définie. Elle permet de sauvegarder une copie de l'ensemble d'un système de fichiers et elle est communément utilisée par les administrateurs système pour sauvegarder des groupes de fichiers volumineux, tels que tous les comptes utilisateurs, dans le répertoire **/home**. Une sauvegarde de système de fichiers permet d'effectuer aisément des sauvegardes incrémentielles. Une sauvegarde incrémentielle sauvegarde tous les fichiers modifiés depuis la dernière sauvegarde.

Les commandes **compress** et **pack** permettent de compresser des fichiers à stocker et les commandes **uncompress** et **unpack** permettent de décompresser les fichiers stockés. La compression et la décompression des fichiers prennent du temps, mais une fois compressées, les données occupent moins d'espace sur le support de stockage.

Plusieurs commandes permettent de créer des sauvegardes et des archives. Pour cette raison, les données sauvegardées doivent porter une étiquette qui indique la commande de sauvegarde utilisée et le type de la sauvegarde (par nom ou par système de fichiers).

sauvegarde	Sauvegardes des fichiers par nom ou par système de fichiers
mksysb	Crée une image installable du groupe de volumes rootvg.
cpio	Copie les fichiers vers ou depuis un stockage d'archives.
dd	Convertit et copie un fichier. Communément utilisée pour convertir et copier des données entre des systèmes exécutant des systèmes d'exploitation différents, tels que des mainframes. La commande dd ne place pas plusieurs fichiers dans une archive ; elle permet de manipuler et de déplacer des données.
tar	Crée ou manipule les archives de format tar .
rdump	Sauvegarde les fichiers par système de fichiers sur un périphérique d'une machine distante.
pax	(Utilitaire d'archivage compatible POSIX) Lit et écrit des archives tar et cpio .

Choix d'une politique de sauvegarde

Aucune stratégie de sauvegarde ne répond aux besoins de tous les utilisateurs. Une stratégie adaptée à un système et un utilisateur, par exemple, peut ne pas être adaptée à un système qui prend en charge cent utilisateurs. De même, une politique développée pour un système dont nombre de fichiers sont modifiés chaque jour ne convient pas à un système avec des données rarement modifiées. Quelle que soit la stratégie de sauvegarde sur votre site, l'important est qu'elle existe et qu'elle soit appliquée fréquemment et régulièrement. A défaut de stratégie efficace et opérationnelle, il peut s'avérer difficile de restaurer des données perdues.

C'est vous qui ferez le choix de la politique à adopter. Voici toutefois quelques règles qui vous guideront :

- Prévenir les pertes importantes de données

La poursuite de l'activité du système est-elle possible après la panne d'un disque dur, quel qu'il soit ? La reprise du système est-elle possible si tous les disques fixes sont en panne ? La reprise du système est-elle possible si vous égarez vos disquettes ou bandes de sauvegarde ? Pour récupérer des données en cas de perte, pouvez-vous mesurer les difficultés rencontrées ? Elaborez ensuite une politique de sauvegarde qui prenne toutes ces questions en compte.

- Vérifier régulièrement les sauvegardes

Les supports et unités matérielles de sauvegarde ne sont pas toujours fiables. Une bibliothèque volumineuse sur bandes ou disquettes de sauvegarde n'a de valeur que si les données sont restituables et lisibles sur disque. Pour vérifier que vos sauvegardes sont exploitables, affichez régulièrement la table des matières à partir de la bande de sauvegarde (avec la commande **restore -T** ou **tar -t** pour les bandes d'archives). Pour les sauvegardes sur disquettes, lisez régulièrement les disquettes, si possible à partir d'une unité de disquette différente de celle utilisée pour la création des sauvegardes. Pour plus de sécurité, vous pouvez doubler les sauvegardes de niveau 0 sur un deuxième support. Sur les bandes en continu contenant des sauvegardes, appliquez la commande **tapechk** qui vérifie sommairement la cohérence.

- Conserver les anciennes sauvegardes

Prévoyez des recyclages réguliers des supports de sauvegarde, sans toutefois les réutiliser tous. L'absence ou l'endommagement d'un fichier important n'est pas toujours détectée en temps réel. Il est donc conseillé de conserver quelques sauvegardes anciennes. Les recyclages suivants des bandes ou disquettes de sauvegarde sont indiqués à titre d'exemple :

- Une fois par semaine, recyclez les disquettes quotidiennes, excepté celle du vendredi.
- Une fois par mois, recyclez toutes les disquettes hebdomadaires, excepté la plus récente ; les sauvegardes des quatre derniers vendredis sont ainsi toujours disponibles.
- Tous les trimestres, recyclez les disquettes mensuelles, à l'exception de la dernière. Conservez en permanence la dernière disquette mensuelle de chaque trimestre sur un autre site.

- Vérifiez les systèmes de fichiers avant la sauvegarde

La sauvegarde d'un système de fichiers endommagés risque d'être inexploitable. Avant de faire des sauvegardes, il est bon de vérifier l'intégrité du système de fichiers avec la commande **fsck**.

- S'assurer que les fichiers ne sont pas en cours d'utilisation pendant la sauvegarde

Pendant les sauvegardes, le système ne doit pas être en cours d'utilisation. Sinon, les fichiers sont susceptibles d'être modifiés auquel cas les sauvegardes seraient inexactes.

- Sauvegarder le système avant toute modification majeure

La sauvegarde complète du système est toujours conseillée avant tout test ou réparation matérielle, avant l'installation d'une unité, d'un programme ou d'autres fonctions système.

Remarque : La sauvegarde des tubes désignés (fichiers spéciaux FIFO) fonctionne, que ceux-ci soient fermés ou ouverts. Toutefois, la restauration est impossible lorsque la sauvegarde est faite sur des tubes ouverts. Lors de la restauration d'un fichier spécial FIFO, son inode est le seul élément indispensable pour le recréer car il contient toutes ses caractéristiques. Le contenu d'un tube désigné n'est pas nécessaire à la restauration. C'est pourquoi, la taille du fichier lors de la sauvegarde doit être zéro (tous les FIFO fermés) avant de lancer cette procédure.

Attention : La sauvegarde et la restauration d'un système doivent être effectuées sur le même type de plate-forme. Les cartes principales CPU et d'E/S notamment doivent être de même type.

Description des supports de sauvegarde

Il existe divers types de supports de sauvegarde. Les types de supports de sauvegarde disponibles pour votre système dépendent de votre logiciel et de votre matériel. Des bandes, des disquettes, des archives distantes et des disques durs locaux sont généralement utilisés. Si vous ne définissez pas une autre unité à l'aide de la commande **backup -f**, la commande **backup** écrit automatiquement sa sortie sur **/dev/rfd0**, à savoir le lecteur de disquette. **Attention :** L'exécution de la commande **backup** provoque la perte de toutes les données stockées sur le support de sortie sélectionné.

Restauration des données

Il existe plusieurs méthodes. Choisissez-en une compatible avec celle utilisée pour la sauvegarde.

Vous devez connaître la méthode adoptée pour la sauvegarde ou l'archivage effectué. Chaque procédure de sauvegarde fournit des informations sur la restauration des données. Par exemple, si vous utilisez la commande **backup**, vous pouvez spécifier une sauvegarde par système de fichiers ou par nom. Une sauvegarde effectuée par système de fichiers ou par nom doit être restaurée de la même manière.

Voici les différentes commandes relatives à la restauration des données :

restore	Copie les fichiers créés par la commande backup .
rrestore	Copie les systèmes de fichiers sauvegardés sur une machine distante vers la machine locale.
cpio	Copie les fichiers vers ou depuis un stockage d'archives.
tar	Crée ou manipule les archives tar .
pax	(Utilitaire d'archivage compatible POSIX) Lit et écrit des archives tar et cpio .

Développement d'une stratégie de sauvegarde

Il existe deux méthodes de sauvegarde de grandes quantités de données :

- la sauvegarde intégrale du système,
- la sauvegarde incrémentale.

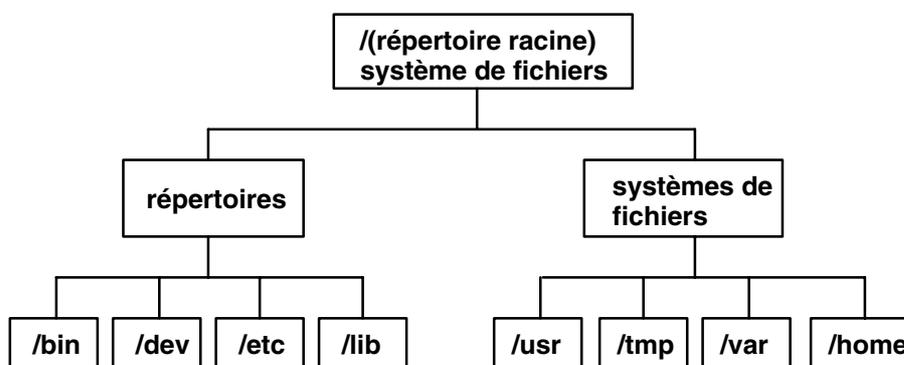
Pour choisir la méthode la plus adaptée à votre site ou à votre système, il est important de bien appréhender la structure du système de fichiers et le placement des données. La stratégie de placement des données doit être déterminée avant de développer une stratégie de sauvegarde. Reportez-vous à la section "Planification des sauvegardes" dans le manuel *AIX - Guide d'administration : système d'exploitation et unités* pour un exemple de stratégie combinant la sauvegarde intégrale, hebdomadaire et la sauvegarde incrémentale quotidienne.

Structure du système de fichiers

Notez bien la différence entre un système de fichiers et un répertoire. Un système de fichiers est une section du disque affectée aux données. L'accès à cette section se fait par montage du système de fichiers sur un répertoire. Du point de vue utilisateur, ce système de fichiers, une fois monté, ressemble à un répertoire. Toutefois, en raison des différences structurelles entre les systèmes de fichiers et les répertoires, les données à l'intérieur de ces deux entités peuvent être gérées séparément.

Lorsque le système d'exploitation est installé pour la première fois, il est chargé dans une structure de répertoire, comme indiqué dans l'illustration de l'arborescence du système de fichiers **/root**.

Figure 11. Arborescence de système de fichiers / (racine) Cette arborescence montre une structure de répertoires avec le système de fichiers / (racine) et les ramifications vers les répertoires et systèmes de fichiers. Les répertoires se ramifient à **/bin**, **/dev**, **/etc** et **/lib**. Les systèmes de fichiers se ramifient à **/usr**, **/tmp**, **/var** et **/home**.



Les répertoires de droite (**/usr**, **/tmp**, **/var** et **/home**) sont tous des systèmes de fichiers, affectés d'une section du disque. Ces systèmes de fichiers sont montés automatiquement à l'amorçage du système ; c'est pourquoi l'utilisateur ne les différencie pas des répertoires de gauche (**/bin**, **/dev**, **/etc** et **/lib**).

Données système et données utilisateur

Les données (programmes ou texte) sont réparties ici en deux catégories :

- Les données système, qui établissent la relation au système d'exploitation et à ses extensions. Elles doivent toujours figurer dans les systèmes de fichiers système, / (racine), **/usr**, **/tmp**, **/var**, etc.
- Les données utilisateur, qui sont exploitées localement par les utilisateurs pour effectuer des tâches spécifiques. Elles doivent être stockées dans le système de fichiers **/home** ou dans des systèmes de fichiers créés à cet effet.

Les applications utilisateur et le texte ne doivent en aucun cas être placés dans des systèmes de fichiers dédiés aux données système. Elles doivent être placées plutôt, par exemple, dans un système de fichiers créé par l'administrateur et monté sur un répertoire nommé **/local.tmp**, qui est utilisé pour le stockage temporaire des données systèmes et utilisateur, constitue une exception.

Sauvegarde

Les sauvegardes de données système et utilisateur sont conservées en cas de destruction accidentelle ou de défaillance d'un disque. Il est plus facile de gérer des sauvegardes distinctes pour les données système et les données utilisateur. Les raisons sont les suivantes :

- Les données utilisateur sont beaucoup plus souvent modifiées que les données du système d'exploitation. En outre, les images de sauvegarde sont beaucoup plus petites quand les deux types de données ne sont pas sur la même image. Par ailleurs, le nombre d'utilisateurs affecte le support et la fréquence du stockage nécessaires.
- La restauration des données utilisateur est plus facile et plus rapide quand la sauvegarde est séparée des données système. La restauration du système d'exploitation en même temps que des données utilisateur prend plus de temps et demande plus d'efforts. En effet, pour restaurer les données système, il faut amorcer le système à partir d'un support amovible (bande ou CD-ROM) puis installer la sauvegarde système.

Pour sauvegarder les données système, démontez tous les systèmes de fichiers utilisateur, y compris **/home**, à l'aide de la commande **umount**. Si ces systèmes de fichiers sont en cours d'utilisation, vous ne pouvez pas les démonter. Planifiez les sauvegardes aux heures creuses pour qu'ils puissent être démontés. Si vous ne démontez pas les systèmes de fichiers des données utilisateur, ils sont sauvegardés avec les données du système d'exploitation. Utilisez la commande **mount** pour ne monter que les systèmes de fichiers du système d'exploitation.

Les seuls systèmes de fichiers montés sont **/**, **/usr**, **/var** et **/tmp** et la sortie de la commande **mount** doit se présenter comme suit :

node	mounted	mounted over	vfs	date	options
	/dev/hd4	/	jfs	Jun 11 10:36	rw,log=/dev/hd8
	/dev/hd2	/usr	jfs	Jun 11 10:36	rw,log=/dev/hd8
	/dev/hd9var	/var	jfs	Jun 11 10:36	rw,log=/dev/hd8
	/dev/hd	/tmp	jfs	Jun 11 10:36	rw,log=/dev/hd8

Une fois tous les systèmes de fichiers utilisateur démontés, reportez-vous à *Backing Up the System Image and User-Defined Volume Groups*, pour plus de détails sur la sauvegarde des données du système d'exploitation.

Quand la sauvegarde du système d'exploitation est terminée, montez le système de fichiers utilisateur avec la commande **smit mount**. Vous pouvez ensuite sauvegarder fichiers, systèmes de fichiers ou autres groupes de volumes, en fonction de vos besoins. Les procédures correspondantes sont décrites plus loin dans ce chapitre.

Reproduction d'un système (clonage)

Le clonage permet de sauvegarder les données de configuration avec les données utilisateur ou les données système. La reproduction d'un système ou d'un groupe de volumes est parfois appelée clonage. L'image obtenue est installable sur un autre système et donc exploitable comme sur le premier système. La commande **mksysb** sert au clonage du groupe de volumes rootvg, qui contient le système d'exploitation tandis que la commande **savevg** sert au clonage des autres groupes de volumes. Les procédures de sauvegarde du système et des groupes de volumes utilisateur sont décrites plus loin dans ce chapitre.

Sauvegarde des fichiers des utilisateurs ou des Systèmes de fichiers

Pour sauvegarder les fichiers utilisateur et les systèmes de fichiers utilisateur, vous pouvez utiliser Web-based System Manager, les raccourcis SMIT **smit backupfile**, ou **smit backupfilesys** ou les commandes listées dans les méthodes de sauvegarde.

Vous pouvez utiliser l'interface SMIT pour sauvegarder un seul fichier et des petits systèmes de fichiers par nom, tel que **/home**, sur votre système local. Notez que SMIT ne peut créer des archives que dans le format fourni par la commande **backup**. En outre, les options de la commande **backup** ne sont pas toutes disponibles dans SMIT. SMIT peut se bloquer si plusieurs bandes ou disques sont nécessaires au cours de la sauvegarde. (Pour plus d'informations, reportez-vous à la description de la commande **backup** dans le document *AIX 5L Version 5.2 Commands Reference*.)

Utilisez la commande **backup** pour sauvegarder des systèmes de fichiers volumineux. Vous pouvez définir un numéro de niveau pour contrôler la quantité de données à sauvegarder (sauvegarde complète, 0; sauvegarde incrémentielle, 1–9). Seule la commande **backup** permet de définir le numéro de niveau des sauvegardes.

La commande **backup** crée des copies dans l'un des deux formats de sauvegarde suivants :

- Fichiers spécifiques sauvegardés par nom à l'aide de l'option **-i**.
- Systèmes de fichiers entiers sauvegardés par i-node à l'aide des paramètres **-Level** et **FileSystem**. Le système de fichiers est défragmenté lors de la restauration depuis la sauvegarde. **Attention** : La sauvegarde par i-node ne fonctionne pas correctement avec les fichiers dont l'ID utilisateur (UID) ou l'ID de groupe (GID) est supérieur à 65 535. Ces fichiers sont sauvegardés avec l'D tronqué et les attributs seront donc erronés lors de la restauration. Dans ce cas, vous devez effectuer une sauvegarde par nom.

Sauvegarde de l'image système et des groupes de volumes définis par l'utilisateur

Le groupe de volumes est stocké sur un disque dur ou un groupe de disques durs, et contient les fichiers de démarrage, BOS, les informations de configuration et les produits logiciels optionnels. Un *groupe de volumes défini par l'utilisateur* (appelé également *groupe de volumes nonrootvg*) contient généralement des fichiers de données et des applications. Vous pouvez sauvegarder une image du système et des groupes de volumes à l'aide de WSM (Web-based System Manager), SMIT ou de procédures de commandes.

Le *groupe de volumes rootvg* est un disque ou un groupe de disques contenant les fichiers de démarrage du système (BOS), les données de configuration et tout autre produit logiciel en option. Un *groupe de volumes utilisateur* (ou *groupe de volumes non rootvg*) contient les fichiers de données et les logiciels d'application.

Les procédures SMIT et Web-based System Manager font appel à la commande **mksysb** pour créer une image de sauvegarde, stockée sur bande ou dans un fichier. Si vous optez pour la bande, le programme de sauvegarde écrit une *image d'amorçage* sur la bande et l'image obtenue pourra donc servir à l'installation.

Remarque :

1. Vous ne pouvez pas créer des bandes de démarrage ou en utiliser pour démarrer un ordinateur personnel PowerPC.
2. Si vous optez pour SMIT, installez d'abord l'ensemble de fichiers **sysbr** dans le progiciel **bos.sysmgt**. Reportez-vous à "Installation de logiciels en option et de mises à jour de service à l'aide de SMIT" dans *AIX 5L Version 5.2 Guide d'Installation*.

Configuration du système source

Configurez le système source avant de créer son image de sauvegarde. Faites-le, excepté si cette image est destinée à l'installation d'autres systèmes (cible) dont les configurations prévues sont différentes.

Le système *source* est celui à partir duquel vous créez la copie de sauvegarde. Le système *cible* est celui sur lequel vous installez cette copie.

Le programme d'installation n'installe que le support logiciel d'unité correspondant à la configuration matérielle de la machine. Aussi, si vous prévoyez d'utiliser une copie du système pour installer d'autres machines, vous aurez probablement d'autres unités à installer sur le système source avant d'en faire l'image de sauvegarde.

Utilisez Web-based System Manager ou le raccourci SMIT **smit devinst** pour installer un support de périphérique supplémentaire sur le système source.

- Si le système source et cible disposent de suffisamment d'espace disque, installez l'ensemble du support logiciel d'unité.
- Si l'espace disque est limité, n'installez que les supports indispensables.

Pour plus d'informations sur l'installation d'un logiciel optionnel, reportez-vous à la section *Installing Optional Software and Service Updates Using SMIT* du document *AIX 5L Version 5.2 Installation Guide and Reference*.

Sont transférées par la sauvegarde, du système source vers le système cible, les données relatives :

- à l'espace de pagination,
- aux volumes logiques,
- Informations rootvg
- Emplacement des partitions logiques (si vous avez sélectionné l'option Map).

Pour plus d'informations sur la définition des paramètres d'installation lors de l'installation de la machine cible depuis une sauvegarde système, reportez-vous à Customizing the BOS Install Program dans le document *AIX 5L Version 5.2 Installation Guide and Reference*.

Montage et démontage des systèmes de fichiers

La procédure "Sauvegarde du système" est dédiée exclusivement à la sauvegarde des systèmes de fichiers montés dans rootvg. De ce fait, vous devez monter avant la sauvegarde tous les systèmes de fichiers que vous voulez inclure dans la sauvegarde. A l'inverse, vous devez démonter tous ceux que vous ne souhaitez *pas* sauvegarder.

La procédure effectue une double sauvegarde des fichiers figurant dans un répertoire local monté sur un autre répertoire local du même système de fichiers. Par exemple, si **/tmp** est monté sur **/usr/tmp**, les fichiers de **/tmp** sont sauvegardés deux fois. Cette duplication peut provoquer le dépassement du nombre de fichiers admis et, par conséquent, l'échec de la future installation de l'image de sauvegarde.

Remarques sur la sécurité

Si vous installez une image de sauvegarde sur d'autres systèmes, pour des raisons de sécurité, les mots de passe et les adresses de réseau ne doivent pas être copiés sur les systèmes cible. Et ce d'autant que des adresses en double peuvent provoquer l'interruption des communications sur le réseau.

Restauration d'une image de sauvegarde

Pendant l'installation de l'image de sauvegarde, le système vérifie si l'espace disque du système cible est suffisant pour créer tous les volumes logiques stockés sur la sauvegarde. Si cet espace est suffisant, la restauration complète de l'image est possible. Sinon, la procédure s'arrête et le système vous invite à sélectionner des disques supplémentaires.

Les systèmes de fichiers créés sur le système cible ont la même taille que sur le système source, sauf si la variable **SHRINK** était définie à **oui** dans le fichier **image.data** avant l'exécution de la sauvegarde. Une exception cependant : le répertoire **/tmp**, qui peut être agrandi pour réserver suffisamment d'espace à la commande **bosboot**. Pour plus d'informations sur la définition des variables, reportez-vous au fichier **image.data** dans le document *AIX 5L Version 5.2 Files Reference*.

Une fois que le système a terminé d'installer l'image de sauvegarde, le programme d'installation reconfigure ODM sur le système cible. Si le système cible n'a pas exactement la même configuration matérielle que le système source, le programme modifie des attributs d'unités dans les fichiers suivants du système cible :

- Tous les fichiers qui commencent par "Cu" dans **/etc/objrepos**.
- Tous les fichiers du répertoire **/dev**.

Pour plus d'informations sur l'installation (ou la restauration) d'une image de sauvegarde, reportez-vous à la section Installing BOS from a System Backup dans le document *AIX 5L Version 5.2 Installation Guide and Reference*.

Chapitre 7. Environnement système

A la base, l'environnement système est l'ensemble de variables qui définissent ou contrôlent certains aspects de l'exécution des processus. Ces variables sont définies ou redéfinies à chaque démarrage d'un shell. Du point de vue de l'administrateur système, il est important de garantir des valeurs correctes pour la connexion de l'utilisateur. La plupart de ces variables sont définies lors de l'initialisation du système, soit par défaut, soit en fonction des valeurs lues dans le fichier **/etc/profile**.

Les sujets abordés sont les suivants :

- "Profils - Généralités"
- Services de manipulation des données sur l'heure
- Activation de la fonction de Mise hors service dynamique d'un processeur

Profils - généralités

Lors de la connexion au système d'exploitation, le shell utilise deux types de fichiers de profils. Il analyse les commandes figurant dans ces fichiers puis les exécute pour définir votre environnement système. Ces fichiers ont des fonctions similaires à ceci près que **/etc/profile** contrôle les variables de profil concernant l'ensemble des utilisateurs du système tandis que **.profile** permet de personnaliser votre propre environnement.

Ce chapitre traite des points suivants :

- Fichier **/etc/profile**,
- Fichier **.profile**,
- Modification de la date/heure système dans le manuel ,
- Modification du message du jour dans le manuel,
- Services de manipulation des données sur l'heure.

Fichier **/etc/profile**

/etc/profile est le premier fichier qu'utilise le système d'exploitation au moment de la connexion. Il contrôle les variables par défaut à l'échelle du système, telles que :

- les variables d'exportation,
- le masque de création de fichier (umask),
- les types de terminaux,
- les messages signalant l'arrivée du courrier.

L'administrateur système configure le fichier **profile** pour tous les utilisateurs du système. Il est le seul à pouvoir modifier ce fichier.

fichier **.profile**

.profile est le deuxième fichier qu'utilise le système d'exploitation au moment de la connexion. Ce fichier est présent dans votre répertoire personnel (**\$HOME**) ; il vous permet de personnaliser votre environnement de travail. Les commandes de **.profile** ont la priorité sur celles de **/etc/profile**. Etant donné que le fichier **.profile** est caché, utilisez la commande **ls-a** pour le répertorier. Utilisez le fichier **.profile** pour contrôler les défauts suivants :

- les shells à ouvrir,
- l'apparence de l'invite,
- les variables d'environnement (par exemple, les variables de chemin d'accès),
- le son du clavier.

L'exemple suivant illustre un fichier **.profil** courant :

```
PATH=/usr/bin:/etc:/home/bin1:/usr/lpp/tps4.0/user:/home/gsc/bin::
epath=/home/gsc/e3:
export PATH epath
csh
```

Dans cet exemple, deux chemins d'accès ont été définis (`PATH` et `epath`) puis exportés et un shell C a été ouvert (`csH`).

Le fichier **.profile** (ou, à défaut, le fichier **profile**) sert aussi à déterminer les variables shell de connexion. En outre, vous pouvez personnaliser les autres environnements shell. Par exemple, utilisez les fichiers **.chsrc** et **.kshrc** pour personnaliser un shell C et un shell Korn au démarrage de ces deux shells.

Services de manipulation des données sur l'heure

Les fonctions de date/heure servent à accéder à la date et l'heure courantes du système et à modifier leur format. Aucun indicateur n'est à spécifier au compilateur pour exploiter ces fonctions.

Ajoutez le fichier d'en-tête de ces fonctions dans le programme. Pour inclure un fichier d'en-tête, procédez comme suit :

```
#include <time.h>
```

Voici la liste des services de date/heure :

adjtime ajuste l'heure pour synchroniser l'horloge système.

ctime, localtime, gmtime, mktime, difftime, asctime, tzset

convertit la date et l'heure selon la représentation de la chaîne.

getinterval, incinterval, absinterval, resinc, resabs, alarm, ualarm, getitimer, setitimer

gère l'heure d'expiration de plusieurs horloges.

gettimer, settimer, restimer, stime, time

recherche ou définit la valeur courante de l'horloge spécifiée à l'échelle du système.

gettimerid affecte une horloge par processus.

gettimeofday, settimeofday, ftime

recherche et définit la date et l'heure.

nsleep, usleep, sleep

met un processus en veille.

reltimerid libère une horloge affectée.

Désallocation dynamique de processeur

Depuis l'existence de la machine de type 7044, modèle 270, le matériel de tous les systèmes dotés de plus de deux processeurs peut détecter les erreurs de connexion collectées par le microcode. Ces erreurs ne sont pas irrémédiables, tant qu'elles ne sont pas récurrentes, et peuvent être ignorées. Toutefois, lorsqu'un type d'échec semble se développer sur un processeur, le type d'incident peut indiquer que le composant est sur le point de connaître une erreur irrémédiable. Cette prévision est effectuée par le microcode en fonction du taux d'échec et de l'analyse des seuils.

Sur ces systèmes, AIX surveille en continu le matériel et appelle régulièrement le microcode pour identifier les erreurs matérielles. Lorsque le nombre d'erreurs de processeur atteint un seuil et que le microcode détermine que le composant système va vraisemblablement connaître une défaillance, le microcode envoie un rapport d'erreur. Dans tous les cas, l'erreur est consignée dans le journal des erreurs du système. En outre, sur les systèmes à plusieurs processeurs, selon le type d'échec, AIX tente d'arrêter en utilisant le processeur non fiable et le désalloue. Cette fonction s'appelle la *désallocation dynamique de processeur*.

A ce stade, le processeur est également marqué par le microcode pour indiquer qu'il est désalloué en permanence pour les réamorçages suivants jusqu'à ce que le personnel de maintenance le remplace.

Impact potentiel sur les applications

La désallocation de processeur est transparente pour la très grande majorité des applications, y compris pour les pilotes et les extensions du noyau. Toutefois, vous pouvez utiliser les interfaces publiées pour déterminer si une application ou une extension du noyau est exécutée sur une machine à plusieurs processeurs, identifier le nombre de processeurs et lier les threads à des processeurs spécifiques.

L'interface `bindprocessor` de liaison des processus ou des threads aux processeurs utilise des numéros de liaison UC. Ces numéros sont compris entre $[0.. N-1]$ où N correspond au nombre total d'UC. Pour éviter de rompre les applications ou les extensions de noyau qui n'acceptent pas de "trous" dans la numérotation des UC, AIX fait apparaître l'UC aux applications comme s'il s'agissait de la "dernière" UC liée (celle qui porte le numéro le plus grand) à désallouer. Par exemple sur un SMP à 8 voies, les numéros d'UC liées sont $[0..7]$. Si un processeur est désalloué, le nombre total d'UC disponibles devient 7 et elles sont numérotées $[0..6]$. En externe, il semble que l'UC 7 ait disparue, quel que soit le processeur physique qui connaît une défaillance.

Remarque : Dans le reste de cette description, le terme *UC* désigne l'entité logique et le terme *processeur* désigne l'entité physique.

Potentiellement, les applications ou les extensions de noyau qui sont des processus de liaison ou des threads peuvent être rompues si AIX termine silencieusement leurs threads de liaison ou force leur transfert vers une autre UC lorsque des processeurs doivent être désalloués. La désallocation dynamique de processeur fournit des interfaces de programmation pour que les applications et les extensions de noyau puissent savoir qu'une désallocation de processeur va se produire. Lorsque les applications et les extensions de noyau sont averties, elles doivent transférer leurs threads liées et les ressources associées (tels que les blocs de demande de synchronisation) depuis le dernier ID d'UC liée et s'adapter à la nouvelle configuration d'UC.

Après la notification, si des threads restent liées au dernier ID d'UC liée, la désallocation est abandonnée, la désallocation abandonnée est consignée dans le journal des erreurs et AIX continue en utilisant le processeur défaillant. Lorsque le processeur tombe en panne, il provoque une erreur générale du système. En conséquence, il est important que les applications et les extensions de noyau soient informées de l'imminence de la désallocation d'un processeur et puissent réagir de manière appropriée.

Même dans les cas rares où la désallocation n'aboutit pas, la désallocation dynamique de processeur avertit toujours à l'avance les administrateurs système. En enregistrant l'erreur dans le journal des erreurs, ils ont la possibilité de planifier une opération de maintenance sur le système pour remplacer le composant défaillant avant qu'une erreur générale du système se produise.

Désallocation de processeur

Le flux type d'événements pour la désallocation de processeur est le suivant :

1. Le microcode détecte qu'un processeur a atteint un seuil d'erreur irrémédiable.
2. Le rapport d'erreur du microcode est consigné dans le journal des erreurs du système et, lorsque AIX est exécuté sur une machine qui prend en charge la désallocation de processeur, il lance la désallocation.
3. AIX signale les processus non-noyau et les threads liées à la dernière UC liée.
4. AIX attend jusqu'à 10 minutes que toutes les threads liées soit transférées depuis la dernière UC liée. Si des threads restent liées, AIX abandonne la désallocation.
5. Si tous les processeurs ou threads ne sont plus liés au processeur défaillant, les descripteurs HAEH (High Availability Event Handlers) déjà enregistrés sont appelés. Un HAEH peut retourner une erreur qui fait échouer la désallocation.
6. Si elle n'est pas abandonnée, le processus de désallocation arrête le processeur défaillant.

En cas d'échec au cours de la désallocation, l'échec et sa cause sont consignés. L'administrateur système peut consulter le journal des erreurs, exécuter les actions appropriées (si possible) et relancer la désallocation. Par exemple, si la désallocation a été abandonnée parce qu'une application n'a pas pu délier ses threads liées, l'administrateur système peut arrêter l'application, relancer la désallocation, puis redémarrer l'application.

Activation et désactivation de la désallocation de processeur

La désallocation dynamique de processeur peut être activée ou désactivée en changeant la valeur de l'attribut **cpuguard** de l'objet ODM `sys0`. Les valeurs possibles de l'attribut sont **enable** et **disable**.

Depuis la version AIX 5.2, la valeur par défaut est **enabled** (l'attribut **cpuguard** est affecté de la valeur **enable**). Les administrateurs système qui veulent désactiver cette fonction doivent utiliser les menus système Web-based System Manager, le menu SMIT System Environments ou la commande **chdev**. Dans les versions précédentes de AIX, la valeur par défaut était **disabled**.)

Remarque : Les erreurs sont toujours consignées, même lorsque la désallocation de processeur est désactivée. Le journal des erreurs contient une erreur telle que `CPU_FAILURE_PREDICTED` qui indique que AIX a été averti d'un problème d'UC.

Relance d'une désallocation de processeur abandonnée

Parfois, la désallocation de processeur échoue parce qu'une application n'a pas transféré ses threads liées depuis la dernière UC logique. Une fois ce problème résolu, par la suppression de liaison (lorsque cette opération est sûre) ou par l'arrêt de l'application, l'administrateur système peut redémarrer la désallocation, de processus avec la commande **ha_star**.

La syntaxe de la commande est la suivante :

```
ha_star -C
```

où **-C** correspond à l'événement de défaillance UC prévisible.

Considérations relatives à l'état du processeur

Les processeurs physiques sont représentés dans la base de données ODM par des objets **proc n** où *n* est un nombre décimal qui correspond au numéro du processeur physique. A l'instar des autres unités représentées dans la base de données ODM, les objets processeur ont un état, tel que Défini/Disponible, et des attributs.

L'état d'un objet **proc** est toujours Disponible tant que le processeur est présent, qu'il soit ou non utilisable. L'attribut **state** d'un objet **proc** indique si un processeur est utilisé, et s'il ne l'est pas, il en indique la raison. Cet attribut peut avoir trois valeurs :

enable	Le processeur est utilisé.
disable	Le processeur a été désalloué dynamiquement.
faulty	Le processeur est déclaré défaillant par le microcode au démarrage.

Si un processeur défaillant est désalloué, son état passe de **enable** à **disable**. Indépendamment de AIX, le processeur est marqué comme défaillant dans le microcode. Lors de l'amorçage, le processeur désalloué n'est pas disponible et il a l'état **faulty**. L'objet ODM **proc**, toutefois, est toujours disponible. Vous devez généralement retirer l'UC défaillante de la carte système ou retirer la carte UC (si possible) de l'objet **proc** pour définir l'état Défini.

Exemple

Dans le scénario suivant, le processeur **proc4** fonctionne correctement et il est utilisé par le système d'exploitation, comme l'indique la sortie suivante :

```
# lsattr -EH -l proc4
attribute value description user_settable

state enable Processor state False
type PowerPC_RS64-III Processor type False
#
```

Lorsque le processeur **proc4** reçoit un échec prévisible, il est désalloué par le système d'exploitation, comme indiqué ci-dessous :

```
# lsattr -EH -l proc4
attribute value description user_settable

state disable Processor state False
type PowerPC_RS64-III Processor type False
#
```

Lors du redémarrage, le microcode signale que le processeur **proc4** est défaillant, comme indiqué ci-dessous :

```
# lsattr -EH -l proc4
attribute value description user_settable

state faulty Processor state False
type PowerPC_RS64-III Processor type False
#
```

Mais l'état du processeur **proc4** est toujours Disponible, comme indiqué ci-dessous :

```
# lsdev -CH -l proc4
name status location description

proc4 Available 00-04 Processor
#
```

Entrées du journal des erreurs

Trois messages d'erreurs sont associés à la désallocation d'UC dans le journal des erreurs. Exemples :

errpt short format – summary

Voici un exemple d'entrées affichées par la commande **errpt** (sans options) :

```
# errpt
IDENTIFIER      TIMESTAMP      T    C    RESOURCE_NAME
DESCRIPTION
804E987A        1008161399    I    O    proc4          CPU
DEALLOCATED
8470267F        1008161299    T    S    proc4          CPU
DEALLOCATION ABORTED
1B963892        1008160299    P    H    proc4          CPU
FAILURE PREDICTED
#
```

- Si la désallocation de processeur est active, le message CPU FAILURE PREDICTED est toujours suivi du message CPU DEALLOCATED ou du message CPU DEALLOCATION ABORTED.
- Si la désallocation de processeur n'est pas active, seul le message CPU FAILURE PREDICTED est présent. L'activation de la désallocation de processeur, après qu'un ou plusieurs messages CPU FAILURE PREDICTED ont été consignés, démarre le processus de désallocation et génère une entrée de succès ou d'échec dans le journal des erreurs, comme indiqué ci-dessus, pour chaque processeur signalé défaillant.

errpt long format – detailed description

Voici la sortie obtenue avec **errpt -a**:

• CPU_FAIL_PREDICTED

Error description: Predictive Processor Failure

Cette erreur indique que le matériel a détecté qu'un processeur va vraisemblablement connaître une défaillance très bientôt. Elle est toujours consignée, que la désallocation de processeur soit active ou non.

DETAIL DATA: *Physical processor number, location*

Exemple d'entrée du journal des erreurs – forme longue

LABEL: CPU_FAIL_PREDICTED
IDENTIFIÉRIER: 1655419A

Date/Time: Thu Sep 30 13:42:11
Sequence Number: 53
Machine Id: 00002F0E4C00
Node Id: auntbea
Class: H
Type: PEND
Resource Name: **proc25**
Resource Class: processor
Resource Type: proc_rspc
Location: **00-25**

Description
CPU FAILURE PREDICTED

Probable Causes
CPU FAILURE

Failure Causes
CPU FAILURE

Recommended Actions
ENSURE CPU GARD MODE IS ENABLED
RUN SYSTEM DIAGNOSTICS.

Detail Data
PROBLEM DATA

0144	1000	0000	003A	8E00	9100	1842	1100
1999	0930	4019					
	0000	0000	0000	0000	0000		
	0000	0000	0000	0000	0000	0000	0000
4942	4D00	5531					
	2E31	2D50	312D	4332	0000		
	0002	0000	0000	0000	0000	0000	0000
0000	0000	0000					
	0000	0000	0000	0000	0000		
	0000	0000	0000	0000	0000	0000	0000
0000	0000	0000					
	0000	0000	0000	0000	0000		
	0000	0000	0000	0000	0000		
...			

. CPU_DEALLOC_SUCCESS

Description de l'erreur : Un processeur a été désalloué après la détection d'un échec de processeur prévisible. Ce message est consigné lorsque la désactivation de processeur est active et que l'UC a été désallouée.

DETAIL DATA: Logical CPU number of deallocated processor.

Exemple : Exemple d'entrée du journal des erreurs – forme longue :

```
LABEL:          CPU_DEALLOC_SUCCESS
IDENTIFIER:     804E987A

Date/Time:     Thu Sep 30 13:44:13
Sequence Number: 63
Machine Id:    00002F0E4C00
Node Id:      auntbea
Class:        O
Type:         INFO
Resource Name:      proc24
```

```
Description
CPU DEALLOCATED
```

```
Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
```

```
Detail Data
LOGICAL DEALLOCATED CPU NUMBER
```

0

Dans cet exemple, **proc24** a été désalloué et correspondait à l'UC logique **0** lors de la défaillance.

. CPU_DEALLOC_FAIL

Description de l'erreur : La désallocation d'un processeur, à la suite d'un échec de processeur prévisible, n'a pas abouti. Ce message est consigné lorsque la désactivation de processeur est active et que l'UC a été désallouée.

DETAIL DATA: *Code d'erreur, numéro d'UC logique, informations complémentaires en fonction du type d'échec.*

Le code d'erreur est une valeur numérique hexadécimale. Les codes d'erreur possibles sont :

- 2 Un(e) ou plusieurs processus/threads sont toujours lié(e)s à la dernière UC logique. Dans ce cas, les données détaillées indiquent les PID des processus concernés.
 - 3 Un pilote ou une extension de noyau enregistrée ont renvoyé un code d'erreur lorsqu'ils ont été notifiés. Dans ce cas, le champ des données détaillées contient le nom du pilote ou de l'extension de noyau concerné (en ASCII).
 - 4 En désallouant un processeur, la machine a moins de deux UC disponibles. Ce système d'exploitation ne désalloue pas plus de $N - 2$ processeurs sur une machine N voies pour éviter de perturber les applications ou les extensions de noyau qui utilisent le nombre total de processeurs disponibles pour déterminer s'ils fonctionnent sur un système à un seul processus sur lequel il est possible de ne pas utiliser les verrous multiprocesseur ou un processeur SMP (Symmetric Multi Processor).
- 200 (0xC8)** La désallocation de processeur est désactivée (l'attribut ODM **cpuguard** a la valeur **disable**). Cette erreur n'apparaît généralement pas lorsque vous ne démarrez pas **ha_star** manuellement.

Exemples : entrée du journal des erreurs – forme longue :

Exemple 1 :

```
LABEL:          CPU_DEALLOC_ABORTED
IDENTIFIER:     8470267F
Date/Time:     Thu Sep 30 13:41:10
Sequence Number:  50
Machine Id:     00002F0E4C00
Node Id:       auntbea
Class:         S
Type:         TEMP
Resource Name:  proc26

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0003
DEALLOCATION ABORTED CAUSE
6676 6861 6568 3200
```

Dans cet exemple, la désallocation de **proc26** a échoué. Le code d'erreur **3** indique qu'une extension de noyau a retourné une erreur à la routine de notification du noyau. DEALLOCATION ABORTED DATA ci-dessus indique **fvhaeh2** qui correspond au nom de l'extension utilisée pour l'enregistrement avec le noyau.

Exemple 2 :

```
LABEL:          CPU_DEALLOC_ABORTED
IDENTIFIER:     8470267F
Date/Time:     Thu Sep 30 14:00:22
Sequence Number:  71
Machine Id:     00002F0E4C00
Node Id:       auntbea
Class:         S
Type:         TEMP
Resource Name:  proc19

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0002
DEALLOCATION ABORTED CAUSE
0000 0000 0000 4F4A
```

Dans cet exemple, la désallocation de **proc19** a échoué. Le code d'erreur **2** indique que des threads sont liées au dernier processeur logique et qu'elles n'ont pas été déliées après avoir reçu le signal SIGCPUFAIL. DEALLOCATION ABORTED DATA indique ces threads appartenaient au processus **0x4F4A**.

Les options de la commande **ps** (`-o THREAD`, `-o BND`) permettent de lister tous les threads et processus avec le numéro de l'UC à laquelle ils sont liés, le cas échéant.

Exemple 3 :

```

LABEL:                CPU_DEALLOC_ABORTED
IDENTIFIER:           8470267F

Date/Time:            Thu Sep 30 14:37:34
Sequence Number:      106
Machine Id:           00002F0E4C00
Node Id:              auntbea
Class:                S
Type:                 TEMP
Resource Name:        proc2

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0004
DEALLOCATION ABORTED CAUSE
0000 0000 0000 0000
```

Dans cet exemple, la désallocation de **proc2** a échoué parce qu'il existait au moins deux processeurs actifs au moment de l'incident (code d'erreur **4**).

Chapitre 8. Gestion des processus

Le processus est l'entité que le système d'exploitation utilise pour contrôler l'utilisation des ressources système. Les *threads* peuvent contrôler le temps processeur, mais la plupart des outils de gestion de système impose toujours de faire référence au processus dans lequel la thread est exécutée et non à la thread elle-même.

Reportez-vous au document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* pour obtenir des informations générales sur la gestion de vos propres processus, tels que redémarrer ou arrêter un processus que vous avez démarré, ou planifier un processus pour plus tard.

Pour les procédures des tâches, reportez-vous à la section Process Management du document *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*

Des outils sont disponibles, qui permettent de :

- Surveiller la création, l'annulation, l'identité des processus, et leur consommation de ressources par le biais de :
 - **ps** qui rend compte des ID processus, des utilisateurs, de la consommation du temps CPU et d'autres attributs.
 - **who -u** qui rend compte de l'ID processus shell des utilisateurs connectés.
 - **svmon** qui rend compte de la consommation par le processus de la mémoire réelle. Pour en savoir plus sur cette commande, reportez-vous à *Performance Tools 2 and 3 for AIX: User's Guide*.
 - **acct** (mécanisme) qui écrit des articles en fin de processus résumant l'utilisation des ressources par le processus. Pour la mise en oeuvre d'un système de comptabilité, reportez-vous à "Comptabilité - généralités".
- Contrôler le niveau de priorité auquel prétend le processus pour le CPU par le biais de :
 - **nice** qui lance une commande spécifiant la priorité d'un processus. Reportez-vous à *AIX 5L Version 5.2 - Guide de l'utilisateur : système d'exploitation et unités*.
 - **renice** qui change la priorité d'un processus donné.
- Mettre fin aux processus incontrôlables par le biais de :
 - **kill** qui adresse un signal au(x) processus concerné(s).
- Régler les mécanismes de gestion de processus du système d'exploitation par le biais de :
 - **schedtune** qui permet de modifier les paramètres du programmeur de processus. Pour en savoir plus sur cette commande, reportez-vous à *AIX 5L Version 5.2 - Guide d'optimisation*.

Chapitre 9. Gestion de la charge de travail

WLM (Workload Manager) permet à l'administrateur système d'exercer un plus grand contrôle sur l'allocation des ressources aux processus par VMM (Virtual Memory Manager) et le sous-système E/S de disque. Vous pouvez utiliser WLM pour empêcher les classes de travail d'entrer mutuellement en conflit et allouer des ressources en fonction des besoins des groupes d'utilisateurs. **Attention** : Pour utiliser efficacement WLM, vous devez bien connaître les processus système et les performances. Si l'administrateur système configure WLM avec des valeurs extrêmes ou inexactes, les performances en seront affectées.

WLM est essentiellement destiné aux gros systèmes. Ces systèmes sont souvent utilisés pour la consolidation des serveurs regroupant les charges de travail de plusieurs systèmes serveurs différents (tels que des systèmes d'imprimantes, de bases de données, d'utilisateurs généraux et de traitement des transactions) afin de réduire le coût de maintenance du système. Ces charges de travail interfèrent souvent entre elles. Elles ont des objectifs et des protocoles d'accord différents.

WLM permet également de compartimenter les communautés d'utilisateurs présentant des comportements très différents en termes de système. Cette fonction permet d'éviter tout déséquilibre entre des tâches interactives ou à faible sollicitation du CPU et des tâches de traitement par lots ou à forte sollicitation du CPU.

WLM s'intègre également au sous-système de comptabilité (reportez-vous à la section Comptabilité – généralités) qui permet aux utilisateurs d'effectuer une comptabilité d'usage de ressource par classe WLM en plus de la comptabilité standard par utilisateur ou par groupe.

Concepts et généralités sur WLM

WLM offre la possibilité de créer différentes classes de services pour les travaux, et de définir des attributs pour ces classes. Lesdits attributs fixent les quantités minimale et maximale de CPU, de mémoire physique et de débit d'E/S disque affectés à ces classes. WLM ordonne ensuite automatiquement les travaux dans des classes, en fonction de règles d'affectation fournies par l'administrateur système. Ces règles d'affectation reposent sur les valeurs d'un ensemble d'attributs d'un process. L'administrateur système ou un utilisateur privilégié peut également affecter des travaux à des classes, choix qui prévaut sur l'affectation automatique.

Définitions

classe	Ensemble de process et des routines associées, possédant un ensemble unique de valeurs de limitation de ressources et auxquels sont appliqués des partages cibles. Le terme classe désigne à la fois les sous-classes et les superclasses.
superclasse	Une superclasse est une classe à laquelle sont associées des sous-classes. Aucun process ne peut appartenir à une superclasse sans également appartenir à une sous-classe. Une superclasse possède un jeu de règles d'affectation de classe qui déterminent quels process lui seront affectés. Elle possède également un jeu de valeurs de limitation de ressources et de partages cibles de ressources qui déterminent la quantité de ressources susceptibles d'être utilisées par les process lui appartenant. Ces ressources sont réparties entre les sous-classes en fonction de leurs valeurs de limitation de ressource et de leurs partages cibles de ressources.
sous-classe	Une sous-classe est une classe associée à une seule et unique superclasse. Chaque process d'une sous-classe est également membre d'une superclasse. Les sous-classes n'ont accès qu'aux ressources mises à la disposition de la superclasse. Une sous-classe possède un ensemble de règles d'affectation de classe qui déterminent quels process affectés à la superclasse lui appartiennent. Elle possède également un jeu de valeurs de limitation de ressources et de partages cibles de ressources qui déterminent les ressources susceptibles d'être utilisées par les process lui appartenant. Ces valeurs de limitation de ressources et partages cibles de ressources indiquent quelle proportion des ressources dont dispose la superclasse (cible de la superclasse) peuvent être exploitées par les process de la sous-classe. L'administration de WLM peut être assurée à l'aide de l'outil de type Web Web-based System Manager, de SMIT ou de l'interface à ligne de commande de WLM.
mécanisme de classification	Un mécanisme de classification est un ensemble de règles d'affectation de classe qui détermine les process qui sont affectés aux différentes classes (superclasses ou sous-classes au sein de superclasses).

règle d'affectation de classe	Une règle d'affectation de classe indique quelles valeurs d'un ensemble d'attributs d'un process entraîneront son affectation à une classe particulière (superclasse ou sous-classe d'une superclasse).
valeur d'attribut de process	La valeur d'attribut d'un process est la valeur que prend un de ses attributs. Ces attributs sont par exemple : ID utilisateur, ID de groupe et chemin d'accès de l'application.
valeurs de limitation de ressources	Les valeurs de limitation de ressources sont les séries de valeurs que Workload Manager doit tenter de préserver pour un ensemble de valeurs d'utilisation de ressources. Ces limites sont totalement indépendantes des limites de ressources spécifiées avec setrlimit() .
partage cible de ressource	Les partages cibles de ressources sont les partages d'une ressource qui doivent être à disposition d'une classe (sous-classe ou superclasse). Ces partages sont utilisés avec d'autres partages de classe (sous-classe ou superclasse) au même niveau pour déterminer la répartition des ressources souhaitée entre les classes de ce niveau.
valeur d'utilisation de ressource	Une valeur d'utilisation de ressource est la quantité d'une ressource qu'un process ou ensemble de process utilise actuellement dans un système. Le fait qu'il s'agisse d'un process ou d'un ensemble de process est déterminé par la portée de la collecte de ressource de process.
portée de la collecte de ressource	La portée de la collecte de ressource est le niveau auquel l'utilisation des ressources est collectée et le niveau auquel les valeurs de limitation de ressource sont appliquées. Cela peut intervenir au niveau de chaque process d'une classe, au niveau de la somme des process d'une classe détenus par chaque utilisateur, ou au niveau de la somme des process d'une classe. La seule portée actuellement prise en charge est la dernière.
propriétés des classes de process	Ce sont les ensembles de propriétés attribuées à un process en fonction des classes (sous-classes ou superclasses) auxquelles il est affecté.

autorisations de classe	Les autorisations de classe sont les ensembles de règles qui indiquent quels utilisateurs et groupes sont autorisés à effectuer des opérations sur une classe ou sur les process et les routines d'une classe. Cela comprend l'autorisation d'affecter manuellement les process à une classe ou de créer des sous-classes d'une superclasse.
rang de classe	La valeur de rang d'une classe est sa position dans la hiérarchie de limitation de ressources souhaitée pour toutes les classes. Les limites de ressources (y compris les cibles de ressources) pour toutes les classes d'un rang doivent être satisfaites avant qu'une ressource quelconque soit fournie aux classes de rang inférieur. Les rangs sont définis au niveau des superclasses et des sous-classes. Les ressources sont fournies aux superclasses en fonction de leur rang. Au sein d'une superclasse, les ressources sont attribuées aux sous-classes en fonction de leurs valeurs de rang dans la superclasse. De ce fait, le rang de superclasse est l'élément de différenciation principal dans la répartition des ressources, le rang de sous-classe offrant un autre élément de différenciation inférieur au sein d'une superclasse.

Classes

WLM permet aux administrateurs système de définir des classes et, pour chaque classe, un ensemble d'attributs et de limites de ressources. Les process sont affectés aux classes en fonction de critères fournis par l'administrateur système. Les droits à ressources et les limites sont appliqués au niveau de la classe. Cette solution permet de définir des classes de service et de réguler l'utilisation des ressources de chaque classe d'application, afin d'éviter que des applications ayant des modèles d'utilisation des ressources très différents n'interfèrent les unes avec les autres lorsqu'elles partagent un seul serveur.

WLM prend en charge une hiérarchie de classes à deux niveaux :

- Les ressources du système sont réparties entre les superclasses en fonction des droits à ressource de chacune d'elles. Ces droits à ressources sont définis par l'administrateur système.
- Chaque superclasse peut à son tour avoir des sous-classes. Les ressources affectées aux superclasses sont réparties entre les sous-classes en fonction des droits à ressources attribués à chacune d'elles.
- L'administrateur système peut déléguer la gestion des sous-classes de chaque superclasse à un *administrateur de superclasses* ou à un groupe d'administrateurs.
- Dans AIX 5.2 et les versions supérieures, WLM prend en charge jusqu'à 69 superclasses (64 définies par l'utilisateur) et 64 sous-classes par superclasse (61 définies par l'utilisateur).
- Selon les besoins de l'entreprise, l'administrateur système peut décider d'utiliser uniquement les superclasses ou les superclasses et les sous-classes. Il peut également n'utiliser les sous-classes que pour certaines superclasses.

Remarque : Tout au long de cette discussion sur WLM, le terme *classe* s'applique aux superclasses et aux sous-classes. Si la discussion ne s'applique qu'à un type de classe spécifique, ce type est explicitement mentionné.

Affectation de process aux classes

Les process sont affectés à une classe selon les règles d'affectation établies par l'administrateur système. Les critères de classification reposent sur la valeur d'un ensemble d'attributs du process, tels que l'id utilisateur, l'id de groupe, le nom du fichier d'application, le type de process et l'étiquette d'application.

La détermination de la superclasse à laquelle un process doit être affecté s'effectue à partir d'un ensemble de règles ; de même, si des sous-classes sont définies dans cette superclasse, un autre ensemble de règles servira à déterminer la sous-classe à laquelle il doit être affecté. Ce processus d'affectation automatique tient également compte des attributs d'**héritage** de la superclasse et de la sous-classe. Pour plus d'informations sur les attributs de classe, reportez-vous à la section Attributs de classe WLM.

L'affectation automatique de classe intervient lorsqu'un process exécute l'appel système **exec**. L'affectation de classe est réévaluée lorsqu'un process emploie un appel système susceptible de modifier un attribut de process servant à la classification. C'est le cas par exemple de **setuid**, **setgid**, **setpri** et **plock**.

Outre cette affectation automatique de classe, un utilisateur disposant des droits appropriés peut affecter manuellement des process ou groupes de process à certaines superclasses ou sous-classes.

Contrôle des ressources

WLM permet de gérer les ressources de deux manières : sous la forme d'un pourcentage des ressources disponibles ou en fonction de l'usage total des ressources. Les ressources qui peuvent être contrôlées sous la forme d'un pourcentage incluent :

- Utilisation du CPU par les routines d'une classe. Il s'agit de la somme de tous les cycles CPU consommés par chaque routine de la classe.
- Utilisation de la mémoire physique par les process d'une classe. Il s'agit de la somme des pages mémoires appartenant aux process de la classe.
- Bande passante d'E/S disque de la classe. Il s'agit de la bande passante (en blocs de 512 octets par seconde) de toutes les E/S lancées par des routines de la classe sur chaque unité de disque à laquelle la classe accède.

Les ressources qui peuvent être contrôlées en fonction de l'utilisation total sont regroupées dans deux catégories : totaux de classe ou totaux de processus Les totaux de classe incluent :

le nombre de processus dans une classe

Il s'agit du nombre de processus actifs dans une classe à un moment donné.

le nombre de threads dans une

Il s'agit du nombre de threads actives dans une classe à un moment donné.

le nombre de connexions dans une

Il s'agit du nombre de sessions de connexions dans une classe à un moment donné.

Les totaux de processus incluent :

le temps UC total

Il s'agit du temps total UC cumulé associé à un processus.

E/S disque totales

Il s'agit du nombre total de blocs cumulés d'E/S de disque associé à un processus.

Durée de connexion totale

Il s'agit du délai total d'activité d'une session de connexion.

Droits à ressources

WLM permet aux administrateurs système de spécifier les droits à ressource par classe indépendamment pour chaque type de ressource. Ces droits peuvent être spécifiés en indiquant ce qui suit :

- La cible d'utilisation des différents types de ressources. Cette cible est indiquée à l'aide de partages, spécifiés sous forme de quantités relatives d'utilisation des différentes classes. Ainsi, si deux classes ont respectivement 1 et 3 partages de CPU et si ce sont les seules classes actives au moment concerné, WLM adoptera un objectif de régulation du CPU de 25 % et 75 %, respectivement. Les pourcentages cibles sont calculés pour les classes dans chaque niveau en fonction du nombre de partages actifs dans un niveau et de la quantité de ressource disponible pour le niveau.
- Limites minimale et maximale. Elles sont indiquées sous forme de pourcentage du total de ressources disponibles. WLM accepte deux types de limites maximales :
 - Une limite maximale souple (soft), qui indique la quantité maximale de la ressource qui peut être rendue disponible en cas de conflit d'accès la concernant. Cette limite peut être dépassée en l'absence de conflit, c'est-à-dire si la ressource ne fait pas l'objet d'autres demandes.
 - Une limite maximale ferme, qui indique la quantité maximale de la ressource qui peut être rendue disponible, même en l'absence de conflit d'accès la concernant.
- Limites totales Les limites totales sont appliquées de manière stricte. Si un processus dépasse l'une de ses limites de consommation totale, il est arrêté. Si une classe atteint l'une de ses limites totales, toute opération qui génère une autre instance de la ressource échoue.

Dans la plupart des cas, des limites maximales logicielles sont suffisantes pour garantir l'affectation des ressources. En utilisant des limites maximales matérielles, des ressources système peuvent ne pas être utilisées du fait qu'elles sont appliquées strictement, même s'il n'existe pas de conflit pour la ressource. Vous devez utiliser les limites maximales matérielles avec précaution du fait qu'elles peuvent affecter considérablement les performances du système ou les application si les limites sont trop basses. Les limites totales doivent être également utilisées avec précaution du fait qu'elles peuvent provoquer l'arrêt des processus ou ne pas fonctionner comme prévu.

En mode actif, WLM tente de maintenir les classes actives proches de leurs cibles. Du fait qu'il existe quelques contraintes sur les valeurs des diverses limites, la somme des limites dans toutes les classes peut dépasser largement 100%. Dans ce cas, si toutes les classes sont actives, la limite ne peut pas être atteinte par toutes les classes. WLM régule la consommation UC en ajustant les priorités de planification des threads dans le système en fonction des performances de la classe à laquelle elles appartiennent, par rapport à ses limites et sa cible. Cette approche garantit une consommation UC moyenne calculée sur une période donnée et non une consommation UC sur une courte période (tics de 10 ms, par exemple).

Si la classe A, par exemple, est la seule classe active, avec une consommation UC minimum de 0 % et une cible UC de 60 partages, elle dispose de 100 % de l'UC. Si la classe B, avec une limite minimale UC de 0 % et une cible UC de 40 partages, devient active, l'utilisation UC de la classe A diminue progressivement pour atteindre 60 % et l'utilisation UC de la classe B passe de 0 à 40 %. Le système se stabilise à une utilisation UC de 60 % et 40 % respectivement en quelques secondes.

Cet exemple suppose qu'il n'existe pas de conflit de mémoire entre les classes. Dans des conditions de travail normales, les limites que vous définissez pour l'UC et la mémoire sont interdépendantes. Par exemple, une classe peut ne pas pouvoir atteindre sa cible ou même son allocation UC minimale si la limite maximale d'utilisation de la mémoire est trop basse comparée à son ensemble d'exploitation.

Pour redéfinir la classe et les limites de classe pour un groupe d'applications, WLM fournit l'outil de génération de rapports **wlmstat** qui indique la quantité de ressource utilisée par chaque classe. L'outil graphique **wlmon** est également fourni pour contrôler le système.

Modes de fonctionnement WLM

WLM peut être utilisé pour réguler la consommation des ressources sous forme de pourcentages par classe, de totaux par classe ou de totaux par processus. La régulation de tous les types de ressources peut être activée en exécutant WLM en mode actif.

Vous pouvez également démarrer un mode WLM qui classe les processus nouveaux et existants et contrôle l'utilisation des ressources des diverses classes sans tenter de réguler cette utilisation. Ce mode s'appelle le *mode passif*.

Le mode passif peut être utilisé lors de la configuration de WLM sur un nouveau système pour vérifier les règles de classification et d'affectation et pour établir une base d'utilisation des ressources pour les diverses classes lorsque WLM *ne* régule pas l'allocation UC et de mémoire. Cela permet aux administrateurs système de définir la manière d'appliquer les partages de ressources et les limites de ressources (si nécessaire) pour favoriser les applications importantes et limiter les applications moins importantes pour leur permettre d'atteindre leurs objectifs professionnels.

Si le temps UC est la seule ressource à réguler, vous pouvez décider d'exécuter WLM en mode actif pour l'UC et en mode passif pour toutes les autres ressources. Ce mode s'appelle le mode *uc uniquement*. Si vous voulez réguler les pourcentages par classe, mais pas les types de ressources totales, la comptabilité et la régulation des ressources totales peut être désactivée pour les totaux par classe, les totaux par processus ou les deux. Dans tous les modes, vous pouvez désactiver la liaison des groupes de ressources.

Contrôle dynamique

Lorsque WLM est actif, les paramètres de la configuration en cours peuvent être modifiés à tout moment ; c'est notamment le cas des attributs d'une classe, de ses partages et limites de ressources, de ses règles d'affectation et de l'ajout ou de la suppression de classes existantes. A cet effet, diverses solutions sont disponibles :

- Modification des fichiers de propriétés pour la configuration actuellement active (répertoire pointé par le lien symbolique **/etc/wlm/current**) et rafraîchissement de WLM avec **wlmcntrl** pour permettre la prise en compte des nouveaux paramètres.
- Création d'une autre configuration avec un jeu de paramètres différent et mise à jour de WLM afin qu'il charge ces paramètres et en fasse la configuration active.
- Modification de certains des paramètres de la configuration active à l'aide de l'interface à ligne de commande de WLM (**mkclass**, **chclass**, et **rmclass**).
- Modification de certains paramètres de la configuration active en cours dans une application à l'aide des API WLM.

Le passage automatique à une nouvelle configuration à des moments donnés de la journée peut être réalisé à l'aide de *jeux de configuration*. Les jeux de configuration permettent à l'administrateur de définir un groupe de configurations à utiliser ainsi que la période pendant laquelle chaque configuration sera active.

Outils de contrôle

WLM dispose de trois commandes qui affichent les statistiques de WLM et permettent à l'administrateur système de surveiller le fonctionnement de ce logiciel :

- **wlmstat** est une commande orientée texte qui affiche des statistiques sous forme de texte (pourcentage d'utilisation des ressources par classe pour tous les types de ressources gérés par WLM).
- **wlmon** offre une vue graphique de l'utilisation des ressources par classe et de la régulation de WLM.

- **wlmp perf** est un outil en option disponible avec la Performance Toolbox ; il offre plus de possibilités, par exemple l'enregistrement et la réexécution à long terme, ainsi que d'autres fonctions.

API WLM

Une API (interface de programmation d'applications) permet aux applications d'effectuer toute opération susceptible d'être exécutée à l'aide de l'interface à ligne de commande de WLM, comme la création, la modification ou la suppression de classes, le changement d'attributs des classes ou des partages et limites de ressources, et l'affectation manuelle de process aux classes.

De plus, l'API permet aux applications de définir un attribut de classification qui leur est propre, appelé *étiquette (tag)*. La définition de cette étiquette à l'aide d'un jeu de valeurs connu de l'administrateur système (via la documentation utilisateur de l'application) permet de faire la distinction entre plusieurs instances de la même application. Sa classification dans différentes classes, avec différents droits à ressources, est ainsi possible.

Comptabilité par classe

Avec l'utilitaire de système de comptabilité d'AIX, vous pouvez rassembler et rendre compte de l'utilisation de plusieurs ressources de système par utilisateur, groupe ou classe WLM. Lorsque le relevé de processus est activé, le système d'exploitation enregistre des statistiques sur l'utilisation de la ressource du processus dans un fichier de comptabilité lorsque le processus s'arrête. À partir de AIX 5.1, cet enregistrement comptable comprend une touche numérique de 64 octets représentant le nom de la classe WLM à laquelle le processus appartenait. (Pour en savoir plus sur l'utilitaire de système de comptabilité, reportez-vous à la section Comptabilité – généralités, page 11-2.)

Le sous-système de comptabilité utilise une touche de 64 octets à la place du nom de classe complet de 34 caractères pour économiser de l'espace (sinon la modification doublerait pratiquement la taille de l'enregistrement comptable). Lorsque la commande de comptabilité est exécutée pour extraire les données au niveau du processus, la touche est traduite en nom de classe à l'aide de la routine mentionnée ci-dessus. Cette traduction utilise les noms de classe actuellement dans les fichiers de configuration WLM. Donc si une classe a été supprimée entre le moment où l'enregistrement comptable a été écrit, lorsque le processus s'est terminé, et le moment où le compte-rendu de comptabilité est exécuté, le nom de classe correspondant à la touche est introuvable, et la classe s'affiche comme `Unknown` (inconnue).

Pour garder des comptes-rendus appropriés de l'usage de ressource de classes supprimées pendant un exercice financier, utilisez l'une des procédures suivantes :

- Au lieu de supprimer la classe, gardez le nom de la classe dans le fichier des classes et supprimez la classe du fichier de règles afin qu'aucun processus ne puisse lui être attribué. Vous pouvez ensuite supprimer la classe après avoir généré le compte-rendu de comptabilité à la fin de l'exercice financier.
- Vous pouvez également supprimer la classe de la configuration à laquelle elle appartient et garder le nom de la classe dans le fichier des classes dans une configuration « factice » (jamais activée) jusqu'à la génération des comptes-rendus de comptabilité pour l'exercice.

Généralités sur les classes WLM

Workload Manager vous aide à contrôler l'allocation des ressources système en définissant des classes de service et en allouant des ressources à chacune de ces classes. Chaque classe dispose d'un groupe d'attributs qui déterminent les allocations de ressources et d'autres comportements. Chaque processus du système appartient à une classe de service et l'allocation de ressources et les comportements associés à la classe lui sont appliqués. Les processus sont affectés à une classe manuellement à l'aide de l'affectation manuelle ou automatiquement en fonction des règles de classification définies par l'utilisateur.

WLM prend en charge deux niveaux de classes : les *superclasses* et les *sous-classes*. Les superclasses sont affectées de ressources en fonction des ressources système disponibles et les sous-classes sont affectées de ressources en fonction des affectations de la superclasse qui leur est associée. Vous pouvez également définir des sous-classes pour mieux contrôler les processus d'une superclasse. Vous pouvez également déléguer la définition des superclasses en définissant un utilisateur administrateur ou un groupe d'administrateurs pour une superclasse.

Pour les superclasses et les sous-classes, vous pouvez définir des classes, des partages de ressources, des limites et des règles en utilisant SMIT, Web-based System Manager ou l'interface de ligne de commande. Les applications peuvent utiliser les API WLM. Les définitions de configurations sont stockées dans des fichiers texte appelés *fichiers de propriétés WLM*.

Un nom de classe contient jusqu'à 16 caractères qui peuvent correspondre à des majuscules et des minuscules, des chiffres et le caractère souligné (_). Pour chaque configuration WLM, chaque nom de superclasse doit être unique. Chaque nom de sous-classe doit être unique dans les superclasses, mais une sous-classe peut porter le nom d'une sous-classe d'une autre superclasse. Pour identifier chaque sous-classe de manière unique, le nom complet d'une sous-classe est constitué du nom de la superclasse et du nom de la sous-classe séparés par un point. Par exemple : *Super.Sub*.

Superclasses

L'administrateur système peut définir jusqu'à 64 superclasses. De plus, cinq superclasses sont créées automatiquement, comme suit :

Superclasse *Default* (par défaut)

Il s'agit de la superclasse par défaut, qui est toujours définie. Tous les process non-racine qui ne sont pas affectés automatiquement à une superclasse spécifique sont affectés à la superclasse par défaut. Les autres process peuvent également être affectés à la superclasse *Default* à l'aide de règles d'affectation spécifiques.

Superclasse *System* (système)

Cette superclasse se voit affecter tous les process privilégiés (root) si ceux-ci ne sont pas affectés à une classe spécifique par des règles. Cette superclasse collecte également les pages mémoire appartenant aux segments mémoire, aux process et aux routines du noyau. D'autres process peuvent également être affectés à la superclasse *System* à l'aide de règles d'affectation spécifiques. Par défaut, la limite minimale de mémoire de cette superclasse est de 1 %.

Superclasse *Shared* (partagée)

Cette superclasse reçoit toutes les pages mémoire partagées par des process appartenant à plusieurs superclasses. Cela comprend les pages dans des régions de mémoire partagée et celles des fichiers utilisés par des process appartenant à plusieurs superclasses (ou à des sous-classes de superclasses distinctes). La mémoire partagée et les fichiers utilisés par plusieurs process appartenant tous à une seule superclasse (ou à des sous-classes de la même superclasse) sont associés à cette superclasse. Ce n'est que lorsqu'un process d'une superclasse différente accède à la région de mémoire partagée ou au fichier que les pages sont placées dans la superclasse *Shared*. Seuls des partages et limites de mémoire physique peuvent être appliqués à cette superclasse. Les partages et limites des autres types de ressources, sous-classes ou règles d'affectation spécifiées ne peuvent pas lui être appliqués. Qu'un segment de mémoire partagé par des processus dans différentes sous-classes de la même superclasse soit classé dans la sous-classe *Shared* (partagé) ou reste dans sa sous-classe d'origine dépend de la valeur de l'attribut **localshm** de la sous-classe d'origine.

Superclasse *Unclassified* (non classée)

Les process existants au moment du lancement de WLM sont classés en fonction des règles d'affectation de la configuration de WLM en cours de chargement. Pendant cette classification initiale, toutes les pages mémoire sont affectées soit à la superclasse à laquelle le process appartient (si elles ne sont pas partagées, ou si elles sont partagées par les process de la même superclasse), soit à la superclasse *Shared* si elles sont partagées par des process de différentes superclasses.

Toutefois, quelques pages ne peuvent pas être directement liées à un process quelconque (et donc à une classe quelconque) au moment de la classification ; la mémoire correspondante est affectée à la superclasse *Unclassified*. La majeure partie de cette mémoire finit par être correctement reclassée au cours des temps, lorsqu'un process y accède, ou lorsqu'elle est libérée et réaffectée à un process après le démarrage de WLM. La superclasse *Unclassified* ne contient aucun process. Des partages et des limites de mémoire physique peuvent lui être appliqués. Elle ne peut pas avoir de partages ou de limites pour les autres types de ressources, sous-classes ou règles d'affectation spécifiées.

Superclasse *Unmanaged* (non gérée)

Cette superclasse est toujours définie. Aucun process ne lui est affecté. Elle sert à accumuler l'utilisation du CPU pour tous les process à priorité fixe et l'utilisation de mémoire pour toutes les pages réservées du système. L'utilisation du CPU pour le waitprocs n'est accumulée dans aucune classe. Ce fait est volontaire, car sinon le système semblerait toujours être à 100 % d'utilisation CPU, ce qui pourrait être trompeur pour les utilisateurs examinant les statistiques WLM.

Sous-classes

Jusqu'à 61 sous-classes peuvent être définies par l'administrateur système ou un administrateur de superclasse. De plus, deux sous-classes spéciales, *Default* et *Shared* sont toujours définies.

Sous-classe *Default*

Il s'agit de la sous-classe par défaut, qui est toujours définie. Tous les process qui ne sont pas automatiquement affectés à une sous-classe particulière de la superclasse sont affectés à la sous-classe *Default*. Vous pouvez également affecter d'autres process à la sous-classe *Default*, à l'aide de règles d'affectation spécifiques.

Sous-classe *Shared*

Cette classe reçoit toutes les pages mémoire utilisées par des process appartenant à plusieurs sous-classes de la superclasse. Il s'agit notamment des pages figurant dans des zones de mémoire partagée et des pages se trouvant dans des fichiers utilisés par des process appartenant à plusieurs sous-classes de la même superclasse. La mémoire partagée et les fichiers utilisés par plusieurs process appartenant tous à une seule sous-classe sont associés à cette sous-classe. Ce n'est que lorsqu'un process d'une sous-classe différente, appartenant à la même superclasse, accède à la zone de mémoire partagée ou au fichier que les pages sont placées dans la sous-classe *Shared* de la superclasse. La sous-classe *Shared* ne contient pas de process. Seuls des partages et des limites de mémoire physique peuvent lui être appliqués. Elle ne peut pas avoir de partages ou de limites pour les autres types de ressources ou les règles d'affectation spécifiées. Qu'un segment de mémoire partagé par des processus dans différentes sous-classes de la même superclasse soit classé dans la sous-classe *Shared* (partagé) ou reste dans sa sous-classe d'origine dépend de la valeur de l'attribut **localshm** de la sous-classe d'origine.

Attributs de classe

Les attributs d'une classe sont :

Class name (nom de la classe) :

peut comporter jusqu'à 16 caractères, ne contenant que des lettres majuscules et minuscules, des chiffres et des caractères de soulignement (_).

Tier (rang) : chiffre entre 0 et 9 servant à définir la priorité d'affectation de ressources entre les classes.

Inheritance (héritage) :

spécifie si un process enfant hérite l'affectation de classe de son process parent.

localshm Empêche les segments de mémoire d'une classe de migrer vers la classe *Shared*.

Administrator (adminuser, admingroup, authgroup) (superclasse uniquement)
Délègue l'administration d'une superclasse.

Authorization (authuser, authgroup)

Délègue le droit d'affecter manuellement un processus à une classe.

Resource Set (rset)

Limite le jeu de ressources auquel une classe peut accéder en terme d'UC (groupe de processeurs).

Attribut de niveau

Les niveaux correspondent à l'ordre d'allocation des ressources système dans les classes WLM. L'administrateur peut définir des classes sur 10 niveaux maximum, numérotés de 0 à 9, 0 correspondant au niveau le plus haut ou le plus important. La quantité de ressources disponible pour le niveau 0 correspond à toutes les ressources système disponibles. La quantité de ressources disponible pour les niveaux inférieurs (nombre supérieur) correspond au nombre de ressources non utilisées par tous les niveaux supérieurs. Les pourcentages de consommation cibles des classes reposent sur le nombre de partages actifs dans ses niveaux et la quantité de ressources disponible pour le niveau. Du fait que le niveau 0 est le seul niveau qui ait toujours des ressources disponibles, il est recommandé de classer dans ce niveau les processus indispensables au fonctionnement du système. Si aucun niveau n'est défini pour une classe, le niveau 0 est utilisé.

Un niveau peut être défini dans la superclasse et dans la sous-classe. Les niveaux de superclasse permettent de vérifier la priorité d'allocation de ressources entre les superclasses. Les niveaux de sous-classe permettent de vérifier la priorité de l'allocation de ressources entre les sous-classes d'une superclasse. Il n'existe pas de relation entre les sous-niveaux de différentes superclasses.

Attribut d'héritage

L'attribut d'héritage d'une classe indique si les processus de la classe doivent être reclassés automatiquement lorsque l'un des attributs de classification du processus change. Lors de la création d'un processus avec la sous-routine **fork**, le processus hérite automatiquement de la classe de son parent, que l'héritage soit activé ou non, sauf lorsque le processus parent a une option, que **inherit tag at fork** est désactivé et que l'héritage de classe de la classe parent est désactivé. Dans ce cas, le processus enfant est reclassé en fonction des règles de classification.

Lorsque l'héritage de classe n'est pas activé sur une classe, tout processus de la classe est classifié automatiquement en fonction des règles de classification après l'appel d'un service qui change un attribut de processus utilisé dans la règle. L'appel le plus courant de ces appels est la sous-routine **exec**, mais les autres sous-routines qui peuvent changer la classification incluent **setuid**, **setgid**, **plock**, **setpri** et **wlm_set_tag**. Lorsque l'héritage est actif, le processus n'est pas reclassé en fonction des règles de classification et il reste dans sa classe. L'affectation manuelle est prioritaire par rapport à l'héritage et peut être utilisée pour reclasser les processus d'une classe dont l'héritage est activé.

La valeur définie pour cet attribut peut être yes ou no. Si aucune valeur n'est définie, l'héritage n'est pas activé sur la classe.

Cet attribut peut être défini au niveau de la superclasse et de la sous-classe. Pour une sous-classe d'une superclasse :

- Si l'attribut d'héritage est affecté de la valeur yes au niveau de la superclasse et de la sous-classe, un enfant d'un processus de la sous-classe reste dans la même sous-classe.
- S'il est affecté de la valeur yes pour la superclasse et de la valeur no (non défini) pour la sous-classe, un enfant d'un processus dans la sous-classe reste dans la même superclasse et il sera classé dans l'une de ses sous-classes en fonction des règles d'affectation de la superclasse.
- S'il est affecté de la valeur no (non défini) pour la superclasse et de la valeur yes pour la sous-classe, un enfant de processus dans la sous-classe est soumis aux règles d'affectation automatique des superclasses.

- Si le process est classé par les règles de la même superclasse, il reste dans la sous-classe (il n'est pas soumis aux règles d'affectation des sous-classes).
- Si le process est classé dans une superclasse différente par les règles des superclasses, les règles d'affectation de sous-classe de la nouvelle superclasse lui sont appliquées pour déterminer à quelle sous-classe de la nouvelle superclasse le process est affecté.
- Si les attributs d'héritage de la superclasse et de la sous-classe ont la valeur non (no) ou ne sont pas spécifiés, lors de l'exec(), l'enfant d'un process d'une sous-classe est soumis aux règles d'affectation automatique standard.

Attribut localshm

L'attribut localshm peut être spécifié aux niveaux de la superclasse et de la sous-classe. Il est utilisé pour éviter que des segments de mémoire appartenant à une classe migrent vers la superclasse ou vers la sous-classe *Shared* (partagé) lorsque des processus y accèdent dans d'autres classes. Les valeurs possibles de l'attribut sont oui ou non. La valeur *oui* signifie que des segments de mémoire dans cette classe doivent y rester et ne pas migrer vers la classe *Shared* (partagé) appropriée. La valeur *non* est la valeur par défaut lorsque l'attribut n'est pas spécifié.

Les segments de mémoire sont classés dans les défaillances de page. Lors de la création d'un segment, ce dernier est signalé comme appartenant à la superclasse *Unclassified* (non classé). Sur la première défaillance de page du segment, il est classé dans la même classe que le processus défaillant. Si ensuite un processus appartenant à une classe différente de la page du segment est défaillant sur ce segment, WLM décide si le segment doit être reclassé dans la classe *Shared* (partagé) (superclasse ou sous-classe) appropriée. Si le processus défaillant et le segment appartiennent à des superclasses différentes, l'une des possibilités suivantes se produit :

- Si l'attribut localshm de la superclasse du segment est défini sur *oui*, le segment reste dans sa superclasse actuelle. Si l'attribut localshm de la sous-classe du segment est défini sur *oui*, le segment reste dans sa sous-classe actuelle. Si l'attribut localshm de la superclasse est défini sur *oui* mais que son attribut de sous-classe est défini sur *non*, il va dans la sous-classe *Shared* (partagé) de la superclasse actuelle.
- Si l'attribut localshm de la superclasse du segment est défini sur *non*, le segment va à la superclasse *Shared* (partagé). Il s'agit là de l'action par défaut.

Si le processus défaillant et le segment appartiennent à différentes sous-classes de la même superclasse, et que l'attribut localshm de la sous-classe du segment est défini sur *oui*, le segment reste dans la classe actuelle (superclasse et sous-classe). Sinon, le segment va dans la sous-classe *Shared* (partagé) de la superclasse.

Bien sûr, si le processus défaillant et le segment appartiennent à la même classe (même superclasse et même sous-classe), le segment n'est pas reclassé sans que les valeurs des attributs localshm soient prises en compte.

Attributs Administrator

Remarque : Ces attributs ne sont valables que pour les superclasses.

Les attributs suivants servent à déléguer l'administration d'une superclasse à un utilisateur ou un groupe d'utilisateurs :

- `adminuser` spécifie le nom de l'utilisateur (tel qu'il est mentionné dans `/etc/passwd`) autorisé à effectuer des tâches d'administration sur la superclasse.
- `admingroup` spécifie le nom d'un groupe d'utilisateurs (tel qu'il est mentionné dans `/etc/group`) autorisés à effectuer des tâches d'administration sur la superclasse.

Une seule valeur (utilisateur ou groupe) est autorisée pour chaque attribut. Il est possible de spécifier l'un ou l'autre attribut, les deux ou aucun d'eux. L'utilisateur ou groupe aura le droit de :

- créer et supprimer des sous-classes ;
- modifier les attributs et les partages et limites de ressources pour les sous-classes ;
- définir, supprimer ou modifier les règles d'affectation de sous-classes ;
- rafraîchir (actualiser) la configuration WLM active pour la superclasse.

Attributs Authorization

Ces attributs sont valables pour toutes les classes. Ils permettent d'indiquer l'utilisateur ou le groupe autorisé à affecter manuellement des process à une classe (superclasse ou sous-classe). Lors de l'affectation manuelle d'un process (ou d'un groupe de process) à une superclasse, les règles d'affectation de la superclasse servent à déterminer à quelle sous-classe de cette superclasse chaque process sera affecté.

Une seule valeur (utilisateur ou groupe) est autorisée pour chaque attribut. Il est possible de spécifier l'un ou l'autre attribut, les deux ou aucun d'eux.

Attribut Resource Set

L'attribut Resource set (appelé *rset*) peut être défini pour n'importe quelle classe. Sa valeur correspond à un nom de groupe de ressources défini par l'administrateur. L'attribut *rset* est un sous-ensemble de la ressource UC disponible dans le système (groupe de processeurs). La valeur par défaut est "system" qui donne accès aux ressources UC disponibles dans le système. La seule restriction réside dans le fait que si *rset* est défini pour une sous-classe, les UC du groupe d'UC doivent correspondre à un sous-ensemble des UC disponibles dans la superclasse. (Pour plus d'informations, reportez-vous à la description de la commande **mkrset** dans le document *AIX 5L Version 5.2 Commands Reference*.)

Remarque : Veillez à affecter les groupes de ressources à une classe n'appartenant pas au niveau 0, du fait que les niveaux inférieurs ont uniquement accès aux ressources non utilisées par les niveaux supérieurs. La restriction d'une classe ne correspondant pas à une classe de niveau 0 à un sous-ensemble d'UC du système peut provoquer un manque d'UC s'il n'existe pas de temps UC sur ces UC.

Affectation de process à des classes

Dans Workload Manager, les process sont classés de deux façons :

- Un processus est automatiquement affecté en utilisant les règles d'affectation lorsque les attributs de classification de processus changent. Lorsque WLM fonctionne en mode actif, cette affectation automatique est toujours en vigueur (elle peut être désactivée). Il s'agit de la procédure la plus courante de classification des processus.
- Un processus sélectionné ou un groupe de processus peuvent être affectés manuellement à une classe par un utilisateur avec la priorité appropriée sur les processus et la classe cible. L'affectation manuelle peut être effectuée en utilisant une commande WLM qui peut être appelée directement ou exécutée via SMIT ou Web-based System Manager ou bien par une application utilisant une fonction de l'interface API WLM. Cette affectation manuelle remplace l'affectation automatique et l'héritage.

Affectation automatique

L'affectation automatique de process à des classes s'effectue selon un ensemble de règles d'affectation de classe définies par l'administrateur WLM. Deux niveaux de règles d'affectation sont disponibles :

- Un premier ensemble de règles au niveau de la configuration de WLM, qui sert à déterminer à quelle superclasse un process particulier doit être affecté.
- Un deuxième ensemble, situé au niveau des superclasses pour lesquelles des sous-classes sont définies. Il détermine à quelle sous-classe le process doit être affecté.

Les règles d'affectation aux deux niveaux repose sur les valeurs d'un groupe d'attributs de processus. Ces attributs sont les suivants :

- L'ID utilisateur du process
- L'ID de groupe du process
- Le chemin d'accès de l'application (programme) exécutée
- Type of processus (32 bits ou 64 bits, par exemple)
- L'étiquette (tag) du process

L'étiquette (tag) est un attribut, défini sous forme de chaîne de caractères, qu'une application peut définir par programme, en faisant appel à l'API (interface de programmation d'application) de WLM.

La classification a lieu chaque fois qu'un attribut change en comparant la valeur des attributs des processus à des listes de valeurs possibles dans les règles d'affectation de classe (appelées **règles**). La comparaison détermine la règle qui correspond à la valeur en cours des attributs de processus.

Une règle d'affectation de classe est une chaîne de caractères qui contient les champs suivants séparés par au moins un espace :

Une affectation de classe est une chaîne de texte contenant différents champs séparés par un ou plusieurs espaces ou tabulations. Les champs sont les suivants :

Name (nom)	Ce champ doit contenir le nom d'une classe défini dans le fichier de classe correspondant au niveau du fichier rules (superclasse ou sous-classe). Les noms de classe ne peuvent contenir que des lettres majuscules ou minuscules, des chiffres et des caractères de soulignement. Ils peuvent comporter jusqu'à 16 caractères. Aucune règle d'affectation ne peut être spécifiée pour les classes définies par le système Unclassified, Unmanaged et Shared.
Reserved (réservé)	Ce champ est réservé à des extensions futures. Sa valeur doit être un tiret (-), et sa présence est obligatoire.
User (utilisateur)	Peut contenir un trait d'union (-) ou au moins un nom d'utilisateur (tel que défini dans le fichier /etc/passwd) La liste contient un ou plusieurs noms séparés par une virgule (,). Vous pouvez utiliser un point d'exclamation (!) avant le nom pour exclure un utilisateur de la classe. Des modèles peuvent être définis pour correspondre à un groupe de noms d'utilisateurs en utilisant la syntaxe de correspondance de modèles du shell Korn. S'il n'existe pas de noms d'utilisateur valides, la règle est ignorée.
Group (groupe)	Peut contenir un trait d'union (-) ou au moins un nom de groupe (tel que défini dans le fichier /etc/group) La liste contient un ou plusieurs noms séparés par une virgule (,). Vous pouvez utiliser un point d'exclamation (!) avant le nom pour exclure un groupe de la classe. Des modèles peuvent être définis pour correspondre à un groupe de noms d'utilisateurs en utilisant la syntaxe de correspondance de modèles du shell Korn. S'il n'existe pas de noms de groupes valides, la règle est ignorée.
Application	<p>Ce champ peut contenir un tiret (-) ou une liste de noms de chemin d'accès d'applications. Il contient le chemin d'accès des applications (programmes) exécutées par les process inclus dans la classe. Les noms d'application sont soit des chemins d'accès complets, soit des modèles du shell Korn correspondant à des chemins d'accès. La liste est composée d'un ou plusieurs chemins d'accès, séparés par une virgule. Pour exclure une application particulière, il suffit d'indiquer un point d'exclamation devant son nom.</p> <p>Au moins une application de la liste doit être détectée lors du chargement sinon la règle est ignorée. Les règles ignorées initialement pour cette raison peuvent devenir effectives plus tard si un système de fichiers est monté et qu'il contient une ou plusieurs applications de la liste.</p>

Type	<p>Ce champ peut contenir un tiret (–) ou une liste d’attributs du process. Les valeurs possibles de ces attributs sont les suivantes :</p> <ul style="list-style-type: none"> • 32bit: il s’agit d’un process 32 bits • 64bit: il s’agit d’un process 64 bits • plock: le process a exécuté l’appel système plock pour réserver de la mémoire • fixed: il s’agit d’un process à priorité fixe (SCHED_FIFO ou SCHED_RR). La valeur du champ type peut être une combinaison d’un ou plusieurs des attributs ci-dessus, séparés par le signe plus (+). 32bit et 64bit s’excluent mutuellement.
Tag (étiquette)	<p>Ce champ peut contenir un tiret (–) ou une liste d’étiquettes d’application. Une étiquette d’application est une chaîne d’au plus 30 caractères alphanumériques. La liste est composée d’une ou plusieurs étiquettes d’application séparées par des virgules.</p>

Les attributs User, Group, Application et Tag peuvent être un groupe de valeurs d’attributs.

Lors de la création d’un process (fourchette), celui-ci reste dans la même classe que son parent. La reclassification se produit lorsque le nouveau process modifie l’un de ses attributs servant à la classification : *exec*, *setuid* (et appels associés), *setgid* (et appels associés), *setpri* et *plock*, par exemple.

Afin de classer le process, WLM examine le fichier **rules** de niveau supérieur de la configuration active pour trouver à quelles superclasses le process doit appartenir. Pour chaque règle du fichier, WLM compare les valeurs en cours des attributs du process aux valeurs et listes de valeurs spécifiées dans la règle. Les règles sont vérifiées dans l’ordre dans lequel elles apparaissent dans le fichier. En cas de correspondance, le process est affecté à la superclasse nommée dans le premier champ de la règle. Ensuite, le fichier de règles de la superclasse est examiné de la même façon, afin de déterminer à quelles sous-classes le process doit être affecté.

Pour qu’un process corresponde à l’une des règles, chacun de ses attributs doit correspondre au champ adéquat de la règle. La liste suivante répertorie les critères servant à déterminer si la valeur d’un attribut correspond aux valeurs du champ du fichier **rules** :

- Si le champ du fichier de règles contient un tiret (–), toute valeur de l’attribut de process en rapport lui correspond.
- Pour tous les attributs à l’exception de *type*, si la valeur de l’attribut de process correspond à l’une des valeurs de la liste du fichier de règles qui n’est pas exclue (précédée d’un “!”), il y a concordance.
- Pour l’attribut *type*, si l’une des valeurs de la règle comprend deux valeurs ou plus séparées par un signe plus (+), il n’y aura concordance du process que si toutes ses caractéristiques correspondent à toutes les valeurs mentionnées ci-dessus.

Tant au niveau des superclasses que des sous-classes, WLM parcourt les règles dans l’ordre dans lequel elles apparaissent dans le fichier **rules**, et range les process dans les classes qui correspondent à la première règle pour laquelle il y a concordance. En d’autres termes, l’ordre dans lequel les règles figurent dans le fichier de règles est très important : il convient d’agir avec la plus grande précaution lors de la création ou de la modification du fichier de règles.

Affectation manuelle

Un process ou un groupe de process peut être affecté manuellement à une superclasse et/ou une sous-classe soit à l'aide des interfaces d'administration de WLM Web-based System Manager, SMIT ou de la commande **wlmassign**, soit via une application utilisant la fonction de l'API WLM **wlm_assign**.

Pour affecter manuellement des process à une classe ou annuler une affectation manuelle existante, un utilisateur doit disposer du niveau de droits adéquat. La section suivante, qui traite des particularités de l'affectation manuelle, adopte ce principe comme acquis. Il sera abordé en détail ultérieurement dans la section sur la sécurité.

L'affectation manuelle peut être réalisée ou annulée séparément au niveau de la superclasse, de la sous-classe, ou aux deux niveaux. Elle est spécifiée par des paramètres de l'interface de programmation et un ensemble d'options de l'interface à ligne de commande utilisée par les outils d'administration de WLM. Ainsi, un process peut être affecté à une superclasse seulement, une sous-classe seulement ou une superclasse et une sous-classe de cette dernière. Dans ce dernier cas, la double affectation peut être effectuée simultanément (via une seule commande ou un seul appel à l'API) ou à différents moments, par des utilisateurs distincts.

Ce système est très souple, mais peut prêter à confusion. Voici deux exemples de cas possibles.

Exemple 1 : Première affectation de process

L'administrateur système affecte manuellement *Process1* de la *superclasseA* à la *superclasseB* (affectation n'intervenant qu'au niveau des superclasses). WLM fera appel aux règles d'affectation automatique des sous-classes de la *superclasseB* pour déterminer à quelle sous-classe le process sera finalement affecté. *Process1* sera affecté à *superclasseB.sous-classeA* et sera désigné comme ayant une affectation de type "superclasse uniquement".

Un utilisateur muni des droits appropriés affecte le process *Process2* appartenant actuellement à la classe *superclasseA.sous-classeA* à une nouvelle sous-classe de la même superclasse, *superclasseA.sous-classeB*. *Process2* est affecté à sa nouvelle sous-classe et désigné comme ayant une affectation de type "sous-classe uniquement".

L'administrateur WLM des sous-classes de *superclasseB* peut décider de réaffecter manuellement le process *Process1* à une autre sous-classe de *superclasseB*, *sous-classeC* par exemple. *Process1* sera reclassé dans *superclasseB.sous-classeC* et sera désigné comme ayant une affectation de niveau superclasse et sous-classe.

Exemple 2 : Réaffectation et annulation de process

La réaffectation et l'annulation d'une affectation manuelle au niveau d'une sous-classe est simple et n'a d'implications que sur l'affectation au niveau de cette sous-classe.

Supposons que l'administrateur système souhaite que *Process2* figure dans une superclasse disposant de plus de ressources, et décide d'affecter manuellement *Process2* à *superclasseC*. *Process2* avait été affecté manuellement à la *sous-classeB* de *superclasseA*, via une affectation de type "sous-classe uniquement". *Process2* étant affecté à une superclasse différente, l'affectation manuelle précédente perd sa signification et est annulée. *Process2* a désormais une affectation manuelle de type "superclasse uniquement" à la *superclasseC*, et est affecté à une sous-classe de cette dernière conformément aux règles d'affectation automatiques.

Maintenant, l'administrateur système décide de mettre fin à l'affectation manuelle du *Processus 1* vers la *superclasse B*. L'affectation manuelle "niveau superclasse" du *Processus 1* est annulée et en l'absence d'héritage, le *Processus 1* est affecté à une superclasse en utilisant les règles d'affectation automatique de premier niveau.

Si les règles n'ont pas changé, *Process1* est affecté à *superclasseA* et son affectation manuelle de niveau sous-classe à *superclasseB.sousclasseC* perd sa signification et est annulée.

Si, pour une raison quelconque, les règles de niveau supérieur affectent *Process1* à *superclasseB*, l'affectation de niveau sous-classe à *superclasseB.sousclasseC* reste valide et est appliquée. *Process1* a désormais une affectation manuelle de type "sous-classe" uniquement.

Mise à jour de WLM

Lors de la mise à jour de WLM (avec la commande **wlmcntrl -u**), la configuration mise à jour peut charger un nouveau groupe de règles de classification. Dans ce cas, les processus sont souvent reclassifiés à l'aide de nouvelles règles. WLM ne reclassifie pas les processus affectés manuellement ou qui se trouvent dans une classe dont l'héritage est activé, à moins que leur classe n'existe pas dans la nouvelle configuration.

Sécurité

Pour affecter un process à une classe ou annuler une affectation manuelle antérieure, l'utilisateur doit disposer de droits sur le process et sur la classe cible. Ces contraintes se traduisent comme suit :

- L'utilisateur root peut affecter tout process à n'importe quelle classe.
- Un utilisateur possédant des droits d'administration sur les sous-classes d'une superclasse donnée (c'est-à-dire que le nom d'utilisateur ou du groupe correspond à celui spécifié dans les attributs **adminuser** et **admingroup** de la superclasse) peut réaffecter manuellement tout process de l'une des sous-classes de cette superclasse à une autre de ses sous-classes.
- Un utilisateur peut affecter manuellement ses process (même ID utilisateur réel ou en vigueur) à une sous-classe pour laquelle il dispose de droits d'affectation manuelle (c'est-à-dire que le nom d'utilisateur ou du groupe correspond à celui spécifié dans les attributs **authuser** et **authgroup** de la superclasse ou de la sous-classe).

Il y a donc trois niveaux de droits parmi les personnes qui peuvent affecter manuellement des process à des classes, root étant le plus élevé. Pour qu'un utilisateur puisse modifier ou mettre fin à une affectation manuelle, il doit avoir au moins le même niveau de droit que la personne à l'origine de la dernière affectation manuelle.

Gestion des ressources à l'aide de WLM

WLM contrôle et régule l'utilisation des ressources des routines et processus actifs du système. Le contrôle et la régulation s'effectuent par classe. Vous pouvez définir des limites minimum ou maximum par classe pour chaque type de ressource gérée par WLM. En outre, il est possible d'affecter une valeur cible par classe pour chaque ressource. Cette valeur représente le quota de ressources optimal pour les tâches de cette classe.

Les partages et les limites au niveau des superclasses font référence à la quantité totale de chaque ressource disponible sur le système. Au niveau des sous-classes, elles font référence à la quantité de chaque ressource mise à la disposition de la superclasse à laquelle appartient la sous-classe (superclasse cible). La hiérarchie des classes est un moyen de diviser les ressources système entre groupes d'utilisateurs (superclasses) et de déléguer l'administration de ce partage des ressources aux administrateurs de superclasses. Chaque administrateur de superclasse peut alors redistribuer cette quantité de ressources entre les utilisateurs du groupe en créant des sous-classes et en définissant les droits à ressources pour ces sous-classes.

Types de ressources

WLM gère trois types de ressources en fonction d'une consommation en pourcentage :

Utilisation du CPU par les routines d'une classe	Somme de tous les cycles CPU consommés par chaque routine de la classe.
Utilisation de la mémoire physique par les processus d'une classe	Somme de toutes les pages mémoire qui appartiennent aux processus de la classe.
Bande passante d'E/S disque de la classe	Bande passante (en blocs de 512 Ko/seconde) de toutes les E/S lancées par des routines de la classe sur chaque unité de disque à laquelle la classe accède.

Chaque seconde, WLM calcule l'utilisation par classe de chaque ressource au cours de la dernière seconde, sous forme de pourcentage des ressources totales disponibles.

- Pour le CPU, la quantité totale de temps CPU disponible chaque seconde est égale à une seconde multipliée par le nombre de CPU du système. Par exemple, sur un SMP à huit voies, si toutes les routines combinées d'une classe ont consommé 2 secondes de temps CPU pendant la dernière seconde, cela représente un pourcentage de $2/8 = 25\%$. Le pourcentage utilisé par WLM est une moyenne étalée sur quelques secondes de cette utilisation des ressources par seconde "instantanée".
- Pour la mémoire physique, la quantité totale de mémoire physique disponible pour les processus à un moment donné est égale au nombre total de pages mémoire physiquement présentes sur le système, moins le nombre de pages réservées. Ces dernières ne sont pas gérées par WLM, puisqu'elles ne peuvent être prises dans une classe et données à une autre pour réguler l'utilisation de mémoire. L'utilisation de mémoire d'une classe est le rapport entre le nombre de pages mémoire non réservées détenues par tous les processus de la classe et le nombre de pages disponibles dans le système, exprimé sous forme de pourcentage.
- Pour les E/S disque, le problème principal est celui de déterminer un "bande passante" disponible significative pour un périphérique. Lorsqu'un disque est occupé à 100 %, son débit en blocs par seconde est très différent lorsqu'une application effectue des E/S séquentielles et lorsque plusieurs applications génèrent des E/S aléatoires. L'emploi du débit maximal mesuré avec des E/S séquentielles comme valeur de bande passante d'E/S disponible afin de calculer le pourcentage d'utilisation du périphérique, peut laisser penser que ce dernier est utilisé à 20 %, alors qu'en fait il l'est à 100 %.

Pour obtenir des pourcentages d'utilisation disque par classe plus précis et fiables, WLM fait appel aux statistiques fournies par les pilotes de disque (affichées à l'aide de la commande **iostat**), qui donnent pour chaque unité de disque le pourcentage de temps pendant lequel le périphérique a été occupé au cours de la dernière seconde. WLM compte le nombre total de blocs qui ont été lus ou écrits sur un périphérique pendant la dernière seconde par toutes les classes accédant au périphérique, le nombre de blocs qui ont été lus ou écrits par chaque classe, et le pourcentage d'utilisation du périphérique. WLM calcule ensuite le pourcentage de débit du disque consommé par chaque classe.

Par exemple, si le nombre total de blocs lus ou écrits au cours de la dernière seconde était de 1000 et si le périphérique était utilisé à 70 %, cela signifie qu'une classe qui lit ou écrit 100 blocs utilise 7 % de la bande passante du disque. Comme pour le temps CPU (qui est également une ressource renouvelable), les valeurs utilisées par WLM pour la régulation des E/S disque sont également une moyenne étalée sur quelques secondes de ces pourcentages par seconde.

Pour les ressources d'E/S disque, les partages et limites s'appliquent à chaque unité de disque à laquelle la classe accède individuellement. La régulation est effectuée indépendamment pour chaque périphérique. En d'autres termes, une classe peut avoir dépassé ses droits sur un périphérique, et les E/S sur ce dernier seront régulées, tout en étant en deçà de ses droits sur un autre disque, ses E/S sur cet autre disque n'étant alors pas limitées.

Dans la version AIX 5.2 et les versions supérieures, WLM prend en charge la comptabilité et la régulation des ressources en fonction de la consommation totale. Il existe deux types de ressources qui peuvent être régulées de cette manière : totaux de classe ou totaux de processus

Totaux de classe

Des limites par classe peuvent être définies pour le nombre de processus, de threads et de sessions de connexion dans la classe. Ces limites sont définies sous forme de valeurs absolues pour chaque ressource qui existe dans la classe à un moment donné. Ces limites sont appliquées de manière stricte. Lorsqu'une classe a atteint sa limite pour l'une de ces ressources, toute tentative de création d'une autre instance de la ressource échoue. L'opération échoue toujours pour tout processus de la classe jusqu'à ce que la classe tombe en dessous de sa limite définie pour la ressource.

totaux de processus

Des limites par processus peuvent être définies pour le temps UC total, les blocs d'E/S de disque et le délai de connexion pour une session de connexion. Ces limites sont définies au niveau de la classe, mais s'appliquent à chaque processus de la classe (chaque processus peuvent utiliser ce délai). Ces valeurs de consommation sont cumulatives et représentent donc la quantité totale associée à chaque ressource qui peut être consommée par le processus au cours de sa durée de vie. Lorsqu'un processus dépasse sa limite totale pour une ressource, le processus prend fin. Le processus reçoit un signal SIGTERM et s'il le détecte et ne se termine pas après une période de grâce de 5 secondes, il reçoit un signal SIGKILL. Lorsqu'une session de connexion atteint 90 % de sa limite de délai, un message d'avertissement est envoyé au terminal de contrôle pour signaler à l'utilisateur que la session va prendre fin.

Partages cibles

Le pourcentage de consommation de la ressource cible (ou désirée) d'une classe est déterminé par le nombre de partages dont il dispose pour la ressource. Les partages indiquent la quantité d'une ressource dont peut disposer une classe, par rapport aux autres classes de son niveau. Un pourcentage cible de classe pour une ressource correspond simplement au nombre de partages divisé par le nombre de partages actifs dans son niveau. Si des limites sont également utilisées, la cible est limitée à la plage [minimum, maximum logiciel]. Si la cible calculée est en dehors de la plage, elle est affectée de la

limite supérieure/inférieure appropriée (voir Limites de ressource). Le nombre de partages actifs correspond au nombre total de partages de toutes les classes contenant au moins un processus actif. Du fait que le nombre de partages actifs est dynamique, la cible l'est aussi. Si une classe est la seule classe active d'un niveau, sa cible correspond à 100 % de la quantité de ressource disponible pour le niveau.

Supposons trois classes de superclasses actives A, B, et C dans le niveau 0, avec des partages pour une ressource de 15, 10 et 5, respectivement. Les cibles sont :

$$\text{target}(A) = 15/30 = 50\%$$

$$\text{target}(B) = 10/30 = 33\%$$

$$\text{target}(C) = 5/30 = 17\%$$

Si ultérieurement, la classe B devient inactive (aucun processus actif), les cibles des classes A et C sont ajustées automatiquement :

$$\text{target}(A) = 15/20 = 75\%$$

$$\text{target}(C) = 5/20 = 25\%$$

Comme vous pouvez le constater, les partages représentent un pourcentage ajusté automatiquement qui permet aux ressources allouées à une classe d'être réparties de manière égale entre les autres classes ou d'être prises aux autres classes, lorsqu'il devient actif/inactif.

Pour plus de souplesse, le nombre de partages d'une classe peut correspondre à un nombre compris entre 1 et 65535. Des partages peuvent être définis pour les superclasses et les sous-classes. Pour les superclasses, les partages sont relatifs à toutes les autres superclasses actives du même niveau. Pour les sous-classes, les partages sont relatifs à toutes les autres sous-classes actives de la même superclasse du même niveau. Les partages d'une sous-classe d'une superclasse n'ont pas de relation avec les partages d'une sous-classe d'une autre superclasse.

Dans certains cas, il peut être nécessaire de rendre la cible d'une classe indépendante du nombre de partages actifs. Pour ce faire, définissez la valeur "–" pour le nombre de partages. Dans ce cas, la classe de la ressource n'est pas régulée, ce qui implique qu'elle n'a pas de partages et que sa cible ne dépend pas du nombre de partages actifs. Sa cible correspond à (ressource disponible pour le niveau – somme des minimaux de toutes les autres classes du niveau). Cette cible, ou la consommation actuelle (qui est inférieure), est issue de ce qui est disponible dans les autres classes du même niveau.

Supposons que les classes A, B, C et D aient des partages pour une ressource donnée, qui correspondent à "–", 200, 150 et 100, respectivement. Toutes les classes sont actives et la classe A consomme 50 % de la ressource :

$$\text{target}(A) = \text{unregulated} = 100\%$$

$$\text{target}(B) = 200/450 * \text{available} = 44\% * 50\% = 22\%$$

$$\text{target}(C) = 150/450 * \text{available} = 33\% * 50\% = 17\%$$

$$\text{target}(D) = 100/450 * \text{available} = 22\% * 50\% = 11\%$$

Du fait que la classe A n'est pas régulée et qu'elle consomme 50 % de la ressource disponible, les autres classes ne disposent que de 50 % de la ressource et leurs cibles sont calculées en fonction de ce pourcentage. Du fait que la classe A est toujours en dessous sa cible (100 %), elle a toujours une priorité supérieure par rapport à toutes les autres classes du même niveau qui ont atteint leur cible ou qui sont supérieures à leur cible (pour plus d'informations, voir Description de la priorité des classes).

Remarque : Ne pas réguler une classe pour une ressource ne revient pas à la mettre dans un niveau supérieur. Les comportements suivants sont vrais pour une classe non régulée (dans le même niveau) et ne sont pas vrais si la classe est placée dans un niveau supérieur :

- Du fait que les partages sont définis pour chaque ressource, une classe peut être dérégulée pour une ou plusieurs ressources et régulée pour d'autres.

- . Les limites minimales des autres classes du même niveau sont respectées. Les niveaux supérieurs ne respectent pas les minimaux définis dans les niveaux inférieurs.
- . Même en l'absence de limites minimales pour les classes avec des partages, la consommation des classes non régulées dépend quelque peu des classes avec des partages du fait qu'elles entrent en compétition pour obtenir de la ressource disponible pour le niveau. Il est nécessaire de faire des tests pour identifier le comportement avec une charge de travail donnée.

Si le nombre de partages n'est pas défini, la valeur par défaut "–" est utilisée et la classe est dérégulée pour la ressource. Notez que dans la première version de WLM, la valeur de partage par défaut, si elle n'est pas définie, est 1.

Les partages sont définis pour chaque classe pour tous les types de ressources. Les partages sont définis dans des strophes du fichier des **partages**. Par exemple :

```
shares
  classname :
  CPU       = 2
  memory    = 4
  diskIO    = 3
```

Limites de ressources

WLM utilise non seulement des partages pour définir des affectations de ressource, mais permet également de définir des limites de ressource pour une classe. Les limites de ressource permettent à l'administrateur de mieux contrôler l'affectation de ressources. Ces limites sont définies sous forme de pourcentages et sont relatives à la quantité de ressources disponible pour le niveau de la classe.

Il existe trois types de limites de régulation par pourcentage :

- | | |
|--------------|--|
| Minimum | Il s'agit de la quantité minimale de ressources qui peut être disponible pour la classe. Si la consommation réelle de la classe est inférieure à cette valeur, la classe est affectée d'une priorité d'accès supérieure à la ressource. Les valeurs possibles sont comprises entre 0 et 100, 0 étant la valeur par défaut (si non définie). |
| Soft Maximum | Il s'agit de la quantité de ressource maximale qu'une classe consomme lorsqu'il existe un conflit pour cette ressource. Si la consommation de la classe est supérieure à cette valeur, la classe est affectée de la priorité la plus basse dans son niveau. S'il n'existe pas de conflit pour la ressource (avec des classes du même niveau), la classe peut consommer une quantité de ressources illimitée. Les valeurs possibles sont comprises entre 1 et 100, 100 étant la valeur par défaut (si non définie). |
| Hard Maximum | Il s'agit de la quantité de ressources maximale qu'une classe consomme même lorsqu'il n'existe pas de conflit. Si la classe atteint cette limite, elle ne peut pas consommer plus de la ressources tant que son pourcentage de consommation ne tombe pas en dessous de la limite. Les valeurs possibles sont comprises entre 1 et 100, 100 étant la valeur par défaut (si non définie). |

Les valeurs de limites de ressource sont définies dans le fichier des limites de ressources par type de ressource dans les strophes de chaque classe. Les limites sont définies comme un minimum dans la plage logicielle maximale, séparées par un trait d'union (–), les espaces étant ignorés. Les valeurs maximales matérielles lorsqu'elles sont définies suivent les valeurs maximales logicielles en étant séparées par un point-virgule (;). Chaque valeur de limite est suivie par un symbole de pourcentage (%).

Voici des exemples d'utilisation des fichiers de règles :

- Si l'utilisateur Jean du groupe acct3 exécute **/bin/vi**, le processus est placé dans la superclasse acctg.

- Si l'utilisateur Jeanne du groupe dev exécute **/bin/emacs**, le processus est placé dans la superclasse devlt (correspondance d'ID de groupe), mais il n'est pas classé dans la sous-classe editors, car l'utilisateur Jeanne est exclu de cette classe. Le processus est placé dans devlt. Default
- Lorsqu'un administrateur DB démarre **/usr/sbin/oracle** avec l'ID utilisateur oracle et l'ID de groupe dbm, pour servir la base de données DB1, le processus doit être classé dans la superclasse Default. Le processus n'est affecté à la superclasse db1 que lorsqu'il affecte _DB1 à son option.

Les limites sont définies pour tous les types de classes, par classe, dans les strophes du fichier des **limites**. Par exemple :

```
shares
  classname :
    CPU      =   0%-50%;80%
    memory   =   10%-30%;50%
```

Dans cet exemple, aucune limite n'est définie pour les E/S de disque. En utilisant les valeurs par défaut système, on obtient :

```
diskIO     =   0%-100%;100%
```

Tous les exemples précédents supposent que l'attribut d'héritage des superclasses et des sous-classes décrites est affecté de la valeur yes. Autrement, les nouveaux processus héritent simplement de la superclasse ou de la sous-classe de leur parent.

Compatibilité ascendante

Partages de ressources

Dans les versions précédentes de WLM, la valeur par défaut pour les partages de ressources non spécifiés était de 1 partage. Dans la version actuelle, la valeur par défaut est tiret (-), ce qui signifie que la consommation de ressources de la classe pour la ressource concernée n'est pas régulée par WLM. Il s'agit d'un changement de sémantique et il est conseillé aux administrateurs système de contrôler leurs configurations actuelles pour s'assurer que la nouvelle valeur par défaut convient pour leurs classes, ou s'il est préférable de définir une valeur par défaut de 1 pour revenir à l'ancien comportement, ou encore s'il vaut mieux définir des valeurs explicites pour certaines des classes. Pour plus d'informations sur la définition de valeurs utilisateur, reportez-vous à *shares* dans *AIX Files Reference*.

Les versions antérieures de WLM n'avaient qu'une limite maximale. Il s'agissait d'une limite "souple" pour le CPU et d'une limite "ferme" pour la mémoire. Les limites indiquées dans l'ancien format, `min%-max%`, seront interprétées comme un maximum souple pour le CPU et un maximum ferme pour la mémoire. Pour les limites qui sont définies avec la version actuelle, toutes les interfaces à l'exception de l'édition directe des fichiers convertiront les données existantes au nouveau format, ou créeront de nouvelles limites conformes au nouveau format. Les administrateurs faisant appel à l'édition de texte devront convertir manuellement le fichier **limits**.

Nouvelle ressource

La ressource d'E/S disque est nouvelle pour la version actuelle, de sorte que lorsque vous activez la nouvelle version de WLM avec des fichiers de configuration de l'ancienne version, les valeurs pour les partages et les limites sont les valeurs par défaut. Les valeurs par défaut du système sont les suivantes :

- `shares = -`
- `min = 0%`, `softmax = 100%`, `hardmax = 100%`

Pour les configurations WLM existantes, la ressource d'E/S disque n'est pas régulée par WLM, ce qui doit entraîner le même comportement pour la classe que dans la version précédente.

Les contraintes suivantes sont les seules que WLM place sur les valeurs des limites de ressources.

- La limite minimum doit être inférieure ou égale à la limite logicielle maximum.
- La limite logicielle maximum doit être inférieure ou égale à la limite matérielle maximum.
- La somme des minimaux de toutes les superclasses d'un niveau ne peut pas être supérieure à 100.
- La somme des minimaux de toutes les sous-classes d'un niveau ne peut pas être supérieure à 100.

Lorsqu'une classe avec une limite de mémoire matérielle atteint sa limite et demande plus de pages, l'algorithme de remplacement de page VMM est démarré et "vole" des pages dans la classe limitée en ramenant ainsi le nombre de pages sous la limite matérielle maximum avant de fournir de nouvelles pages. Ce comportement est correct, mais l'augmentation des activités de pagination, qui peut se produire même lorsqu'il existe un grand nombre de pages libres, affecte les performances générales du système. Des limites de mémoire minimum pour les autres classes sont recommandées avant d'imposer un maximum de mémoire matérielle pour une classe.

Du fait que les classes en dessous de leur minimum ont la priorité la plus haute dans leur niveau, la somme des minimaux doit correspondre à un niveau raisonnable, en fonction des besoins en ressources des autres classes du même niveau.

La contrainte de la somme des limites minimum dans un niveau étant égale ou inférieure à 100, implique qu'une classe dans le niveau de priorité le plus haut est toujours autorisée à recevoir des ressources jusqu'à hauteur de sa limite minimum. WLM ne garantit pas que la classe atteindra sa limite minimum. Cela dépend de la manière dont les processus de la classe utilisent leurs ressources et des autres limites en vigueur. Par exemple, une classe peut ne pas atteindre sa limite de temps UC minimum, car elle ne peut pas obtenir une quantité de mémoire suffisante.

Pour la mémoire physique, la définition d'une limite de mémoire minimum permet de protéger les pages des processus de la classe (au moins pour ceux du niveau de priorité le plus haut). Une classe ne doit pas voir ses pages "volées" lorsqu'elle se trouve en dessous de sa limite minimum, à moins que toutes les classes actives soient en dessous de leur limite minimum et que l'une d'entre elles demande plus de pages. Une classe dans le niveau de priorité le plus haut ne doit pas voir ses pages "volées" lorsqu'elle se trouve en dessous de sa limite minimum. La définition de limites de mémoire minimum pour une classe de travaux interactifs permet d'empêcher toutes leurs pages d'être "volées" entre des activations consécutives (même lorsque la quantité de mémoire disponible est faible) et d'améliorer les temps de réponse.

Attention : L'utilisation inappropriée de limites maximales matérielles peut avoir un impact significatif sur les performances du système ou des applications. Du fait qu'en définissant des limites matérielles, des ressources peuvent ne pas être utilisées, dans la plupart des cas, il est préférable d'utiliser des limites maximales logicielles. Les limites maximales matérielles peuvent être utilisées pour limiter la consommation d'un niveau supérieur pour rendre disponible une ressource à un niveau inférieur, mais il est préférable de placer les applications qui ont besoin de ressources dans un niveau supérieur.

Depuis la version AIX 5.2, des limites totales peuvent être définies dans le fichier des limites avec les valeurs et les unités figurant dans le tableau suivant :

Table 2. Limites de ressources pour Workload Manager

Ressource	Unités autorisées	Unité par défaut	Valeur maximale	Valeur minimale
totalCPU	s, m, h, d, w	s	$2^{30} - 1$ s	10 s
totalDiskIO	KB, MB, TB, PB, EB	KB	$(2^{63} - 1) * 512/1\ 024$ KB	1 MB
totalConnect Time	s, m, h, d, w	s	$2^{63} - 1$ s	5 m
totalProcesses	–	–	$2^{63} - 1$	2
totalThreads	–	–	$2^{63} - 1$	2
totalLogins	–	–	$2^{63} - 1$	1

Remarque : Les spécificateurs d'unité ne tiennent pas compte de la casse. s = secondes, m = minutes, h = heures, d = jours, w = semaine, KB = kilo-octets, MK = mégaoctets, etc.

Exemple de strophe de limites :

```
BadUserClass:
    totalCPU = 1m
    totalConnectTime = 1h
```

Le nombre total de limites peut être défini en utilisant n'importe quelle valeur du tableau ci-dessus avec les restrictions suivantes :

- Si définie, la valeur totalThreads doit correspondre au minimum à la valeur de totalProcesses.
- Si totalThreads est défini et que totalProcesses ne l'est pas, la limite de totalProcesses correspond à la valeur de totalThreads.

Les limites totales peuvent être définies au niveau de la superclasse et de la sous-classe. Lors de la vérification des limites, la limite de la sous-classe est vérifiée avant la limite de la superclasse. Si les deux limites sont définies, la limite la plus basse est utilisée. Si la limite de la sous-classe est supérieure à la limite de la superclasse associée, un avertissement est envoyé lorsque la configuration est chargée. Cela est significatif pour les limites totales de classe du fait que la limite absolue (et non par rapport à la superclasse) et une sous-classe peuvent consommer toutes les ressources disponibles pour la superclasse. Si non définie, la valeur par défaut de toutes les limites totales est "–", ce qui correspond à aucune limite. Par défaut, la comptabilité et la régulation de total de classes et de processus sont activées lorsque WLM est actif. L'option **–T [class]proc** de la commande **wlmcntrl** peut être utilisée pour désactiver la comptabilité et la régulation.

Description de la priorité de classe

WLM alloue des ressources aux classes en affectant une priorité à chaque classe de chaque ressource. La priorité d'une classe est dynamique et repose sur le niveau, les partages, les limites et la consommation actuelle de la classe. A tout moment, la classe ou les classes ayant la priorité la plus haute ont un accès préférentiel aux ressources. Au niveau le plus haut, les niveaux représentent les plages non entrelacées de la priorité de classe. Les classes du niveau 0 ont toujours une priorité supérieure à celle des classes du niveau 1 (sauf si supérieure à la valeur maximale matérielle)

Lors de la détermination de la priorité d'une classe, WLM utilise ses contraintes avec la priorité suivante (de la plus haute vers la plus basse) :

limites matérielles

Si la consommation de la classe est supérieure à la limite maximale matérielle d'une ressource, la classe est affectée de la priorité la plus basse et ne peut pas accéder à la ressource tant que son niveau de consommation ne tombe pas en dessous de cette limite.

niveau

En l'absence de limites matérielles, une priorité de classe est limitée par les priorités minimale et maximale autorisées pour son niveau.

limites logicielles

Si la consommation de la classe est inférieure à la valeur minimale des limites maximales logicielles, la classe est affectée de la priorité la plus haute dans le niveau. Si la consommation de la classe est supérieure à la valeur maximale logicielle, la classe est affectée de la priorité la plus basse du niveau.

partages

Les partages permettent de calculer les cibles de consommation de classe pour chaque ressource. La priorité de classe augmente alors que la consommation de classe tombe sous la valeur cible, et diminue alors que la consommation de la classe est supérieure à la valeur cible. Notez que les limites logicielles ont une plus haute priorité et que la priorité d'une classe est déterminée en fonction de ces limites, lorsque nécessaire.

Même si les partages et les limites peuvent être utilisés pour chaque classe et pour chaque ressource, les résultats sont plus prévisibles si l'un ou l'autre est utilisé pour chaque classe.

Groupes de ressources

WLM utilise des groupes de ressources (ou *rsets*) pour restreindre les processus dans une classe à un sous-ensemble des ressources physiques du système. Dans WLM, les ressources physiques gérées sont la mémoire et les processeurs. Un groupe de ressources valide est constitué de la mémoire et d'au moins un processeur.

En utilisant SMIT ou Web-based System Manager, un administrateur système peut définir et nommer des groupes de ressources contenant un sous-ensemble des ressources disponibles sur le système. Ensuite, en utilisant les interfaces d'administration WLM, l'utilisateur root ou un administrateur de superclasse désigné peut utiliser le nom du groupe de ressources sous la forme de l'attribut **rset** d'une classe WLM. Ensuite, chaque processus affecté à la classe WLM est envoyé uniquement à l'un des processeurs du groupe de ressources, en séparant efficacement les charges de travail de la ressource UC.

Tous les systèmes actuels ont uniquement un domaine de mémoire partagé par tous les groupes de ressources. En conséquence, cette méthode ne sépare pas physiquement les charges de travail en mémoire.

Registre Rset

Les services de registre **rset** permettent à l'administrateur système de définir et de nommer des groupes de ressources pour permettre à d'autres utilisateurs ou applications de les utiliser. Pour diminuer les risques de conflits de noms, le registre prend en charge un schéma d'appellation à deux niveaux. Le nom du groupe de ressources se présente sous la forme *espace_nom / nom_rset*. *espace_nom* et *nom_rset* peuvent comporter jusqu'à 255 caractères, tiennent compte de la casse et peuvent contenir des majuscules, des minuscules, des chiffres, le caractère souligné et des points (.). L' *espace de nom* de **sys** est réservé par le système d'exploitation et utilisé pour les définitions **rset** qui représentent les ressources du système.

Les noms de définition **rset** sont uniques dans l'espace de nom de registre. Si vous ajoutez au registre une définition **rset** dont le nom correspond à une définition **rset** existante, la définition existante est remplacée par la nouvelle définition en affectant l'autorisation et le privilège appropriés. Seul l'utilisateur root peut créer, modifier et supprimer des groupes de ressources et mettre à jour la base de données in-core **rset** à l'aide de SMIT ou Web-based System Manager.

Chaque définition **rset** a ses propres propriétaires (ID utilisateur), groupe (ID de groupe) et autorisations d'accès. Ces données sont spécifiques lors de la création de la définition **rset** et existent pour contrôler les accès. Comme c'est le cas pour les fichiers, des autorisations d'accès distinctes existent pour le propriétaire, le groupe et les autres qui définissent si des autorisations de lecture et/ou d'écriture ont été accordées. L'autorisation de lecture permet d'extraire une définition **rset** alors que l'autorisation d'écriture permet de modifier ou de supprimer une définition **rset**.

Les définitions **rset** définies par l'administrateur système sont stockées dans le fichier de strophe */etc/rsets*. Le format de ce fichier n'est pas décrit et les utilisateurs doivent manipuler les définitions **rsets** via SMIT ou les interfaces Web-based System Manager pour éviter tout problème de compatibilité ultérieur si le format de fichier est modifié. Comme c'est le cas avec les définitions de classes WLM, les définitions **rset** doivent être chargées dans les structures de données du noyau pour que WLM puisse les utiliser.

Au lieu de donner une longue définition des groupes de ressources et d'expliquer comment les créer depuis des blocs de base appelés *RAD système* (Resource Access Domains). Pour plus d'informations sur ce qu'est un groupe de ressources et la manière d'en créer un, reportez-vous à la section *Creating a Resource Set* du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

Configuration de WLM

Les définitions de classe, les attributs de classe, les partages et les limites et les règles d'affectation de classe automatique peuvent être entrés avec Web-based System Manager, SMIT ou l'interface de ligne de commande WLM. Ces définitions et règles sont stockées dans des fichiers texte qui peuvent être également créés ou modifiés dans un éditeur de texte.

Ces fichiers (appelés *fichiers de propriétés WLM*) sont stockés dans les sous-répertoires de **/etc/wlm**. Un groupe de fichiers décrivant les superclasses et leurs sous-classes définissent une configuration WLM. Les fichiers de la configuration WLM **Config** sont stockés dans **/etc/wlm/Config**. Ce répertoire contient les définitions des paramètres WLM des superclasses. Les fichiers s'appellent **description**, **classes**, **shares**, **limits** et **rules**. Ce répertoire peut également contenir des sous-répertoires avec le nom des superclasses où sont stockées les définitions des sous-classes. Par exemple, pour la superclasse *Super* de la configuration WLM **Config**, le répertoire **/etc/wlm/Config/Super** contient les fichiers de propriétés des sous-classes de la superclasse *Super*. Les fichiers s'appellent **description**, **classes**, **shares**, **limits** et **rules**.

Une fois que l'administrateur système a défini une configuration WLM, la configuration peut être activée à l'aide de Web-based System Manager, du raccourci **smit wlmmanage** ou de la commande **wlmcntrl**.

Vous pouvez définir plusieurs groupes de fichiers de propriétés qui définissent différentes configurations de gestion de la charge de travail. Ces configurations sont généralement stockées dans les sous-répertoires de **/etc/wlm**. Un lien symbolique **/etc/wlm/current** pointe vers le répertoire qui contient les fichiers de configuration en cours. Ce lien est mis à jour par la commande **wlmcntrl** au démarrage de WLM, avec un groupe défini de fichiers de configuration.

La création d'une configuration WLM est décrite dans la section Configure Workload Manager (WLM) to Consolidate Workloads du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*. Cette section décrit les concepts et les caractéristiques de chaque phase de la procédure.

Identification des besoins des applications

La première étape de la définition d'une configuration repose sur l'identification des besoins des utilisateurs et informatiques ainsi que sur les applications du système, leurs besoins en ressources et les besoins de votre entreprise (par exemple, quelles tâches sont importantes et quelle est celle qui doit recevoir la priorité la plus basse). À partir de ces informations, vous pouvez définir les superclasses, puis les sous-classes.

La définition des priorités dépend des fonctions que WLM dessert dans votre organisation. Dans le cas de consolidation de serveurs, il se peut que vous connaissiez déjà les applications, les utilisateurs et leurs besoins en ressources ; vous pourrez alors ignorer ou raccourcir certaines des étapes.

WLM vous permet de classer les processus par utilisateur ou groupe, application, type, étiquette ou selon une combinaison de ces attributs. Comme WLM régule l'utilisation des ressources entre les classes, groupez les applications et les ressources ayant les mêmes modèles d'utilisation des ressources dans les mêmes classes. Par exemple, il peut s'avérer judicieux de séparer les travaux interactifs, qui généralement consomment très peu de temps CPU mais exigent des temps de réponse rapides, des travaux de traitement par lots, qui consomment beaucoup de CPU et de mémoire. La situation est identique dans un environnement de bases de données où vous devez séparer le trafic de type OLTP des requêtes d'exploration de données, qui représentent une lourde charge.

Cette étape s'effectue à l'aide des interfaces d'administration de WLM, Web-based System Manager, SMIT ou interface à ligne de commande. Pour les premières occasions, il est probablement préférable d'utiliser Web-based System Manager ou SMIT. Ces outils vous guideront dans les étapes de création de votre première configuration WLM, dont la définition des superclasses et de leurs attributs. Lors de la première passe, vous pouvez définir certains des attributs et laisser les autres à leur valeur par défaut. Il en va de même des partages de ressources et des limites. Toutes ces caractéristiques peuvent être modifiées dynamiquement par la suite.

Vous pouvez alors démarrer WLM en mode passif, vérifier votre classement et commencer à consulter les trames d'utilisation des ressources de vos applications.

Vérifiez votre configuration à l'aide de la commande **wlmcheck** ou des menus correspondants de SMIT ou de Web-based System Manager, puis lancez WLM en mode passif sous la nouvelle configuration. WLM classera tous les processus existants (et tous ceux créés par la suite) et commencera à compiler des statistiques sur le CPU, la mémoire et l'utilisation des E/S disque des différentes classes. Il ne tentera pas de réguler cette utilisation des ressources.

Vérifiez que les différents processus sont classés dans la bonne classe, tel que cela a été prévu par l'administrateur système (à l'aide du paramètre **-o** de la commande **ps**). Si certains processus ne sont pas classés conformément à vos attentes, vous pouvez modifier vos règles d'affectation ou définir le bit d'héritage pour certaines des classes (si vous souhaitez que les nouveaux processus restent dans la même classe que leur parent) et mettre WLM à jour. Vous pouvez répéter ce processus jusqu'à ce que ce premier niveau de classification vous convienne (superclasses).

L'exécution de WLM en mode passif et sa régénération (toujours en mode passif) entraîne peu de risques, une faible surcharge système et peut être lancée en toute sécurité sur un système de production sans perturber le fonctionnement normal du système. L'activation et la régénération de WLM sont effectuées avec la commande **wlmcntrl**, appelée soit directement, soit via SMIT ou Web-based System Manager.

L'exécution de WLM en mode passif permet de collecter des statistiques à l'aide de la commande **wlmstat**. **wlmstat** peut être utilisée à intervalles réguliers pour afficher l'utilisation des ressources par classe sous forme de pourcentage du total des ressources disponibles, pour les superclasses. Ceci vous permet de surveiller votre système pendant des périodes prolongées pour vérifier l'utilisation des ressources par vos principales applications.

Définition des niveaux, des partages et des limites

En utilisant les données collectées en exécutant WLM en mode passif et en fonction de vos objectifs professionnels, déterminez le nombre de niveaux à associer à chaque superclasse et quel partage pour chaque ressource doit être affecté aux diverses classes. Pour certaines classes, vous pouvez définir des limites minimales et maximales. Ajustez les partages et le nombre de niveaux pour atteindre vos objectifs d'allocation de ressources. Réservez des limites pour les cas qui ne peuvent pas être résolus qu'avec des partages. En outre, vous pouvez être amené à ajouter des sous-classes.

Dans certains cas, il peut être nécessaire d'utiliser les limites minimales et maximales. Ajustez les partages et les numéros de niveau pour parvenir à vos objectifs en matière d'affectation des ressources. Réservez les limites aux cas que vous ne pouvez pas résoudre avec les seuls partages.

- Utilisez les limites minimales pour les applications qui ont en général une faible utilisation des ressources mais ont besoin d'un temps de réponse court lorsqu'elles sont activées par un événement externe. L'un des problèmes auxquels sont confrontés les travaux interactifs lorsque la mémoire devient insuffisante est que leurs pages leur sont retirées pendant les périodes d'inactivité. Une limite de mémoire minimale peut servir à protéger certaines des pages des travaux interactifs si leur classe est dans le niveau 0.

- Faites appel aux limites maximales pour restreindre certains travaux de faible priorité consommant beaucoup de ressources. Sauf si vous partitionnez vos ressources système pour d'autres raisons, une limite ferme prend son sens généralement pour une ressource non renouvelable telle que la mémoire. Ceci est dû au temps nécessaire à l'écriture de données hors de l'espace de pagination lorsqu'une classe de priorité plus élevée a besoin de pages que la première classe aurait utilisée. Pour l'utilisation du CPU, vous pouvez utiliser des niveaux ou des maximums souples pour garantir qu'une classe de priorité plus élevée se voie affecter immédiatement du temps CPU.

Lorsque vous créez et ajustez les paramètres des sous-classes, vous pouvez actualiser WLM uniquement pour les sous-classes d'une superclasse qui n'affectent pas les utilisateurs et les applications des autres superclasses, jusqu'à ce que le comportement du système vous convienne.

Vous pouvez également définir d'autres configurations avec des paramètres différents en fonction de vos besoins. Dans ce cas, vous pouvez gagner du temps en copiant et en modifiant les configurations existantes.

Ajustement de la configuration

Maintenant, vous êtes prêt à démarrer WLM en mode actif. Contrôlez le système avec la commande **wlmstat**, vérifiez que la régulation effectuée par WLM correspond à vos objectifs et assurez-vous qu'il ne supprime pas des ressources pour des applications et qu'il n'en affecte pas plus à d'autres. Si tel est le cas, ajustez les partages et actualisez WLM.

Lorsque vous contrôlez et ajustez les partages, les limites et le nombre de niveaux, déterminez si vous voulez déléguer l'administration des sous-classes ou de toutes les superclasses. L'administrateur peut ensuite contrôler et définir les partages de sous-classes, les limites et le nombre de niveaux.

L'administrateur de chaque superclasse peut répéter ce processus pour les sous-classes de chaque superclasse. La seule différence est qu'il n'est pas possible de lancer WLM en mode passif au niveau de la sous-classe seulement. La configuration et l'optimisation de la sous-classe doivent être effectuées lorsque WLM est en mode actif. Pour éviter tout impact sur les utilisateurs et les applications de la superclasse, vous pouvez lancer le numéro de niveau, ainsi que les partages et les limites des sous-classes avec leur valeur par défaut ('-' (tiret) pour les partages, 0 % pour les minimums, et 100 % pour les maximums souples et fermes). Avec ces options, WLM ne régulera pas l'affectation de ressources entre les sous-classes.

API de Workload Manager

Les applications peuvent utiliser les API WLM, un groupe de sous-routines de la bibliothèque `/usr/lib/libwlm.a`, pour exécuter toutes les tâches qu'un administrateur WLM peut effectuer à l'aide des commandes WLM, notamment :

- créer des classes ;
- modifier des classes ;
- supprimer des classes ;
- affecter manuellement des process à des classes ;
- extraire des statistiques WLM.

De plus, la routine `wlm_set_tag` permet à une application de définir l'étiquette (tag) d'une application et de spécifier si cette étiquette doit être héritée par les process enfants au niveau de `fork` ou d'`exec`. La bibliothèque permet la prise en charge d'applications 32 ou 64 bits multi-routines.

Etiquette d'application

Cette étiquette est une chaîne de caractères pouvant servir de critère pour la classification automatique des process (à l'aide du fichier `rules`). Elle offre un critère de classification défini par application, en complément des critères définis par le système tels que `user`, `group`, `application` et `type`.

Lorsqu'un process d'application définit son étiquette, il est immédiatement reclassé en fonction des règles de superclasse et de sous-classe en vigueur pour la configuration WLM active. WLM vérifie alors les règles d'affectation, recherchant une correspondance à partir de tous les attributs de process, y compris la nouvelle étiquette.

Pour avoir un effet, cette étiquette doit apparaître dans une ou plusieurs des règles d'affectation. Cela signifie que le format et l'utilisation des différentes étiquettes par chaque application doivent être clairement indiqués dans la documentation d'administration de ces applications et être connus des administrateurs WLM, afin qu'ils puissent exploiter les différentes valeurs des étiquettes dans leurs règles d'affectation en vue de faire la distinction entre les différentes instances de la même application.

Le jeu de caractéristiques des process de l'application dont ils ont besoin pour classer ces process pouvant varier selon les utilisateurs, il est conseillé de faire en sorte que l'application fournisse un jeu d'attributs de configuration ou d'exécution pouvant servir à constituer l'étiquette. L'administrateur de l'application aura ainsi la possibilité de spécifier à l'application le format de l'étiquette. Les attributs susceptibles d'être utilisés et la syntaxe servant à spécifier le format de l'étiquette WLM dépendent de l'application et sont sous la responsabilité de son fournisseur.

Par exemple, une instance d'un serveur de bases de données est capable de déterminer sur quelle base de données elle travaille (`db_name`) et via quel port TCP (`port_num`) un utilisateur donné est connecté. Les administrateurs peuvent avoir les priorités suivantes :

- créer différentes classes pour les process accédant à différentes bases de données, afin de donner à chaque classe des droits à ressource distincts ;
- séparer les process desservant des demandes distantes d'origines différentes et employer le numéro de port comme attribut de classification ;
- créer une superclasse pour chaque base de données et une sous-classe par numéro de port dans chaque superclasse.

Pour répondre à ces différents besoins, une solution consiste à spécifier le contenu et le format de l'étiquette. Dans cet exemple, nous supposons que l'étiquette peut être transmise à l'application dans un fichier de configuration ou un paramètre d'exécution tel que `WLM_TAG=$db_name` ou `WLM_TAG=$db_name_$port_num`.

Lors de la définition de cette étiquette, une application peut spécifier si ses enfants en hériteront, afin que tous les process issus d'une instance particulière d'une application puissent être classés dans la même classe. Dans la plupart des cas, l'étiquette d'application sert à définir l'étiquette d'héritage.

L'exemple suivant montre comment il est possible d'utiliser les étiquettes d'application. Dans cet exemple, l'étiquette de la base de données est identique au nom de la base de données. Deux instances du serveur travaillant sur des bases de données différentes définiront deux étiquettes distinctes, *db1* et *db2*.

L'administrateur système peut créer deux classes distinctes, *dbserv1* et *dbserv2*, puis classer les deux serveurs de bases de données (ainsi que leurs enfants, en cas de recours à l'héritage d'étiquette) dans ces classes à l'aide des étiquettes. Il est alors possible de donner à chaque classe des droits à ressource distincts, en fonction des objectifs de l'entreprise.

Les règles d'affectation correspondantes seront semblables à ceci :

```
* class   resvd  user  group   application   type  tag
*
dbserv1   -      -    dbadm   /usr/sbin/dbserv -   db1
dbserv2   -      -    dbadm   /usr/sbin/dbserv -   db2
```

API de gestion de classe

L'API WLM permet aux applications de :

- demander les noms et les caractéristiques des classes existant dans une configuration WLM donnée (**wlm_read_classes**) ;
- créer une nouvelle classe pour une configuration WLM particulière, définir les valeurs des différents attributs de la classe (rang, héritage, etc.) ainsi que les partages et limites des ressources gérées par WLM, telles que le CPU, la mémoire physique et les E/S par blocs (**wlm_create_class**) ;
- modifier les caractéristiques d'une classe existant dans un configuration WLM particulière, notamment les attributs de la classe et ses partages et limites de ressources (**wlm_change_class**) ;
- supprimer une classe existante d'une configuration donnée (**wlm_delete_class**).

Les modifications ne sont appliquées qu'au fichier de propriétés de la configuration WLM spécifiée. Il est également possible, en indiquant une chaîne vide comme nom de configuration, de n'appliquer les modifications qu'aux classes chargées en mémoire centrale, ce qui entraîne une mise à jour immédiate de l'état de la configuration active.

Pour effectuer des appels à l'API, l'appelant doit disposer des mêmes droits que pour une exécution à partir des interfaces à ligne de commande, SMIT et Web-based System Manager :

- tout utilisateur peut lire les noms et les caractéristiques de la classe,
- seul l'utilisateur root peut créer, modifier ou supprimer des superclasses,
- seul root ou des administrateurs de superclasse désignés (attribut de superclasse *adminuser* ou *admingroup*) peuvent créer, modifier ou supprimer des sous-classes d'une superclasse donnée.

Lorsque l'administration de WLM est assurée simultanément par les administrateurs de WLM via les outils à ligne de commande et d'administration ainsi que par des applications via l'API, il convient d'être particulièrement vigilant. Les deux interfaces partagent le même espace de noms pour les noms des superclasses et des sous-classes, et le nombre total de superclasses et de sous-classes.

De surcroît, lorsque l'API modifie directement les données WLM en mémoire centrale (en créant de nouvelles classes par exemple), les administrateurs WLM n'en sont avertis que lorsqu'ils voient apparaître des classes qu'ils n'ont pas créées dans le résultat de commandes telles que **wlmstat**. Pour éviter tout conflit susceptible de perturber les applications utilisant cette API lorsque l'administrateur met à jour WLM, les classes créées via l'API qui ne sont pas définies dans les fichiers de propriétés de WLM ne sont pas automatiquement retirées des données en mémoire centrale. Elles restent en vigueur jusqu'à ce qu'elles soient explicitement supprimées par la routine `wlm_delete_class` ou par une exécution de la commande **rmclass** (appelée par l'administrateur système soit directement, soit via SMIT ou Web-based System Manager).

API de gestion de WLM

L'API de WLM permet également aux applications de :

- demander ou modifier le mode de fonctionnement de WLM, à l'aide de la fonction **wlm_set** ;
- demander l'état actuel de WLM ;
- arrêter WLM ;
- basculer du mode actif au mode passif et vice versa ;
- activer et désactiver la liaison **rset** ;
- démarrer ou mettre à jour WLM avec la configuration en cours ou une autre, à l'aide de la routine **wlm_load** ;
- affecter un process ou un groupe de process à une classe, à l'aide de la routine **wlm_assign**.

L'API doit avoir le même niveau de droits que les commandes de l'interface à ligne de commande **wlmcntrl** et **wlmassign** :

- Tout utilisateur peut demander l'état de WLM.
- Seul root peut changer le mode de fonctionnement de WLM.
- Seul root peut mettre à jour ou actualiser une configuration complète.
- root ou un administrateur de superclasse autorisé (*adminuser* ou *admingroup*) peut mettre à jour WLM en ce qui concerne les sous-classes d'une superclasse donnée.
- root, un utilisateur autorisé (spécifié à l'aide de *authuser* ou de **authgroup**), ou un administrateur autorisé de superclasse (*adminuser* ou *admingroup*) peut affecter des process à une superclasse ou une sous-classe. Reportez-vous à **wlmassign** pour plus de détails.

API de statistiques de WLM

Les routines de l'API de WLM et **wlm_get_bio_stats** permettent aux applications d'accéder aux statistiques de WLM affichées par les commandes **wlmstat**.

API de classification de WLM

La routine **wlm_check** permet à l'utilisateur de vérifier les définitions de classes et les règles d'affectation d'une configuration WLM. La routine API **wlm_classify** permet à une application de déterminer la classe dans laquelle un processus avec un groupe d'attributs défini doit être placé.

Compatibilité binaire

Afin d'assurer une compatibilité binaire en cas de changements ultérieurs des structures de données, chaque appel à l'API se voit transmettre comme paramètre un numéro de version qui permettra à la bibliothèque de déterminer avec quelle version des structures de données l'application a été bâtie.

Commandes WLM

WLM dispose de commandes qui permettent à l'administrateur système de :

- créer, modifier et supprimer des superclasses et des sous-classes en utilisant les commandes **mkclass**, **chclass** et **rmclass**. Ces commandes mettent à jour les *classes*, *partages* et *limites* du fichier.
- Utilisez la commande **wlmcntrl** pour démarrer, arrêter et mettre à jour WLM.
- Vérifier les fichiers de propriétés WLM d'une configuration et déterminer la classe (superclasse et sous-classe) à laquelle est affecté un processus avec un groupe d'attributs donné en utilisant la commande **wlmcheck**.
- Contrôler l'utilisation des ressources par classe à l'aide de la commande **wlmstat** (ASCII). La majorité des outils de contrôle des performances, tels que ceux démarrés par les commandes **svmon** et **topas**, ont des extensions pour prendre en compte les classes WLM et fournissent des statistiques sur les classes et les niveaux en utilisant de nouvelles options de ligne de commande.
- Les options de la commande **ps** permettent à l'utilisateur d'afficher la classe à laquelle appartient un processus et son option d'application. La commande **ps** permet également de lister tous les processus qui appartiennent à une superclasse ou sous-classe.
- Il n'existe pas d'interface de ligne de commande pour gérer les règles d'affectation. Vous devez utiliser SMIT ou les outils d'administration Web-based System Manager ou un éditeur de texte.

Exemple de classification, de règles et de limites WLM

Il existe plusieurs méthodes pour classer un processus et ces méthodes fonctionnent toutes simultanément. Un algorithme de première correspondance vertical est utilisé pour fournir une souplesse de configuration maximale. Vous pouvez organiser le regroupement des processus par utilisateur avec des cas spéciaux pour les programmes portant un certain nom, par nom de chemin avec des cas spéciaux pour certains utilisateurs et en fonction d'autres arrangements.

Exemples d'affectation de règles

Voici un exemple de fichier de **règles** de haut niveau pour la configuration *Config* (*/etc/wlm/Config/fichier de règles*) :

```
* Ce fichier contient les règles utilisées par WLM pour
* affecter un processus à une superclasse
*
* class resvd user      group  application      type  tag
db1      -      -      -      /usr/bin/oracle*  _DB1
db2      -      -      -      /usr/bin/oracle*  _DB2
devlt    -      -      dev      -      -      -
VPs      -      bob,ted  -      -      -      -
acctg    -      -      acct*    -      -      -
System   -      root    -      -      -      -
Default  -      -      -      -      -      -
```

Voici un exemple de fichier **règles** pour la superclasse **devlt** dans */etc/wlm/Config/devlt/fichier de règles* :

```

* Ce fichier contient les règles utilisées par WLM pour
* affecter un processus à une sous-classe de la
* superclasse devlt
*
* class resvd user      group      application      type      tag
hackers  -      jim,liz    -      -      -      -
hogs     -      -          -      -      64bit+plock -
editors  -      !sue      -      /bin/vi, /bin/emacs -
build   -      -          -      /bin/make, /bin/cc  -
Default -      -          -      -      -      -

```

Remarque : L'astérisque (*) est le caractère de commentaire utilisé dans le fichier des règles.

Voici des exemples d'utilisation de ce fichier de règles : Tous les exemples suivants supposent que l'attribut d'héritage des superclasses et des sous-classes décrites est affecté de la valeur yes. Si l'héritage était activé, les nouveaux processus hériteraient de la superclasse ou de la sous-classe de leurs processus parents.

- Si l'utilisateur Jean du groupe acct3 exécute **/bin/vi**, le processus est placé dans la superclasse acctg.
- Si l'utilisateur Jeanne du groupe dev exécute **/bin/emacs**, le processus est placé dans la superclasse devlt (correspondance d'ID de groupe), mais il n'est pas classé dans la sous-classe editors, car l'utilisateur Jeanne est exclu de cette classe. Le processus est placé par défaut dans devlt.
- Lorsqu'un administrateur de base de données avec l'ID utilisateur oracle et l'ID de groupe dbm démarre **/usr/sbin/oracle** pour servir la base de données DB1, le processus doit être classé dans la superclasse Default. Le processus n'est affecté à la superclasse db1 que lorsqu'il affecte _DB1 à son option.

Exemples de partages et de limites

Supposons que les classes A, B, C et D aient les parages 3, 2, 1 et 1, respectivement. Si les classes A, C et D sont actives, les cibles calculées sont :

target(A) = 3/5 = 60%

target(C) = 1/5 = 20%

target(D) = 1/5 = 20%

Si lors du test, il s'avère que les applications de la classe A fonctionnent correctement lorsqu'elles sont autorisées à utiliser 50 % de la ressource, il peut être préférable de fournir les 50 % restant de la ressource aux autres classes. Cela peut être effectué en fournissant à la classe A un maximum logiciel de 50 % pour la ressource. Du fait que la cible calculée actuelle 60 % est au-dessus de cette limite, elle est ramenée à la valeur logique maximum. Dans ce cas, la consommation cible ou actuelle (selon la plus basse des deux) de la classe A est soustraite de la quantité de ressource disponible. Du fait que la classe a désormais une cible contrainte par sa limite (et non ses partages), les partages de la classe sont également soustraits du nombre de partages actifs. En supposant que la consommation de la classe A est de 48 %, les cibles sont maintenant :

target(A) = 3/5 = 60%, softmax = 50, = 50%

target(C) = 1/2 * (100 - 48) = 26%

target(D) = 1/2 * (100 - 48) = 26%

Un peu plus tard, toutes les classes peuvent devenir actives et les cibles sont là encore ajustées automatiquement :

target(A) = 3/7 = 42%

target(B) = 2/7 = 28%

target(C) = 1/7 = 14%

target(D) = 1/7 = 14%

Exemple avec des limites d'UC

L'exemple suivant examine l'allocation d'UC, en supposant que chaque classe consomme toute l'UC qui lui est fournie.

Les classes A et B sont dans le même niveau. Les limites d'UC de A sont [30 % – 100 %]. Les limites d'UC de B sont [20 % – 100 %]. Lorsque les deux classes sont actives et qu'elles utilisent suffisamment d'UC, WLM vérifie qu'elles obtiennent leurs pourcentages minimum de chaque seconde (moyenne de plusieurs secondes). Ensuite WLM distribue les cycles UC restants en fonction des valeurs de partages cibles UC.

Si les partages cibles UC de A et B correspondent respectivement à 60 % et 40 %, l'utilisation de l'UC pour A et B se stabilise respectivement à 60 % et 40 %.

La troisième classe C est ajoutée. Cette classe est un groupe de travaux UC et doit être exécutée avec la moitié du temps (ou plus) UC disponible. La classe a les limites [20% – 100%] et les partages cibles d'UC 100 %. Si C se trouve dans le même niveau que A et B, quand C démarre, l'allocation d'UC de A et B diminue franchement et les trois classes se stabilisent à 30 %, 20 % et 50 %, respectivement. Dans ce cas, leurs cibles correspondent également au minimum de A et B.

Un administrateur système peut décider d'empêcher les travaux par lot de consommer 50 % de l'UC lorsque d'autres travaux, avec une priorité supérieure éventuellement, sont également actifs. Dans le cas de l'exemple précédent, C est placé dans le niveau de priorité le plus bas. C reçoit le temps UC qui reste après que A et B ont reçu le temps UC qui leur est nécessaire. Dans l'exemple ci-dessus, C ne reçoit aucun temps UC, car A et B peuvent consommer 100 % du temps UC. Dans la plupart des cas, toutefois, A et B, dans un niveau de priorité élevé, sont constitués de travaux interactifs ou orientés transaction qui n'utilisent pas tout le temps UC. C reçoit du temps UC pour lequel il est en compétition avec d'autres classes du même niveau ou des niveaux inférieurs.

Exemple avec des limites de mémoire

L'exemple suivant examine l'allocation de mémoire à des groupes de processus avec des cibles de mémoire variables. Trois groupes de processus doivent être actifs : un groupe de processus interactifs qui doivent être exécutés chaque fois qu'ils sont utilisés (PEOPLE), un travail par lot qui doit être toujours exécuté en arrière-plan (BATCH1) et un second travail par lot plus important qui est exécuté toutes les nuits (BATCH0).

PEOPLE a un minimum de mémoire de 20 %, une cible de mémoire de 50 partages et une valeur de niveau de classe de 1. Une limite de 20 % minimum garantit que les applications bureautiques de cette classe reprennent leur exécution rapidement lorsque l'utilisateur touche le clavier.

BATCH1 a un minimum de mémoire de 50 %, une cible de mémoire de 50 partages et une valeur de niveau de 3.

BATCH0 a un minimum de mémoire de 80 %, une cible de mémoire de 50 partages et une valeur de niveau de 2.

Les classes PEOPLE et BATCH1 ont une limite de mémoire totale minimum de 70. Dans des conditions normales (lorsque BATCH0 est actif), ces deux classes peuvent utiliser l'ensemble de la mémoire qui leur est réservée. Elles partagent le reste de la mémoire dans la machine à hauteur de 50 %, même si elles se trouvent dans des niveaux différents. A minuit, lorsque BATCH0 démarre, le total de mémoire minimum atteint 150. WLM ignore le minimum du niveau le plus bas jusqu'à ce que les processus des niveaux supérieurs se terminent. BATCH0 prend de la mémoire de la réserve de 50 % de BATCH1, mais pas de la réserve de 20 % de PEOPLE. Une fois BATCH0 terminé, les réserves de mémoire pour les processus du niveau 3 sont de nouveau honorées et le système revient à son équilibre de mémoire normale.

Compatibilité amont

Lorsque vous démarrez WLM avec des configurations créées avec des versions antérieures à AIX 5.1, seules les superclasses sont utilisées. La sortie par défaut de la commande **wlmstat** liste uniquement les superclasses qui sont similaires à celles des versions précédentes. Par exemple :

```
# wlmstat
      CLASS  CPU  MEM  DKIO
Unclassified  0   0   0
  Unmanaged  0   0   0
    Default  0   0   0
    Shared   0   2   0
    System   2  12   0
    class1   0   0   0
    class2   0   0   0
#
```

Si des superclasses contiennent des sous-classes définies par un administrateur WLM, les sous-classes sont affichées. Par exemple :

```
# wlmstat
      CLASS  CPU  MEM  DKIO
Unclassified  0   0   0
  Unmanaged  0   0   0
    Default  0   0   0
    Shared   0   2   0
    System   3  11   7
    class1   46   0   0
class1.Default 28   0   0
class1.Shared  0   0   0
  class1.sub1 18   0   0
    class2   48   0   0
#
```

La sortie est identique lorsque vous utilisez la commande **ps**. Pour les processus d'une superclasse sans sous-classes, la commande **ps** affiche le nom de la superclasse comme nom de classe de processus.

Chapitre 10. SRC et sous-systèmes

Ce chapitre présente le contrôleur de ressources système (SRC) et les différents sous-systèmes qu'il gère. Pour les procédures relatives aux tâches, reportez-vous à la section SRC et sous-systèmes dans le manuel *AIX 5L Version 5.2 Guide de Gestion du Système: Système d'exploitation et unités*.

SRC - généralités

Le contrôleur SRC (System Resource Controller) fournit un ensemble de commandes et de sous-routines rendant le travail de création et de contrôle des sous-systèmes plus facile pour l'administrateur et le programmeur. Un *sous-système* est un programme, un processus, un ensemble de programmes ou un ensemble de processus capables d'opérer indépendamment ou avec un système de contrôle. Le sous-système est conçu comme une unité remplissant une fonction donnée.

Le SRC a été conçu pour minimiser l'intervention de l'opérateur. Il fournit un mécanisme de contrôle des processus du sous-système opérant à partir d'une ligne de commande et avec l'interface C. Ce mécanisme comprend :

- Une interface utilisateur cohérente pour lancer, arrêter et générer des états.
- La journalisation des arrêts anormaux des sous-systèmes.
- Un programme de notification appelé lors de l'arrêt système anormal des processus associés.
- Le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
- La prise en charge du contrôle des opérations sur un système distant.
- Le rafraîchissement d'un sous-système (par exemple, après en avoir modifié la configuration).

SRC représente un moyen simple de lancer et arrêter la collecte d'informations sur l'état des processus.

Composants du sous-système

Un sous-système peut être doté des propriétés suivantes :

- Il possède un nom connu du système.
- Il requiert un environnement d'exécution plus complexe qu'un sous-routine ou un programme non privilégié.
- Il intègre des programmes d'application, des bibliothèques et un code sous-système.
- Il contrôle des ressources qu'il peut démarrer et arrêter par leur nom.
- Il a besoin d'être notifié de l'échec d'un processus associé pour effectuer un nettoyage ou restaurer des ressources.
- Il requiert un contrôle opérationnel plus important qu'un simple processus démon.
- Il a besoin d'être contrôlé à distance par un opérateur.
- Il met en oeuvre des sous-serveurs pour gérer des ressources spécifiques.
- Il ne se place pas lui-même en arrière-plan.

Voici quelques sous-systèmes : ypserv, ntsd, qdaemon, inetd, syslogd et sendmail.

Remarque : Pour en savoir plus sur les fonctions SRC d'un sous-système spécifique, reportez-vous à ce sous-système.

Pour la liste des sous-systèmes actifs et inactifs de votre système, exécutez la commande **lssrc -a**.

Groupe de sous-systèmes

Un *groupe de sous-systèmes* est un ensemble de sous-systèmes spécifiés. Regrouper des sous-systèmes permet de les contrôler ensemble et en une seule opération. Voici quelques exemples de groupes de sous-systèmes : TCP/IP, SNA Services, NIS (Network Information System) et NFS (Network File Systems).

Sous-serveur

Un *sous-serveur* est un programme ou un processus appartenant à un sous-système. Un sous-système peut posséder plusieurs sous-serveurs, auquel cas il est responsable du démarrage et de l'arrêt des sous-serveurs ; il doit en outre fournir leurs états. Les sous-serveurs ne peuvent être définis que pour un sous-système communiquant par sockets et files de message IPC. Les sous-systèmes communiquant par signaux sont incompatibles avec les sous-serveurs.

Les sous-serveurs sont automatiquement démarrés en même temps que leurs sous-systèmes parent. Si vous tentez de démarrer un sous-serveur dont le parent est inactif, la commande **startsrc** démarre également le sous-système.

Structure hiérarchique de SRC

SRC présente une structure hiérarchique. En début de structure, se trouve le système d'exploitation, suivi d'un groupe de sous-systèmes (tel que **tcpip**) lui-même contenant un sous-système (tel que le démon **inetd**), qui, à son tour, peut posséder plusieurs sous-serveurs (tels que le démon **ftp** et la commande **finger**).

Commandes d'administration SRC

démon srcmstr	Démarre SRC.
commande startsrc	Démarre un sous-système, un groupe de sous-systèmes ou un sous-serveur.
commande stopsrc	Arrête un sous-système, un groupe de sous-systèmes ou un sous-serveur.
commande refresh	Rafraîchit un sous-système.
commande traceson	Active le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
commande tracesoff	Désactive le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
commande lssrc	Recherche l'état d'un sous-système.

Chapitre 11. Comptabilité système

L'utilitaire de comptabilité système permet de collecter des informations et de générer des rapports sur l'exploitation (de groupe ou individuelle) des différentes ressources système.

Les sujets traités sont les suivants :

- Généralités sur la comptabilité
- Collecte et rapports de données système
- Collecte des données de comptabilité
- Rapports de données de comptabilité
- Commandes de comptabilité
- Fichiers de comptabilité

Comptabilité - généralités

Avec l'utilitaire de système de comptabilité d'AIX, vous pouvez rassembler et rendre compte de l'utilisation individuelle, de groupe et de classe Workload Manager (WLM) de différentes ressources système.

Ces informations peuvent servir à facturer l'exploitation des ressources aux utilisateurs et à contrôler certains aspects de l'exploitation du système. Pour faciliter la facturation, le système de comptabilité fournit le cumul par membre du groupe adm de l'utilisation des ressources et, avec la commande **chargefee** (le cas échéant), les éléments de facturation.

En outre, le système de comptabilité permet d'évaluer l'adéquation des affectations de ressources en cours, de définir des limites et des quotas de ressources, de planifier les besoins futurs et de commander des fournitures pour les imprimantes et autres unités.

Pour la mise en œuvre de l'utilitaire de comptabilité sur votre système, reportez-vous à :

- Collecte et rapport de données système
- Collecte de données comptables
- Rapport de données comptables
- Commandes comptables
- Fichiers comptables

Collecte et rapport de données système

Pour la collecte automatique de données, un membre du groupe adm doit suivre les procédures décrites à "Mise en œuvre d'un système de comptabilité" dans le manuel "Setting Up an Accounting System". Ces procédures permettent au démon **cron** d'exécuter des commandes générant des données sur :

- le temps de connexion de chaque utilisateur sur le système,
- l'exploitation du processeur, de la mémoire et des ressources d'entrée-sortie,
- l'espace disque occupé par chaque fichier utilisateur,
- l'exploitation des imprimantes et des traceurs,
- l'exploitation des commandes.

Chaque session et chaque processus terminé fait l'objet d'un enregistrement écrit par le système. Ces enregistrements sont convertis en enregistrements comptables cumulés (tacct), regroupés par utilisateur puis fusionnés dans un rapport quotidien.

Un rapprochement des rapports quotidiens est effectué régulièrement pour générer les cumuls concernant l'exercice fiscal défini. Les méthodes de collecte des données et de rapport sur celles-ci, ainsi que les différents fichiers et commandes comptables sont décrits ci-après.

Toutefois, pour obtenir des informations spécifiques, un membre du groupe adm peut entrer certaines commandes (au clavier). Ces commandes sont décrites à "Commandes clavier", page 11-7.

Collecte de données comptables

Il existe différents types de données comptables :

Données sur les durées de connexion,

Les processus,

L'utilisation disque,

L'utilisation de l'imprimante et

La taxation.

Chacun de ces types est décrit dans les paragraphes suivants.

Durée de connexion

Ce sont les commandes **init** et **login** qui collectent ce type de données. Lors de la connexion, le programme **login** écrit un enregistrement dans le fichier **/etc/utmp**. Dans cet enregistrement figurent le nom de l'utilisateur, la date, l'heure et le port de connexion. Les commandes, telles que **who**, utilisent ce fichier pour rechercher les utilisateurs connectés sur les différentes stations. Si le fichier comptable sur l'heure de connexion **/var/adm/wtmp** existe, la commande **login** y intègre une copie de l'enregistrement de connexion.

A la fin du programme de connexion (c'est-à-dire généralement lorsque vous vous déconnectez), la commande **init** enregistre la fin de session dans un autre enregistrement du fichier **/var/adm/wtmp**. A la différence des enregistrements de connexion, ces enregistrements ne reprennent pas le nom de l'utilisateur. Le format de ces deux types d'enregistrements est décrit dans le fichier **utmp.h**.

La commande **acctwtmp** écrit également des entrées spécifiques dans le fichier **/var/adm/wtmp**, relatives aux démarrages et fermetures du système.

Pour en savoir plus, reportez-vous à "Rapports de données comptables", page 11-5.

Processus

Le système collecte des données sur l'utilisation par chaque processus en cours des différentes ressources. Ces données comprennent :

- les numéros des utilisateurs et groupes exécutant des processus,
- les huit premiers caractères du nom de la commande,
- le temps écoulé et le temps processus utilisé par le processus,
- la mémoire utilisée,
- le nombre de caractères transférés,
- le nombre de blocs disque écrits ou lus pour l'exécution du processus.

La commande **accton** enregistre ces données dans un fichier spécial, généralement dans le fichier **/var/adm/pacct**.

Les commandes associées sont **startup**, **shutacct**, **dodisk**, **ckpacct** et **turnacct**.

Pour en savoir plus, reportez-vous à "Rapport de données comptables".

Utilisation disque

Un grand nombre de données comptables sont collectées pendant l'exploitation des ressources. La commande **dodisk**, exécutée comme défini par le démon **cron**, écrit périodiquement dans le fichier **/var/adm/acct/nite/dacct** un enregistrement par utilisateur sur l'utilisation des disques. Pour ce faire, **dodisk** appelle d'autres commandes. Selon que la recherche comptable est pointue ou non, c'est la commande **diskusg** ou **acctdusg** qui collecte les données. La commande **acctdisk** écrit l'enregistrement comptable cumulé. Cet enregistrement est exploité ensuite par la commande **acctmerg** pour préparer le rapport comptable quotidien.

dodisk facture à l'utilisateur les liens aux fichiers trouvés dans son répertoire de connexion et répartit équitablement le coût de chaque fichier entre ces liens. Ainsi, le coût d'utilisation d'un fichier est partagé également entre tous ceux qui l'utilisent et ne grève pas les utilisateurs qui renoncent à l'accès au fichier concerné.

Pour en savoir plus, reportez-vous à "Utilisation disque", page 11-3.

Utilisation de l'imprimante

La collecte des données sur l'utilisation de l'imprimante résulte de la coopération de la commande **enq** et du démon de mise en file d'attente. **enq** met en file d'attente le nom de l'utilisateur, le numéro du travail d'impression et le nom du fichier à imprimer. Après l'impression du fichier, la commande **qdaemon** écrit un enregistrement ASCII dans un fichier, généralement le fichier **/var/adm/qacct**, contenant le nom et numéro de l'utilisateur et le nombre de pages imprimées. Vous pouvez trier ces enregistrements et les convertir en un enregistrement comptable cumulé.

Pour en savoir plus, reportez-vous à "Utilisation disque".

Taxation

Avec la commande **chargefee**, vous pouvez générer dans le fichier **/var/adm/fee** un enregistrement comptable cumulé en ASCII. Ce fichier peut être intégré aux rapports quotidiens avec la commande **acctmerg**.

Pour en savoir plus, reportez-vous à "Comptabilité - généralités", page 11-3.

Rapport de données comptables

Une fois les différents types de données comptables collectées, les enregistrements sont traités et convertis en rapports.

Les commandes comptables convertissent automatiquement les enregistrements en notation scientifique dès lors que les nombres sont élevés. Les nombres sont représentés en notation scientifique au format suivant :

Base e+ Exp

OU

Base e- Exp

qui est égal au nombre *Base* multiplié par 10 puissance + *Exp* ou - *Exp*. Par exemple, la notation scientifique 1,345e+9 est égale à 1,345x10⁹ ou à 1 345 000 000. Et la notation scientifique 1,345e-9 est égale à 1,345x10⁻⁹ ou à 0,000000001345.

Durée de connexion

La commande **runacct** appelle deux commandes, **acctcon1** et **acctcon2**, pour traiter les enregistrements de connexion, de déconnexion et de fermeture du système collectés dans le fichier **/var/adm/wtmp**. La commande **acctcon1** convertit ces enregistrements en enregistrements de session et les inscrit dans le fichier **/var/adm/acct/nite/lineuse**. Ensuite la commande **acctcon2** convertit les enregistrements de session en un enregistrement comptable cumulé, **/var/adm/logacct**, intégré par la commande **acctmerg** aux rapports quotidiens.

Si vous exécutez **acctcon1** depuis la ligne de commande, vous devez l'assortir de l'indicateur **-l** pour générer le rapport correspondant, **/var/adm/acct/nite/lineuse**. Pour produire un rapport global sur la session pour la période comptable, **/var/adm/acct/nite/reboots**, utilisez la commande **acctcon1** avec l'indicateur **-o**.

La commande **lastlogin** produit un rapport donnant la dernière date à laquelle chaque utilisateur s'est connecté.

Processus

Deux commandes traitent les données de facturation collectées dans le fichier **/var/adm/pacct** ou dans un autre fichier spécifié. La commande **acctprc1** traduit l'ID utilisateur en un nom d'utilisateur et écrit des enregistrements ASCII contenant les éléments facturables : La commande **acctprc2** convertit ces enregistrements en enregistrements comptables cumulés intégrés aux rapports quotidiens par la commande **acctmerg**.

Les données comptables sur les processus fournissent des informations exploitables pour contrôler l'utilisation des ressources système. La commande **acctcms** en résume l'utilisation par nom de commande. Ce résumé indique combien de fois la commande a été exécutée, le temps processeur et la mémoire utilisés, et l'intensité d'exploitation de ces ressources (appelé *hog factor* = facteur de monopolisation). La commande **acctcms** génère des statistiques à long terme sur l'utilisation du système, dont l'utilisation totale et la fréquence d'utilisation des commandes.

La commande **acctcom** gère les mêmes données que **acctcms**, mais fournit des informations détaillées sur chaque processus. Vous pouvez afficher tous les enregistrements comptables de processus ou sélectionner ceux qui vous intéressent. Les critères de sélection comprennent la charge imposée par le processus, l'heure à laquelle le processus s'est terminé, le nom de la commande, l'utilisateur ou le groupe qui a appelé le processus, le nom de la classe WLM à laquelle le processus appartenait et le port sur lequel le processus s'est exécuté. Contrairement à d'autres commandes de comptabilité, **acctcom** peut être exécutée par tous les utilisateurs.

Utilisation disque

Les enregistrements d'utilisation disque collectés dans le fichier **/var/adm/acct/nite/dacct** sont fusionnés dans les rapports comptables quotidiens par la commande **acctmerg**.

Utilisation de l'imprimante

L'enregistrement ASCII dans le fichier **/var/adm/qacct** peut être converti en un enregistrement comptable cumulé à intégrer au rapport quotidien avec la commande **acctmerg**.

Taxation

Si vous avez utilisé la commande **chargefee** pour facturer aux utilisateurs des services tels que les restaurations et consultations de fichiers, ou des matériels, un enregistrement comptable cumulé en ASCII est inscrit au fichier **/var/adm/fee**. Ce dernier est intégré aux rapports quotidiens par la commande **acctmerg**.

Rapport quotidien

Les données comptables brutes sur les durées de connexion et des processus, l'utilisation disque et imprimante, et la taxation sont fusionnées en rapports quotidiens par la commande **acctmerg**. Appelée par la commande **runacct** dans le cadre des routines quotidiennes, **acctmerg** génère les rapports suivants :

/var/adm/acct/nite/dacct

Rapport intermédiaire généré lorsqu'un des fichiers d'entrée est plein.

/var/adm/acct/sum/tacct

Rapport de total cumulatif dans le format **tacct**. Ce fichier est utilisé par la commande **monacct** pour produire le résumé mensuel ASCII.

La commande **acctmerg** peut convertir les enregistrements de format ASCII en format binaire et fusionner les enregistrements de différentes sources pour créer un seul enregistrement pour chaque utilisateur.

Rapport mensuel

Appelé par le démon **cron**, la commande **monacct** génère le rapport suivant :

/var/adm/acct/fiscal produit par le rapport **/var/adm/acct/sum/tacct** avec la commande **monacct**. Selon la définition de cette commande, elle est exécutée tous les mois ou en fin d'exercice fiscal.

Commandes comptables

Les commandes comptables ne fonctionnent pas toutes de la même façon. Certaines :

- collectent des données ou produisent des rapports pour un type de comptabilité spécifique : durées des connexions ou des processus, utilisation disque ou imprimante, ou utilisation des commandes.
- appellent d'autres commandes. Par exemple, la commande **runacct**, généralement exécutée automatiquement par le démon **cron**, appelle nombre des commandes qui collectent et traitent les données comptables et préparent les rapports. Pour le traitement automatique de la comptabilité, vous devez tout d'abord configurer le démon **cron** pour exécuter **runacct**. Reportez-vous à la commande **crontab** pour plus de détails sur la configuration du démon **cron** en vue de soumettre des commandes à intervalles réguliers.
- exécutent des fonctions de maintenance et garantissent l'intégrité des fichiers de données actifs.
- permettent aux membres du groupe adm d'exécuter des tâches occasionnelles, telles que afficher des enregistrements spécifiques au moyen d'une commande entrée au clavier.
- permettent à l'utilisateur d'afficher des informations spécifiques. **acctcom** est l'unique commande utilisateur ; elle affiche la synthèse comptable des processus.

Commandes exécutées automatiquement

Plusieurs commandes, généralement exécutées par le démon **cron**, collectent automatiquement des données comptables.

- runacct** Gère la principale procédure comptable quotidienne. Généralement initiée par le démon **cron** en dehors des heures d'exploitation, **runacct** appelle plusieurs autres commandes comptables pour traiter les fichiers de données actifs et produire des synthèses sur l'utilisation des commandes et des ressources, triées par nom d'utilisateur. En outre, elle appelle la commande **acctmerg** pour générer une synthèse quotidienne et la commande **ckpacct** pour maintenir l'intégrité des fichiers de données actifs.
- ckpacct** Gère les tailles de fichiers **pacct**. Il est préférable de disposer de plusieurs petits fichiers **pacct** si vous devez redémarrer la procédure **runacct** après l'échec du traitement de ces enregistrements. La commande **ckpacct** vérifie la taille du fichier de données actif **/var/adm/pacct** et si le fichier contient plus de 500 blocs, la commande appelle la commande **turnacct switch** pour désactiver temporairement la comptabilité des processus. Les données sont transférées vers un nouveau fichier **pacct**, **/var/adm/pacct x**. (*x* est un entier dont la valeur augmente chaque fois qu'un fichier **pacct** est créé.) Si le nombre de blocs libres sur le disque tombe en dessous de 500, la commande **ckpacct** appelle la commande **turnacct off** pour désactiver la comptabilité des processus.
- dodisk** Appelle la commande **acctdisk** et la commande **diskusg** ou la commande **acctdusg** pour écrire les enregistrements d'utilisation du disque dans le fichier **/var/adm/acct/nite/dacct**. Ces données sont ensuite fusionnées dans les rapports quotidiens.
- dodisk** Appelle la commande **acctdisk** et la commande **diskusg** ou la commande **acctdusg** pour écrire les enregistrements d'utilisation du disque dans le fichier **/var/adm/acct/nite/dacct**. Ces données sont ensuite fusionnées dans les rapports quotidiens.
- monacct** Génère un résumé périodique à partir des rapports quotidiens.
- sa1** Collecte et stocke les données binaires dans le fichier **/var/adm/sa/sa dd**, où *dd* correspond au jour du mois.
- sa2** Écrit un rapport quotidien dans le fichier **/var/adm/sa/sa dd**, où *dd* correspond au jour du mois. La commande supprime du fichier **/var/adm/sa/sa dd** les rapports de plus d'une semaine.

Les autres commandes sont exécutées automatiquement par les procédures autres que le daemon **cron** :

- startup** Ajoutée au fichier **/etc/rc**, la commande **startup** lance les procédures de démarrage du système de comptabilité.
- shutacct** Enregistre l'heure de désactivation de la comptabilité en appelant la commande **acctwtmp** pour écrire une ligne dans le fichier **/var/adm/wtmp**. Elle appelle ensuite la commande **turnacct off** pour désactiver la comptabilité des processus.

Commandes disponibles à partir du clavier

Un membre du groupe adm peut entrer les commandes suivantes depuis le clavier :

- ac** Imprime les enregistrements d'heure de connexion. Cette commande est fournie pour la compatibilité avec les systèmes BSD (Berkeley Software Distribution).
- acctcom** Affiche les résumés de comptabilité des processus. Cette commande est également accessible aux utilisateurs.

acctcon1	Affiche les résumés des heures de connexion. L'option -l ou -o doit être utilisée.
accton	Active ou désactive la comptabilité des processus.
chargefee	Impute à l'utilisateur une charge prédéterminée pour les unités de travail effectuées. Les charges sont ajoutées au rapport quotidien par la commande acctmerg .
fwtmp	Convertit les fichiers binaires en fichiers ASCII et vice versa.
last	Affiche des informations sur les connexions précédentes. Cette commande est fournie pour la compatibilité avec les systèmes BSD.
lastcomm	Affiche des informations sur les dernières commandes exécutées. Cette commande est fournie pour la compatibilité avec les systèmes BSD.
lastlogin	Affiche l'heure de la dernière connexion de chaque utilisateur.
pac	Prépare les enregistrements de comptabilité d'imprimante/traceur. Cette commande est fournie pour la compatibilité avec les systèmes BSD (Berkeley Software Distribution).
prctmp	Affiche un enregistrement de session.
prtacct	Affiche les fichiers de comptabilité totale.
sa	Résume les informations de comptabilité brutes pour faciliter la gestion des informations de comptabilité volumineuses. Cette commande est fournie pour la compatibilité avec les systèmes BSD.
sadc	Fournit des rapports sur diverses actions système local, telles que l'utilisation des tampons, les E/S de disque et de bande, les compteurs d'activité du terminal TTY et les compteurs d'accès aux fichiers.
sar	Ecrit dans un format standard le contenu des compteurs d'activité cumulatifs sélectionnés dans le système d'exploitation. La commande sar fournit des rapports uniquement sur les activités locales.
time	Imprime l'heure réelle, l'heure utilisateur et l'heure système nécessaires pour exécuter une commande.
timex	Signale en secondes le délai écoulé, le délai utilisateur et le délai d'exécution.

Fichiers comptables

Il existe deux répertoires principaux de comptabilité : **/usr/sbin/acct** où sont stockés tous les programmes en C et les procédures shell indispensables pour lancer la comptabilité système et **/var/adm** contenant les fichiers de données, de rapports et de synthèse.

Ce sont les membres du groupe adm qui possèdent ces fichiers de données comptables ; tous les fichiers de données actifs (tels que **wtmp** et **pacct**) résident dans le répertoire personnel adm **/var/adm**.

Fichiers de données

Les fichiers du répertoire **/var/adm** sont les suivants :

/var/adm/diskdiag	sortie des diagnostics pendant l'exécution de programmes comptables sur disque.
/var/adm/dtmp	sortie de la commande acctdusg .
/var/adm/fee	sortie de la commande chargefee , sous forme d'enregistrements tacct en ASCII.
/var/adm/acct/	fichier comptable de processus actif.
/var/adm/wtmp	fichier comptable de processus actif.
/var/adm/Spacct.mddd	fichier comptable de processus actif.

Fichiers de rapport et de synthèse

Les fichiers de rapports et de synthèse résident dans un sous-répertoire **/var/adm/acct**. Avant d'activer le système de comptabilité, vous devez créer les sous-répertoires suivants. Pour en savoir plus, reportez-vous à "Mise en œuvre d'un système de comptabilité".

/var/adm/acct/nite	contient les fichiers réutilisés chaque jour par la commande runacct .
/var/adm/acct/sum	contient les fichiers de synthèse avec cumuls quotidiennement mis à jour par la commande runacct .
/var/adm/acct/fiscal	contient les fichiers de synthèse mensuelle créés par la commande monacct .

Fichiers de commande runacct

Les fichiers de rapport et de synthèse suivants, générés par la commande **runacct**, présentent un intérêt tout particulier :

/var/adm/acct/nite/lineuse	contient des statistiques sur l'utilisation de chaque ligne de terminal du système. Ce rapport est particulièrement utile pour détecter les lignes défectueuses. Une différence de plus de 3 à 1, le cas échéant, entre le nombre de déconnexions et de connexions signifie vraisemblablement qu'une ligne est défectueuse.
/var/adm/acct/nite/daytacct	contient le fichier comptable cumulé du jour précédent.
/var/adm/acct/sum/tacct	contient le cumul de chaque fichier nite/daytacct quotidien et peut servir à la facturation. La commande monacct réinitialise ce fichier chaque mois ou au moment de l'exercice fiscal.
/var/adm/acct/sum/cms	contient le cumul des synthèses quotidiennes de commandes. La commande monacct lit cette version binaire du fichier et la purge. Une version ASCII figure dans nite/cms .
/var/adm/acct/sum/daycms	contient la synthèse quotidienne de commandes dont une version ASCII est stockée dans nite/daycms .

/var/adm/acct/sum/loginlog	contient un enregistrement de la dernière date/heure à laquelle chaque ID utilisateur a été employé.
/var/adm/acct/sum/rprt mmdd	contient une copie du rapport quotidien sauvegardé par la commande runacct .

Fichiers du répertoire **/var/adm/acct/nite**

active	utilisé par la commande runacct pour enregistrer la progression et imprimer les messages d'erreur et les avertissements. Le fichier active.mmjj est une copie du fichier active effectuée par le programme runacct après une détection d'erreur.
cms	synthèse cumulée en ASCII sur les commandes utilisées par la commande prdaily .
ctacct. mmjj	enregistrements comptables cumulés sur les connexions.
ctmp	enregistrements sur les sessions de connexion.
daycms	synthèse quotidienne en ASCII sur les commandes, utilisée par la commande prdaily .
daytacct	enregistrements comptables cumulés sur un jour.
dacct	enregistrements comptables cumulés sur les disques, créés par la commande dodisk .
accterr	sortie des diagnostics générée pendant l'exécution de la commande runacct .
lastdate	dernière exécution de runacct , sous la forme date +%m%d .
lock1	contrôle l'exploitation en série de la commande runacct .
lineuse	rapport sur l'utilisation de la ligne tty, exploité par la commande prdaily .
log	sortie de la commande acctconl .
log mmjj	semblable à log après détection d'une erreur par la commande runacct .
reboots	contient les dates de début et fin de wtmp , et un listing des redémarrages du système.
statefile	enregistre l'état courant pendant l'exécution d'une commande runacct .
tmpwtmp	fichier wtmp corrigé par la commande wtmpfix .
wtmperror	Contient les messages d'erreur wtmpfix .
wtmperr mmjj	semblable à wtmperror après détection d'une erreur par la commande runacct .
wtmp. mmjj	Fichier wtmp du jour précédent. Supprimée pendant le nettoyage de la commande runacct .

Fichiers du répertoire **/var/adm/acct/sum**

cms	fichier de synthèse de toutes les commandes pour l'exercice fiscal en cours, en format binaire.
cmsprev	fichier de synthèse des commandes sans la dernière mise à jour.
daycms	fichier de synthèse des commandes pour le jour précédent, en format binaire.
lastlogin	fichier créé par la commande lastlogin .
pacct. mmjj	version concaténée de tous les fichiers pacct pour mmjj . Ce fichier est supprimé par la commande remove après le démarrage du système.
rprt mmjj	sortie sauvegardée de la commande prdaily .

tacct	fichier comptable de cumuls pour l'exercice fiscal en cours.
tacctprev	semblable à tacct sans la dernière mise à jour.
tacct mmjj	fichier comptable cumulé pour <i>mmjj</i> .

Fichiers du répertoire `/var/adm/acct/fiscal`

cms ?	fichier de synthèse de toutes les commandes pour l'exercice fiscal, spécifié par <i>?</i> , en format binaire.
fiscrpt ?	rapport semblable à celui de la commande prdaily pour l'exercice fiscal, spécifié par <i>?</i> , en format binaire.
tacct ?	fichier comptable cumulé pour l'exercice fiscal, spécifié par <i>?</i> , en format binaire.

Formats des fichiers comptables

Les sorties générées par les fichiers comptables et les formats correspondants sont décrits ci-après :

wtmp	fichier comptable de processus actifs, au format défini dans le fichier utmp.h .
ctmp	enregistrements sur les sessions de connexion. Le format est décrit dans le fichier ctmp.h .
pacct*	enregistrements comptables de processus actifs, au format défini dans le fichier /usr/include/sys/acct.h .
Spacct*	fichier comptable de processus actif. Le format de ces fichiers est défini dans le fichier sys/acct.h .
daytacct	enregistrements comptables cumulés sur un jour. Le format du fichier est défini dans le fichier tacct .
sum/tacct	fichier binaire cumulant les synthèses quotidiennes des commandes. Le format de ce fichier est défini dans le fichier d'en-tête /usr/include/sys/acct.h .
ptacct	versions concaténées des fichiers pacct . Le format de ces fichiers est défini dans le fichier tacct .
ctacct	enregistrements comptables cumulés sur les connexions. Le format de ce fichier est défini dans le fichier tacct .
cms	synthèse comptable cumulée des commandes exploitée chaque jour par la commande prdaily au format binaire. Une version ASCII figure dans nite/cms .
daycms	synthèse quotidienne des commandes exploitée par la commande prdaily au format binaire. Une version ASCII figure dans nite/daycms .

Chapitre 12 Web-based System Manager

Web-based System Manager est une application client–server qui fournit à l'utilisateur une interface graphique puissante pour accéder à plusieurs hôtes et les gérer. Avec Web-based System Manager, vous pouvez afficher les utilisateurs et les groupes, installer des logiciels, des imprimantes et des unités, gérer les volumes logiques, les utilisateurs, les groupes et les ressources, monter et démonter des systèmes de fichiers, configurer le réseau et exécuter de nombreuses autres tâches d'administration de système. Une architecture plug-in facilite l'extension de la suite. En outre, Web-based System Manager prend en charge le contrôle dynamique des événements système et la notification à l'administrateur.

L'interface offre un contrôle Pointer et cliquer sur les objets qui fournit une alternative à l'apprentissage des commandes ou de SMIT. Vous pouvez sélectionner la machine à gérer et lorsque vous naviguez vers l'opération à exécuter, les panneaux actualisent les données pour afficher les options autorisées.

Pour plus d'informations sur l'utilisation de Web-based System Manager et les procédures associées, reportez-vous au document *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

Chapitre 13. SMIT (System Management Interface Tool)

Bien que Web-based System Manager soit la principale interface de gestion de système, SMIT (System Management Interface Tool) fournit une interface alternative en langage naturel orientée tâches. L'outil SMIT s'exécute dans deux interfaces : ASCII (non graphique) ou AIXwindows (graphique).

SMIT vous aide à exécuter la tâche appropriée à l'aide de menus, de sélecteurs et de boîtes de dialogue, ce qui vous évite d'avoir à utiliser des syntaxes de commandes complexes et des valeurs de paramètres et les erreurs d'orthographe des commandes système. En outre, SMIT crée des fichiers journaux que vous pouvez utiliser pour dupliquer une configuration système ou pour apprendre à utiliser des commandes.

Dans l'interface SMIT, les options du menu principal permettent d'accéder à des sous-menus qui fournissent les options appropriées pour l'exécution d'une tâche. Pour ignorer le menu principal et accéder directement à un sous-menu ou une boîte de dialogue, vous pouvez utiliser la commande **smit** avec le paramètre *Fast Path*.

Pour en savoir plus sur SMIT, vous pouvez :

- Démarrer SMIT et sélectionner **Using SMIT (Information Only)** dans le menu principal SMIT.
- Dans les boîtes de dialogue SMIT, sélectionnez **On Context (Ctrl+F1)** dans le menu Help et placez le curseur sur l'option de menu appropriée ou dans le champ sur lequel vous voulez afficher des informations complémentaires.

Le tableau suivant répertorie les tâches SMIT de base :

Tâches de base SMIT

<i>Tâche</i>	<i>Raccourci SMIT</i>	<i>Sélection (ASCII)</i>	<i>Sélection (AIXwindows)</i>
Entrer dans SMIT	smit		
Quitter SMIT.		F12	F12 ou option Exit SMIT dans le menu Exit
Commande Show		F6	F6 ou option Command dans le menu Show
Raccourci Show		F8	F8 ou option FastPath dans le menu Show

Chapitre 14. Unités

Les unités incluent les composants matériels, tels que les imprimantes, les périphériques, les cartes, les bus et les emplacements d'extension, ainsi que les pseudo-unités telles que le fichier spécial d'erreur et le fichier spécial null. Les pilotes d'unités se trouvent dans le répertoire `/usr/lib/drivers`.

Ce chapitre présente les méthodes utilisées par le système d'exploitation pour gérer ces unités :

- Configuration d'un grand nombre d'unités
- Nœuds d'unités
- Codes d'emplacement des unités
- Gestion de la connexion à chaud PCI
- E/S à plusieurs chemins

Configuration d'un grand nombre d'unités

Le nombre d'unités que peut prendre en charge AIX varie en fonction du système et dépend de plusieurs facteurs. Les facteurs suivants ont un impact sur les systèmes de fichiers qui prennent en charge les unités :

- La configuration d'un grand nombre d'unités nécessite de stocker plus d'informations dans la base de données de configuration des unités ODM. Elle peut également nécessiter plusieurs fichiers spéciaux d'unités. En conséquence, un espace plus important et plus d'i-nodes sont nécessaires pour le système de fichiers.
- Certaines unités nécessitent plus d'espace que d'autres dans la base de configuration des unités ODM. Le nombre de fichiers spéciaux ou d'i-nodes varie également en fonction des unités. En conséquence, la quantité d'espace et le nombre d'i-nodes du système de fichiers dépendent du type des unités du système.
- Les unités MPIO (Multipath I/O) nécessitent plus d'espace que les unités non-MPIO, car les informations sont stockées dans l'ODM de l'unité elle-même, et pour chaque chemin d'accès à l'unité. En règle générale, l'espace de chaque chemin correspond à 1/5 de l'espace de l'unité. Par exemple, une unité MPIO avec cinq chemins utilise un espace équivalent à celui de deux unités non-MPIO.
- AIX inclut des unités logiques et des unités physiques dans la base de données de configuration des unités ODM. Les unités logiques incluent les groupes de volumes, les volumes logiques, les interfaces réseau, etc. Dans certains cas, la relation entre les unités logiques et les unités physiques peut affecter considérablement le nombre total d'unités prises en charge. Si, par exemple, vous définissez un groupe de volumes avec deux volumes logiques pour chaque disque physique connecté au système, cela correspond à quatre unités AIX pour chaque disque. En revanche, si vous définissez un groupe de volumes avec six volumes logiques pour chaque disque physique, il existera huit unités AIX pour chaque disque. En conséquence, moitié moins de disques peut être connectée.
- La modification des attributs par défaut augmente la taille de la base de données de configuration des unités ODM et peut empêcher de prendre en charge un plus grand nombre d'unités.
- La mémoire réelle nécessaire est proportionnelle au nombre d'unités.

AIX utilise deux fichiers système pour prendre en charge les unités :

- Le système de fichiers RAM est utilisé au cours de l'amorçage dans un environnement sans espace de pagination et sans système de fichiers disque monté. La taille du système de fichiers RAM correspond à 25 % de la taille de la mémoire système, jusqu'à 128 Mo. Un i-node est alloué à chaque Ko dans le système de fichiers RAM. La mémoire système minimum pour AIX 5.2 est de 128 Mo, ce qui se traduit par un système de fichiers d'une taille minimale de 32 Mo avec 32 768 i-nodes. Si la mémoire système est de 512 Mo ou plus, la taille maximale du système de fichiers RAM est de 128 Mo avec 131 072 i-nodes. Si la taille de l'espace du système de fichiers RAM ou le nombre d'i-nodes nécessaire pour prendre en charge les unités connectées est supérieure à l'allocation associée au disque RAM, le système ne démarre pas. Dans ce cas, vous devez supprimer des unités.
- La taille et le nombre d'i-nodes du système de fichiers root (rootvg) sur le disque peuvent être augmentés si le groupe de volumes rootvg contient des partitions non allouées. Avec AIX 5.2 et la taille minimale de système de fichiers RAM, il est possible de configurer jusqu'à 5 000 unités AIX. Avec la taille maximale de système de fichiers RAM, il est possible de configurer jusqu'à 25 000 unités AIX. Ces chiffres incluent les unités physiques et les unités logiques. Selon les divers facteurs mentionnés dans cette section, votre système peut configurer plus ou moins d'unités.

Remarque : Un grand nombre d'unités allonge le délai de configuration et donc le délai d'amorçage.

Nœuds d'unité

Les unités sont structurées en grappes appelées *nœuds*. Chaque nœud représente un sous-système logique d'unités, les unités de niveau inférieur étant dépendantes de celles de niveau supérieur dans la hiérarchie (enfant-parent). Par exemple, le nœud système est en haut de la hiérarchie et comprend l'ensemble des unités physiques du système. Le nœud système est le nœud le plus élevé dans la hiérarchie, suivi du bus et des cartes qui en dépendent. En bas de la hiérarchie sont situées les unités auxquelles ne sont pas connectées d'autres unités. Ces unités sont dépendantes de toutes les autres dans la hiérarchie.

Lors de l'amorçage, les dépendances parent-enfant servent à configurer toutes les unités formant un nœud. La configuration commence par le nœud supérieur et continue vers le bas. Les unités qui dépendent d'une unité d'un niveau supérieur ne peuvent être configurées qu'après celle-ci.

MPIO (Multiple-path I/O) est une fonction disponible dans AIX 5.2 et les versions supérieures. Si une unité dispose d'un pilote compatible MPIO, elle peut avoir plusieurs parents dans cette hiérarchie, ce qui permet de disposer de plusieurs chemins de communication simultanée entre l'unité et une machine donnée ou une partition logique dans une machine. Pour plus d'informations sur le fonctionnement de cette fonction, reportez-vous à la section Base de données de configuration des unités.

Classes d'unité

La gestion d'unité suppose la compréhension par le système d'exploitation des connexions d'unité autorisées. Le système d'exploitation classe les unités en trois groupes :

- classes fonctionnelles,
- sous-classes fonctionnelles,
- types d'unité.

Une classe fonctionnelle représente des unités exécutant la même fonction. Par exemple, les imprimantes constituent une classe fonctionnelle. Les classes fonctionnelles sont réparties en sous-classes, tenant compte de certaines similitudes des unités. Par exemple, les imprimantes peuvent être divisées en deux sous-classes : les imprimantes série et les imprimantes parallèles. Les types d'unités sont classés en fonction des modèles et du fabricant.

Les classes d'unité définissent des connexions parent-enfant pour le système d'exploitation. La hiérarchie définit les sous-classes pouvant être connectées pour chaque emplacement potentiel de connexion enfant. Par exemple, le terme carte 8 ports RS-232 indique que seules les unités de la sous-classe RS-232 peuvent être connectées à un des ports de la carte.

Les classes d'unité et leurs dépendances hiérarchiques sont actualisées dans une base de configuration d'unités ODM (Object Data Manager).

Base de configuration d'unités

Les données relatives aux unités figurent dans une base prédéfinie ou une base personnalisée constituant la base de configuration d'unités. La base prédéfinie regroupe les données de configuration de toutes les unités possibles prises en charge par le système. Les données sur la hiérarchie de la classe d'unité font partie de cette base de données. La base personnalisée contient des données de configuration concernant chaque unité définie et configurée du système. Un enregistrement de chaque unité connectée au système est conservé.

Le gestionnaire de configuration est un programme qui configure automatiquement les unités pendant l'amorçage du système et le temps d'exécution. Pendant ce processus, il fait appel aux données de la base prédéfinie et de la base personnalisée et actualise ensuite cette dernière.

Depuis la version AIX 5.2, la fonction MPIO (Multiple-path I/O) utilise un identificateur d'unité unique (UDID) pour identifier chaque unité MPIO, quel que soit le chemin dans lequel elle est découverte. L'identificateur UDID est enregistré dans la base de données de configuration des unités. Lorsqu'une unité est découverte, les UDID de la base de données sont vérifiés pour déterminer s'il s'agit d'une nouvelle unité ou d'un autre chemin d'accès à une unité existante. Lorsque plusieurs chemins d'accès à une unité sont détectés, le pilote d'unité ou l'extension de noyau Path Control Manager détermine le chemin à utiliser pour une demande donnée. Pour plus d'informations sur la gestion des unités MPIO, reportez-vous à la section *Managing MPIO-Capable Devices* du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

Etats des unités

Quatre états peuvent caractériser les unités connectées au système :

Undefined	Le système ne connaît pas l'unité (état non défini).
Defined	Les données spécifiques de l'unité sont enregistrées dans la base personnalisée sans toutefois être disponibles (état défini).
Available	Une unité définie est associée au système d'exploitation ou l'unité définie est configurée (état disponible).
Stopped	L'unité n'est pas disponible mais est toutefois connue de son gestionnaire (état arrêté).

Quand une unité tty et une imprimante utilisent alternativement le même connecteur tty, dans la base de configuration d'unités, l'unité tty et l'imprimante sont définies sur le même parent et le même port. Une seule unité peut être configurée à la fois. Pendant la configuration du connecteur tty, les données d'installation de l'imprimante sont retenues en attendant d'être à nouveau configurées. L'unité n'est pas supprimée mais est à l'état défini. Maintenir une unité à l'état défini retient les données de la base personnalisée pour une unité qui n'est pas couramment utilisée, avant sa première mise à disposition ou pendant sa suppression temporaire du système.

C'est le gestionnaire d'unité, s'il existe, qui rend l'unité disponible.

Certaines unités, en particulier les pseudo-unités TCP/IP, ont recours à l'état arrêté.

Gestion des unités

Vous pouvez utiliser l'application Web-based System Manager Devices, SMIT, ou les commandes du système d'exploitation pour exécuter les tâches de gestion des unités, telles que supprimer, ajouter ou configurer des unités.

Codes d'emplacement des unités

Le *code d'emplacement* représente le chemin d'accès de l'unité centrale ou du tiroir CPU à la carte, aux câbles de signaux et, le cas échéant, au multiplexeur asynchrone de l'unité ou de la station. Ce code représente un autre moyen d'identifier les unités physiques.

Le code d'emplacement est formé de une à quatre zones de données de deux caractères, selon le type d'unité. Ces zones représentent le tiroir, l'emplacement de carte, le connecteur et le port. Chacune de ces zones comporte deux caractères.

Le code d'emplacement du tiroir est un code de deux caractères situé dans la zone correspondante. Celui de la carte figure dans la zone du tiroir et celle de l'emplacement de carte au format AA-BB, AA correspondant au code d'emplacement du tiroir et BB au code du bus et de l'emplacement de la carte. Les autres unités ont des codes d'emplacement au format AA-BB-CC ou AA-BB-CC-DD, où AA-BB est le code d'emplacement de la carte à laquelle l'unité est connectée, CC le code du connecteur de la carte et DD un numéro de port ou une adresse d'unité SCSI.

Pour trouver les étiquettes mentionnant les codes d'emplacement sur le matériel, reportez-vous au guide de l'opérateur.

Carte

Le code d'emplacement d'une carte est un code au format AA-BB, où AA représente le code d'emplacement du tiroir logeant la carte et BB le bus d'E/S et l'emplacement de la carte.

La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système. Dans cette zone, toute autre valeur indique que la carte loge dans une unité d'extension d'E/S. Dans ce cas, la valeur de AA identifie le bus d'E/S et le numéro d'emplacement du tiroir CPU où se trouve la carte d'extension asynchrone. Pour le premier chiffre, la valeur 0 correspond au bus d'E/S standard et la valeur 1 au bus d'E/S en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S indiqué.

Le premier chiffre de la zone BB identifie la carte d'E/S contenant la carte d'adaptation. Si elle loge dans le tiroir CPU ou l'unité centrale, la valeur de ce chiffre est 0 pour le bus d'E/S standard et 1 pour celui en option. Si la carte est située dans un tiroir d'extension d'E/S, ce chiffre est 0. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S indiqué (ou dans le tiroir d'extension d'E/S).

Le code d'emplacement 00-00 identifie la carte d'E/S standard.

Exemples :

00-05	carte à l'emplacement 5 de la carte d'E/S standard, située dans le tiroir CPU ou l'unité centrale, selon le type du système.
00-12	carte à l'emplacement 2 du bus d'E/S en option, située dans le tiroir CPU.
18-05	carte à l'emplacement 5 d'un tiroir d'extension d'E/S. Ce tiroir est connecté à la carte d'extension asynchrone située à l'emplacement 8 du bus d'E/S en option dans le tiroir CPU.

Imprimante/traceur

Les codes d'emplacement 00-00-S1-00 ou 00-00-S2-00 représentent l'imprimante/traceur connecté au port série s1 ou s2 de la carte d'E/S standard. Le code 00-00-0P-00 représente l'imprimante parallèle connectée au port parallèle de la carte d'E/S standard.

Tout autre code d'emplacement indique que l'imprimante ou le traceur n'est pas connecté à la carte d'E/S standard mais à une autre carte. Le code a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte pilotant l'unité.

AA	La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système. Toute autre valeur indique que la carte loge dans un tiroir d'extension d'E/S ; dans ce cas, le premier chiffre identifie le bus d'E/S et le second le numéro d'emplacement sur le bus dans le tiroir CPU contenant la carte d'extension asynchrone à laquelle le tiroir d'extension d'E/S est connecté.
BB	Le premier chiffre de cette zone identifie le bus d'E/S contenant la carte. Si la carte loge dans le tiroir CPU ou l'unité centrale, la valeur 0 représente le bus d'E/S standard et 1 celui en option. Si la carte se trouve dans un tiroir d'extension d'E/S, ce chiffre est 0. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S (ou dans le tiroir d'extension d'E/S) contenant la carte.
CC	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur asynchrone est connecté. Les valeurs possibles de cette zone sont 01, 02, 03 et 04.
DD	Cette zone identifie le numéro de port sur le multiplexeur asynchrone auquel l'imprimante ou le traceur est connecté.

Unité tty

Les codes d'emplacement 00-00-S1-00 ou 00-00-S2-00 indiquent que l'unité tty est connectée aux ports série d'E/S standard s1 ou s2.

Tout autre code indique que l'unité tty n'est pas connecté à la carte d'E/S standard mais à une autre carte. Le code a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte pilotant l'unité.

AA	La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système. Toute autre valeur indique que la carte loge dans une unité d'extension d'E/S. Dans ce cas, le premier chiffre identifie le bus d'E/S et le second le numéro d'emplacement sur le bus dans le tiroir CPU contenant la carte d'extension asynchrone à laquelle le tiroir d'extension d'E/S est connecté.
BB	Le premier chiffre de la zone BB identifie le bus d'E/S contenant la carte d'adaptation. Si elle loge dans le tiroir CPU ou l'unité centrale, la valeur de ce chiffre est 0 pour le bus d'E/S standard et 1 pour celui en option. Si la carte est située dans un tiroir d'extension d'E/S, ce chiffre est 0. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S indiqué (ou dans le tiroir d'extension d'E/S).
CC	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur asynchrone est connecté. Les valeurs possibles de cette zone sont 01, 02, 03 et 04.
DD	Cette zone identifie le numéro de port sur le multiplexeur asynchrone auquel l'unité tty est connectée.

Unité SCSI

Toute unité SCSI, y compris :

- CD-ROM,
- disques,
- unités Initiator,
- unités optiques de lecture-écriture,
- bandes,
- mode Target.

Le format du code d'emplacement est AA-BB-CC-S, L. AA-BB identifie le code d'emplacement de la carte SCSI pilotant l'unité SCSI.

AA	La valeur 00 dans la zone AA signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre représente le bus d'E/S standard. La valeur 00 représente le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte. La valeur 00 dans cette zone indique le contrôleur SCSI standard.
CC	Cette zone identifie le bus SCSI de la carte à laquelle l'unité est connectée. Pour une carte ne fournissant qu'un seul bus SCSI, la valeur de cette zone est 00. Sinon, la valeur 00 indique une unité reliée au bus SCSI interne de la carte et la valeur 01 une unité reliée au bus SCSI externe de la carte.
S, L	Cette zone identifie l'ID SCSI et le numéro d'unité logique (LUN) de l'unité SCSI. La valeur S identifie l'ID SCSI et L le numéro d'unité logique (LUN).

Codes d'emplacement des disques connectés directement au bus Pour une unité de disque connectée directement,

le format du code d'emplacement est AA-BB. Le champ AA contient la valeur 00 qui indique que le disque se trouve dans l'unité système. Le champ BB indique le bus E/S et le numéro d'emplacement d'extension contenant le disque. Le premier chiffre est toujours 0, qui indique que le disque est connecté au bus E/S standard. Le second chiffre indique le numéro de l'emplacement d'extension du bus E/S standard auquel le disque est connecté.

Disque série

Le code d'emplacement des unités de disque en série a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte pilotant l'unité.

Les différentes zones s'interprètent comme suit :

AA	La valeur 00 dans la zone AA signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, 0 pour le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.

CC	Cette zone identifie le connecteur de la carte à laquelle le tiroir pilotant l'unité est connecté. Les valeurs possibles de cette zone sont 00, 01, 02 et 03.
DD	Cette zone identifie le numéro d'unité logique (LUN) du disque. Il correspond à l'emplacement du tiroir logeant le disque.

Unité de disquette

Les codes d'emplacement des unités de disquette sont 00-00-0D-01 ou 00-00-0D-02, indiquant qu'elles sont reliées au port 0 ou 1 de la carte principale d'E/S standard.

Rotateur/clavier LPFK

Pour une carte d'entrée graphique reliée à une unité rotateur/clavier LPFK, le code d'emplacement a le format AA-BB-CC.

Les différentes zones s'interprètent comme suit :

AA	La valeur 00 dans la zone AA signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, 0 pour le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.
CC	Cette zone indique le connecteur de carte auquel l'unité est connectée. La valeur est 01 ou 02, selon que l'unité est reliée au port 1 ou 2 sur la carte.

Remarque : En série, ces unités n'indiquent pas de codes d'emplacement. Elles sont supposées reliées à une unité tty. Cette dernière est spécifiée par l'utilisateur lors de la définition des rotateurs/claviers LPFK.

Port multiprotocole

Le code d'emplacement d'un port multiprotocole a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte multiprotocole.

Les différentes zones s'interprètent comme suit :

AA	La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, 0 pour le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.
CC	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur multiprotocole est connecté. La valeur est toujours 01.
DD	Cette zone identifie le numéro de port physique sur le multiplexeur multiprotocole. Les valeurs possibles de cette zone sont 00, 01, 02 et 03.

Gestion des connexions à chaud PCI

Vous pouvez connecter à chaud une nouvelle carte PCI dans un emplacement d'extension PCI à connexion à chaud alors que le système d'exploitation est actif. Il peut s'agir d'une carte de même type que celle qui est installée ou d'une carte d'un type différent. Les nouvelles ressources sont disponibles pour le systèmes d'exploitation et les applications sans avoir à redémarrer le système d'exploitation. Vous pouvez être amené à ajouter une carte connectable à chaud dans les cas suivants :

- Ajout de fonction au matériel et au microcode existant ou extension de leur capacité.
- Migration de cartes PCI d'un système qui ne nécessite plus la fonction fournie par les cartes.
- Installation d'un nouveau système pour lequel les cartes deviennent disponibles après la configuration initiale des sous-systèmes matériels, notamment des cartes PCI, et l'installation et le démarrage du système d'exploitation.

Remarque : Si vous ajoutez une carte en exécutant une opération de remplacement ou d'ajout de carte PCI à chaud, la carte et ses unités enfants ne sont pas disponibles comme spécifications d'unité d'amorçage à l'aide de la commande **bootlist**. Vous devez redémarrer la machine pour que le système d'exploitation reconnaisse toutes les unités d'amorçage potentielles.

Vous pouvez également supprimer une carte PCI connectable à chaud défailante ou la remplacer par une autre de même type sans avoir à arrêter le système ou le mettre hors tension. Lorsque vous remplacez la carte, le pilote d'unité existant prend en charge la carte, car il s'agit d'une carte de même type. Les informations de configuration des unités et de configuration des unités sous la carte sont conservées pour la carte de remplacement. Vous pouvez être amené à remplacer une carte dans les cas suivants :

- Remplacement temporaire d'une carte pour déterminer un problème ou isoler une FRU défailante.
- Remplacement d'une carte défailante en continu ou par intermittence.
- Remplacement d'une carte redondante défailante dans une configuration HACMP ou I/S à plusieurs chemins.

Lorsque vous retirez une carte connectable à chaud, les ressources fournies par la carte ne sont plus disponibles pour le système d'exploitation et les applications. Vous pouvez être amené à retirer une carte dans les cas suivants :

- Retrait des sous-systèmes E/S existants.
- Retrait d'une carte qui n'est plus nécessaire ou défailante et carte de remplacement non disponible.
- Migration d'une carte vers un autre système lorsque la fonction de la carte n'est plus nécessaire sur le système contenant la carte.

Pour pouvoir retirer ou remplacer une carte, vous devez la déconfigurer. Le pilote d'unité associé doit libérer les ressources système qu'il a allouées à l'unité. Cela inclut la libération de la mémoire, l'annulation de la définition de l'interruption et des descripteurs EPOW, la libération de la zone DMA et des ressources de synchronisation, et toutes les autres tâches nécessaires. Les interruptions, la mémoire de bus et les E/S de bus doivent être également désactivées dans le pilote.

L'administrateur système doit exécuter les tâches suivantes avant et après le retrait d'une carte :

- Terminer et restaurer les applications, daemons ou processus qui utilisent l'unité.
- Démonter et remonter les systèmes de fichiers.

- Retirer et recréer les définitions d'unités et exécuter d'autres opérations nécessaires pour libérer une unité utilisée.
- Placer le système dans un état qui permette d'effectuer les opérations de maintenance.
- Obtenir et installer les pilotes d'unités nécessaires.

Les opérations de retrait et de remplacement échouent si l'unité connectée à l'emplacement d'extension identifié n'a pas été déconfiguré et ne se trouve pas dans l'état défini. Vous pouvez effectuer cette opération avec la commande `rmdev`. Avant de placer la carte dans l'état défini, fermez toutes les application qui utilisent la carte afin que la commande aboutisse.

Dans certains cas, vous pouvez effectuer les tâches suivantes :

- Préparez une carte PCI connectable à chaud à insérer, retirer ou remplacer.
- Identifiez les emplacements d'extension PCI impliqués dans l'opération de connexion à chaud.
- Retirez ou insérez les cartes PCI connectables à chaud. **Attention** : Bien que la gestion PCI de connexion à chaud permette d'ajouter, de retirer et de remplacer des cartes PCI sans mettre le système hors tension ou redémarrer le système d'exploitation, les cartes connectables à chaud ne peuvent pas être toutes gérées de cette manière. Par exemple, le disque dur qui constitue le groupe de volumes ou le contrôleur E/S auquel il est connecté ne peut pas être supprimé ou remplacé sans mettre le système hors tension car il est nécessaire pour exécuter le système d'exploitation. Avant de retirer ou d'insérer une carte PCI connectable à chaud, reportez-vous à l'annexe *PCI Adapter Placement Reference* de la documentation de votre unité système pour déterminer si la carte peut être retirée à chaud. Reportez-vous à la documentation de l'unité système pour plus d'informations sur l'installation ou le retrait des cartes.

Pour plus d'informations sur la gestion d'une carte PCI connectable à chaud, reportez-vous aux sections suivantes du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

- Adding a PCI Hot Plug Adapter
- Removing or Replacing a PCI Hot Plug Adapter

Unconfiguring PCI Communications Adapters

Cette section contient des informations générales sur la déconfiguration des cartes de communication PCI, à savoir les cartes Ethernet, Token-ring, FDDI et ATM. Pour plus d'informations sur la procédure, reportez-vous à la section Unconfiguring Communications Adapters du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*

Si votre application utilise le protocole TCP/IP, vous devez retirer l'interface TCP/IP de la carte de la liste des interfaces réseau avant de placer la carte dans l'état défini. Utilisez la commande `netstat` pour déterminer si la carte est configurée en fonction du protocole TCP/IP et identifier les interfaces réseau actives de la carte.

Une carte Ethernet peut avoir deux interfaces : Standard Ethernet (en *X*) ou IEEE 802.3 (et *X*). *X* correspond au même numéro dans le nom de carte **ent X**. Une seule des ces interfaces peut utiliser TCP/IP à la fois. Par exemple, la carte Ethernet **ent0** peut avoir les interfaces `en0` et `et0`.

Une carte Token ring peut avoir une seule interface : Token-ring (tr *X*). *X* correspond au même numéro dans le nom de carte **tok X**. Par exemple, la carte Token-ring **tok0** a une interface `tr0`.

Une carte ATM peut avoir une seule interface `atm` : ATM (at *X*). *X* correspond au même numéro dans le nom de carte **atm X**. Par exemple, la carte ATM **atm0** a une interface `at0`. Toutefois, les cartes ATM peuvent avoir plusieurs clients émuloés fonctionnant sur une seule carte.

La commande `ifconfig` supprime une interface du réseau. La commande `rmdev` déconfigure l'unité PCI en conservant sa définition dans la classe d'objet CDOC (Customized Devices Object Class). Lorsque la carte se trouve dans l'état défini, vous pouvez utiliser la commande `drslot` pour retirer la carte.

Exemple

1. Pour déconfigurer les enfants du bus PCI `pci1` et toutes les autres unités en dessous d'eux en conservant leurs définitions d'unité dans la classe d'objet CDOC, tapez :

```
rmdev -p pci1
Le système affiche un message similaire au message suivant :

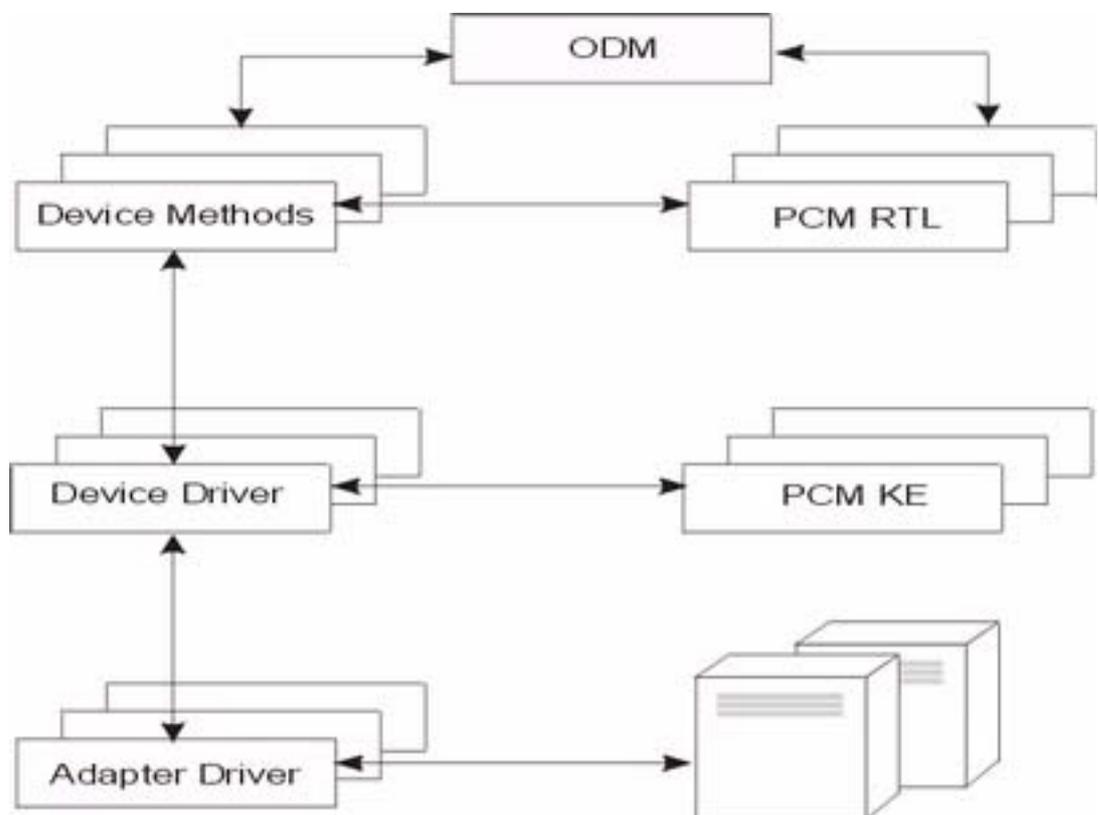
rmt0 Defined
hdisk1 Defined
scsi1 Defined
ent0 Defined
```

E/S à plusieurs chemins

Avec les E/S à plusieurs chemins (MPIO), une unité peut être détectée via une ou plusieurs connexions physiques ou *chemins*. Un module PCM (Path-Control Module) fournit les fonctions de gestion de chemins.

Un pilote d'unité compatible MPIO peut contrôler plusieurs types d'unités cibles. Un module PCM peut prendre en charge une ou plusieurs unités spécifiques. En conséquence, un pilote d'unité peut être interfacé avec plusieurs modules PCM qui contrôlent les E/S sur les chemins d'accès aux unités cibles.

Figure 12. Interaction des composants MPIO Cette illustration montre l'interaction entre les différents composants d'une solution MPIO. Dans cette figure, le pilote d'unité MPIO contrôle plusieurs types d'unités cibles nécessitant chacun un module PCM différent. (KE=Kernel Extension, RTL=Run-time Loadable).



Pour qu'une unité puisse tirer parti de MPIO, le pilote, les méthodes et les attributs prédéfinis de l'unité dans ODM (Object Data Manager) doivent être modifiés pour prendre en charge la détection, la configuration et la gestion de plusieurs chemins. Depuis la version AIX 5.2 avec 5200-01, les pilotes SCSI parallèle et canal fibre optique et leurs méthodes d'unité ont été modifiés pour prendre en charge les unités MPIO. En outre, les attributs prédéfinis de certaines unités dans ODM ont été modifiés pour MPIO.

Le module AIX PCM est constitué : du module de configuration PCM RTL et de l'extension de noyau PCM KE. Le module PCM RTL est un module chargeable à l'exécution qui permet aux méthodes d'unité de détecter d'autres attributs d'unité PCM KE ou de chemin ODM nécessaires à l'extension de noyau PCM KE. Le module PCM RTL est chargé par une méthode d'unité. Une ou plusieurs routines dans le module PCM RTL sont utilisées pour exécuter des opérations spécifiques qui initialisent ou modifient des variables PM KE.

Le module PCM KE fournit les fonction de gestion de chemins aux pilotes qui prennent en charge l'interface MPIO. Le module PCM KE dépend de la configuration de l'unité pour détecter les chemins et communiquer les informations correspondantes au pilote de l'unité. Chaque pilote d'unité compatible MPIO ajoute les chemins à une unité depuis son ou ses parents immédiats. La maintenance et la planification des E/S sur différents chemins sont fournies par le module PCM KE et elles ne sont pas apparentes pour le pilote d'unité MPIO.

Le module PCM KE peut fournir plusieurs algorithmes de routage qui peuvent être sélectionnés par l'utilisateur. Le module PCM KE peut également permettre de collecter des informations qui peuvent être utilisées pour déterminer et sélectionner le meilleur chemin d'une demande E/S. Le module PCM KE peut sélectionner le meilleur chemin en fonction de divers critères, notamment de l'équilibre de la charge, du débit de la connexion, de l'échec de connexion, etc.

Le module AIX PCM dispose d'une fonction de vérification d'intégrité qui peut être utilisée pour :

- Vérifier les chemins et déterminer ceux qui peuvent être utilisés pour envoyer des E/S.
- Activer un chemin marqué comme défaillant suite à un dysfonctionnement temporaire (par exemple, lorsqu'un câble connecté à une unité a été retiré, puis déconnecté).
- Identifier les chemins non utilisés qui pourraient être utilisés en cas de basculement (par exemple, lorsque la valeur d'attribut d'algorithme est `failover`, la vérification d'intégrité peut tester des chemins alternatifs).

Tous les disques ne peuvent pas être détectés et configurés avec le module AIX PCM. Si l'unité n'est pas détectée, contactez le fournisseur pour savoir si un module PCM est disponible pour l'unité.

Configuration d'une unité MPIO

La configuration d'une unité MPIO fait appel aux mêmes commandes que celles d'une unité non-MPIO. Depuis la version AIX 5.2 avec 5200-01, les commandes **cfgmgr**, **mkdev**, **chdev**, **rmdev** et **lsdev** prennent en charge la gestion des instances d'unités MPIO et l'affichage de leurs attributs. Une instance d'unité MPIO dispose également d'instances de chemins associés à l'instance d'unité. Les commandes **mkpath**, **chpath**, **rmpath** et **lspath** gèrent les instances de chemins et affichent leurs attributs.

Une instance de chemin peut être ajoutée ou supprimée d'une unité MPIO sans avoir à déconfigurer l'unité. Pour plus d'informations sur ces commandes, reportez-vous à la section Managing MPIO-Capable Devices du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Une unité MPIO peut avoir des attributs supplémentaires, mais les attributs obligatoires que toutes les unités MPIO doivent prendre en charge sont **reserve_policy** et **algorithm**. L'attribut **reserve_policy** détermine le type de méthodologie de réserve que met en œuvre le pilote d'unité lorsque l'unité est ouverte et il peut être utilisé pour limiter l'accès à l'unité depuis d'autres cartes sur le même système ou un autre système. L'attribut **algorithm** définit la technologie qu'utilise le module PCM pour gérer les E/S sur les chemins configurés de l'unité. Pour plus d'informations sur l'attribut **reserve_policy**, reportez-vous à la section Attributs d'unité MPIO. Pour plus d'informations sur l'attribut **algorithm**, reportez-vous à la section Attributs de module PCM (Path Control Module).

Unités à plusieurs chemins prises en charge

Le module AIX PCM prend en charge un groupe d'unités définies dans le fileset **devices.common.ibm.mpio.rte**. Pour les autres types d'unités, le fournisseur de l'unité doit fournir les attributs prédéfinis dans ODM, un module PCM et le code nécessaire pour déterminer si l'unité est compatible MPIO.

Pour déterminer les unités prises en charge par le module AIX PCM, exécutez le script suivant :

```

line          odmget -qDvDr=aixdiskpcmke PdDv | grep uniquetype | while read
              do
                  utype='echo $line | cut -d'"'"' -f2`
                  dvc='odmget -q"uniquetype=$utype AND
attribute=dvc_support" PdAt`
                  echo $dvc | grep values | cut -d'"'"' -f2
              done

```

Le script affiche la liste des types d'unités uniques pris en charge par le module AIX PCM. Les deux types d'unités pris en charge par AIX 5.2 PCM sont (**disk/scsi/scsd**) et *MPIO autre FC* (**disk/fcp/mpioosdisk**).

Un disque *MPIO other FC* est un sous-ensemble d'autres disques à canal fibre optique. Une unité est prise en charge comme disque *MPIO other FC* uniquement s'il n'existe pas d'attributs prédéfinis ODM fournis par le vendeur et que l'unité a été certifiée fonctionner avec le module AIX PCM. La certification ne garantit pas que toutes les fonctions de l'unité sont prises en charge lorsque l'unité est configurée comme disque *MPIO other FC*.

Pour déterminer les unités prises en charge comme disques *MPIO other FC*, exécutez les script suivant :

```
odmget -qattribute=mpio_model_map PdAt | grep deflt | cut -d'"'"' -f2
```

Le script affiche la liste des données de requête qui contiennent le nom du fournisseur et le modèle de l'unité.

Pour afficher toutes les unités MPIO installées sur le système, exécutez le script suivant :

```
odmget -qattribute=unique_id PdAt | grep uniquetype | cut -d'"'"' -f2
```

Le script affiche la liste des types d'unités MPIO uniques des fournisseurs et AIX PCM.

Attributs d'unité MPIO

Cette section décrit les attributs pris en charge uniquement par les unités à plusieurs chemins. Vous pouvez afficher ou modifier les attributs avec Web-based System Manager, SMIT ou des commandes (notamment **lsattr** et **chdev**).

L'attribut obligatoire que toutes les unités MPIO doivent prendre en charge est **reserve_policy**. Généralement, une unité à plusieurs chemins dispose également de l'attribut **PR_key**. Une unité à plusieurs chemins peut avoir d'autres attributs. Les autres attributs d'unité sont les suivants :

reserve_policy Indique si une méthodologie de réservation est utilisée lorsque l'unité est ouverte. Les valeurs sont les suivantes :

no_reserve N'applique pas une méthodologie de réservation pour l'unité. L'unité est accessible à d'autres initiateurs qui peuvent se trouver sur d'autres systèmes hôtes.

single_path_reserve Applique une méthodologie de réservation SCSI2 pour l'unité, ce qui implique que l'unité est accessible uniquement à l'initiateur qui a émis la réserve. Cette stratégie empêche d'autres initiateurs du même hôte ou d'autres hôtes d'accéder à l'unité. La stratégie utilise la réservation SCSI2 pour réserver l'unité à un seul initiateur (chemin) et les commandes routées via les autres chemins créent un conflit de réservation.

Des algorithmes de sélection qui alternent les commandes dans plusieurs chemins peuvent provoquer une augmentation des échanges (thrashing) lorsque la valeur **single_path_reserve** est sélectionnée. Supposons qu'un module PCM d'unité dispose d'un attribut obligatoire affecté d'une valeur qui distribue les E/S dans plusieurs chemins. Lorsque **single_path_reserve** est en vigueur, le pilote d'unité doit émettre une réinitialisation BDR (Bus Device Reset), puis une réservation en utilisant un nouveau chemin pour envoyer la commande suivante pour arrêter la réservation précédente. Chaque fois qu'un chemin différent est sélectionné, une augmentation des échanges se produit et les performances se dégradent du fait de l'envoi d'une réinitialisation BDR et d'une réservation vers l'unité cible. Le module AIX PCM ne permet pas de sélectionner un algorithme qui pourrait provoquer une augmentation des échanges.

PR_exclusive Applique une réservation persistante SCSI3, une méthodologie d'hôte exclusive lorsque l'unité est ouverte. La valeur de l'attribut **PR_key** doit être unique pour chaque système hôte. L'attribut **PR_key** permet d'empêcher les initiateurs d'autres systèmes hôtes d'accéder à l'unité.

PR_shared Applique une réservation persistante SCSI3, une méthodologie d'hôte partagée lorsque l'unité est ouverte. La valeur de l'attribut **PR_key** doit être unique pour chaque système hôte. Les initiateurs des autres systèmes hôtes doivent s'enregistrer pour pouvoir accéder à l'unité.

PR_key Nécessaire uniquement si l'unité prend en charge les stratégies de réservation persistante (**PR_exclusive** ou **PR_shared**).

Attributs de module PCM (Path Control Module)

Outre le module par défaut AIX PCM, un module PCM d'unité spécifique peut être fourni par le fournisseur de l'unité. Les attributs qui peuvent être modifiés par l'utilisateur sont définis par le fournisseur de l'unité. Un module PCM d'unité spécifique peut avoir des attributs d'unité et de chemin.

Les attributs d'unité pour le module AIX PCM sont les suivants :

algorithm Détermine la méthodologie qui distribue les E/S dans les chemins d'une unité. Cet attribut a les valeurs suivantes :

failover Envoie toutes les E/S dans un seul chemin. Si le chemin est défaillant, un autre chemin est sélectionné pour envoyer toutes les E/S. Cet algorithme suit tous les chemins activés dans une liste ordonnée. Si le chemin utilisé pour envoyer les E/S connaît un dysfonctionnement, le chemin suivant actif de la liste est sélectionné. La séquence dans la liste est déterminée par la priorité de chemin dans l'attribut de page 14-15.

- round_robin** Distribue les E/S dans tous les chemins actifs. La priorité de chemin est déterminée par la priorité de chemin de la valeur d'attribut de page 14-15. Si un chemin connaît une défaillance ou est désactivé, il ne peut plus envoyer les E/S. La priorité des autres chemins est recalculée pour déterminer le pourcentage d'E/S qui doit être envoyé dans chaque chemin. Si tous les chemins ont la même valeur, les E/S sont envoyées à part égale dans tous les chemins actifs.
- hcheck_mode** Détermine les chemins à vérifier lorsque la fonction de vérification d'intégrité est utilisée. L'attribut prend en charge les modes suivants :
- enabled** Envoie la commande **healthcheck** dans les chemins actifs.
 - failed** Envoie la commande **healthcheck** dans les chemins défaillants.
 - nonactive** (valeur par défaut) Envoie la commande **healthcheck** dans les chemins sans E/S actives, y compris les chemins défaillants. Si l'algorithme sélectionné est failover, la commande **healthcheck** est également envoyée dans les chemins actifs sans E/S actives. Si l'algorithme sélectionné est round_robin, la commande **healthcheck** est envoyée uniquement dans les chemins défaillants, car cet algorithme maintient actifs tous les chemins avec des E/S.
- hcheck_interval** Définit la fréquence d'exécution de la vérification de l'intégrité sur les chemins d'une unité. L'attribut peut avoir une valeur comprise entre 0 et 3 600 secondes. Lorsque la valeur est 0, la vérification de l'intégrité est désactivée.

L'attribut de chemin pour le module AIX PCM est le suivant :

- path priority** Modifie le comportement de la méthodologie de l'algorithme de la liste des chemins.

Lorsque la valeur de l'attribut **algorithm** est failover, les chemins sont placés dans une liste. La séquence de la liste détermine le chemin qui est sélectionné en premier et s'il est défaillant, le chemin suivant sélectionné. La séquence est définie par la valeur de l'attribut de priorité de chemin. La priorité 1 est la priorité la plus haute. Plusieurs chemins peuvent avoir la même priorité, mais si tous les chemins ont la même priorité, la sélection est basée sur le moment où chaque chemin a été configuré.

Lorsque la valeur de l'attribut **algorithm** est **round_robin**, la séquence est déterminée par le pourcentage d'E/S. La priorité de chemin détermine le pourcentage d'E/S qui doit être envoyé dans chaque chemin. Les E/S sont distribuées dans les chemins actifs. Un chemin est sélectionné jusqu'à ce qu'il atteigne son pourcentage. L'algorithme marque ensuite le chemin comme étant défaillant ou désactivé pour distribuer les demandes E/S en fonction de la priorité de chemin.

Chapitre 15. Unités de bande

Ce chapitre présente les fonctions de gestion des unités de bande. La plupart de ces fonctions exploitent les informations de la base de données de configuration des unités du système, cette base de données regroupe deux bases : la base de données de configuration prédéfinie qui contient les informations sur tous les types d'unité admis par le système et la base de données personnalisée qui contient les informations sur chaque unité installée sur le système. Pour que le système d'exploitation puisse utiliser une unité (de bande ou non), elle doit être déclarée dans cette base personnalisée et relever d'un type défini dans la base prédéfinie.

Les sujets abordés sont les suivants :

- Attributs des unités de bande
- Fichiers spéciaux pour unités de bande

Les tâches de base des unités de bande sont répertoriées dans la section Tape Drives du document *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Attributs des unités de bande

Cette section décrit les attributs modifiables des unités de bande. Vous pouvez les afficher ou les modifier à l'aide de l'application **Web-based System Manager Devices**, de **SMIT** ou de commandes (notamment **lsattr** et **chdev**).

Chaque type d'unité de bande n'utilise qu'une partie des attributs.

Présentation générale

Taille de bloc

L'attribut taille de bloc indique la taille de bloc à utiliser pour la lecture ou l'écriture d'une bande. Les données sont inscrites sous forme de blocs de données délimités par des espaces interblochs. Sur les bandes non formatées, il est préférable d'utiliser des blocs de grande taille pour réduire le nombre d'espaces interblochs et disposer ainsi de davantage d'espace pour l'inscription des données. La valeur **0** indique une taille de bloc variable. Les valeurs par défaut et les valeurs admises varient en fonction de l'unité de bande.

Mémoires tampon

Lorsque vous positionnez l'attribut mémoires tampon sur **yes** (avec **chdev**, l'attribut **mode**), les applications reçoivent un message de confirmation d'écriture dès le transfert des données en mémoire tampon sur l'unité de bande, même si l'écriture de bande n'est pas encore réalisée. Avec la valeur **no**, l'écriture n'est notifiée qu'une fois les données inscrites sur la bande. La valeur **no** n'est pas compatible avec la lecture et l'écriture sur bande en mode continu. La valeur par défaut est **yes**.

Lorsque cet attribut est positionné sur **no**, l'unité de bande est moins rapide mais elle garantit une meilleure intégrité des données en cas de coupure de courant ou de défaillance du système et facilite le traitement des fins de support.

Marques de fichier étendues

Lorsque cet attribut est positionné sur **no** (avec **chdev**, l'attribut **extfm**), une marque de fichier standard est inscrite sur la bande chaque fois que nécessaire. La valeur **yes** provoque l'inscription d'une marque de fichier étendue. Pour les unités de bande, cet attribut peut être activé. La valeur par défaut est **no**. Par exemple, les marques de fichiers étendus sur unités de bande 8 mm mobilisent 2,2 Mo et nécessitent pour leur inscription jusqu'à 8,5 secondes. Les marques de fichiers standard utilisent 184 Ko et environ 1,5 secondes.

Lorsque vous utilisez des bandes 8 mm en mode adjonction, il est préférable d'utiliser les marques de fichier étendus pour un meilleur positionnement après des opérations inverses sur marques de fichier. Ceci permet de réduire les risques d'erreur.

Tension

La valeur **yes** (avec **chdev**, l'attribut **ret**) qu'après chaque insertion ou réinitialisation d'une bande, la bande est automatiquement retendue. Cela signifie que la bande est déroulée jusqu'à la fin puis entièrement rebobinée. Cette opération, qui demande plusieurs minutes, diminue le risque d'erreurs. Avec la valeur **no**, l'unité de bande ne retend pas automatiquement la bande. La valeur par défaut est **yes**.

Densité

L'attribut Densité égale à #1 (avec **chdev**, l'attribut **density_set_1**) définit la densité appliquée par l'unité de bande pour l'utilisation de fichiers spéciaux **/dev/rmt***, **/dev/rmt*.1**, **/dev/rmt*.2** et **/dev/rmt*.3**. L'attribut Densité égale à #2 (avec **chdev**, l'attribut **density_set_2**) définit la densité appliquée par l'unité de bande pour l'utilisation de fichiers spéciaux **/dev/rmt*.4**, **/dev/rmt*.5**, **/dev/rmt*.6** et **/dev/rmt*.7**. Pour plus de détails, reportez-vous à "Fichiers spéciaux pour unités de bande".

Les attributs de densité sont représentés par des nombres décimaux compris entre **0** et **255**. La valeur **0** demande l'application de la densité par défaut pour l'unité de bande, généralement la densité maximale. Les valeurs admises et leur signification varient en fonction du type d'unité de bande. Ces attributs n'ont aucune répercussion sur la capacité de lecture de l'unité pour des bandes écrites dans des densités admises par l'unité. Habituellement, l'attribut densité égale à #1 est positionné à la valeur maximale possible pour l'unité de bande et l'attribut densité égale à #2, à la seconde valeur maximale possible pour l'unité de bande.

Réservation

Pour les unités de bande qui acceptent cet attribut (avec **chdev**, l'attribut **res_support**), la valeur `res_support=yes` réserve l'unité de bande sur le bus SCSI à son ouverture. Lorsque plusieurs cartes SCSI partagent l'unité de bande, l'activation de cet attribut permet de limiter l'accès à une seule carte lorsque l'unité est ouverte. Certaines unités SCSI ne prennent pas en charge cette fonction. D'autres ont une valeur prédéfinie pour cette fonction et la prennent toujours en charge.

Taille de bloc de longueur variable

Cet attribut (avec **chdev**, l'attribut **var_block_size**) spécifie la taille de bloc requise par l'unité de bande lors de l'écriture d'articles de longueur variable. Sur certaines unités de bande SCSI, une taille de bloc non nulle doit être spécifiée (dans les données Mode Select) lors de l'écriture d'articles de longueur variable. La taille de bloc est positionnée à **0** pour indiquer des blocs de longueur variable. Reportez-vous aux informations spécifiques de l'unité de bande SCSI pour déterminer le positionnement requis.

Compression de données

La valeur `yes` de cet attribut (avec **chdev**, l'attribut **compress**) passe l'unité de bande en mode compression, si l'unité offre cette fonction. Dans ce cas, elle inscrit les données sur la bande dans un format compressé pour stocker plus d'informations. La valeur `no` force l'unité de bande à écrire les données en mode natif (non compressé). Cet attribut est sans incidence sur les opérations de lecture. La valeur par défaut est `yes`.

Autochargement

La valeur `yes` de cet attribut (avec **chdev**, l'attribut **autoload**) active la fonction d'autochargement, si l'unité offre cette fonction. Dans ce cas, si la fin de la bande est atteinte lors d'une opération de lecture et d'écriture, la bande suivante est automatiquement chargée pour poursuivre l'opération. Cette fonction est sans incidence sur les commandes applicables uniquement à une seule bande en cartouche. La valeur par défaut est `yes`.

Délai entre deux tentatives

Cet attribut définit le délai d'attente en secondes au-delà duquel le système relance une commande qui n'a pas abouti. Le système peut effectuer quatre tentatives maximum. Cet attribut ne s'applique qu'aux unités de bande de type ost. La valeur par défaut est 45.

Délai de lecture/écriture

Cet attribut définit le délai maximal (en secondes) accordé au système pour exécuter avec succès une commande de lecture (READ) ou d'écriture (WRITE). Cet attribut ne s'applique qu'aux unités de bande de type ost. La valeur par défaut est 144.

Renvoyer erreur sur changement de bande

Lorsque l'attribut Renvoyer erreur sur changement de bande ou réinitialisation est sélectionné, une erreur est renvoyée à l'ouverture lorsque l'unité de bande a été réinitialisée ou que la bande a été changée. Une opération ayant laissé la bande au milieu de la bande à la fermeture doit avoir eu lieu. L'erreur renvoyée est un `-1` et `errno` a la valeur `EIO`. Une fois présentée à l'application, la situation d'erreur est annulée. De même, la reconfiguration de l'unité de bande annule la situation d'erreur.

Attributs pour unités de bande 4 mm 2 Go (type 4mm2gb)

Taille de bloc

La valeur par défaut est 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité. Les valeurs de densité sont prédéfinies car l'unité de bande écrit toujours en mode 2 Go.

Attributs pour unités de bande 4 mm 4 Go (type 4mm4gb)

Taille de bloc

La valeur par défaut est 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

L'utilisateur ne peut pas modifier la densité appliquée par cette unité. L'unité module automatiquement la densité utilisée en fonction du type de support DDS (Digital Data Storage) installé :

Type de support	Configuration de l'unité
DDS	Lecture seulement
DDS	Lecture/écriture en mode 2 Go uniquement
DDS2	Lecture dans l'une ou l'autre des densités, écriture en mode 4 Go seulement
non-DDS	Non pris en charge ; cartouche éjectée

Compression de données

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 8 mm 2,3 Go (type 8mm)

Taille de bloc

La valeur par défaut est 1024. Une valeur inférieure réduit le volume de données stockées sur bande.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité. Les valeurs de densité sont prédéfinies car l'unité de bande écrit toujours en mode 2,3 Go.

Attributs pour unités de bande 8 mm 5 Go (type 8mm5gb)

Taille de bloc

La valeur par défaut est 1024. Pour une bande inscrite en mode 2,3 Go, une valeur inférieure réduit la quantité de données stockées.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendus

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
140	Mode 5 Go (compression possible)
21	Mode 5 Go (compression impossible)
20	Mode 2,3 Go
0	Valeur par défaut (mode 5 Go)

Les valeurs par défaut sont 140 pour l'attribut densité égale à #1 et 20 pour l'attribut densité égale à #2. La valeur 21 associée à l'un de ces attributs autorise la lecture ou l'écriture en mode 5 Go non compressé.

Compression de données

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 8 mm 20000 Mo (autoconfiguration)

Taille de bloc

La valeur par défaut est 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendus

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

L'unité peut lire et écrire sur des cartouches de format 20 Go. Pendant la lecture, l'unité détermine automatiquement le format des données inscrites sur la bande. Pendant l'écriture, la valeur de la densité détermine le format des données inscrites sur la bande.

Valeurs possibles :

Valeur	Signification
39	Mode 20 Go (compression possible)
0	Valeur par défaut (mode 20 Go)

La valeur par défaut est 39 pour les attributs densité égale à #1 et densité égale à #2.

Compression de données

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 35 Go (type 35gb)

Taille de bloc

La capacité de traitement du 7205 Modèle 311 est affectée par la taille de bloc. Pour cette unité, la taille de bloc minimale recommandée est de 32 Ko. Toute valeur inférieure réduit le débit des données (temps de sauvegarde/restauration). Le tableau ci-après répertorie les tailles de bloc recommandées par les commandes :

Commande prise en charge	Taille de bloc par défaut (octets)	RECOMMANDATION
BACKUP	32 Ko ou 51,2 Ko (par défaut)	32 Ko ou 51,2 Ko, selon que la commande Backup est par nom ou pas. Aucune modification de l'utilisateur n'est requise.
TAR	10 Ko	Il y a erreur dans le manuel qui indique une taille de bloc de 512 Ko. Définissez le paramètre de taille de bloc à -N64 .
MKSYSB	Voir BACKUP	MKSYSB utilise la commande BACKCUP. Aucune modification de l'utilisateur n'est requise.
DD	n/a	Définissez le paramètre de taille de bloc à bs=32K .
CPIO	n/a	Définissez le paramètre de taille de bloc à -C64 .

Remarque : Vous devez connaître la puissance et la capacité de traitement lorsque vous sélectionnez une taille de bloc. Les tailles de bloc réduites affectent les performances, mais pas la puissance de traitement. Les puissances de traitement des formats 2,6 Go (densité) et 6 Go (densité) sont affectées si vous utilisez une taille de bloc inférieure à la taille recommandée. Par exemple: la sauvegarde de 32 Go dure environ 22 heures avec une taille de bloc de 1024 octets. La même sauvegarde dure environ 2 heures avec une taille de bloc de 32 Ko.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Le tableau ci-après présente le type de cartouche et les valeurs de densité (décimal et hexadécimal) pris en charge par l'unité de bande 7205–311. Lors d'une opération de restauration (lecture), l'unité règle automatiquement la densité sur celle de l'écriture. Lors d'une opération de sauvegarde (écriture), vous devez régler la densité sur celle de la cartouche de données que vous utilisez.

Cartouches de données prises en charge	Capacité native	Capacité des données compressées	Web-based System Manager ou SMIT	Valeur de densité hexadécimale
DLTtape III	2,6 Go	2,6 Go (sans compression)	23	17h
	6,0 Go	6,0 Go (sans compression)	24	18h
	10,0 Go	20,0 Go (par défaut pour l'unité)	25	19h
DLTapeIIIxt	15,0 Go	30,6 Go (par défaut pour l'unité)	25	19h
DLTapeIV	20,0 Go	40,0 Go	26	1Ah
	35,0 Go	70,0 Go (par défaut pour l'unité)	27	1Bh

Remarque : Si vous demandez une capacité native non prise en charge pour la cartouche de données, l'unité utilise la puissance de traitement maximale prise en charge pour la cartouche chargée dans l'unité.

Compression de données

La compression réelle dépend du type de données écrites. (voir le tableau ci-dessus) Un rapport de compression de 2/1 est adopté pour cette capacité des données compressées.

Attributs à valeur fixe

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs pour unités de bande 1/4 pouce 150 Mo (type 150mb)

Taille de bloc

La taille de bloc par défaut est 512. Pour les blocs de longueur variable, la seule taille de bloc possible est 0.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rebobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–150) ou dernière valeur de densité utilisée par le système.

Les valeurs par défaut sont 16 pour l'attribut densité égale à #1 et 15 pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 1/4 pouce 525 Mo (type 525mb)

Taille de bloc

La taille de bloc par défaut est 512. Les autres valeurs possibles sont 0 pour des blocs de longueur variable et 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rebobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
17	QIC–525*
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–525) ou dernière valeur de densité utilisée par le système.

* QIC–525 est le seul mode qui accepte une taille de bloc de 1024.

Les valeurs par défaut sont 17 pour l'attribut densité égale à #1 et 16 pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 1/4 pouce 1200 Mo (type 1200mb-c)

Taille de bloc

La taille de bloc par défaut est 512. Les autres valeurs possibles sont 0 pour des blocs de longueur variable et 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rebobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120
0	Valeur par défaut (QIC-1000) ou dernière valeur de densité utilisée par le système.

* QIC-525 et QIC-1000 sont les seuls modes qui acceptent une taille de bloc de 1024.

Les valeurs par défaut sont 21 pour l'attribut densité égale à #1 et 17 pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 4 mm 12 000 Mo (autoconfiguration)

Taille de bloc

La capacité de traitement du 12 000 Mo 4 mm est affectée par la taille de bloc. Pour cette unité, la taille de bloc minimale recommandée est de 32 Ko. Toute valeur inférieure réduit le débit des données (temps de sauvegarde/restauration). Le tableau ci-après répertorie les tailles de bloc recommandées par les commandes :

Commande prise en charge	Taille de bloc par défaut (octets)	RECOMMANDATION
BACKUP	32 Ko ou 51,2 Ko (par défaut)	32 Ko ou 51,2 Ko, selon que la commande Backup est par nom ou pas. Aucune modification de l'utilisateur n'est requise.
TAR	10 Ko	Il y a erreur dans le manuel qui indique une taille de bloc de 512 Ko. Définissez le paramètre de taille de bloc à -N64 .
MKSYSB	Voir BACKUP	MKSYSB utilise la commande BACKUP. Aucune modification de l'utilisateur n'est requise.
DD	n/a	Définissez le paramètre de taille de bloc à bs=32K
CPIO	n/a	Définissez le paramètre de taille de bloc à -C64 .

Remarque : Vous devez connaître la puissance et la capacité de traitement lorsque vous sélectionnez une taille de bloc. Les tailles de bloc réduites affectent les performances, mais pas la puissance de traitement.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Le tableau ci-après présente le type de cartouche et les valeurs de densité (décimal et hexadécimal) pris en charge par l'unité de bande 12 000 Mo 4 mm. Lors d'une opération de restauration (lecture), l'unité règle automatiquement la densité sur celle de l'écriture. Lors d'une opération de sauvegarde (écriture), vous devez régler la densité sur celle de la cartouche de données que vous utilisez.

Cartouches de données prises en charge	Capacité native	Capacité des données compressées	Valeur de densité Web-based System Manager ou SMIT	Valeur de densité hexadécimale
DDS III	2,0 Go	4,0 Go	19	13h
DDS2	4,0 Go	8,0 Go	36	24h
DDS3	12,0 Go	24,0 Go	37	25h

Remarque : Si vous demandez une capacité native non prise en charge pour la cartouche de données, l'unité utilise la puissance de traitement maximale prise en charge pour la cartouche chargée dans l'unité.

Compression de données

La compression réelle dépend du type de données écrites. (voir le tableau ci-dessus) Un rapport de compression de 2/1 est adopté pour cette capacité des données compressées.

Attributs à valeur fixe

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs pour unités de bande 1/4 pouce 13 000 Mo (autoconfiguration)

Taille de bloc

La taille de bloc par défaut est 512. Les autres valeurs possibles sont 0 pour des blocs de longueur variable et 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rebobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
33	QIC-5010-DC*
34	QIC-2GB*
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120
0	Valeur par défaut (QIC-5010-DC)*

* QIC-525, QIC-1000, QIC-5010-DC et QIC-2GB sont les seuls modes qui acceptent une taille de bloc de 1024.

Les valeurs par défaut sont 33 pour l'attribut densité égale à #1 et 34 pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation et de taille de bloc variable.

Attributs pour unités de bande 9 pistes 1/2 pouce (type 9trk)

Taille de bloc

La valeur par défaut est 1024.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
3	6 250 bits par pouce (bpp)
2	1 600 bpp
0	Densité précédemment utilisée

Les valeurs par défaut sont 3 pour l'attribut densité égale à #1 et 2 pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de tension, de réservation, de taille de bloc variable et de compression.

Attributs pour cartouche 1/2 pouce 3490e (type 3490e)

Taille de bloc

La valeur par défaut est 1024. Cette unité offre un débit de transfert de données élevé et la taille de bloc peut se révéler critique pour certaines opérations. La vitesse d'exploitation peut être sensiblement améliorée avec des blocs de grande taille. De façon générale, il est conseillé d'opter pour la plus grande taille de bloc possible.

Remarque : Augmenter la taille de bloc peut entraîner des incompatibilités avec d'autres programmes installés sur le système. Dans ce cas, vous en êtes averti lors de l'exécution des programmes concernés par le message :

```
Un appel système a reçu un paramètre incorrect.
```

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Compression

Reportez-vous aux informations générales fournies pour cet attribut.

Autochargement

Cette unité est équipée d'un séquenceur de bande, fonction d'autochargement qui charge et éjecte séquentiellement les cartouches de bande d'une série à partir d'un chargeur de cartouche. Pour cette opération, le commutateur situé sur le panneau avant de l'unité doit être positionné sur AUTO et l'attribut d'autochargement sur *yes*.

Attributs pour autres bandes SCSI (type ost)

Taille de bloc

La valeur par défaut est 512, mais elle peut être ajustée à la taille de bloc par défaut de votre unité de bande. Les valeurs les plus courantes sont 512 et 1024. Les unités de bande 8 et 4 mm utilisent généralement une taille de bloc de 1024. L'espace sur bande est mal exploité si l'attribut taille de bloc est laissé à 512. La valeur 0 indique une taille de bloc variable sur certaines unités.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

La valeur par défaut est 0 pour les deux densités. Les valeurs possibles et leur signification varient en fonction du type d'unité de bande.

Réservation

La valeur par défaut est no. Elle peut être basculée sur yes si l'unité accepte la fonction de réservation. En cas de doute, conservez la valeur no.

Taille de bloc de longueur variable

La valeur par défaut est 0. Les valeurs non nulles sont utilisées sur des unités QIC (Quarter Inch Cartridge). Pour plus de précisions, reportez-vous aux informations relatives à votre unité de bande.

Délai entre deux tentatives

Cet attribut ne s'applique qu'aux unités de bande de type ost.

Délai de lecture/écriture

Cet attribut ne s'applique qu'aux unités de bande de type ost.

Attributs à valeur fixe

Pour les unités de bande déclarées de type ost, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de tension et de compression.

Fichiers spéciaux pour unités de bande

L'écriture et la lecture de fichiers sur bande se fait à l'aide de fichiers spéciaux **rmt**. Plusieurs types de fichiers spéciaux sont associés à chaque unité de bande connue du système d'exploitation. Ces fichiers sont **/dev/rmt***, **/dev/rmt*.1**, **/dev/rmt*.2**, ... **/dev/rmt*.7** où **rmt*** représente le nom logique d'une unité de bande, par exemple **rmt0** ou **rmt1**.

Sélectionner l'un de ces fichiers spéciaux revient à choisir le mode d'exécution des opérations d'E/S sur l'unité de bande.

Densité	Vous pouvez opter pour une densité égale à #1 ou à #2. Ces densités sont définies dans les attributs de l'unité de bande. La densité maximale possible est généralement attribuée à la densité #1 et la valeur maximale suivante possible à la densité #2. C'est pourquoi, par abus de langage, les fichiers spéciaux utilisant la densité égale à #1 sont parfois assortis du qualificatif Haute densité et les fichiers spéciaux utilisant la densité égale à #2 du qualificatif Faible densité. Lors de la lecture de la bande, le paramètre de densité est ignoré.
Rebobinage à la fermeture	Vous pouvez demander le rebobinage automatique complet de la bande à la fermeture du fichier spécial relatif à l'unité de bande. Dans ce cas, le positionnement en début de bande est intégré au processus de fermeture du fichier.
Tension à l'ouverture	Vous pouvez demander que la bande soit retendue à l'ouverture du fichier, c'est-à-dire déroulée jusqu'à la fin, puis entièrement rebobinée. Cette précaution réduit le risque d'erreurs. Dans ce cas, le positionnement en début de bande est intégré au processus d'ouverture du fichier.

Le tableau ci-dessous donne la liste des fichiers spéciaux **rmt** et de leurs caractéristiques.

Fichier spécial	Rebobinage à la fermeture	Tension à l'ouverture	Densité
/dev/rmt*	oui	non	#1
/dev/rmt*.1	non	non	#1
/dev/rmt*.2	oui	oui	#1
/dev/rmt*.3	non	oui	#1
/dev/rmt*.4	oui	non	#2
/dev/rmt*.5	non	non	#2
/dev/rmt*.6	oui	oui	#2
/dev/rmt*.7	non	oui	#2

Si, par exemple, vous souhaitez écrire trois fichiers sur bande dans l'unité de bande **rmt2**, le premier en début de bande et les deux autres à la suite, avec la densité égale à #1 pour l'unité de bande, vous pouvez utiliser, dans l'ordre, les fichiers spéciaux suivants :

1. `/dev/rmt2.3`
2. `/dev/rmt2.1`
3. `/dev/rmt2`

Explication :

- Le fichier **/dev/rmt2.3** est choisi comme premier fichier car il est doté de l'option de rebobinage à l'ouverture qui garantit l'écriture du premier fichier en début de bande. L'option de rebobinage à la fermeture n'est pas retenue car l'opération d'E/S suivante doit commencer à la fin de ce fichier. Si la bande est déjà positionnée au début, l'utilisation du fichier **/dev/rmt2.1** comme premier fichier se révèle plus rapide, la phase de retension de la bande étant omise.
- Le fichier **/dev/rmt2.1** est choisi comme deuxième fichier car il ne comporte ni l'option de retension à l'ouverture, ni l'option de rebobinage à la fermeture. Or, le repositionnement en début de bande à l'ouverture ou à la fermeture du fichier est inutile.
- Le fichier **/dev/rmt2** est choisi comme troisième et dernier fichier car l'option de retension à l'ouverture n'est pas souhaitée, ce fichier étant précédé du deuxième fichier. En revanche, l'option de rebobinage à la fermeture est sélectionnée car aucune opération d'écriture n'est prévue à la suite du troisième fichier. La prochaine utilisation de la bande commencera au début de la bande.

Le choix du fichier spécial **rmt** n'est pas le seul moyen de contrôle des opérations sur bande ; vous disposez également de la commande **tctl**.

Annexe A. Comparaisons pour les gestionnaires de systèmes BSD

Rubriques de cette annexe :

- Comparaison de AIX et de BSD pour les gestionnaires de systèmes
- Présentation de AIX pour les gestionnaires de systèmes BSD
- Principales différences entre 4.3 BSD et ce système d'exploitation
- Comptabilité pour les gestionnaires de systèmes BSD 4.3
- Sauvegarde pour les gestionnaires de systèmes BSD 4.3
- Démarrage pour les gestionnaires de systèmes BSD 4.3
- Commandes d'administration de système pour les gestionnaires de systèmes BSD 4.3
- Commande Cron pour les gestionnaires de systèmes BSD 4.3
- Unités pour les gestionnaires de systèmes BSD 4.3
- Tableau de comparaison de fichiers pour 4.3 BSD, SVR4 et ce système d'exploitation
- Systèmes de fichiers pour les gestionnaires de systèmes BSD 4.3
- Recherche et analyse des fichiers pour les gestionnaires de systèmes BSD 4.3
- Espace de pagination pour les gestionnaires de systèmes BSD 4.3
- Mise en réseau pour les gestionnaires de systèmes BSD 4.3
- Documentation en ligne et commande man pour les gestionnaires de systèmes BSD 4.3
- NFS et NIS (Yellow Pages auparavant) pour les gestionnaires de systèmes BSD 4.3
- Mots de passe pour les gestionnaires de systèmes BSD 4.3
- Mesure des performances et réglage pour les gestionnaires de systèmes BSD 4.3
- Imprimantes pour les gestionnaires de systèmes BSD 4.3
- Terminaux pour les gestionnaires de systèmes BSD 4.3
- UUCP pour les gestionnaires de systèmes BSD 4.3

Comparaison entre ce système d'exploitation et BSD pour les administrateurs système

Les sections suivantes sont d'ordre général :

- Introduction to this operating system for BSD System Managers
- Major Differences Between 4.3 BSD Systems and this Operating System
 - Comptabilité pour administrateurs système BSD 4.3
 - Sauvegarde pour administrateurs système BSD 4.3
 - Amorçage et lancement pour administrateurs système BSD 4.3
 - Commandes d'administration d'AIX pour administrateurs système BSD 4.3
 - Cron pour administrateurs système BSD 4.3
 - Unités pour administrateurs systèmes BSD 4.3
 - Tableau de comparaison de fichiers BSD 4.3, SVR4 et AIX
 - Systèmes de fichiers pour administrateurs système BSD 4.3
 - Recherche et examen de fichiers pour administrateurs système BSD 4.3
 - Espace de pagination pour administrateurs système BSD 4.3
 - Réseau pour administrateurs système BSD 4.3
 - Documentation en ligne et commande man pour administrateurs système BSD 4.3
 - NFS et NIS (ex"Yellow Pages") pour administrateurs système BSD 4.3
 - Mots de passe pour administrateurs système BSD 4.3
 - Mesure et affinement des performances pour administrateurs système BSD 4.3
 - Imprimantes pour administrateurs système BSD 4.3
 - Terminaux pour administrateurs système BSD 4.3
 - UUCP pour administrateurs système BSD 4.3

Introduction à ce système d'exploitation pour administrateurs système BSD

Voici quelques conseils qui vous aideront à démarrer l'administration du système :

- Commencez par vous connecter, en tant qu'utilisateur racine, sur la console graphique.
- Exécutez les tâches de gestion à partir de la console système tant que vous n'êtes pas complètement à l'aise avec le système : il est plus simple de travailler depuis cette console qu'à partir d'un terminal distant. Une fois qu'AIX n'aura plus de secret pour vous, vous pourrez sans problème travailler à distance depuis un terminal xterm ou ASCII.
- Plusieurs utilitaires de ce système d'exploitation sont proposés pour la gestion du système. Ces utilitaires sont les suivants :
 - SMIT (System Management Interface Tool) : fournit une interface entre l'administrateur et les commandes de configuration et de gestion. SMIT facilite l'exécution de nombreuses tâches d'administration. Pour plus d'informations, voir Outil SMIT (System Management Interface Tool).
 - Le gestionnaire ODM (Object Data Manager) : fournit des routines d'accès aux objets des bases de données ODM. Ces bases contiennent des informations sur la configuration des unités. Pour en savoir plus, reportez-vous à "Unités - généralités".
 - Le contrôleur SRC (System Resource Controller) : donne accès et contrôle les démons et aux autres ressources système, et permet le contrôle, via une interface unique. Pour en savoir plus, reportez-vous à "Contrôleur SRC – généralités".

Principales différences entre BSC 4.3 et ce système d'exploitation

Cet article récapitule les principales différences entre le présent système d'exploitation et BSD 4.3. Pour obtenir plus d'informations sur ces thèmes, reportez-vous à la liste d'articles dans la section Comparaison entre AIX et BSD pour les administrateurs système.

Stockage des données de configuration

BSD 4.3 stocke généralement les données de configuration dans des fichiers ASCII. Les informations apparentées se trouvent sur une même ligne et le traitement des enregistrements (tri et recherche) peut être effectué sur le fichier ASCII lui-même. Ces enregistrements, de longueur variable, sont terminés par un saut de ligne. BSD 4.3 offre des outils permettant de convertir les fichiers ASCII volumineux en format base de données (dbm). Les fonctions de bibliothèque correspondantes explorent la paire de fichiers dbm s'ils existent, ou, dans le cas contraire, le fichier ASCII d'origine.

Certaines données de configuration du présent système d'exploitation sont stockées dans des fichiers ASCII, mais le plus souvent sous forme de *strophes*. Une strophe est un ensemble d'éléments d'information apparentés, stockés dans un groupe de lignes. Chaque élément est doté d'une étiquette, simplifiant l'appréhension du contenu du fichier.

Le présent système d'exploitation prend également en charge les versions dbm des mots de passe et des informations utilisateur. Les fichiers **/etc/passwd**, **/etc/group** et **/etc/inittab** sont en outre des exemples de fichiers de ce système d'exploitation où les informations sont stockées sous forme traditionnelle et non sous forme de strophes.

Les autres données de configuration de ce système d'exploitation sont stockées dans des fichiers maintenus par le gestionnaire d'objet ODM (Object Data Manager). Web-based System Manager ou SMIT (System Management Interface Tool) peut manipuler et afficher les informations des fichiers ODM. Vous pouvez également faire appel directement aux commandes ODM pour visualiser ces fichiers. Pour interroger les fichiers ODM, vous disposez des commandes :

- **odmget**,
- **odmshow**.

Pour modifier ces fichiers, des commandes :

- **odmadd**,
- **odmcreate**,
- **odmdrop**,
- **odmchange**,
- **odmdelete**.

Attention : Modifier les fichiers ODM de manière incorrecte peut provoquer l'arrêt du système, avec impossibilité de le relancer. Ne faites pas appel aux commandes ODM que si des commandes spécifiques (telles que celles générées par SMIT ou Web-based System Manager) échouent.

Gestion de la configuration

Au démarrage d'un système sous le présent système d'exploitation, un ensemble de commandes de configuration sont appelées par le gestionnaire de configuration. Ces commandes sont appelées *méthodes*. Elles identifient les unités du système et mettent à jour les fichiers ODM appropriés dans le répertoire **/etc/objrepos**.

Les fichiers unité spéciaux du répertoire **/dev** ne sont pas préinstallés. Certains fichiers spéciaux (fichiers disque, par exemple) sont créés automatiquement au cours du processus de configuration du démarrage. D'autres fichiers spéciaux (ceux des terminaux ASCII, par exemple) doivent être créés par l'administrateur système par le biais du menu Unités de SMIT ou de l'application Web-based System Manager Devices. Ces informations sont conservées dans ODM en vue d'un usage ultérieur.

Gestion de disque

Sous le présent système d'exploitation, les unités de disque sont des *volumes physiques*. Les partitions forment des *volumes logiques*. Comme dans BSD 4.3, un volume physique peut être associé à plusieurs volumes logiques. Mais, contrairement à BSD 4.3, un seul volume de ce système d'exploitation peut s'étendre sur plusieurs volumes physiques. Pour ce faire, il faut regrouper les volumes physiques dans un *groupe de volumes* et créer les volumes logiques sur ce groupe.

Voici quelques-unes des commandes relatives aux systèmes de fichiers et à la gestion des volumes :

- **crfs**
- **varyonvg**
- **varyoffvg**
- **lsvg**
- **importvg**
- **exportvg**

Les commandes BSD 4.3 suivantes sont également disponibles :

- **mkfs**
- **fsck**
- **fsdb**
- **mount**
- **umount**

Les différences entre ces commandes pour BSD 4.3 et pour ce système d'exploitation sont expliquées dans la section Systèmes de fichiers pour administrateurs système BSD 4.3, page A-19 .

BSD 4.3 maintient la liste des systèmes de fichiers dans le fichiers **/etc/fstab**. Le présent système d'exploitation maintient une strophe pour chaque système de fichiers dans le fichier **/etc/filesystems**.

Nouvelles commandes

Pour prendre en charge les nouveaux systèmes de configuration et de gestion de disque, le présent système d'exploitation propose maintenant environ 150 commandes nouvelles (nouvelles aussi pour les administrateurs BSC 4.3). Pour plus d'informations, reportez-vous à la section Commandes d'administration pour administrateurs système BSD 4.3.

Amorçage et lancement

Le présent système d'exploitation prend en charge l'identification et la configuration automatiques des unités. Par conséquent, le démarrage est très différent de celui des systèmes BSD 4.3. Outre le noyau, une image d'un système de fichiers d'amorçage, ainsi que les données de configuration (des unités) antérieures, sont chargés sur un disque RAM. Au cours de la première phase du lancement, un nombre de données de configuration suffisant pour permettre l'accès aux volumes logiques est chargé et vérifié. L'unité d'espace de pagination est identifiée auprès du noyau et le système de fichiers racine du disque est vérifié. Le système d'exploitation déplace alors le système de fichiers racine du disque RAM vers le disque dur, et achève la procédure de lancement, en configurant notamment les autres unités.

Autorisation utilisateur

BSD 4.3, et les systèmes UNIX version AT&T antérieures à SVR4, stockent toutes les données d'authentification utilisateur (mots de passe chiffrés compris) dans le fichier **/etc/passwd**. Normalement, ce fichier est lisible par tous.

Sur les systèmes SVR4, les mots de passe chiffrés ne se trouvent plus dans le fichier **/etc/passwd**, mais dans le fichier **/etc/shadow**. Ce fichier est accessible aux seuls utilisateurs racine et aux programmes sécurisés (**/bin/login** par exemple).

Le présent système d'exploitation enregistre les mots de passe chiffrés dans le fichier **/etc/security/passwd**. Le répertoire **/etc/security** contient deux autres fichiers, **user** et **limits**. Ces trois fichiers déterminent les droits d'accès d'un utilisateur au système (accès aux commandes **rlogin** ou **telnet**, par exemple) et les limites des ressources utilisateur (taille de fichier, espace d'adressage, etc.).

Impression

La plupart des commandes d'impression BSD 4.3 sont acceptées. Parmi les différences, minimales, notez que **/etc/qconfig** est le fichier de configuration de ce système d'exploitation.

Les systèmes d'impression ligne de ce système d'exploitation et de BSC 4.3 peuvent coopérer : il est possible de soumettre des travaux à des systèmes BSD 4.3 et d'imprimer des travaux soumis par des systèmes BSC 4.3.

Shells

Le présent système d'exploitation prend en charge les shells Bourne, C et Korn. Le chemin d'accès complet au shell Bourne est **/bin/bsh**. Le fichier **/bin/sh** est un lien fixe au fichier **/bin/ksh**. Ce fichier est modifiable par l'administrateur.

AIX ne prend pas en charge **setuid** ou **setgid** comme scripts shell dans quelque shell que ce soit.

Remarques :

1. Ce système d'exploitation ne dispose pas de scripts shell basés sur **/bin/sh**. Mais de nombreux scripts d'autres systèmes sont basés sur **/bin/sh** (le shell Bourne).
2. Malgré la similarité des shells Bourne et Korn, le shell Korn n'est pas exactement un surensemble du shell Bourne.

Comptabilité pour administrateurs système BSD 4.3

Les fichiers de comptabilité du répertoire **/usr/lib/acct** et les outils de suivi de l'activité système du répertoire **/usr/lib/sa** de ce système d'exploitation sont identiques à ceux de AT&T System V Release 4 (SVR4) combinés avec les utilitaires de comptabilité BSC 4.3.

La plupart des commandes de comptabilité se trouvent dans le répertoire **/usr/lib/acct**. Pour démarrer le système de comptabilité, exécutez la commande **/usr/lib/acct/startup**. Sinon, aucune commande de comptabilité (**lastcomm** (1), par exemple) ne renverra d'informations.

Ce système d'exploitation fournit les fonctions de compatibilité BSC 4.3 suivantes :

last (1)	Indique les dernières connexions utilisateur et terminale.
lastcomm (1)	Affiche, dans l'ordre inverse, les dernières commandes exécutées.
acct (3)	Active/désactive la comptabilité des processus.
ac (8)	Comptabilité de connexion.
accton (8)	Active/désactive la comptabilité système.
sa (8)	Maintient les fichiers de comptabilité système.

Ce système d'exploitation fournit également les commandes de comptabilité SVID (System V Interface Definition) Issue II et les fonctions de bibliothèque suivantes :

acctcms (1)	Génère un rappel de la syntaxe des commandes pour les enregistrements de comptabilité.
acctcms (1)	Affiche les récapitulatifs sélectionnés des enregistrements de comptabilité des processus.
acctcon1 (1)	Convertit les enregistrements de connexion/déconnexion en enregistrements de session.
acctcon2 (1)	Convertit les enregistrements de connexion/déconnexion en enregistrements de cumul.
acctdisk (1)	Génère des enregistrements de cumul à partir des résultats de la commande diskusg (1).
acctmerg (1)	Fusionne des fichiers de cumul dans un fichier intermédiaire.
accton (1)	Active le système de comptabilité.
acctprc1 (1)	Traite les données comptables issues de la commande acct (3).
acctprc2 (1)	Traite les résultats de la commande acctprc1 (1) dans des enregistrements de cumul.
acctwtmp (1)	Manipule les enregistrements de durée de connexion.
chargefee (1)	Impute au nom de connexion.
ckpacct (1)	Contrôle la taille du fichier /usr/adm/pacct .
diskusg (1)	Génère des données comptables relatives au disque.
dodisk (1)	Effectue des opérations comptables sur le disque.
fwtmp (1)	Convertit des enregistrements binaires (fichier wtmp) en enregistrements ASCII formatés.
	Remarque : Le fichier wtmp se trouve dans le répertoire /var/adm .
lastlogin (1)	Met à jour les dates de dernière connexion de chaque utilisateur.

monacct (1)	Crée des fichiers récapitulatifs mensuels.
prctmp (1)	Imprime le fichier d'enregistrement de session issu de la commande acctcon1 (1).
prdaily (1)	Formate l'état comptable de la veille.
prtacct (1)	Formate et imprime un fichier de cumul comptable.
runacct (1)	Exécute la comptabilité quotidienne.
shutacct (1)	Appelée par la commande d'arrêt du système (shutdown) pour interrompre la comptabilité et en consigner la cause.
startup (1)	Appelée par l'initialisation du système pour démarrer la comptabilité.
turnacct (1)	Active/désactive la comptabilité des processus.
wtmpfix (1)	Corrige l'horodate dans un fichier avec le format wtmp .

Sauvegarde pour administrateurs système BSD 4.3

Les commandes **tar** et **cpio** peuvent transférer des données d'un système à un autre. La commande **tar** du présent système d'exploitation n'est pas totalement compatible avec la commande **tar** de BSD 4.3. La commande **tar** de ce système d'exploitation requiert l'option **-B** (entrée bloquée) si la lecture est effectuée à partir d'un tube. La version AT&T de la commande **cpio** est compatible avec cette version.

Le présent système d'exploitation peut lire et écrire le format des commandes **dump** et **restore**. Par exemple, la commande **backup** de ce système d'exploitation avec la syntaxe :

```
backup -0uf Unité SystèmeFichiers
```

équivalent à la commande BSD 4.3 **dump** avec la syntaxe :

```
dump 0uf Unité SystèmeFichiers
```

De même, la commande **restore** de ce système d'exploitation avec la syntaxe :

```
restore -mivf Unité
```

équivalent à la commande BSD 4.3 **restore** avec la syntaxe :

```
restore ivf Unité
```

Le présent système d'exploitation propose également les commandes BSD 4.3 **rdump** et **rrestore**. La seule différence est que, sous le présent système d'exploitation, chaque argument doit être précédé d'un **-** (tiret). Par exemple, la commande :

```
rdump -0 -f orca:/dev/rmt0 /dev/hd2
```

équivalent à la commande BSD 4.3 :

```
rdump 0f orca:/dev/rmt0 /dev/hd2
```

La commande **backup** de ce système d'exploitation, avec la syntaxe :

```
backup -0f /dev/rmt0 /dev/hd2
```

équivalent à la commande BSD 4.3 **dump** avec la syntaxe :

```
dump 0f /dev/rmt0 /dev/hd2
```

Support de bande SCSI non IBM

Le présent système d'exploitation ne prend pas directement en charge les dérouleurs de bande SCSI non IBM. Mais vous pouvez y ajouter vos propres en-tête et interface utilisant le dérouleur SCSI IBM. Pour en savoir plus, reportez-vous à Adding an unsupported device to the system dans *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

Amorçage et lancement pour administrateurs système BSD 4.3

Sur les systèmes BSD 4.3, le programme **init** est la dernière étape de la procédure d'amorçage. Le rôle essentiel de ce programme est de créer des processus pour chaque port de terminal disponible. Les ports de terminal disponibles sont repérables en lisant le fichier **/etc/ttys**.

Sur un système System V, le programme **init** est démarré à l'initialisation du système. Le processus **init** lance les processus en fonction des entrées du fichier **/etc/inittab**.

Le présent système d'exploitation adopte la procédure d'initialisation de System V. Vous pouvez éditer le fichier **/etc/inittab**, directement via la commande **telinit** ou par le biais d'une des commandes suivantes :

chitab (1)	modification d'article(s) dans le fichier /etc/inittab .
lsitab (1)	Affiche la liste des enregistrements du fichier /etc/inittab .
mkitab (1)	Crée des enregistrements dans le fichier /etc/inittab .
rmitab (1)	Supprime des enregistrements du fichier /etc/inittab .

Les modifications apportées au fichier **/etc/inittab** prennent effet au réamorçage suivant du système ou après exécution de la commande **telinit q**.

Commandes d'administration pour administrateurs système BSD 4.3

Voici la liste des commandes propres à l'administration de l'environnement du présent système d'exploitation.

bosboot (1)	Initialise une unité d'amorçage.
bootlist (1)	Modifie la liste des unités d'amorçage (ou leur ordre dans cette liste) disponibles pour le système.
cfgmgr (1)	Configure des unités en exécutant les programmes du répertoire /etc/methods .
chcons (1)	Réachemine la console système vers une unité ou un fichier – effectif au démarrage suivant.
chdev (1)	Modifie les caractéristiques d'une unité.
chdisp (1)	Change l'écran utilisé par le sous-système LFT (low-function terminal).
checkcw (1)	Prépare le texte en espacement fixe pour la commande troff .
checkeq (1)	Vérifie les documents formatés par les macros de type memorandum.
checkmm (1)	Vérifie les documents formatés par les macros de type memorandum.
checknr (1)	Contrôle les fichiers nroff et troff .
chfont (1)	Change la police par défaut sélectionnée au moment de l'amorçage.
chfs (1)	Modifie les attributs d'un système de fichiers.
chgroup (1)	Modifie les attributs des groupes.
chgrpmem (1)	Modifie les administrateurs ou les membres d'un groupe.
chhwkbd (1)	Modifie les attributs du clavier LFT (low-function terminal) enregistrés dans la base de données ODM (Object Data Manager).
chitab (1)	Modifie des articles dans le fichier /etc/inittab .
chkbd (1)	Modifie la mappe clavier par défaut utilisée par le LFT (low-function terminal) au moment de l'amorçage.
chkey (1)	Modifie votre clé de chiffrement.
chlang	Définit la variable d'environnement LANG dans le fichier /etc/environment pour la connexion suivante.
chlicense (1)	Il existe deux types de licence utilisateur: Les licences fixes sont toujours activées, leur nombre pouvant être modifié via l'option -u . Les licences flottantes sont activées ou désactivées via l'option -f .
chlv (1)	Modifie les caractéristiques d'un volume logique.
chnamsv (1)	Modifie la configuration d'un service de noms TCP/IP sur un hôte.
chprtsv (1)	Modifie la configuration d'un service d'impression sur une machine client ou serveur.
chps (1)	Modifie les attributs d'un espace de pagination.
chpv (1)	Modifie les caractéristiques d'un volume physique dans un groupe de volumes.
chque (1)	Modifie le nom de la file d'attente.
chquedev (1)	Change le nom d'unité de l'imprimante ou du traceur.
chssys (1)	Modifie la définition d'un sous-système dans la classe d'objets sous-système.
chtcb (1)	Modifie ou interroge l'attribut TCB (Trusted Computing Base) d'un fichier.

chtz	Modifie les informations relatives au temps système.
chuser (1)	Modifie les attributs d'un utilisateur.
chvfs (1)	Modifie des articles dans le fichier /etc/vfs .
chvg (1)	Définit les caractéristiques d'un groupe de volumes.
chvirprt (1)	Modifie la valeur des attributs d'une imprimante virtuelle.
crfs (1)	Ajoute un système de fichiers.
crvfs (1)	Crée des entrées dans le fichier /etc/vfs .
exportvg (1)	Exporte la définition d'un groupe de volumes à partir d'un ensemble de volumes physiques.
extendvg (1)	Ajoute des volumes physiques à un groupe de volumes.
grpck (1)	Vérifie une définition de groupe.
importvg (1)	Importe une nouvelle définition de groupe de volumes à partir d'un ensemble de volumes physiques.
lsallq (1)	Affiche la liste de toutes les files d'attente configurées.
lsallqdev (1)	Affiche la liste de toutes les files d'attente d'imprimante et de traceur configurées dans une file d'attente donnée.
lsdisp (1)	Affiche la liste des écrans disponibles sur le système.
lsfont (1)	Affiche la liste des polices disponibles sur l'écran.
lsfs (1)	Affiche les caractéristiques de systèmes de fichiers.
lsgroup (1)	Affiche les attributs de groupes.
lsitab (1)	Affiche la liste des enregistrements du fichier /etc/inittab .
lskbd (1)	Affiche la liste des mappes clavier disponibles pour le sous-système LFT (low-function terminal).
lslicense (1)	Affiche le nombre de licences fixes et l'état des licences flottantes.
lspp (1)	Affiche la liste des logiciels en option.
lsnamsv (1)	Affiche les informations sur le service de noms, enregistrées dans la base de données.
lsprtsv (1)	Affiche les informations sur le service d'impression, enregistrées dans la base de données.
lsp	Affiche l'espace de pagination et ses attributs.
lsque (1)	Affiche le nom de la strophe de file d'attente.
lsquedev (1)	Affiche le nom de la strophe d'unité.
lssrc (1)	Récupère l'état d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
lsuser (1)	Affiche les attributs des comptes utilisateur.
lsvfs (1)	Affiche la liste des entrées dans le fichier /etc/vfs .
mkcatdefs (1)	Prétraite un fichier source de messages.
runcat (1)	Etablit un tube du résultat de la commande mkcatdefs pour la commande gencat .
mkdev (1)	Ajoute une unité au système.
mkfont (1)	Ajoute au système le code de police associé à un écran.
mkfontdir (1)	Crée un fichier fonts.dir à partir d'un répertoire de fichiers de police.
mkgroup (1)	Crée un groupe.
mkitab (1)	Crée des enregistrements dans le fichier /etc/inittab .
mklv (1)	Crée un volume logique.

mklvcopy (1)	Ajoute des copies à un volume logique.
mknamsv (1)	Configure un service de noms TCP/IP sur un hôte pour un client.
mknotify (1)	Ajoute une définition de méthode de notification à la classe d'objets de notification.
mkprtsv (1)	Modifie la configuration d'un service de noms TCP/IP sur un hôte.
mkps (1)	Ajoute un espace de pagination au système.
mkque (1)	Ajoute une file d'attente d'impression au système.
mkqudev (1)	Ajoute une unité de file d'attente d'impression au système.
mkserver (1)	Ajoute une définition de sous-serveur à la classe d'objets sous-serveur.
mkssys (1)	Ajoute une définition de sous-système à la classe d'objets sous-système.
mksysb	Sauvegarde des systèmes de fichiers montés dans le groupe de volumes rootvg pour les réinstallations ultérieures.
mkzfile	Enregistre la taille des systèmes de fichiers montés dans le groupe de volumes rootvg pour les réinstallations ultérieures.
mktcpip (1)	Définit les valeurs requises pour démarrer TCP/IP sur un hôte.
mkuser (1)	Crée un compte utilisateur.
mkuser.sys (1)	Personnalise un nouveau compte utilisateur.
mkvg (1)	Crée un groupe de volumes.
mkvirprt (1)	Crée une imprimante virtuelle.
odmadd (1)	Ajoute des objets aux classes d'objets créées.
odmchange (1)	Modifie le contenu d'un objet sélectionné dans la classe spécifiée.
odmcreate (1)	Génère les fichiers .c (source) et .h (en-tête) requis pour le développement d'application ODM et crée des classes d'objets vides.
odmdelete (1)	Supprime les objets sélectionnés de la classe d'objets spécifiée.
odmdrop (1)	Supprime une classe d'objets.
odmget (1)	Extrait des objets des classes spécifiées et les place dans le fichier d'entrée odmadd .
odmshow (1)	Affiche la définition d'une classe d'objets.
pwdck (1)	Vérifie les informations d'authentification locale.
redefinevg	Redéfinit l'ensemble de volumes physique d'un groupe de volumes dans la base de données de configuration des unités.
reducevg (1)	Supprime des volumes physiques d'un groupe de volumes. Si tous les volumes physiques d'un groupe sont supprimés, le groupe lui-même l'est également.
reorgvg (1)	Réorganise l'affectation de la partition physique d'un groupe de volume.
restbase (1)	Restaure les informations personnalisées de l'image d'amorçage.
rm del (1)	Supprime un delta d'un fichier SCCS (Source Code Control System).
rmdev (1)	Supprime une unité du système.
rmf (1)	Supprime des dossiers et les messages qu'ils contiennent.
rmfs (1)	Supprime un système de fichiers.
rmgroup (1)	Supprime un groupe.
rm itab (1)	Supprime des enregistrements du fichier /etc/inittab .
rmlv (1)	Supprime des volumes logiques d'un groupe de volumes.
rmlvcopy (1)	Supprime des copies d'un volume logique.

rmm (1)	Supprime des messages.
rmnamsv (1)	Modifie la configuration d'un service de noms TCP/IP sur un hôte.
rmnotify (1)	Supprime une définition de méthode de notification de la classe d'objets de notification.
rmprtsv (1)	Supprime de la configuration un service d'impression sur une machine serveur ou client.
rmpps (1)	Supprime un espace de pagination du système.
rmque (1)	Supprime une file d'attente d'impression du système.
rmqueuedev (1)	Supprime du système une unité de file d'attente imprimante ou traceur.
rmserver (1)	Supprime une définition de sous-serveur de la classe d'objets sous-serveur.
rmssys (1)	Supprime une définition de sous-système de la classe d'objets sous-système.
rmuser (1)	Supprime un compte utilisateur.
rmvfs (1)	Supprime des entrées du fichier /etc/vfs .
rmvirprt (1)	Supprime une imprimante virtuelle.
savebase (1)	Sauvegarde les données d'unité personnalisées ODM sur l'unité d'amorçage.
swapoff (1)	Désactive un ou plusieurs espaces de pagination.
swapon (1)	Spécifie des périphériques supplémentaires pour la pagination et l'échange.
syncvg (1)	Synchronise les copies non courantes de volumes logiques.
usrck (1)	Vérifie une définition utilisateur.
varyoffvg (1)	Désactive un groupe de volumes.
varyonvg (1)	Active un groupe de volumes.

Cron pour administrateurs système BSD 4.3

Le démon **cron** du présent système d'exploitation est semblable à celui du System V Release 2. Une entrée du fichier **/etc/inittab** active le démon **cron**.

Unités pour administrateurs systèmes BSD 4.3

Une unité d'un système BSD 4.3 n'est accessible à une application que si :

- l'unité est physiquement installée et qu'elle fonctionne,
- le pilote de l'unité se trouve dans le noyau,
- les fichiers unité spéciaux correspondants se trouvent dans le répertoire **/dev**,

Une unité de ce système d'exploitation n'est accessible à une application que si :

- l'unité est physiquement installée et qu'elle fonctionne,
- le pilote de l'unité se trouve dans le noyau ou dans une extension chargée,
- les fichiers unité spéciaux correspondants se trouvent dans le répertoire **/dev**,
- la base de données objet du répertoire **/etc/objrepos** contient des entrées pour l'unité qui correspondent à la configuration physique.

Les programmes propres aux unités, appelés *méthodes*, qui se trouvent dans le répertoire **/etc/methods**, maintiennent la base de données objet. Les méthodes sont appelées par le gestionnaire de configuration (accessible via la commande **cfgmgr**) et d'autres commandes.

Si une application ne peut plus accéder à une unité, ce peut être le matériel qui est en cause, ou encore la base de données du répertoire **/etc/objrepos** qui est endommagée.

Les processus de la commande **cfgmgr** de la base de données de configuration (du répertoire **/etc/objrepos**) sont traités au moment du lancement par la commande **cfgmgr** (le gestionnaire de configuration).

Le pseudo-code ci-dessous illustre la logique du gestionnaire de configuration :

```
/* Main */
While there are rules in the Config_Rules database
{
    Get the next rule and execute it
    Capture stdout from the last execution
    Parse_Output(stdout)
}

/* Parse Output Routine */
/* stdout will contain a list of devices found */
Parse_OutPut(stdout)
{
    While there are devices left in the list
    {
        Lookup the device in the database
        if (!defined)
            Get define method from database and
execute
            if (! configured)
            {
                Get config method from database and
execute
                Parse_Output(stdout)
            }
    }
}
```

Tableau de comparaison de fichiers BSD 4.3, SVR4 et de ce système d'exploitation

Le tableau suivant compare noms et fonctions des fichiers dans les trois systèmes BSD 4.3, SVR4 et de ce système d'exploitation.

Tableau de comparaison de fichiers				
Fichier BSD 4.3	Fichier SVR4	Fichier de ce système d'exploitation	Base de données	Type (odm/dbm)
L-Devices	Devices	Devices	non	
L-dialcodes	Dialcodes	Dialcodes	non	
L.cmds	Permissions	Permissions	non	
L.sys	Systems	Systems	non	
USERFILE	Permissions	Permissions	non	
aliases	mail/namefiles	aliases	alias DB/DB	dbm
fstab	vfstab	filesystems	non	
ftpusers	ftpusers	ftpusers	non	
gettytab		N/A		
group	group	group	non	
hosts	hosts	hosts	non	
hosts.equiv	hosts.equiv	hosts.equiv	non	
inetd.conf	inetd.conf	inetd.conf	non	
map3270	N/A	map3270	non	
motd	motd	motd	non	
mtab	mnttab	N/A	non	
named.boot	named.boot	named.boot	non	
named.ca		named.ca	non	
named.hosts		named.data (voir remarque)	non	
named.local		named.local	non	
named.pid	named.pid	named.pid	non	
named.rev		named.rev	non	
networks	networks	networks	non	
passwd	passwd	passwd	non	
printcap	qconfig	qconfig		
protocols		protocols	non	
remote	remote	remote	non	
resolv.conf	resolv.conf	resolv.conf	non	
sendmail.cf	sendmail.cf	sendmail.cf	sendmail.cfDB	aucun
services		services	non	
shells	shells	N/A		

stab		N/A		
syslog.conf		syslog.conf	non	
syslog.pid		syslog.pid	non	
termcap	terminfo	terminfo		
ttys	ttys	N/A	oui	odm
types		N/A		
utmp	utmp	utmp		
vfont		N/A		
vgrindefs		vgrindefs		
wtmp	wtmp	wtmp		

Remarque : Les noms de fichiers **named.ca**, **named.hosts**, **named.local** et **named.rev** peuvent être définis par l'utilisateur dans le fichier **named.boot**. Les noms indiqués ici sont ceux qui ont été retenus dans la documentation de ce système d'exploitation.

Systèmes de fichiers pour administrateurs système BSD 4.3

Cette section propose une comparaison sommaire entre les systèmes de fichiers de ce système d'exploitation et ceux d'autres systèmes et indique les types de systèmes de fichiers pris en charge sur ce système d'exploitation.

Ce système d'exploitation passe par le fichier **/etc/filesystem** pour obtenir les informations sur les unités des systèmes de fichiers et propose des commandes similaires pour le montage et le démontage des systèmes de fichiers.

Fichiers **/etc/filesystems** et **/etc/fstab**

Les systèmes BSD 4.3 stockent les listes d'unités par bloc et de points de montage dans le fichier **/etc/fstab**.

Les systèmes SVR4 stockent les données sur les unités par bloc et sur les points de montage dans le fichier **/etc/vfstab**.

Le présent système d'exploitation stocke les données sur les unités sur bloc et sur les points de montage dans le fichier **/etc/filesystems**. Les commandes **crfs**, **chfs** et **rmfs** mettent à jour le fichier **/etc/filesystems**.

Les administrateurs BSD 4.3 seront sans doute intéressés par la variable **check** du fichier **/etc/filesystems**. Vous pouvez affecter à cette variable la valeur True, False ou une valeur numérique. Par exemple, vous pouvez spécifier `check=2` dans le fichier **/etc/filesystems**. Le nombre précise le passage de la commande **fsck** qui effectuera la vérification du système de fichiers concerné. Le paramètre **check** correspond au cinquième champ d'un enregistrement du fichier **/etc/fstab**.

Aucun paramètre relatif à la fréquence de cliché ne se trouve dans le fichier **/etc/filesystems**.

Support des systèmes de fichiers sur ce système d'exploitation

Ce système d'exploitation prend en charge les quotas disque.

Ce système d'exploitation ne permet pas le montage de disquettes comme systèmes de fichiers.

La syntaxe des commandes **mount** et **umount** dans ce système d'exploitation diffère de celle des versions BSD 4.3 et SVR4 de ces commandes. Le tableau récapitule les différentes syntaxes.

Fonction	Syntaxe de ce système d'exploitation	Syntaxe BSD 4.3	Syntaxe SVR4
monte tous les systèmes de fichiers	mount all	mount -a	mountall
démonte tous les systèmes de fichiers	umount all	umount -a	umountall

Voir Systèmes de fichiers pour plus d'informations.

Recherche et examen de fichiers pour administrateurs système BSD 4.3

Le présent système d'exploitation accepte les commandes BSD 4.3 suivantes :

- **which**,
- **whereis**,
- **what**,
- **file**.

Le présent système d'exploitation n'accepte pas la syntaxe BSD 4.3 **fast find** de la commande **find**. Il n'existe pour le moment pas de fonction de remplacement. Le script **ffind** suivant peut simuler la fonction :

```
#!/bin/bsh
PATH=/bin
for dir in /bin /etc /lib /usr
do
  find $dir -print | egrep $1
done
```

La syntaxe du script **ffind** est la suivante :

```
ffind NomFichie
```

Espace de pagination pour administrateurs système BSD 4.3

Les commandes suivantes aident à gérer l'espace de pagination (également appelé espace de permutation) :

- | | |
|--------------------|---|
| chps (1) | Modifie les attributs d'un espace de pagination. |
| lsps (1) | Affiche la liste des attributs d'un espace de pagination. |
| mkps (1) | Ajoute un espace de pagination au système. |
| rmps (1) | Supprime un espace de pagination du système. |
| swapoff (1) | Désactive un ou plusieurs espaces de pagination. |
| swapon (1) | Spécifie d'autres unités de pagination et de permutation. |

Si vous avez besoin d'un grand espace de pagination, placez un volume logique de pagination sur chaque disque : vous pourrez ainsi planifier la pagination sur plusieurs unités de disque.

Réseau pour administrateurs système BSD 4.3

Cet article traite de l'utilisation de la configuration de réseau BSD 4.3 ASCII, des commandes et des options complémentaires, de la résolution de noms et d'adresses sur ce système d'exploitation et des différences entre la gestion d'un réseau BSD 4.3 et celle de ce système d'exploitation.

Configuration BSD 4.3 : modification du lancement par défaut

Vous pouvez administrer les interfaces réseau de ce système d'exploitation via SMIT et les fichiers ODM, ou encore via les fichiers de configuration BSD 4.3 ASCII.

Pour administrer des interfaces réseau via les fichiers de configuration BSD 4.3 ASCII, annulez, dans le fichier **/etc/rc.net**, la mise en commentaire des commandes sous l'en-tête :

```
# Part II - Traditional
Configuration
```

Puis, si vous souhaitez que soient pris en charge la configuration des fichiers ordinaires et le support SRC, éditez le fichier **/etc/rc.net** et annulez la mise en commentaire des commandes **hostname**, **ifconfig** et **route** avec les paramètres appropriés.

Pour la configuration des fichiers ordinaires sans support SRC, lancez la commande **smit setbootup_option** pour passer à une configuration **rc** de style BSD. Cette option configure le système pour qu'il utilise le fichier **/etc/rc.bsdnet** au moment du lancement. Vous devez également éditer le fichier **/etc/rc.bsdnet** et annuler la mise en commentaire des commandes **hostname**, **ifconfig** et **route** avec les paramètres appropriés.

Autres options des commandes ifconfig et netstat

La commande **ifconfig** de ce système d'exploitation a les options complémentaires suivantes :

- | | |
|----------------|---|
| mtu | La variable <i>mtu</i> définit l'unité MTU (Maximum Transmission Unit) maximale utilisée sur le réseau local (et les sous-réseaux locaux) et l'unité MTU utilisée pour les réseaux distants. Pour optimiser la compatibilité avec Ethernet et les autres réseaux, affectez la valeur 1500 à la variable par défaut <i>mtu</i> Token-Ring et Ethernet. |
| allcast | L'option allcast définit la stratégie de diffusion Token-Ring. Définissez l'option allcast pour optimiser la connectivité via les ponts Token-Ring. Si vous désactivez l'option allcast (en définissant -allcast), vous réduisez l'excès de trafic sur l'anneau. |

La commande **netstat** de ce système d'exploitation utilise l'option **-v**. La commande **netstat -v** imprime les statistiques des pilotes telles que le nombre d'octets transmis, le nombre d'erreurs transmises, le nombre d'octets reçus et le nombre d'erreur reçues.

Autres commandes de gestion de réseau

Le système d'exploitation prend également en charge les commandes suivantes :

```
securetcpip
gated
no
iptrace
ipreport
```

Résolution de noms et d'adresses

Les sous-routines **gethostbyname** et **gethostbyaddr** de la bibliothèque **libc** fournissent le support des services DNS (Domain Name Service), NIS (Network Information Services, ex-Yellow Pages), et la base de données **/etc/hosts**. Si le fichier **/etc/resolv.conf** existe, le serveur de noms est le premier exploré. Si le nom n'est pas résolu et que NIS est actif, NIS est exploré. Si NIS n'est pas actif, c'est le fichier **/etc/hosts** qui est exploré.

Différences entre ce système d'exploitation et BSD 4.3

Sur ce système d'exploitation, les démons réseau sont lancés depuis le fichier **/etc/rc.tcpip**, et non depuis le fichier **/etc/rc.local**. Le script shell **/etc/rc.tcpip** est appelé depuis le fichier **/etc/inittab**, et non depuis le fichier **/etc/rc**.

Si le contrôleur SRC (System Resource Controller) est actif, les démons TCP/IP sont exécutés sous son contrôle. Si vous ne souhaitez pas qu'ils le soient, lancez la commande **smit setbootup_option** pour passer à une configuration **rc** de style BSD.

Les fonctions de gestion de réseau BSD 4.3 acceptées par ce système d'exploitation sont les suivantes :

- fonctions de journalisation SYSLOG au niveau noyau,
- support des systèmes XNS (Xerox Network Systems),
- droits d'accès aux sockets de domaine UNIX.

Commande **tn3270**

La commande **tn3270** est un lien avec la commande **telnet**, mais qui utilise le fichier **/etc/map3270** et la variable d'environnement courante **TERM** pour générer les mappages clavier 3270. Ainsi, la commande **tn3270** opère exactement comme sa version BSD.

Si vous souhaitez modifier les séquences d'échappement par défaut utilisées par les commandes **tn3270**, **telnet** et **tn**, définissez la variable d'environnement **TNESC** avant de lancer ces commandes.

Documentation en ligne et commande man pour administrateurs système BSD 4.3

Le présent système d'exploitation accepte les commandes **man -k**, **apropos** et **whatis**, mais la base de données utilisée par ces commandes doit être créée au préalable via la commande **catman -w**.

La commande **man** recherche d'abord les pages de texte plat dans les fichiers **/usr/man/cat?**. Puis, elle recherche les pages formatées **nroff** dans le fichier **/usr/man/man?**. Les nouvelles pages de manuel peuvent être ajoutées en texte plat ou au format **nroff**.

Remarques :

1. Les pages de texte de la commande **man** ne sont pas fournies avec le système. La base de données correspondante doit être créée via la commande **catman**. Ces pages peuvent être soit du texte plat stocké dans les fichiers **/usr/man/cat ?**, soit des pages formatées **nroff** stockées dans les fichiers **/usr/man/man ?**.
2. Le programme sous licence de formatage de texte doit être installé pour que la commande **nroff** soit à disposition de la commande **man** pour la lecture des pages formatées **nroff**.

NFS et NIS (ex"Yellow Pages") pour administrateurs système BSD 4.3

Les démons NFS (Network File System) et NIS (Network Information Services) sont lancés à partir du fichier **/etc/rc.nfs**. Ils supposent l'activation préalable du démon **portmap** dans le fichier **/etc/rc.tcpip**. Par défaut, le fichier **/etc/rc.nfs** n'est pas appelé par le fichier **/etc/inittab**. Si vous ajoutez une ligne dans le fichier **/etc/inittab** pour appeler le script **/etc/rc.nfs**, il doit être appelé après le script **/etc/rc.tcpip**.

Si NIS est actif, vous devez intégrer une entrée racine avant l'entrée **+::** (signe plus, deux-points, deux-points) dans le fichier **/etc/passwd** et une entrée système avant l'entrée **+::** dans le fichier **/etc/group**. Un administrateur système peut ainsi se connecter comme utilisateur racine et effectuer les modifications requises si le système ne parvient pas à communiquer avec le serveur NIS.

NFS peut être configuré via le Web-based System Manager (tapez **wsm** puis sélectionnez **Network**) ou le raccourci SMIT, **smit nfs**. Les menus Web-based System Manager et SMIT font référence à NIS (ex Yellow Pages) sous la forme NIS. Nombre des commandes NFS et NIS se trouvent dans les répertoires **/etc** et **/usr/etc**.

Certains environnements NFS utilisent une commande **arch** pour identifier les familles et les types de machines. Définissez l'identificateur **power** de la famille (UC) et l'identificateur **ibm6000** pour le type de machine.

Mots de passe pour administrateurs système BSD 4.3

Voici quelques précisions sur les différences de gestion des mots de passe entre ce système d'exploitation et BSD 4.3.

Définition d'un mot de passe utilisateur

Lorsque vous exécutez la commande `/bin/passwd` de ce système d'exploitation comme utilisateur racine, vous êtes invité à fournir le mot de passe racine. Voici un exemple :

```
# passwd cslater
Changing password for "cslater"
Enter root's Password or
cslater's Old password:
cslater's New password:
Re-enter cslater's
new password:
#
```

La version BSD 4.3 ne vous invite pas à entrer le mot de passe racine. En voici un exemple :

```
# passwd cslater
New password:
Retype new password:
#
```

Importation d'un fichier de mots de passe BSD 4.3

Pour importer un fichier de mots de passe BSD 4.3, copiez-le dans le fichier `/etc/passwd`, puis entrez :

```
pwdck -y ALL
```

Le fichier `/etc/security/limits` doit être ensuite mis à jour avec une strophe nulle pour tout nouvel utilisateur. La commande `usrck` le fait, mais elle peut poser problème sauf si le fichier `/etc/group` est importé avec le fichier `/etc/passwd`.

Remarque : Si le fichier `/etc/security/limits` est modifié, la pile ne doit pas dépasser 65 536 octets. Si elle dépasse cette limite, l'exécution de la commande `usrck` risque de poser problème : ramenez la taille de la pile à 65536 et relancez la commande `usrck`.

Exécutez également les commandes `grpck` et `usrck` pour vérifier les attributs groupe et utilisateur.

Edition du fichier de mots de passe (Password)

Sous ce système d'exploitation, les commandes `lsuser`, `mkuser`, `chuser` et `rmuser` permettent de gérer les mots de passe. Toutes ces commandes peuvent être exécutées via SMIT ou Web-based System Manager. Toutefois elles ne traitent qu'un utilisateur à la fois.

Remarque : Passer par un éditeur pour modifier plusieurs noms utilisateur requiert d'éditer simultanément plusieurs fichiers, car les mots de passe sont stockés dans le fichier `/etc/security/passwd`, les informations sur les droits d'accès, dans le fichier `/etc/security/user` et les autres données utilisateur, dans le fichier `/etc/passwd`.

Ce système d'exploitation rejette la commande `vipwn`, mais accepte la commande `mkpasswd`. Mais vous pouvez toujours administrer les mots de passe sur ce système d'exploitation comme vous le feriez sur un système BSD 4.3. Procédez comme suit :

1. Placez un fichier de mots de passe BSD 4.3 dans le fichier `/etc/shadow`.
2. Modifiez les droits d'accès au fichier :

```
chmod 000 /etc/shadow
```

3. Placez le script shell **vipw** suivant dans le répertoire **/etc** :

```
-----
#!/bin/bsh
#
# vipw. Uses pwdck for now. May use usrck someday
#
PATH=/bin:/usr/bin:/etc:/usr/ucb # Add to this if your editor is
                                # some place else
if [ -f /etc/ptmp ] ; then
    echo "/etc/ptmp exists. Is someone else using
vipw?"
    exit 1
fi
if [ ! -f /`which "$EDITOR" | awk '{ print $1 }` ] ; then
    EDITOR=vi
fi
cp /etc/shadow /etc/ptmp
if (cmp /etc/shadow /etc/ptmp) ; then
    $EDITOR /etc/ptmp
else
    echo cannot copy shadow to ptmp
    exit 1
fi
if (egrep "^root:" /etc/ptmp >/dev/null) ; then
    cp /etc/ptmp /etc/shadow ; cp /etc/ptmp /etc/passwd
    chmod 000 /etc/passwd /etc/shadow
    pwdck -y ALL 2>1 >/dev/null # return code 114 may change
    rc=$?
    if [ $rc -eq 114 ] ; then
        chmod 644 /etc/passwd
        rm -f /etc/passwd.dir /etc/passwd.pag
        mkpasswd /etc/passwd
        # update /etc/security/limits, or ftp
        # will fail
    else
        pwdck -y ALL
    fi
fi
else
    echo bad entry for root in ptmp
fi
rm /etc/ptmp
-----
```

4. Si vous utilisez le script shell **vipw** ou la commande **mkpasswd**, n'oubliez pas que Web-based System Manager, SMIT et les commandes **mkuser**, **chuser** et **rmuser** n'utilisent pas la commande **mkpasswd**. Vous devez lancer :

pour mettre à jour les fichiers **/etc/passwd.dir** et **/etc/passwd.pag**.

Attention : L'initialisation de la variable **IFS** et des instructions **trap** permettent de se prémunir contre certaines méthodes exploitant les failles au niveau de la sécurité, inhérentes à la fonction **setuid**. Les scripts shell **vipw** et **passwd** sont toutefois conçus pour des environnements relativement ouverts, où la compatibilité est un élément-clé. Si vous souhaitez un environnement plus sûr, utilisez exclusivement les commandes standard de ce système d'exploitation.

5. Placez le script shell **passwd** suivant dans le répertoire **/usr/ucb** :

```
-----  
#!/bin/ksh  
#  
# matches changes to /etc/security/passwd file with changes to  
#/etc/shadow  
#  
IFS=" "  
PATH=/bin  
trap "exit 2" 1 2 3 4 5 6 7 8 10 12 13 14 15 16 17 18 21 22 \  
      23 24 25 27 28 29 30 31 32 33 34 35 36 60 61 62  
if [ -n "$1" ]; then  
    USERNAME=$1  
else  
    USERNAME=$LOGNAME  
fi  
if [ -f /etc/ptmp ] ; then  
    echo password file busy  
    exit 1  
fi  
    trap "rm /etc/ptmp; exit 3" 1 2 3 4 5 6 7 8 10 12 13 \  
        14 15 16 17 18 21 22 23 24 25 27 28 29 30 31 \  
        32 33 34 35 36 60 61 62  
if (cp /etc/security/passwd /etc/ptmp) ; then  
    chmod 000 /etc/ptmp else  
    rm -f /etc/ptmp exit 1  
fi  
if ( /bin/passwd $USERNAME ) ; then  
    PW=`awk ' BEGIN { RS = "" }  
           $1 == user { print $4 } ' user="$USERNAME:" \  
/etc/security/passwd \  
else  
    rm -f /etc/ptmp  
    exit 1  
fi  
rm -f /etc/ptmp  
awk -F: ' $1 == user { print $1":"pw":"$3 ":"$4":"$5":"$6":"$7 }  
         $1 != user { print $0 }' user="$USERNAME" pw="$PW" \  
    /etc/shadow > /etc/ptmp  
chmod 000 /etc/ptmp  
mv -f /etc/ptmp /etc/shadow  
-----
```

6. Modifiez les droits d'accès au script **passwd** :

```
chmod 4711 /usr/ucb/passwd
```

7. Vérifiez que la variable d'environnement **PATH** de chaque utilisateur spécifie d'explorer le répertoire **/usr/ucb** avant le répertoire **/bin**.

Mesure des performances et réglage pour les gestionnaires systèmes BSD 4.3

Toutes les unités du système d'exploitation disposent d'attributs. Pour afficher les attributs des unités, entrez :

```
lsattr -E -l nom_unité
```

Tout attribut ayant la valeur True peut être modifié avec la commande :

```
chdev -l nom_unité -a attr=valeur
```

Attention : Modifier incorrectement les paramètres d'unité peut endommager le système.

Par défaut, le nombre maximal de processus par utilisateur est de 40. Cette valeur peut se révéler insuffisante pour des utilisateurs ayant ouvert simultanément plusieurs fenêtres. Pour modifier la valeur sur tout le système, entrez :

```
hdev -l sys0 -a maxuproc=100
```

Le maximum est ici porté à 100 (effectif dès le réamorçage du système).

Pour afficher la valeur courante, ainsi que d'autres attributs, entrez :

```
lsattr -E -l sys0
```

L'attribut **maxmbuf** n'est pour le moment pas accepté par les services **mbuf**.

Ce système d'exploitation accepte les commandes **vmstat** et **iostat**, mais non la commande **systat**, ni les moyennes de charge.

Imprimantes pour les gestionnaires de systèmes BSD 4.3 Dans AIX 5.1 et les versions supérieures,

le système d'exploitation prend en charge deux sous-systèmes d'impression : 4.3 BSD et System V. Le sous-système d'impression System V utilise les commandes, les files d'attente et les fichiers System V Release 4 et il est administré de la même manière. Les paragraphes suivants indiquent les connaissances nécessaires à la gestion du sous-système d'impression 4.3 BSD. Vous contrôlez l'activation du système via SMIT. Un seul sous-système peut être actif à la fois.

L'impression est gérée par des programmes et des configurations du répertoire `/usr/lpd`. La conception, la configuration, le mécanisme de mise en file d'attente et le processus démon des sous-systèmes d'impression de BSD 4.3 et de ce système d'exploitation sont différents. Les deux systèmes utilisent néanmoins le protocole **lpd** pour les impressions à distance. Ils utilisent également le fichier `/etc/hosts.lpd`, s'il existe, ou sinon `/etc/host.equiv`. Le sous-système d'impression de ce système d'exploitation offre une passerelle vers les sous-systèmes BSD 4.3 ; les systèmes utilisant ce système d'exploitation peuvent donc soumettre des travaux aux systèmes BSD 4.3 et accepter des travaux soumis par ces systèmes.

Le fichier `/etc/printcap` de BSD 4.3 n'existe pas dans le présent système d'exploitation. Ce fichier combine des informations de configuration du spouleur et de la base de données des capacités d'imprimante. Les utilisateurs doivent bien connaître le format et les mots-clés du fichier **printcap** pour configurer correctement une imprimante.

Le fichier `/etc/qconfig` de ce système d'exploitation ne contient que des informations sur la configuration du spouleur. Les capacités de l'imprimante sont définies dans la base de données prédéfinie/pré-personnalisée ODM. Vous disposez de la commande **mkvirprt** pour définir les capacités d'une imprimante donnée sur le système.

Pour rendre l'imprimante **lp0** disponible pour imprimer sur l'hôte distant `viking`, insérez, dans un fichier système BSD 4.3 `/etc/printcap` :

```
lp0|Print on remote printer attached to
viking:Z
:lp=:rm=viking:rp=lp:st=/usr/spool/lp0d
```

Pour faire de même sur le présent système d'exploitation, insérez les lignes suivantes dans le fichier `/etc/qconfig` :

```
lp0:
    device = dlp0
    host = viking
    rq = lp
dlp0:
    backend = /usr/lib/lpd/rembak
```

Pour en savoir plus sur le sous-système d'impression, reportez-vous à la section relative aux imprimantes (gestion système).

Ce système d'exploitation accepte les commandes d'impression et les fonctions de bibliothèque suivantes :

cancel (1)	Annule les demandes à une imprimante ligne.
chqueuedev (1)	Change le nom d'unité de l'imprimante ou du traceur.
chvirprt (1)	Modifie la valeur des attributs d'une imprimante virtuelle.
disable (1)	Désactive une file d'attente d'imprimante.
enable (1)	Active une file d'attente d'imprimante.
hplj (1)	Posttraite la sortie troff pour HP LaserJetII avec cartouche K.
ibm3812 (1)	Posttraite la sortie troff pour IBM 3812 Mod 2 Pageprinter.

ibm3816 (1)	Posttraite la sortie troff pour IBM 3816 Pageprinter.
ibm5587G (1)	Posttraite la sortie troff pour IBM 5587G avec cartouche 32x32/24x24.
lp (1)	Envoie des demandes à une imprimante ligne.
lpr (1)	Met des travaux d'impression en file d'attente.
lprm (1)	Supprime des travaux de la file de spouillage d'une imprimante ligne.
lpstat (1)	Affiche des informations sur l'état d'une imprimante ligne.
lpstat (1)	Génère la configuration d'impression d'une imprimante ligne.
lsallqdev (1)	Affiche la liste de toutes les unités de file d'attente configurées d'une file d'attente.
lsvirprt (1)	Affiche les attributs d'une imprimante virtuelle.
mkque (1)	Ajoute une file d'attente d'impression au système.
mkqudev (1)	Ajoute une unité de file d'attente d'impression au système.
mkvirprt (1)	Crée une imprimante virtuelle.
pac (1)	Prépare les enregistrements comptables de l'imprimante/du traceur.
piobe (1)	Imprime le gestionnaire des travaux d'impression pour le programme expéditeur de l'imprimante.
pioburst (1)	Génère les pages d'en-tête et de fin pour les sorties.
piocmdout (3)	Sous-routine qui génère une chaîne d'attribut pour un formateur d'impression.
piodigest (1)	Prétraite les valeurs des attributs de définition d'une imprimante virtuelle et les enregistre.
pioexit (3)	Sous-routine existant à partir d'un formateur d'impression.
pioformat (1)	Pilote un formateur d'impression.
piofquote (1)	Convertit certains caractères de contrôle destinés aux imprimantes PostScript.
piogetstr (3)	Sous-routine qui extrait une chaîne d'attribut pour un formateur d'impression.
piogetvals (3)	Sous-routine qui initialise les variables base de données des attributs d'impression pour un formateur d'impression.
piomsgout (3)	Sous-routine qui envoie un message à partir d'un formateur d'impression.
pioout (1)	Programme pilote d'unité du programme expéditeur de l'imprimante.
piopredef (1)	Crée une prédéfinition du flot de données d'impression.
proff (1)	Formate le texte pour les imprimantes avec flots de données personnelles.
qadm (1)	Administre le système de spouillage de l'imprimante.
qconfig (4)	Configure un système de files d'attente d'impression.
qstatus (1)	Fournit l'état de l'imprimante au système de files d'attente d'impression.
restore (3)	Restaure l'imprimante à son état par défaut.
rmque (1)	Supprime une file d'attente d'impression du système.
rmqudev (1)	Supprime du système une unité de file d'attente imprimante ou traceur.
rmvirprt (1)	Supprime une imprimante virtuelle.
splp (1)	Affiche ou modifie les paramètres du pilote d'impression.
xpr (1)	Formate un fichier de cliché de fenêtre pour une sortie imprimante.

Terminaux pour administrateurs système BSD 4.3

Traditionnellement, pour activer/désactiver un port, les administrateurs BSD 4.3 modifient le fichier **/etc/ttys** et envoient un signal **HUP** au programme **init**.

Ce système d'exploitation stocke les informations sur le port du terminal dans le gestionnaire ODM et lance les terminaux lorsque le programme **init** lit le fichier **/etc/inittab**. Dans ce système d'exploitation, vous devriez utiliser l'application Web-based System Manager Devices ou SMIT pour configurer les ports du terminal.

Il n'existe pas de mappage fixe entre le port et le nom de fichier unité spécial dans le répertoire **/dev**. Cela peut engendrer des doutes sur le port à configurer, pour les administrateurs abordant ce système d'exploitation. Dans les menus SMIT, le port série de la première carte (libellé **s1**) est référencé par l'emplacement **00-00-S1**, la carte **sa0** et le port **s1** dans le menu SMIT. Le port série de la seconde carte (libellé **s2**) est référencé par l'emplacement **00-00-S2**, la carte **sa1** et le port **s2**.

Pour activer/désactiver un port, vous disposez des commandes **penable** et **pdisable**.

termcap et terminfo

Comme System V, le présent système d'exploitation se sert des entrées **terminfo** du fichier **/usr/lib/terminfo/?/***. Les utilisateurs BSD 4.3 trouveront sans doute utiles les commandes suivantes :

- | | |
|---------------------|--|
| captoinfo(1) | Convertit un fichier termcap en fichier terminfo |
| tic(1) | Traduit les fichiers terminfo source en format compilé. |

Ce système d'exploitation inclut la source de nombreuses entrées **terminfo**. Certaines doivent être compilées via la commande **tic**. Le fichier **termcap** se trouve dans le fichier **/lib/libtermcap/termcap.src**.

UUCP pour administrateurs système BSD 4.3

Ce système d'exploitation fournit les utilitaires BNU (Basic Networking Utilities) System V (souvent appelés HDB UUCP).

Dialers (4)	Affiche la liste des modems utilisés par les liaisons BNU à distance.
Maxuuxqts (4)	Limite le nombre d'instances de démons BNU uuxqt exécutables simultanément.
Permissions (4)	Spécifie les droits des systèmes distants sur les commandes BNU.
Poll (4)	Spécifie le moment où BNU doit interroger les systèmes distants.
Systems (4)	Affiche la liste des ordinateurs distants avec lesquels peut communiquer le système local.
rmail (1)	Gère le courrier reçu à distance via BNU.
uuchek (1)	Vérifie les fichiers et les répertoires requis par BNU.
uuclean (1)	Supprime les fichiers du répertoire de spoupage BNU.
uucleanup (1)	Supprime les fichiers sélectionnés du répertoire de spoupage BNU.
uucpadm (1)	Entre les informations de configuration BNU de base.
uudemon.admin (1)	Donne régulièrement des informations sur l'état des transferts de fichiers BNU.
uudemon.cleanu (1)	Nettoie les répertoires de spoupage et les fichiers journaux BNU.
uudemon.hour (1)	Lance les appels de transport de fichier vers les systèmes distants via BNU.
uudemon.poll (1)	Interroge les systèmes spécifiés dans le fichier d'interrogation BNU.
uulog (1)	Donne des informations sur les activités de transfert de fichiers BNU sur un système.
uupoll (1)	Force l'interrogation d'un système BNU distant.
uuq (1)	Affiche la file d'attente des travaux BNU et supprime de cette file les travaux spécifiés.
uusnap (1)	Affiche l'état des contacts BNU avec les systèmes distants.
uustat (1)	Consigne l'état et propose un contrôle limité des opérations BNU.

Ce système d'exploitation propose également les commandes BSD 4.3 **uuencode** et **uudecode**. La commande HDB **uugetty** n'est pas acceptée.

Pour en savoir plus, reportez-vous à Fichiers BNU, formats de fichiers et répertoires dans *AIX 5L Version 5.2 - Guide de l'utilisateur : communications et réseaux*.

Index

A

- adressabilité de fragment de système de fichiers, 5-24
- AIX, généralités sur les administrateurs système BSD
 - amorçage et lancement, A-10
 - commandes, A-11
 - comptabilité, A-7
 - cron, A-15
 - espace de pagination, A-21
 - sauvegarde, A-9
 - unités, A-16
 - UUCP, A-33
- allocation, fichier vide (kproc), 5-27
- allocations de fichier vide, 5-27
- amorçage
 - AIX pour administrateurs système BSD, A-10
 - description
 - généralités, 2-3
 - mode de maintenance, 2-6
 - système de fichiers RAM, 2-7
 - traitement de l'amorçage, 2-4

B

- blocs, coûts des performances de , 5-24
- BSD, A-29, A-30
 - comparaison à ce système d'exploitation pour les administrateurs système, comptabilité, sauvegarde, amorçage et lancement, commandes, A-2
 - comparaison avec les administrateurs système AIX
 - amorçage et lancement, A-10
 - commandes, A-11
 - comptabilité, A-7
 - cron, A-15
 - espace de pagination, A-21
 - sauvegarde, A-9
 - unités, A-16
 - UUCP, A-33
 - comparaison pour administrateurs système, A-4
 - comparaison pour les administrateurs système, A-3
 - comparaison de fichiers, A-17
 - documentation en ligne et commande man, A-24
 - mots de passe, A-26
 - NFS et NIS (exYellow Pages), A-25
 - recherche et examen de fichiers, A-20
 - réseau, A-22
 - systèmes de fichiers, A-19
 - terminaux, A-32
 - comparaison pour les gestionnaires de systèmes, A-1
 - imprimantes, A-30
 - performances, A-29

C

- clavier, modification des attributs, utilisation de la commande chhwkbd, A-11
- codes d'emplacement, 14-5
 - défini, 14-5
 - disque connecté directement, 14-7
 - disque série, 14-7
 - imprimante/traceur, 14-6
 - port multiprotocole, 14-8
 - rotateur/clavier LPFK, 14-8
 - tty, 14-6
 - unité de disquette, 14-8
 - unité SCSI, 14-7
- codes d'emplacement des rotateurs/claviers LPFK, 14-8
- Commande man, 1-4
- commande man, pour administrateurs système BSD, A-24
- commandes, AIX pour administrateurs système BSD, A-11
- compression des données, 5-27
 - coûts des performances de , 5-29
 - fragments, 5-20
- comptabilité
 - AIX pour administrateurs système BSD, A-7
 - collecte de données, généralités, 11-3
 - commandes
 - exécution automatique, 11-7
 - généralités, 11-6
 - données d'utilisation de l'imprimante, 11-4, 11-5
 - données de l'utilisation du disque, 11-4
 - rapport, 11-5
 - données de processus
 - collecte, 11-3
 - rapport, 11-5
 - durée de connexion
 - collecte, 11-3
 - rapport, 11-5
 - fichiers
 - fichiers de commande runacct, 11-9
 - fichiers de données, 11-9
 - fichiers de rapport et de synthèse, 11-9
 - formats, 11-11
 - généralités, 11-8
 - généralités, 11-2
 - rapport de données, généralités, 11-4
 - rapports
 - mensuels, 11-6
 - quotidiens, 11-6
 - taxation
 - rapport, 11-5
 - taxation, 11-4
- comptabilité système
 - collecte de données, généralités, 11-3
 - commandes, exécution automatique, 11-7

- données d'utilisation de l'imprimante
 - collecte, 11-4
 - rapport, 11-5
- données de l'utilisation du disque, 11-5
 - collecte, 11-4
- données de processus
 - collecte, 11-3
 - rapport, 11-5
- durée de connexion, 11-3, 11-5
- fichiers
 - fichiers de commande runacct, 11-9
 - fichiers de données, 11-9
 - fichiers de rapport et de synthèse, 11-9
 - formats, 11-11
 - généralités, 11-8
- généralités, 11-2
- rapport de données, généralités, 11-4
- rapports
 - mensuels, 11-6
 - quotidiens, 11-6
- taxation
 - rapport, 11-5
 - taxation, 11-4
- Contrôleur de ressources système
 - commandes, liste, 10-3
 - fonctions, 10-2
 - illustration, 10-3
- cron, AIX pour administrateurs système BSD, A-15

D

- démon cron, collecte de données, 11-2
- Désallocation dynamique de processeur, 7-5
- disponibilité
 - face aux incidents de carte
 - ou d'alimentation, 3-10
 - face aux incidents de disque, 3-10
- durée de connexion, 11-3

E

- E/S à plusieurs chemins, 14-11
- environnement de système, 7-5
 - Désallocation dynamique de processeur, 7-5
- environnement système
 - profil, 7-2
 - services de manipulation des données de l'heure, 7-4
- environnements shell, personnalisation, 7-2
- environnements utilisateur, personnalisation, 7-2
- espace d'échange, voir espace de pagination, 4-1
- espace de pagination, AIX pour administrateurs système BSD, A-21

F

- fermeture, description, 2-8
- fichier .profile, 7-2
- fichier /etc/profile, 7-2
- fichiers
 - montage, 5-14
 - pour administrateurs système BSD, A-17, A-20
- fichiers de connexion
 - fichier .profile, 7-2
 - fichier /etc/profile, 7-2
- fichiers mappe, 3-18

- fragments
 - coûts des performances de , 5-24
 - effet sur l'utilisation du disque, 5-19
 - effet sur la sauvegarde et la restauration, 5-30
 - et un nombre variable d'i-nodes, 5-20
 - limitation pour pilote d'unité, limitation pour pilotes d'unité, 5-31
 - taille de
 - définition, 5-22
 - identification , 5-22

G

- gestion des connexions à chaud PCI, PCI, 14-9
- groupe de sous-systèmes, description, 10-2
- groupes de volumes
 - création de groupes de volumes distincts, 3-9
 - définition de, 3-3
 - haute disponibilité, 3-9
 - mise en œuvre d'une stratégie, 3-22
 - procédure de mise en service , 3-7
 - quorums, 3-7
 - sans quorum, 3-8
 - stratégie, 3-9
- groupes de volumes sans quorum , 3-8

I

- i-nodes, 5-22
 - et fragments, 5-20
 - nombre d'octets par i-node (NBPI)
 - définition, 5-22
 - identification , 5-22
 - nombre variable de, 5-22
- i-nodes, nombre de, 5-24
- idbgen, 7-4
- imprimante, pour les gestionnaires de systèmes BSD, A-30
- imprimantes, codes d'emplacement, 14-6

J

- JFS (Journaled File System)
 - avec un nombre variable d'i-nodes, 5-20
 - compression des données, 5-27
 - fragments, 5-20
 - limitations de taille, 5-23
 - taille maximum de, 5-24
- JFS2 (Enhanced Journaled File System),
 - limitations de taille, 5-23, 5-25
- Journal JFS (Journaled File System), taille de, 5-24

L

- LVM, 3-1
- LVM (Logical Volume Manager), 3-1
 - définition, 3-6

M

- montage
 - distant, définition, 5-14
 - généralités, 5-13
 - local, définition, 5-14
 - montage de station de travail sans disque
 - description de, 5-16

- sécurité, 5-15
- montage des systèmes de fichiers, 5-14
- montages automatiques, 5-14
- montages automatiques
 - de /etc/filesystem, 5-14
- utilisation de plusieurs montages, 5-14
- mots de passe, pour administrateurs système BSD, A-26
- MPIO, 14-11
- MWC (cohérence écrit-miroir), 3-13

N

- NBPI, 5-22
- NFS et NIS, pour administrateurs système BSD, A-25
- NIS, A-25
- nombre d'octets par i-node (NBPI), 5-22
 - allocation, 4-4
 - caractéristiques de création, 4-8
 - commandes de gestion, 4-7
 - généralités, 4-1, 4-4
 - mode d'allocation Early, 4-4
 - mode d'allocation Late, 4-4
- nombre variable d'i-nodes, 5-22 et fragments, 5-20

P

- partitions logiques
 - définition, 3-5
 - règles d'affectation inter-disque, 3-15
- partitions physiques
 - définition, 3-4
 - taille, 3-4
- performances, Gestionnaires de systèmes BSD, A-29
- pilotes d'unités, effet sur l'utilisation de fragments sur la taille de, 5-31
- point de montage, 5-13
- port multiprotocole, codes d'emplacement, 14-8
- procédure de mise en service, 3-7
- processus
 - collecte de données comptables, 11-3
 - génération de rapports comptables, 11-5
 - gestion, 8-1
- profil
 - fichiers, 7-2
 - généralités, 7-2

Q

- quorums
 - définition, 3-7
 - groupes de volumes sans quorum, 3-8

R

- range (option), 3-15
- règles d'affectation inter-disque, 3-14
- règles d'affectation intra-disque, 3-17
- règles de contrôle de l'écriture, 3-19
- règles de programmation des écritures, 3-12
- répartition, 3-19
- répertoire /export, 5-9
- répertoire /usr/share, 5-7

- Répertoire SOPT (Shared Product Object Tree), 5-9
- répertoire SPOT, 5-9
- répertoires, montage, 5-14
- réseau, pour administrateurs système BSD, A-22
- restauration, effet des fragments sur, 5-30

S

- sauvegarde
 - AIX pour administrateurs système BSD, A-9
 - commandes, liste de, 6-2
 - effet des fragments sur, 5-30
 - fichiers utilisateur, 6-8
 - généralités, 6-2
 - groupe de volumes défini par l'utilisateur, image système, 6-9
 - méthodes, 6-2
 - procédure pour les données système et utilisateur, 6-6
 - reproduction d'un système (clonage), 6-7
 - restauration des données, 6-4
 - stratégie de gestion
 - développement, 6-5
 - planification, 6-5
 - politique, 6-3
 - systèmes de fichiers utilisateur, 6-8
 - types de supports, 6-4
 - unités, illustration, 6-4
- services de manipulation des données de l'heure, 7-4
- SMIT (System Management Interface Tool), 13-1
- sous-serveur, description, 10-3
- sous-système, propriétés, 10-2
- stations de travail sans disque, sécurité du montage, 5-15
- stockage de volume logique
 - définition, 3-2
 - groupes de volumes, 3-3
 - groupes de volumes sans quorum, 3-8
 - partitions logiques, 3-5
 - quorums, 3-7
 - systèmes de fichiers, 3-5
 - tailles maximales, 3-5
 - volumes logiques, 3-4
 - volumes physiques, 3-2
- stockage sur volume logique
 - règles d'affectation inter-disque, 3-14
 - règles d'affectation intra-disque, 3-17
 - règles de programmation des écritures, 3-12, 3-13
- Strict (option), 3-16
- système, démarrage, 2-2
- système de comptabilité, commandes, depuis le clavier, 11-7
- système de fichiers, images, 5-30
- système de fichiers / (racine), 5-2
- système de fichiers /var, 5-8
- systèmes de fichiers
 - arborescence
 - généralités, 5-2
 - répertoire /export, 5-9
 - répertoire /usr/share, 5-7

- système de fichiers / (racine), 5-2
 - système de fichiers /usr, 5-5
 - système de fichiers /var, 5-8
- commandes de gestion, 5-11, 5-12
- compression des données, 5-27
- fichiers sporadiques, 5-25
- fichiers volumineux, 5-26
- fragments, 5-20
- généralités, 5-1
- i-nodes, 5-20
- montage, 5-14
- pour administrateurs système BSD, A-19
- sauvegarde des systèmes de fichiers
 - utilisateur, 6-8
- tâches de gestion, 5-11
- techniques de journalisation, 5-1
- types
 - CD-ROM, 5-18
 - DVD-ROM, 5-18
 - JFS (Journaled File System), 5-18
 - JFS2 (Enhanced Journaled File System), 5-18
 - NFS (Network File System), 5-18
- systèmes de fichiers activés, espace libre, 5-26
- systèmes de fichiers compatibles
 - allocations de fichier vide, 5-27
 - créer, 5-26
 - géométrie des fichiers volumineux, 5-26

T

- taille de groupe d'allocation, 5-24
- taxation, 11-4
- terminaux, pour administrateurs
 - système BSD, A-32
- traitement de l'amorçage, phases, 2-4
- tty (teletypewriter), codes d'emplacement, 14-6

U

- unité
 - AIX pour administrateurs système BSD, A-16
 - classes, 14-3
 - états, 14-4
 - noeuds, 14-2
- unité de disquette, codes d'emplacement, 14-8

- unités, 14-1
 - codes d'emplacement, 14-5
 - configuration d'un grand nombre d'unités, 14-1
- unités de bande
 - attributs, modifiable, 15-2, 15-4, 15-5, 15-6, 15-7, 15-8, 15-9, 15-10, 15-11, 15-12
 - fichiers spéciaux, 15-14
 - gestion, 15-1
- unités de disque, disque série, codes d'emplacement, 14-7
- unités de disques (disques durs), connecté directement, 14-7
- unités SCSI, codes d'emplacement, 14-7
- utilisation de l'imprimante, 11-4
- utilisation disque, 11-4
- utilisation du disque, effet des fragments sur , 5-19
- UUCP, AIX pour administrateurs système BSD, A-33

V

- VGDA (Volume Group Descriptor Area), 3-7
- VGSA (Volume Group Status Area), 3-7
- Virtual Memory Manager, 4-2
- VMM, 4-2
- VMM (Virtual Memory Manager), généralités, 4-1
- volumes logiques
 - définition, 3-4
 - fichiers mappe, 3-18
 - règles de contrôle de l'écriture, 3-19
 - répartis, 3-19
 - stratégie, 3-11
 - stratégie de groupe de volumes, 3-22
 - zones actives, 3-20
- volumes physiques, définition, 3-2

W

- Web-based System Manager, 12-1

Y

- Yellow Pages, A-25
 - pour administrateurs système BSD, A-25

Z

- zones actives dans les volumes logiques, 3-20

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull Concepts de gestion de système Système d'exploitation et unités

N° Référence / Reference N° : 86 F2 28EF 03

Date / Dated : Juin 2003

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.

Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.

If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

**BULL CEDOC
357 AVENUE PATTON
BP 20845
49008 ANGERS CEDEX 01
FRANCE**

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:

Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL CEDOC
ATTN / Mr. L. CHERUBIN
357 AVENUE PATTON
B.P. 20845
49008 ANGERS CEDEX 01
FRANCE

Phone / Téléphone : +33 (0) 2 41 73 63 96
FAX / Fax +33 (0) 2 41 73 60 19
E-Mail / Courrier Electronique: srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:

<http://www.logistics.bull.net/cedoc>

<http://www-frec.bull.com> <http://www.bull.com>

CEDOC Reference # N° Référence CEDOC	Qty Qté	CEDOC Reference # N° Référence CEDOC	Qty Qté	CEDOC Reference # N° Référence CEDOC	Qty Qté
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	

[__]: **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

PHONE / TELEPHONE : _____ FAX : _____

E-MAIL : _____

For Bull Subsidiaries / Pour les Filiales Bull :

Identification: _____

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :

Customer Code / Code Client : _____

For Bull Internal Customers / Pour les Clients Internes Bull :

Budgetary Section / Section Budgétaire : _____

For Others / Pour les Autres :

Please ask your Bull representative. / Merci de demander à votre contact Bull.

BULL CEDOC
357 AVENUE PATTON
BP 20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 F2 28EF 03

PLACE BAR CODE IN LOWER
LEFT CORNER



Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

