Bull NovaScale

HPC Linux
Guide d’installation

Linux

ORDER REFERENCE
86 F2 31EG 02

Bull NovaScale

HPC Linux
Guide d’installation

Linux

Logiciel

Juillet 2003

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 F2 31EG 02

Imprimé in 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 rspectifs.

Linux est une marque déeposée Linus Torvalds.

Table des matières

Préface

But du document

Le but de ce document est de montrer comment construire un environnement Linux HPC de logiciels Open Source ou propriétaires testés par Bull pour les matériels Bull d'architecture Itanium-2™

Organisation du document

Après cette introduction,

- Le chapitre 1 décrit les notions de base sur HPC dans un environnement LINUX et de logiciels libres.
- Le chapitre 2 décrit les composants logiciels sélectionnés par Bull.
- Le chapitre 3 décrit l'installation d'un cluster.
- Le chapitre 4 décrit l'installation de composants logiciels à partir du CD fourni par Bull ou à partir des sites des fournisseurs sur NovaScale 4040, NovaScale 5080 et NovaScale 5160.
- Le chapitre 5 décrit la désinstallation des logiciels Intel soumis à licences.
- Le chapitre 6 via une FAQ regroupe quelques questions et leurs réponses.

Chapitre 1. Notions générales sur Linux HPC

1.1 HPC (High Performance Computing)

1.1.1 Définition de HPC

C'est ce qu'on appelle en français le ***calcul de haute performance***.
Ce terme définit les grosses applications scientifiques nécessitant une grande puissance de calcul, une grande précision des résultats et pouvant utiliser de très grandes quantités de données.

1.1.2 Utilisation du HPC

Le calcul de haute performance est utilisé dans différents domaines :

- La recherche (dynamique moléculaire, mécanique des fluides…)
- L'industrie scientifique (automobile, nucléaire, météo …)
- L'imagerie de synthèse (effets spéciaux…)
- Le data mining (exploitation statistique de grosses bases de données type Data Warehouse…)

1.2 LINUX

1.2.1 Définition

Linux est un système d'exploitation de type UNIX, multi-tâches et multi-utilisateurs, disponible sur de nombreuses architectures matérielles, en particulier les machines à base de processeurs ix86 et Itanium™. Il intègre la plupart des technologies les plus récentes (SMP, clustering, RAID).

La principale singularité de Linux est d'être un logiciel libre, développé de façon collaborative et pour une grande part bénévole par des milliers de programmeurs répartis dans le monde.

Linux est un noyau. Pour l'utiliser, il faut des applications, c'est ce que proposent les distributions. Une distribution est un ensemble de programmes plus un noyau à installer sur une machine. Parmi les distributions Linux, on peut citer RedHat, Mandrake, Suse, TurboLinux .

1.2.2 La petite histoire…

Linux est un système d'exploitation conçu par Linus Torvalds, un étudiant finlandais. Le commencement de l'écriture du noyau eu lieu en 1991. La règle de base qui a prévalu tout au long de sa réalisation voulait que la licence soit ouverte. Toutes les parties de ce système d'exploitation ont ainsi été réécrites et améliorées au fil du temps (aujourd'hui il en est à sa version 2.4.20).

Le noyau Linux est diffusé sous forme de « distributions », contenant le noyau ainsi que des programmes souvent accompagnées d'un outil « maison » facilitant l'installation. Elles sont mises à disposition sur des sites FTP et des CD commercialisés. Linux bénéficie à ce titre énormément des logiciels libres mis au point dans le cadre de divers autres projets, en particulier de GNU.

1.2.3 LINUX et HPC

1.2.3.1 Avantages de Linux

- Multi-plateforme : le noyau Linux fonctionne sur PC, Itanium, PowerPC …
- Fiable, robuste
- De nombreux logiciels (y compris le noyau) sont en Open Source et à ce titre téléchargeables gratuitement sur Internet. En outre, cela permet qu'un maximum de personnes travaille sur ces logiciels et accélère les temps de développement.
- Très largement déployé : Linux compte de plus en plus d'adeptes, non seulement dans les Centres de Recherche et les Universités, mais aussi dans l'industrie. La synergie qui existe derrière Linux est immense et explique la qualité et son émergence très rapide.

1.2.3.2 Open Source (Logiciel libre)

Depuis Linus Torvalds et son système Linux, l'Open Source s'est considérablement développé.

Le logiciel Libre, en tant qu'idée politique, a été popularisé par Richard Stallman depuis 1984, année où il a créé la Free Software Foundation (FSF) et son projet GNU. Il a mis sur pied un ensemble de droits dont il estimait que tout utilisateur devait pouvoir jouir, et les a codifiés au sein de la licence publique générale de GNU, ou GPL.
Le code source n'est donc plus la possession privée d'une personne, d'un groupe de personnes, ou d'une société, comme c'était le cas depuis la naissance de l'informatique dans les années 60, jusque dans les années 80/90. Les plus

grandes entreprises emboîtent actuellement le pas des développeurs indépendants et proposent à leur tour des logiciels de qualité professionnelle en Open Source.

Voici quelques uns des principes de l'Open Source et de la licence GPL :

- Libre utilisation
- Libre redistribution
- Le code source doit être à disposition de tout le monde
- La licence doit autoriser les modifications et les travaux dérivés
- Intégrité du code source de l'auteur : diffusion de patchs
- Le redistributeur doit accorder la même licence aux destinataires

Il existe d'autres licences comme la licence BSD. A la différence de la licence GPL , la licence BSD pourra permettre (si indiqué dans le source) la non diffusion des modifications.

Pour plus d'informations consulter les pages :
http://www.gnu.org
http://www.opensource.org
http://www.freebsd.org

1.3 Cluster HPC

1.3.1 Qu'est-ce qu'un cluster ?

Un cluster est un ensemble de machines (appelée nœuds) ***connectées entre elles dans le but de remplir une fonction.*** Chaque nœud peut être monoprocesseur ou multiprocesseur SMP (multitraitement symétrique), les processeurs se partageant dans ce cas la mémoire et les disques.

1.3.2 Les grands types de cluster

Cluster scientifique (HPC) : Le calcul est divisé en plusieurs tâches qui seront effectuées en parallèle sur les différentes machines du cluster. Ces clusters sont surtout utilisés par le monde scientifique, graphique.

Cluster haute-disponibilité (HA) : Il est créé afin d'éviter une interruption de service en cas de dysfonctionnement matériel ou logiciel d'une machine. Par exemple, si un nœud serveur n'est plus apte à assurer ses fonctions, il sera automatiquement remplacé par une autre machine du cluster.

Le load balancing ou répartition de charge est souvent cité lorsque l'on parle de clustering. Il a pour but de ***distribuer l'exécution de processus*** systèmes ou réseau ***sur les différents nœuds***. Ainsi, lorsque le nœud répartiteur de charge reçoit un processus, il regarde la charge et la spécialisation de chacun des nœuds et il affecte ainsi le processus au plus approprié. Ce principe est utilisé dans le domaine des réseaux et plus particulièrement sur celui des services comme les serveurs WEB ou FTP.

1.3.3 L'architecture d'un cluster HPC

Un système Cluster HPC est :

- Un ensemble de machines indépendantes et similaires en terme d'architecture et vitesse. Une des machines (le nœud serveur) répartit les tâches entre toutes les autres (nœuds clients) qui lui renvoient le résultat une fois les calculs terminés.

- Cet ensemble disposera d'un réseau dédié rapide de communication interne pour les applications parallèles(MPI) et d'un autre réseau TCPIP pour le management faisant apparaître l'ensemble comme une seule entité vue de l'extérieur. Cette notion s'oppose à celle de réseau de stations où chaque nœud dispose d'un écran, clavier .

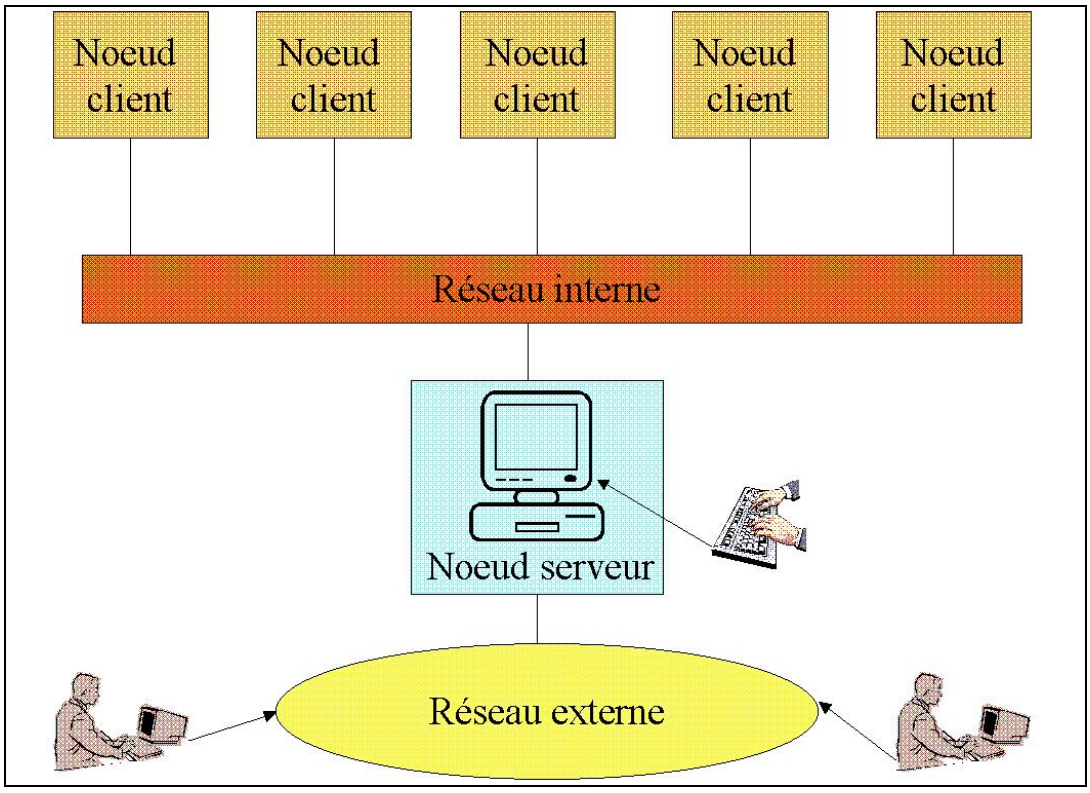


Schéma d'un cluster classique

Le nœud serveur contrôle l'ensemble du cluster. Il est aussi la console du cluster et la passerelle (gateway) vers le monde extérieur. Un grand cluster peut avoir plus d'un nœud serveur et des nœuds clients dédiés à des tâches bien spécifiques (ex : calcul, serveur de fichiers).

1.4 Linux HPC et Bull

Linux aujourd'hui est un système d'exploitation capable de supporter les exigences des environnements HPC demandées par les applications scientifiques en terme de nombre de CPUs, bande passante IO et de support pour la programmation parallèle. C'est pourquoi Bull le propose sur ces plates-formes SMP Itanium-2 4 voies (NovaScale 4040), 8 voies (NovaScale 5080) et 16 voies (NovaScale 5160) ainsi que sur des configurations « cluster » interconnectant des NovaScale 4040 pour accroître la capacité CPU.

Chapitre 2. Description de l'environnement logiciel Linux HPC

2.1 Introduction

Dans ce chapître est décrit un ensemble de logiciels constituant un environnement logiciel HPC pouvant être utilisé sur machines Bull.
Sauf les logiciels dont le nom est souligné (qui sont des logiciels spécifiques clusters), l'ensemble des logiciels cités ci-dessous peut être utilisé sur une machine seule ou sur un cluster.

2.2 Composants logiciels HPC

			Open Source	Produit commercialisé
DEVELOPPEMENT D'APPLICATIONS	**Bibliothèques scientifiques**		Blas, pblas, blacs, lapack, scalapack, fftw, atlas	Libmkl (Intel), NAG
	Librairies parallèles	Mpi	Mpich, Lam-mpi	ScaMpi
		Autres	PVM	OpenMP
	Compilateurs (C, C++, Fortran)		GNU	Intel, NAG
OUTILS	**Exploitation**	Répartition de tâches	OpenPBS, Maui,	LSF, PBSPRO ScaOPBS
		Debogueurs	Gdb, dbx	Totalview (Etnus) Idb (Intel)
		Profiling	Cprof/vprof	Vampir (Pallas)
		Analyses de performances	Papi, pfmon, perfometer	Vtune (Intel)
	Administration Cluster	Déploiement / Installation	Ka-Tools, C3, SystemImager	
		Shell distribué	Ka-Tools, gexec	
		Contrôle et monitoring	Ganglia, top	
OS	**Administration Système**		Webmin, Nagios	
	Système de fichiers		*Distribués :* NFS, PVFS *Locaux:* EXT2, EXT3, Reiserfs, xfs	
	Système d'exploitation		Linux	

Les composants logiciels d'un cluster HPC

2.2.1 Les compilateurs

Les compilateurs jouent un rôle essentiel pour exploiter le potentiel des processeurs Itanium 2. En effet, ces derniers utilisent l'architecture EPIC (Explicit Parallel Instruction set Computing) qui permet l'exécution de plusieurs instructions en parallèle. Le parallélisme doit donc être détecté et exploité au niveau du compilateur. Pour ces raisons, Bull propose les compilateurs (C/C++ et Fortran) d'Intel dont les équipes disposent de toutes les compétences et les connaissances sur l'architecture pour fournir des produits performants. Une version des compilateurs de GNU est également disponible pour les utilisateurs familiers de ces logiciels.

2.2.1.1 Compilateur Intel C/C++

La version courante du compilateur C/C++ d'Intel est la version 7.1.
Les principales caractéristiques de ce compilateur sont les suivantes :

- Optimisation du débit des instructions flottantes
- Optimisation des appels interprocéduraux
- Pré-chargement de données
- Prédiction pour les instructions conditionnelles
- Chargement spéculatif
- Optimisation du pipeline logiciel

Ce compilateur est conforme au standard Ansi C/C++ et au standard ISO C/C++. Il offre également une compatibilité avec les produits GNU. Ainsi un code source ou objet GNU C peut être compilé avec le compilateur Intel. Les outils emacs et gbd peuvent également être utilisés avec ce compilateur.

Le compilateur supporte les fonctionnalités multithreading :

- OpenMP 2.0 pour C/C++ est supporté. Le compilateur accepte les pragmas OpenMP et génère une application multithreadée.
- Parallelisation automatique : une option du compilateur permet de détecter le parallélisme (notamment au niveau des boucles de calcul) et de générer une application multithreadée.

2.2.1.2 Compilateur Intel Fortran

La version du compilateur Fortran95 d'Intel est actuellement la version 7.1. Les principales caractéristiques de ce compilateur (identiques au compilateur C) sont les suivantes :

- Optimisation du débit des instructions flottant
- Optimisation des appels interprocéduraux
- Pré-chargement de données
- Prédiction pour les instructions conditionnelles
- Chargement spéculatif

- Optimisation du pipeline logiciel

Ce compilateur est conforme au standard ISO Fortran 95. Il offre également une compatibilité avec les produits GNU : les outils emacs et gbd peuvent être utilisés avec ce compilateur. Il supporte aussi les fichiers codés en big endian. Enfin, ce compilateur permet le développement d'applications mixant des programmes écrits dans les langages C et Fortran.
Le compilateur supporte les fonctionnalités multithreading :

- OpenMP 2.0 pour Fortran est supporté. Le compilateur accepte les pragmas OpenMP et génère une application multithreadée.
- Parallèlisation automatique : une option du compilateur permet de détecter le parallélisme (notamment au niveau des boucles de calcul) et de générer une application multithreadée.

2.2.1.3 Compilateurs GNU

Gcc, la collection de compilateurs libres pouvant compiler du C/C++ et Fortran fait partie de la distribution LINUX installée.

2.2.2 Les bibliothèques scientifiques

Ce sont des ensembles de fonctions testées, validées et optimisées. Cela permet aux développeurs de ne pas avoir à réinventer ces sous-programmes à chaque fois.

Avantages des ces bibliothèques scientifiques :

- Portabilité
- Supportent différents types de données (réel, complexe, double précision…)
- Prennent en compte différents types de stockage (matrice bande, symétrique …)

Plusieurs bibliothèques existent et sont reconnues dans le monde scientifique. Ce sont par exemple les librairies BLAS, LAPACK…

BLAS = Basic Linear Algebra Subprograms

Il s'agit d'une librairie de base en algèbre linéaire (opérations impliquant des matrices et des vecteurs). Les fonctions sont séparées en 3 parties :

- Routines de niveau 1 pour la représentation des vecteurs et les opérations vecteur/vecteur
- Routines de niveau 2 pour la représentation des matrices et les opérations matrice/vecteur

Routines de niveau 3 principalement pour les opérations matrice/matrice

BLACS = Basic Linear Algebra Communication Subprograms

Blacs est une bibliothèque de communications spécialisées (par passage de messages). Elle permet, après avoir défini une grille de processus, d'échanger des vecteurs ou matrices, des blocs... Elle peut être compilée au dessus de MPI ou de PVM.

PBLAS = Parallel Basic Linear Algebra Subprograms.

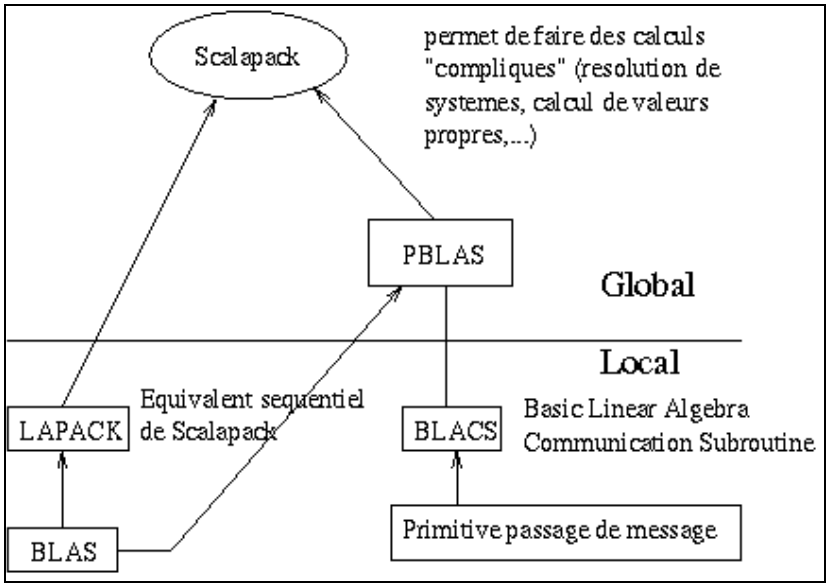
Pblas est la version parallélisée des blas pour les machines à mémoire distribuée. Elle a besoin de la distribution cyclique par bloc des matrices que propose la bibliothèque Blacs.

LAPACK = Linear Algebra PACKage

C'est un ensemble de routines fortran 77 servant à résoudre les problèmes d'algèbre linéaire, tel que la résolution de systèmes linéaires, calcul de valeur propre, calcul matriciel … Cependant elle n'est pas écrite pour une architecture parallèle.

SCALAPACK = SCAlable Linear Algebra PACKage

Cette librairie est la version scalable de LAPACK. Ces 2 librairies utilisent le partionnement par blocs pour minimiser les échanges de données entre les différents niveaux de la mémoire. Scalapack est surtout utilisée pour les problèmes de recherche de valeurs propres, les factorisations (LU, Cholesky et QR). Les matrices sont distribuées avec l'aide de BLACS.



Interdépendance entre les différentes librairies mathématiques standards

Les routines des composantes locales sont appelées par un seul processus avec des arguments résidant en mémoire locale.
Les routines des composantes globales sont synchrones et parallèles. Elles sont appelées avec des arguments qui sont des matrices ou des vecteurs distribués sur l'ensemble des processus.

Libmkl :

Cette librairie optimisée par Intel pour ses processeurs possède les librairies suivantes: blas, lapack et fft.

2.2.3 Les applications parallèles

Il existe 2 types d'architectures parallèles. La première, dite à mémoire distribuée, considère que chaque processeur possède son propre espace de mémoire vive non accessible par les autres processeurs. La deuxième dite à mémoire partagée, permet aux processeurs d'accéder simultanément, et de façon transparente, à un même espace de mémoire vive.

Pour architecture parallèle à mémoire distribuée

Pour ce type d'architecture, ***l'échange de messages*** est le concept clé. L'échange de messages (ou message passing) repose sur une idée simple et logique : un processus envoie un message à un ou plusieurs autres processus qui doivent le recevoir.

Un message doit contenir les informations suivantes :
- L'identificateur du processus émetteur
- L'identificateur du processus récepteur
- Le type de la donnée
- Sa longueur
- Les données

Un message doit être envoyé à une adresse bien déterminée et le processus récepteur doit pouvoir classer et interpréter ce message. Ceci est géré par un environnement tel que PVM ou MPI. Une application PVM ou MPI est un ensemble de processus autonomes exécutant chacun leur propre code et communiquant grâce à des fonctions de la bibliothèque fournie par chacun de ces environnements.

MPI = Message Passing Interface

MPI repose sur le concept d'échange de messages. C'est une tentative de normalisation des bibliothèques de communication par passage de messages. C'est un standard qui possède à l'heure actuelle deux versions de ses spécifications : MPI-1 et MPI-2. La seconde version ajoute des directives pour les entrées/sorties parallèles (MPI-IO), l'interface C++ et le débogage parallèle. MPI propose des interfaces de développement pour les langages C, C++ et Fortran.

Il existe plusieurs implémentations de philosophies très différentes, les deux principales étant Lam-MPI et MPICH.

- LAM-MPI :
  www.lam-mpi.org

  La bibliothèque est développée par l'Université de Notre Dame (Indiana). Cette implémentation se base sur un système de démons qui tournent sur chaque nœud et assurent l'envoi et la réception des messages. De plus, il est possible de lancer différents processus qui communiquent entre eux. L'exemple type est un processus « maître » avec plusieurs processus « esclaves », ils sont tous différents, communiquent via MPI, et sont lancés sur plusieurs nœuds différents.

- MPICH
  www-unix.mcs.anl.gov/mpi/mpich

  MPI-Chameleon est développée conjointement par l'Université du Mississipi et le laboratoire Argonne. Comme LAM-MPI, il s'agit d'une implémentation de MPI.

  MPICH est basée sur des "devices" conditionnant les échanges de messages. Ils ont un impact direct sur les performances. Plusieurs devices sont disponibles, et chacun a sa propre particularité. Les deux principaux sont :

  - ch_p4 : permet une liaison inter-machines (inter-noeuds) par utilisation du protocole TCP/IP..
  - ch_shmem : permet une liaison rapide entre tâches s'exécutant sur une même machine par utilisation de la mémoire partagée pour effectuer les échanges de messages.

PVM = Parallel Virtual Machine
www.csm.ornl.gov/pvm/pvm_home.html
PVM est un logiciel, développé à Oak Ridge National Laboratory, qui permet d'utiliser un ensemble de stations de travail Unix reliées par un réseau comme une machine parallèle.

PVM permet d'utiliser un ensemble de stations de travail Unix reliées par un réseau comme une machine parallèle. Cet ensemble de stations constitue une machine parallèle virtuelle. L'ensemble de machines peut être géographiquement distribué. Les machines peuvent être hétérogènes, c'est-à-dire, composées de processeurs différents, fonctionnant sous des systèmes d'exploitation éventuellement différents. Une fois installé, PVM permet de faire communiquer entres elles ces différentes machines au moyen de messages que s'échangent les processus.

PVM est constitué de 2 éléments :

1) Une interface de programmation disponible pour les langages C et Fortran. Cette bibliothèque dispose des primitives nécessaires à la génération de processus s'exécutant en parallèle sur plusieurs machines.

2) Le démon de la machine virtuelle PVM est un processus tournant sur chaque nœud du cluster. Les démons s'échangent des messages entre eux avec les instructions fournies par les primitives. Ces messages permettent de « charger » un programme sur les différents nœuds du cluster, de passer des paramètres, de collecter des résultats…

Les caractéristiques de PVM sont les suivantes :
C'est un logiciel du domaine public
Il est facile à installer.
Il est souple d'utilisation :

- Il tourne sur un grand nombre d'architectures de machines différentes
- Il tourne sur des réseaux locaux ou non (LAN, WAN ou combinaison des deux)
- C'est l' application de l'utilisateur qui décide où et quand les composants vont être exécutés et qui détermine les contrôles et les dépendances.
- Il permet de programmer dans différents langages (essentiellement C et Fortran)
- La machine virtuelle est facile à définir et à modifier

Pour architecture parallèle à mémoire partagée

OpenMP = Open Multi Processing

www.openmp.org

Défini par un groupe des plus importants fabricants de logiciels et matériels informatiques, OpenMP est une " interface de programmation " portable qui permet de faciliter le développement des applications parallèles pour machines à mémoire partagée.

L'approche de la parallélisation OpenMP, en mémoire partagée, est très différente de celle utilisée par MPI en mémoire distribuée. Il n'y a qu'une seule instance du programme, exécutée en parallèle sur plusieurs processeurs. Des directives insérées dans le programme permettent de gérer la distribution des calculs entre les processeurs.

Ce standard définit des directives pour les compilateurs Fortran, C et C++, une bibliothèque de " routines " et des variables d'environnement.
Les directives étendent les possibilités de programmation séquentielle des compilateurs en permettant de définir les régions parallèles du programme, comment se fait le partage des tâches, les points de synchronisation et permettent de gérer le partage (ou pas) des données.
La bibliothèque de " routines " et les variables d'environnement permettent, elles, de gérer l'environnement du programme lors de son exécution.

Les caractéristiques de ce standard sont les suivantes :

- Scalabilité
- Simplicité d'utilisation : parallélisation incrémentale
- Portabilité au SMP
- Parallélisation de haut niveau
- Flexibilité pour exprimer différents types de parallélisme
- Orienté performances

2.2.4 Les systèmes de fichiers

Il existe 2 types de systèmes de fichiers : les systèmes de fichiers locaux et les systèmes de fichiers distribués. Les premiers sont utilisés localement sur une machine afin de permettre l'accès aux fichiers contenus sur le disque. Les seconds sont nécessaires pour accéder à des fichiers distants via le réseau. Ils ont en plus des systèmes de verrouillage pour maintenir une cohérence entre les différents utilisateurs. Par exemple, si plusieurs personnes travaillent sur le même fichier, ce type de file system va permettre de gérer ce travail « simultané ».

Il y a aussi au-dessus de ces deux types de systèmes de fichiers, les systèmes de fichiers parallèlles qui répartissent les fichiers sur plusieurs disques parallèlisant ainsi l'accès à différentes parties du fichier.

Les systèmes de fichiers locaux

On peut séparer ces files systems en 2 groupes : les journalisés et les non journalisés.

Non journalisés

On peut dire que dans ce cas, on travaille « sans filet ». C'est à dire que s'il y a une panne brutale, il faut tout vérifier avant de pouvoir se resservir de la machine.
C'est par exemple le cas d'Ext2, un système de fichiers standard de Linux.

Journalisés

La plupart des systèmes de fichiers modernes utilisent des techniques de journalisation empruntées au monde des bases de données pour améliorer la reprise sur incident. Les transactions sont écrites séquentiellement sur une partie du disque appelée journal avant de les écrire sur disque à leur place définitive dans le système de fichiers.
Maintenant, si une panne survient, une cohérence est ainsi maintenue et la vérification est très rapide.

- ReiserFS
  www.namesys.com
  C'est un système qui possède plusieurs caractéristiques : il permet des arrêts fréquents du système et de gérer les fichiers par b-tree ce qui est très utile quand il y a beaucoup de petits fichiers, de plus, il optimise le stockage des petits fichiers.

- XFS - eXtended File System
  oss.sgi.com/projects/xfs
  XFS, largement reconnu comme un système de fichiers 64 bits très haute performance, permet un redémarrage rapide après un crash ainsi que la capacité de gérer des systèmes de fichiers extrêmement larges.

- Ext3
  Ce file system correspond à la version journalisée de ext2.

Les systèmes de fichiers distribués

Le calcul scientifique nécessite assez souvent d'avoir accès à des fichiers de données importants. Dès lors, le temps moyen d'accès au fichier va beaucoup influencer la vitesse d'exécution du programme. L'idéal serait d'avoir un programme s'exécutant sur une machine unique doté d'un disque dur à haute performance. Cependant le principe même des clusters est de répartir les calculs sur l'ensemble des nœuds d'un réseau. L'accès à des systèmes de fichiers distants devient incontournable. C'est à ce niveau qu'interviennent les systèmes de répartition de fichiers distribué.

NFS - Network File System

Système fonctionnant sur le mode client/serveur autorisant les accès à un répertoire à plusieurs clients distribués sur un réseau. Il permet donc le partage des données entre les nœuds du cluster. Les limites sont celles du serveur en ce qui concerne la taille du répertoire et des fichiers, et du réseau pour les vitesses d'accès.

Les systèmes de fichiers parallèles

PVFS - Parallel Virtual File System

PVFS offre un système de fichiers réparti au dessus d'un système de fichiers standard tel que ext2 ou ReiserFS. Le principe de PVFS est assez simple à appréhender: au lieu de stocker un fichier sur un seul disque, forçant les machines distantes à faire des requêtes séquentielles sur le réseau, on stocke plutôt le fichier sur plusieurs disques du réseau.

Sur Linux, « The Parallel Virtual File System » est une implémentation du standard PFS (Parallel File System).
Il fournit une haute performance et un système de fichiers parallèle pour des machines en cluster avec:
un domaine des noms uniques
un accès transparent

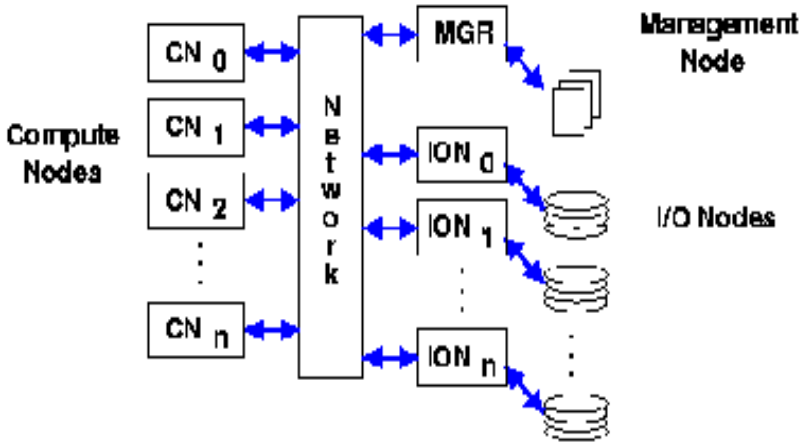


Schéma montrant les Compute Nodes et les Storage Nodes

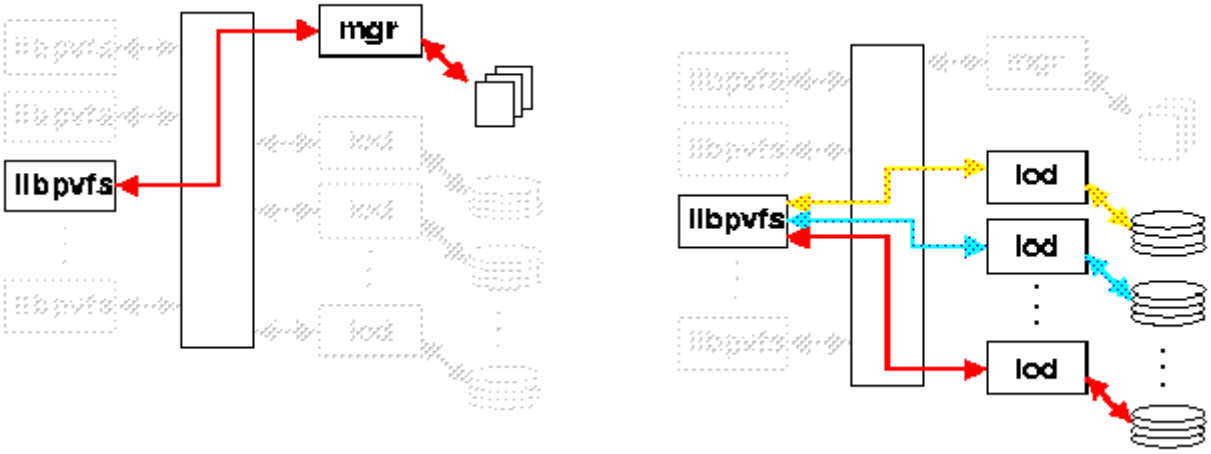


Schéma montrant les accès pour les méta data de localisation des données, et les flux de data.
 Il montre le processus d'initialisation des transferts, suivi des transferts sans interférence avec le manager. Ce mécanisme permet d'obtenir de bonnes performances.

2.2.5 Les outils d'administration système

Ce type d'outil est nécessaire même pour des machines qui ne sont pas en cluster. Ils servent par exemple au bon fonctionnement des machines, à la configuration des réseaux, à la gestion des utilisateurs …

2.2.5.1 Webmin

Webmin possède une interface Web à travers laquelle il est possible d'administrer le système de la machine ainsi que le réseau. Il permet entre autre chose de configurer les paramètres d'un système, de gérer les utilisateurs, les services, de faire du routage, du partage de fichiers… Et tout cela pour des machines de type Unix.

Avec son interface graphique, Webmin permet de configurer un grand nombre de choses sans difficulté. Webmin s'installe au démarrage ou avec l'ajout des paquetages. Une fois le serveur installé, vous pouvez mettre en place des sécurités et des restrictions de droits pour l'accès à ce serveur.
Vous pourrez aussi installer de nombreux serveurs tel que :

- Le serveur http : Apache
- Le serveur de fichier : Samba
- Les serveurs de bases de données : MySQL et PostgreSQL
- Le serveur DNS : Bind
- Le serveur DHCP

Au niveau du réseau, vous aurez la possibilité de préciser les paramètres d'un "ping", d'un "traceroute", mais également ceux du DNS (lookup), de recherche d'un fichier (Whois). En ce qui concerne le matériel, vous aurez accès au réseau, aux imprimantes, aux disques (partitionnement) ainsi qu'à la mise en place du raid.

2.2.5.2 Nagios

L'url d'accès au site web de Nagios® est le suivant : www.nagios.com

2.2.5.2.1 Description

Nagios® est un moniteur, outil de surveillance d'hôtes et de services réseau et autres. Outil de contrôle et prévention, il permet d'avertir les administrateurs (contacts), si des seuils d'alertes sont dépassés et/ou si des incidents surviennent sur le réseau et les serveurs. Nagios est ainsi un outil d'aide aux administrateurs leur permettant de prévenir cad d'éviter aux clients, utilisateurs et managers des perturbations de fonctionnement.
Nagios a été conçu pour fonctionner sur le système d'exploitation Linux, mais il fonctionne aussi très bien sur d'autres systèmes Unix.

Le démon Nagios de monitoring teste de façon intermittente des hôtes (serveurs) et des services tels que vous les avez définis, grâce à des 'plugins' externes. Ces plugins sont des programmes externes. Ces' plugins' renvoient des informations dites de 'status' à Nagios.
Quand survient un problème, le démon peut aussi envoyer des notifications (alertes) à des 'contacts' (généralement des administrateurs) grâce à différents moyens : email, SMS, 'instant message', ….

Nagios stocke les retours d'information des plug-ins dans des fichiers donnant les statuts courants des hôtes et services, dans des fichiers historiques de log et dans des rapports.
Ceux-ci sont accessibles directement , mais aussi visualisables via un browser web.

2.2.5.2.2 Caractéristiques techniques

- Monitoring (surveillance) de services de réseaux (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Monitoring (surveillance) des ressources serveurs (charge processeurs, utilisation des disques et de la mémoire, fonctionnement des processeurs, …).
- Conception simple des 'plugins', vous permettant de développer facilement vos propres tests sur les hôtes et les services.
- Possibilité de définir une hiérarchie réseau, permettant de détecter et de faire la distinction entre les hôtes 'hors service' et ceux que Nagios ne peut atteindre
- Mécanisme de notification à des contacts (interlocuteurs) quand Nagios détecte un problème sur un hôte ou sur un service, et quand le problème est résolu. Notifications possibles par email, pager, ou autre méthode définie par l'utilisateur.
- Mécanisme optionnel d'escalade des notifications vers des groupes de contacts.
- Possibilité de mettre en place un processus de gestionnaire d'évènements s'appuyant sur les évènements services et hôtes générés, pour résoudre les problèmes de façon pro-active.
- Support pour implémenter des serveurs de monitoring redondants et distribués.
- Interface de commandes externes, qui permet d'effectuer des modifications en ligne sur le monitoring, et sur le fonctionnement des notifications, grâce à l'utilisation de gestionnaires d'évènements, de l'interface web, et d'applications utilisateurs.
- Sauvegarde puis récupération des informations status sur les hôtes et les services après un restart de Nagios..
- Temps d'arrêt programmables permettant d'interrompre l'envoi de notifications concernant les hôtes et les services pendant les périodes planifiées de 'hors normes'.

- Interface Web permettant d'accéder à la documentation sur Nagios (en anglais).
- Interface Web permettant de visualiser le statut courant du réseau: l'historique des notifications et problèmes, les fichiers logs
- Possibilité de prendre connaissance des problèmes via l'interface Web.
- Schéma d'architecture d'autorisations simple, vous permettant de définir les utilisateurs qui pourront voir et modifier des éléments grâce à l'interface Web.

2.2.5.2.3 Licence / License

Nagios® is licensed under the terms of the GNU General Public License Version 2 as published by the Free Software Foundation. This gives you legal permission to copy, distribute and/or modify Nagios under certain conditions. Read the 'LICENSE' file in the Nagios distribution or read the online version of the license for more details. Nagios is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

2.2.5.3 NAGAT

Nagat, est l'outil d'administration de Nagios,. C'est une solution fondée sur le web et écrite en php.
Nagat permet de configurer Nagios, moniteur-superviseur d'hôtes et de services..

2.2.5.4 NRPE

NRPE signifie: Nagios Remote Plugin Executor

NRPE permet d'exécuter des plugins 'locaux' (comme check_disk, check_procs ….) sur les hôtes distants du réseau, surveillés par Nagios :
Nagios appelle le plugin check_nrpe. Ce plugin lance une requête sur un plugin local installé sur l'hôte distant. Ce plugin local s'exécute et renvoie un status qui est transmis, via check_nrpe à Nagios.
Pour que cela fonctionne, cela nécessite que nrpe fonctionne sur l'hôte distant. Il peut être activé comme un démon individuel ou comme un service d'inetd ou xinetd.

2.2.5.5 Les Plugins

2.2.5.5.1 Principe

Introduction

Nagios ne dispose pas de mécanisme interne pour vérifier l'état d'un service, d'un hôte, etc. Il utilise des programmes externes (appelés plugins) pour exécuter les tâches d'exécution. Nagios exécutera ainsi un plugin dès qu'il aura besoin de tester un service ou un hôte qui est supervisé. Les plugins font « ce qu'il faut » pour exécuter le contrôle demandé et renvoient le résultat à Nagios. Nagios analyse le résultat reçu du plugin et prend les mesures nécessaires (ex : lancer des gestionnaires d'événements, envoyer des notifications, etc).

Les plugins sont séparés de la logique du programme principal de Nagios.

Avantages

Grâce à cette architecture, vous pouvez contrôler tout ce à quoi vous pensez. Si vous pouvez automatiser le processus de contrôle d'un élément, vous pouvez le superviser avec Nagios. Un certain nombre de plugins ont déjà été créés pour superviser les ressources élémentaires comme : la charge du processeur, l'espace libre du disque dur, les ping effectués, etc. Si vous voulez superviser quelque chose d'autre, regardez la documentation existant sur le web sur comment écrire des plugins .

Périmètre de Nagios

Nagios n'a aucune idée de ce que vous supervisez. Vous pouvez en effet superviser des statistiques de trafic réseau, un taux d'erreur, la température d'une pièce, la tension de la CPU, la vitesse de rotation du ventilateur, la charge du processeur, l'espace libre du disque dur.... Aussi, Nagios ne trace pas de graphes des changements de valeur des ressources surveillées. Par contre, il suit leurs changements d'états.
Ce sont les plugins qui savent exactement ce qu'ils surveillent et comment le faire...

Tous les plugins qui respectent les consignes minimales de développement pour ce projet contiennent une documentation interne. Cette documentation peut être affichée en exécutant le plugin avec le paramétre "-h" ("--help" si les paramètres longs sont activés).

Par exemple, si vous voulez savoir comment fonctionne le plugin check_http ou quels paramètres il accepte, vous devez essayer :

- soit : ./check_httpd --help

- soit : ./check_httpd --h

Exemples de définition de commandes pour des services

La distribution principale des plugins comprend un fichier de configuration (appelé checkcommands.cfg) qui contient des exemples de définitions de commandes de contrôle de service et d'hôte utilisant les derniers plugins.
Les services à surveiller sont définis dans le fichier de configuration services.cfg. Celui-ci s'appuie ensuite sur checkcommands.cfg, pour connaître le plugin à exécuter.

Ex : de service
```
define service{
        use                             generic-service
        host_name                       linux_serv1
        service_description             HTPP-serv1
        ……..
        normal_check_interval  10
        contact_groups          linux_admins
        ….      ……
        check_command                   check_http_com
}
```

Commande et Plugin associés:
```
define command{
        command_name                    check_http_com
        command_line            $USER1$/check_http –H $HOSTADDRESS$
}
```

Où check_http est un plugin livré avec Nagios. Il doit être compilé lors de l'installation de Nagios.

2.2.6 Les outils d'administration cluster

Les outils de contrôle et de monitoring

De façon générale, ce genre d'outil a pour but de surveiller l'activité du cluster et son bon fonctionnement. Ils n'augmentent pas directement les possibilités des machines mais en facilitent la gestion. Voici ce que peut indiquer un outil de monitoring ou de contrôle :

Est-ce qu'un nœud du cluster ne fonctionne pas ?
Encombrement du réseau
Utilisation du cluster
…

Ces outils vont être utilisés par 2 types de personnes :

L'administrateur système dont le but est de :
- Maintenir le cluster en état de fonctionnement
- Diagnostiquer et, si possible, résoudre des comportements anormaux

L'utilisateur :
- Contrôler l'impact de l'exécution d'un programme dans un but d'optimisation
- Regarder si une application tourne bien, que tous les processus ont bien été lancés

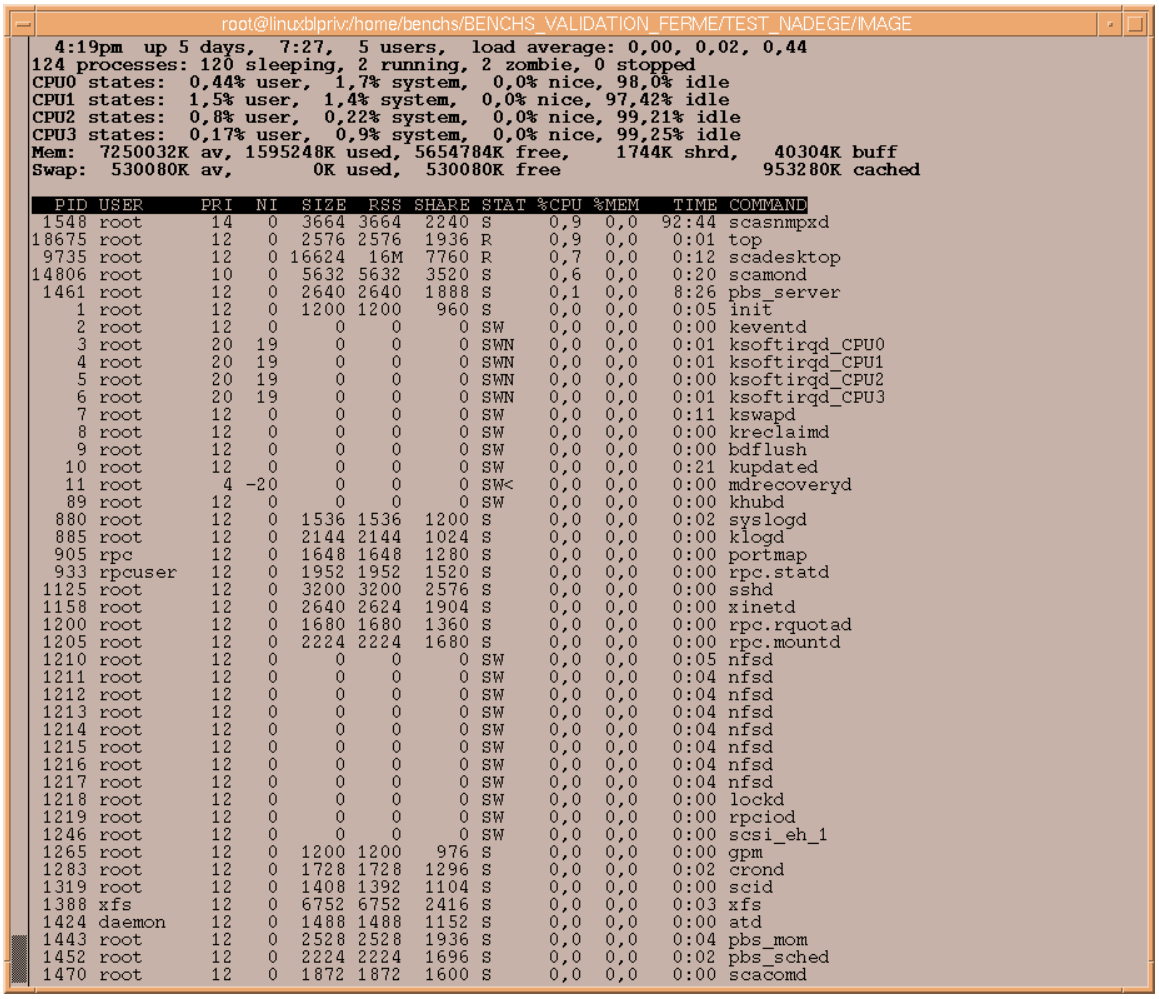
Ganglia :

Ganglia est un outil de monitoring pour les clusters. Il dispose d'une interface graphique au format Web, permettant de visualiser sous forme graphique différents renseignements sur l'utilisation du cluster.
C'est également un outil de contrôle puisqu'il indique lorsqu'un nœud ne fonctionne plus.

Top :

Cette commande Unix permet de connaître différentes informations en temps réel sur l'exécution des processus et sur l'utilisation d'un nœud.

- Voici un aperçu de ce que peut fournir la commande top :
- Liste des processus lancés
- Indique par ordre décroissant le temps machine des processus, les plus gourmands en premier
- Permet de déterminer les processus ayant besoin de beaucoup de ressources
- Permet d'observer la vie du système
- Indique la mémoire utilisée, le pourcentage de swap utilisé ….
- …



Exemple de sortie donnée par la commande top

Les outils permettant de lancer une même commande sur plusieurs nœuds

Ce qu'on appelle un shell distribué est indispensable pour l'utilisation d'un cluster. En effet, il y a toujours des instructions que l'on souhaite effectuer sur différents nœuds du cluster sans avoir pour autant besoin de le faire manuellement sur chacun des noeuds. Plusieurs outils ont donc cette fonctionnalité.

Ka-Tools :

Cet outil a été réalisé pour faciliter l'installation et l'utilisation d'un cluster Linux. Tous ses composants supportent le passage à l'échelle (scalabilité).

Le module ka-run de ce composant permet de lancer une commande sur les machines souhaitées.

Gexec :
Gexec propose la même fonctionnalité.

C3 = Cluster Command and Control :
C3 implémente un certain nombre de commandes pour aider à l'administration du cluster.

Les outils de déploiement

Ce terme désigne des composant logiciels permettant d'installer une distribution ou des packages sur plusieurs machines en même temps. En effet, pour les grands clusters, il est indispensable de disposer d'un tel outil pour ne pas avoir besoin de refaire la même installation un grand nombre de fois. Aujourd'hui, des outils répondent à cette demande et permettent d'installer une distribution Linux par exemple sur plusieurs centaines de machines en quelques dizaines de minutes.

Ka-Tools :
Le module ka-deploy de cet outil permet de cloner des nœuds.

SystemImager :
Il sert à automatiser l'installation de Linux sur un cluster de machines identiques. Il permet la distribution de logiciels, la configuration et la mise à jour de système d'exploitation …

2.2.7 Les outils d'analyse de performances et de profiling

2.2.7.1 Leur fonction

Pour comprendre l'activité de la machine et comparer ses performances avec une autre, le matériel dispose de compteurs comptabilisant les occurrences d'événements associés à des fonctions du processeur : nombre d'Opérations Flottantes, nombre de cycles d'attente d'accès mémoire, nombre de store/load/branch, nombre total de cycles, nombre de défaut de Cache…

L'intérêt de regarder ces divers évènements, est de faire une corrélation entre la structure d'un programme et son adaptation à l'architecture matérielle sous-jacente. Pour le constructeur, cela peut lui permettre d'évaluer son matériel à logiciel fixe ; pour un développeur de logiciel, cela peut fournir des informations utiles à l'explication des manques de performance de certaines parties de codes et donc orienter les optimisations en fonction du matériel cible.

Le suivi de ces événements est particulièrement utile dans le cas de programmes importants comme des compilateurs, des benchmarks, ou de la modélisation de performance.

2.2.7.2 Quel outil dans quel cas ?

2.2.7.2.1 Dans le monde de l'open Source

Toute une hiérarchie d'outils est opérationnelle

- Si vous voulez avoir une vue globale des goulots d'étranglement de votre application logicielle, alors c'est du simple profiling système que vous avez besoin : **vprof** peut répondre à votre besoin
- Si par contre vous voulez caractériser votre architecture matérielle ou si vous voulez comparer le comportement de deux programmes similaires vis à vis des performances hardware, alors **pfmon** permet d'avoir des réponses globales sans toucher au code.
- si, enfin, ce sont des mesures fines qui vous intéressent à des points précis de votre programme que vous connaissez parfaitement, alors **PAPI** vous permet de programmer exactement l'instrumentation qui vous convient.'

Pour résumer le niveau croissant de fonctionnalités disponibles, on trouve :

objet	Nature	fourniture	fonction
perfmon.c	Module source du kernel Sous arch/ia64/kernel	http://www.kernel.org version >=2.4.18 avec patches ia64 valide à la configuration du noyau par CONFIG_PERFMON=y	Procédures de gestion des registres hardware
pfmon	Utilitaire mettant en œuvre perfmon du noyau constitué d'un binaire : pfmon et d une bibliothèque : libfpm	ftp://ftp.hpl.hp.com/pub/linux-ia64/ pfmon-2.0.ia64.rpm	Exploitation des évènements hardware (par appel à perfmon.c) **Interface commande sans modification de code**

PAPI	API : Interface programme intégrant la bibliothèque de pfmon (libpfm)	http://icl.cs.utk.edu/projects/papi/software/ papi 2.3.4	Gestion des évènements hardware par appel à des fonctions spécifiques de plus ou moins haut niveau, au choix. Permet un nombre de compteurs non limité**, se place dans le code du programme** aux endroits précis qui veulent être mesurés.
cprof	Outil de profiling au-dessus de papi	http://aros.ca.sandia.gov/~cljanss/perf/vprof/	Profiling système ou surveillance d'un évènement hardware par PAPI. **Simple édition de liens** avec l'application sans modification du code source. L'application génère ainsi à l'exécution un fichier de données exploite par la commande cprof
vprof	Version visuelle de cprof	http://aros.ca.sandia.gov/~cljanss/perf/vprof/	Même fonction que cprof mais présentation des résultats dans une fenêtre graphique

2.2.7.2.2 Dans le monde commercial

VTUNE :

Intel commercialise un outil du monde windows , capable d'échantillonner les évènements recueillis sur un serveur distant sous Linux. L'avantage de cet outil est l'ergonomie de la présentation graphique.
Son inconvénient est la liaison réseau qui est nécessaire entre les 2 mondes et la station windows supplémentaire.

VAMPIR :

VAMPIR est un outil graphique de profiling commercialisé par la société PALLAS. Il permet de visualiser et d'analyser les performances des codes MPI.

2.2.7.3 Exemples d'utilisation comparée d'outils de profiling

2.2.7.3.1 PFMON

Pfmon est le bon outil pour obtenir rapidement un chiffre élémentaire caractérisant un programme, sans aucune programmation, par exemple le nombre d'opérations en virgule flottante du programme test2:

```
pfmon -e FP_OPS_RETIRED test2
2 421 914 081 FP_OPS_RETIRED
```

Il est possible de capturer 4 évènements a la fois et d'avoir une sortie complète (with header) ; par exemple la commande:

```
pfmon --with-header -k -u -e
        L2_MISSES,L2_REFERENCES,L3_MISSES,IA64_INST_RETIRED
        --outfile=result_pfmon/pfmon_miss_16.2081 runspec ……..
```

donne:

```
#
# date: Fri Feb 21 02:16:14 2003
#
# hostname: chircane
#
# kernel version: Linux 2.5.59 #2 SMP Wed Feb 12 06:17:00 PST 2003
#
# pfmon version: 2.0
```

```
# kernel perfmon version: 1.3
#
# page size: 16384 bytes
# CLK_TCK: 1024 ticks/second
# CPU configured: 4
# CPU online: 4
# physical memory: 4189208576
# physical memory available: 2908225536
#
# host CPUs:  4-way 997MHz Itanium 2 (McKinley, B3)
#       PAL_A: 0.7.31
#       PAL_B: 0.7.36
#       Cache levels: 3 Unique caches: 4
#       L1D:    16384 bytes, line  64 bytes, load_lat   1, store_lat   3
#       L1I:    16384 bytes, line  64 bytes, load_lat   1, store_lat   0
#       L2 :   262144 bytes, line 128 bytes, load_lat   5, store_lat   7
#       L3 :  3145728 bytes, line 128 bytes, load_lat  12, store_lat   7
#
# captured events:
#       PMD4: L2_REFERENCES, kernel+user level(s)
#       PMD5: L2_MISSES, kernel+user level(s)
#       PMD6: L3_MISSES, kernel+user level(s)
#       PMD7: IA64_INST_RETIRED, kernel+user level(s)
#
# monitoring mode: per-process
#
#
# instruction sets:
```

```
#       PMD4: L2_REFERENCES, ia32/ia64
#       PMD5: L2_MISSES, ia32/ia64
#       PMD6: L3_MISSES, ia32/ia64
#       PMD7: IA64_INST_RETIRED, ia32/ia64
#
#
# command: pfmon --with-header -k -u -e
            L2_MISSES,L2_REFERENCES,L3_MISSES,IA64_INST_RETIRED
            --outfile=result_pfmon/pfmon_miss_16.2081 runspec -c il420-linux-v7
            --iterations=1 --tune=base -I --noreportable --rate --users 16 applu
#
#
#
              4234016 L2_MISSES
             69799519 L2_REFERENCES
               527948 L3_MISSES
            772924788 IA64_INST_RETIRED
```

2.2.7.3.2 PAPI

La même chose peut être obtenue avec PAPI mais de manière beaucoup plus fine, par modification de la programmation sur des séquences précises :

On peut distinguer 3 niveaux d'interface:

a) Le plus simple ou très haut niveau:

- Utilisation de la routine PAPI_flops, qui donne le temps réel, le temps CPU et le nombre d'opérations flottantes par seconde

Dans un programme fortran:

```
#include "fpapi.h"

real real_time,cpu_time, mflops
 integer*8 fp_ins

call PAPIf_flops( real_time, cpu_time, fp_ins, mflops, ierr )
```

....traitement....

```
call PAPIf_flops(real_time,cpu_time, fp_ins, mflops, ierr)
```

...édition des résultats:

```
    write (6,120) real_time, cpu_time, fp_ins,mflops
120   format (//'PAPI result'/'       real time (secs) :', f15.3,
  $  /'       CPU time (secs) :',f15.3,
  $  /'floating point instructions:', i15,
  $  /'                    MFLOPS:', f15.3)
```

b) L'interface haut-niveau (non thread-safe):

- Inclure les définitions de PAPI:

```
#include "fpapi.h"
```

- Déclarer les évènements a compter et la récupération des erreurs:

```
integer events(2),numevents, ierr
character*PAPI_MAX_STR_LEN errorstring
```

- Déclarer les variables pour récupérer les compteurs

```
integer*8 values(2)
```

- Positionner chaque événement au type désiré (cf fpapi.h)

```
 numevents= 2
events(1)=PAPI_FP_INS
 events(2)=PAPI_TOT_CYC
```

- Démarrer les compteurs et tester le compte-rendu

```
call PAPIf_start_counters(events, numevents, ierr)
  if ( ierr .NE. PAPI_OK ) then
    call PAPIF_perror(ierr,errorstring,PAPI_MAX_STR_LEN)
    print *, errorstring
  end if
```

- Faire un début de traitement, puis lire et réinitialiser les compteurs sans les arrêter

```
call PAPIf_read_counters(values,numevents,ierr)
if ( ierr .NE. PAPI_OK ) then
    call PAPIF_perror(ierr,errorstring,PAPI_MAX_STR_LEN)
    print *, errorstring
end if
```

- On peut aussi accumuler les compteurs en utilisant la routine

```
PAPIf_accum_counters(values,numevents,ierr)
```

- Continuer le traitement puis arrêter les compteurs:

```
call PAPIf_stop_counters(values, numevents, ierr)
if ( ierr .NE. PAPI_OK ) then
    call PAPIF_perror(ierr,errorstring,PAPI_MAX_STR_LEN)
    print *,errorstring
end if
```

c) L'interface détaillé "low-level"(thread safe)

Il permet de gérer des évènements prédéfinis (au nombre de 100 dans PAPI!) aussi bien que des évènements complètement natifs, définis par leur codage dans les registres.

- Il faut commencer par initialiser la library par

```
call PAPIf_library_init(ierr)
```

- Puis gérer des ensembles d'évènements qui seront utilisés de concert par:

```
call PAPIf_create_eventset(es,ierr)
```

- Ensuite introduire des évènements dans cet ensemble par:

```
call PAPIf_add_event( es,  PAPI_TOT_CYC,, ierr )
```

- Démarrer, lire, arrêter un ensemble

```
call PAPIf_start(es)
call PAPIf_read(es)
call PAPIf_accum(es)
call PAPIf_stop(es)
```

Se reporter au user's guide de PAPI pour l'interface exact.

2.2.7.3.3 VPROF

Si on veut avoir une visibilité de la répartition du taux d'opérations flottantes par seconde sur le programme test2, voila la marche à suivre :
Tout d'abord, il faut compiler le programme (test2.cc) en –g et lui link-editer un module permettant la collecte : vmonauto.o (ce qui évite d'insérer un appel à vprof en début et en fin) ;
ensuite il faut faire une édition de liens avec les bibliothèques vmon et PAPI:

```
make test2
c++ -g -O2 -Wl,-static -o test2 test2.cc ../lib/vmonauto.o ../lib/libvmon.a –L
/opt/envhpc/papi-linux-ia64/lib
```

A l'exécution du programme il faut spécifier

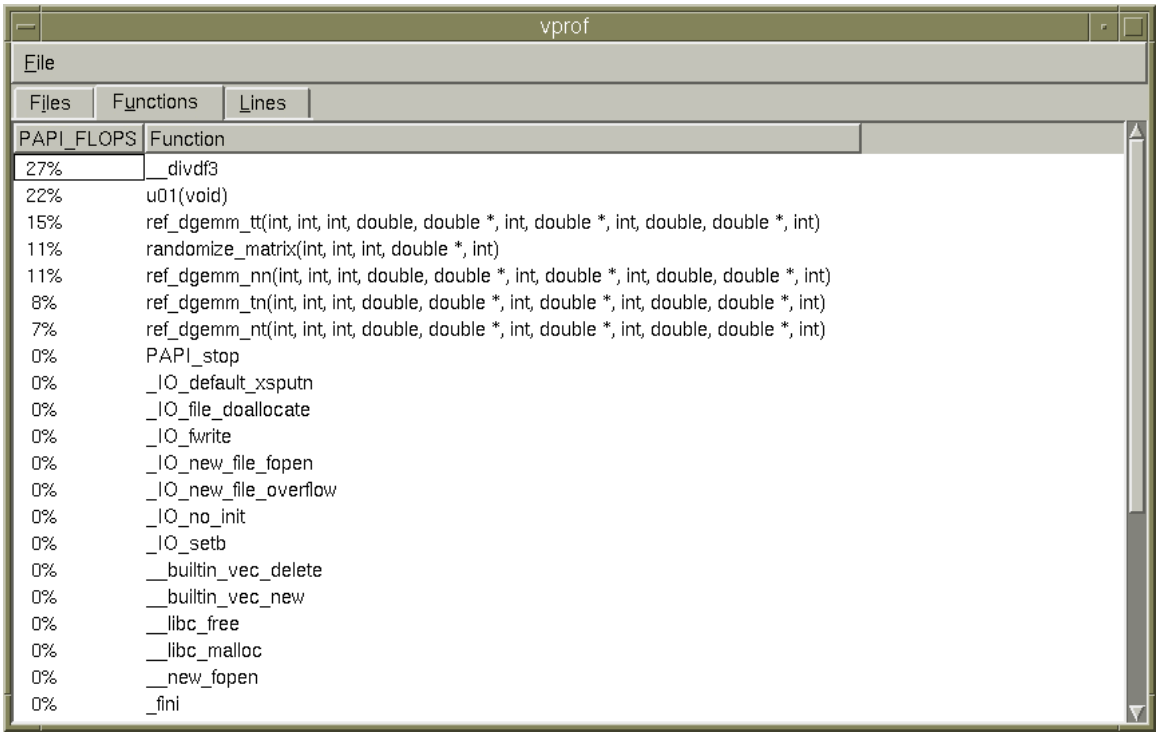
- quelle métrique est étudiée : VMON=,(par défaut c'est le profiling système)
- et quel est le nom du fichier qui va collecter les renseignements :VMON_FILE=, (par défaut c'est vmon.out)

```
VMON=PAPI_FLOPS  test2
```

Ensuite on peut appeler vprof après avoir défini le Display, et spécifier par –d le directory des sources si ce n'est pas le directory courant

```
export DISPLAY=.... :0.0
../bin/vprof –d. test2
```

et voilà la fenêtre obtenue, qui décline les différentes fonctions avec le pourcentage d'opérations flottantes par seconde.



La commande cprof donne les mêmes informations, mais sous forme de texte ;

Exemple, lors d'un autre essai du même programme:

```
 ../bin/cprof -r test2
Columns correspond to the following events:
  PAPI_FLOPS - Floating Point instructions per second (1145301 events)
Function Summary:
  20.6% __divdf3
  15.3% ref_dgemm_tn(int, int, int, double, double *, int, double *, int,
double, double *, int)
  14.8% ref_dgemm_nn(int, int, int, double, double *, int, double *, int,
double, double *, int)
```

```
  13.9% u01(void)
  12.2% ref_dgemm_tt(int, int, int, double, double *, int, double *, int,
double, double *, int)
  11.9% randomize_matrix(int, int, int, double *, int)
  11.3% ref_dgemm_nt(int, int, int, double, double *, int, double *, int,
double, double *, int)
```

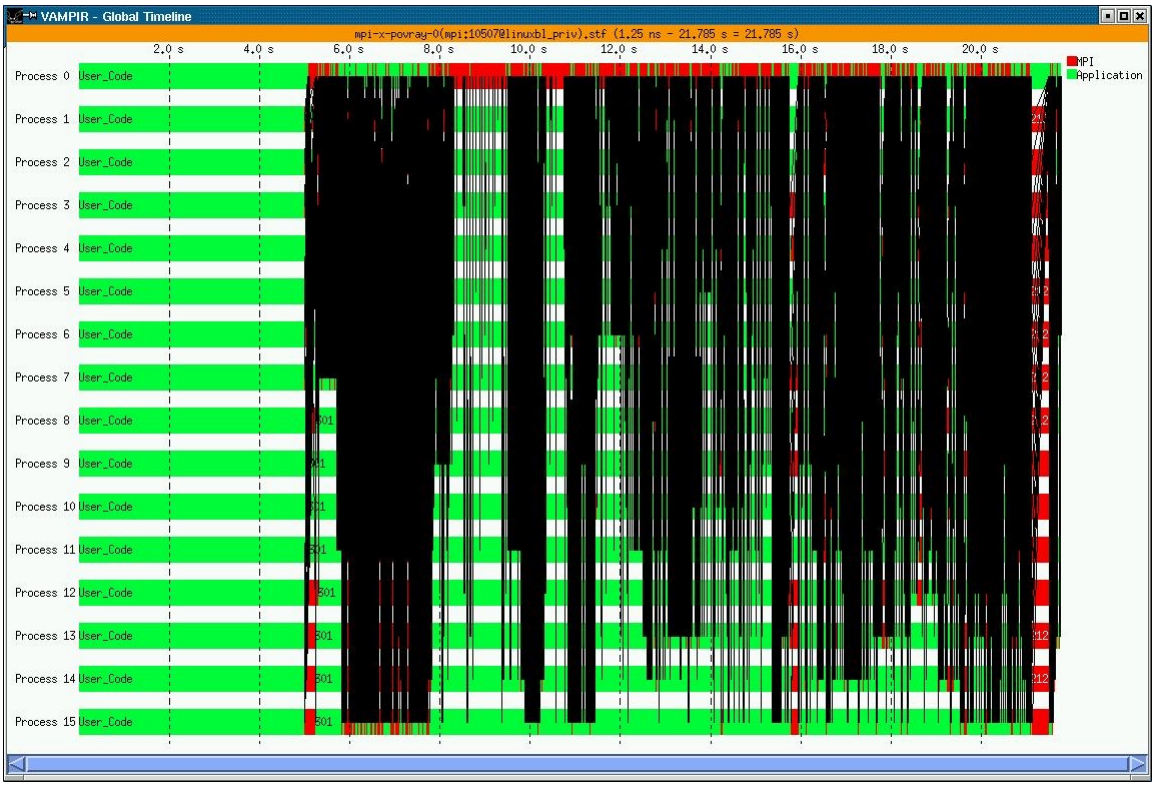
Remarque :
Dans certains cas , on remarque des fautes de segmentation avec PAPI sous forme de bibliothèque partagée ; dans ce cas essayer un link-edit statique

2.2.7.4 VAMPIR

VAMPIR est un outil graphique de profiling commercialisé par la société PALLAS. Il permet de visualiser et d'analyser les performances des codes MPI.

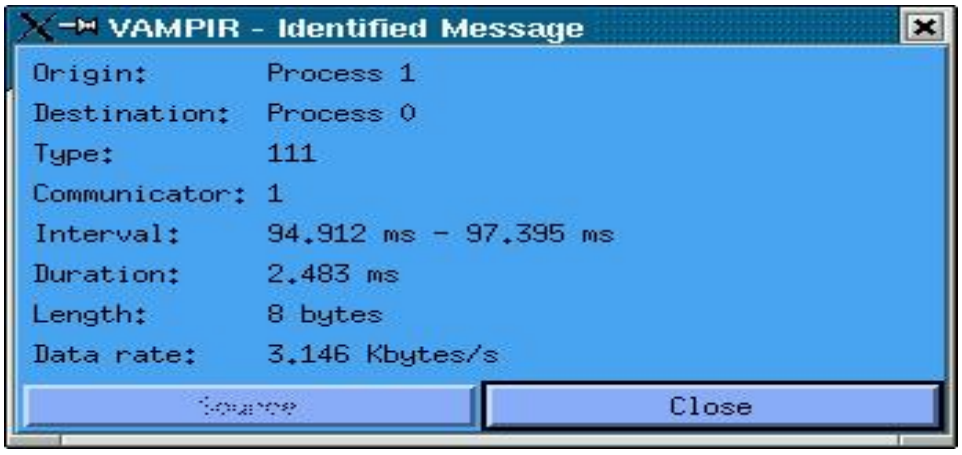
Après avoir créé des fichiers de traces à l'aide du logiciel VampirTrace, il suffit de lancer Vampir pour pouvoir les interpréter de façon graphique.

Voici quelques exemples de graphes proposés :



Vue d'ensemble des communications MPI d'une application

Ce graphique montre l'ensemble des communications MPI survenues au cours de cette application. Il est possible de zoomer sur certaines parties et également d'avoir des renseignements plus précis sur une communication particulière en cliquant dessus.

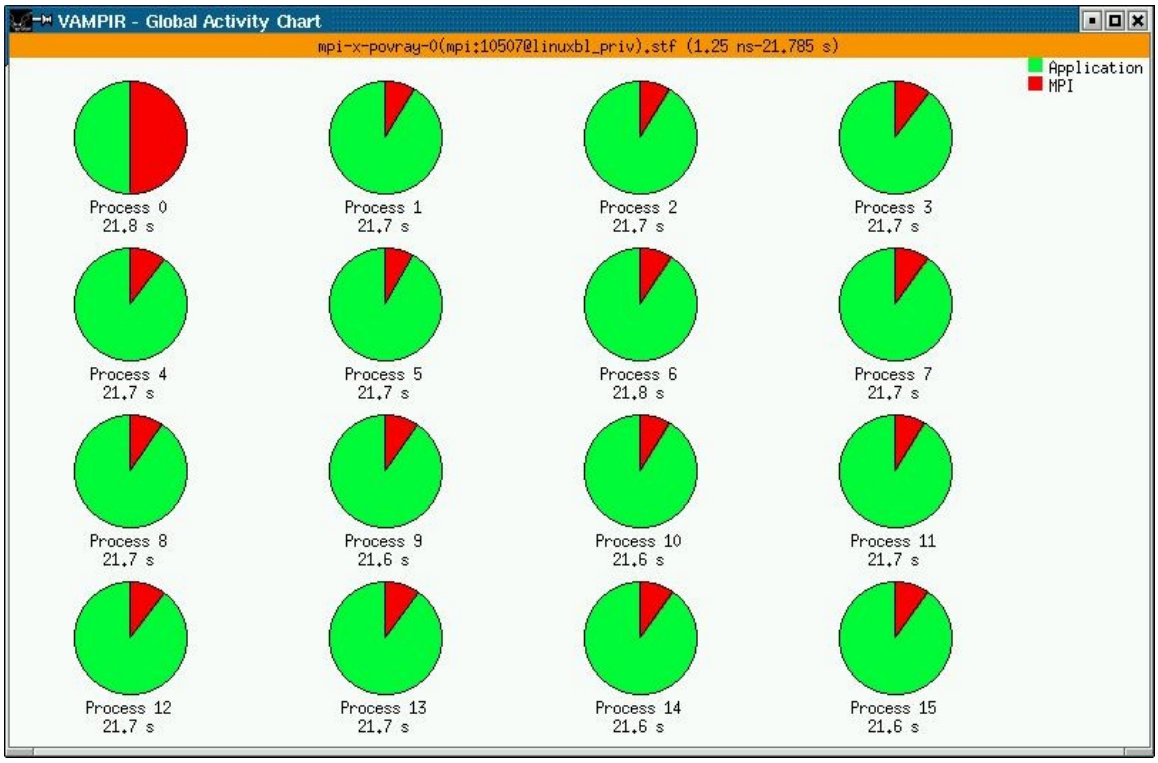


Renseignements sur une communication MPI

Ce message indique que la communication, d'une durée de 2,483 ms, a eu lieu du processus 1 vers le processus 0…

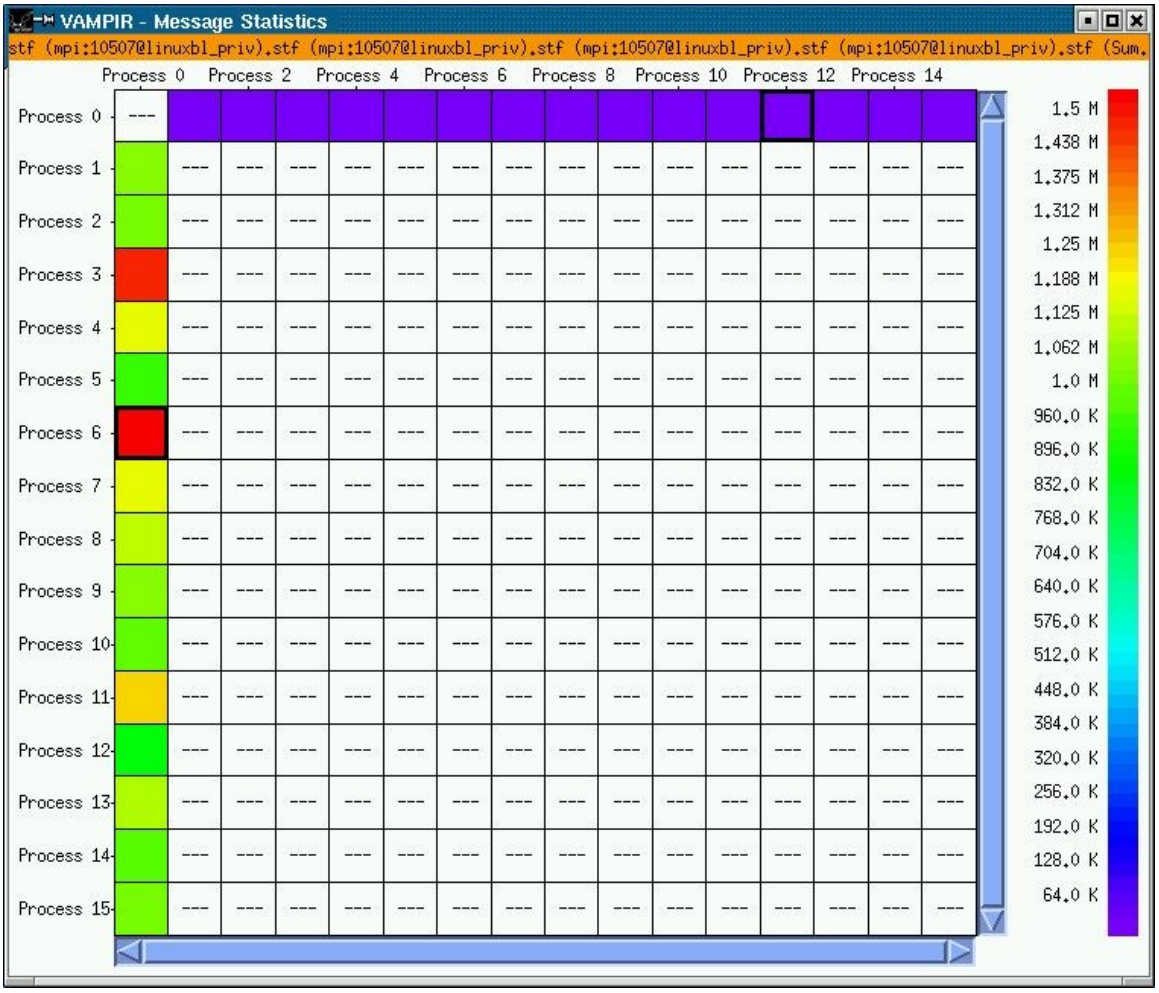
Temps passé dans les différentes activités

En clair, on a le temps total passé au niveau applicatif et en foncé, le temps passé en communication MPI. Pour cet exemple, on se rend ainsi compte que le programme passe plus de temps en application qu'en communications MPI.



Temps passé par chacun des processus en application vs MPI

Ce graphe permet ainsi de faire des comparaisons entre les différents processus et de déterminer ceux qui effectuent le plus de communications.



Statistiques sur les échanges de messages entre les processus

Ce dernier exemple de graphique met en évidence les quantités de messages échangés entre chacun des processus. Ici, on remarque que les processus ne communique que via le processus 0.

Ces différents graphes donnent ainsi un aperçu des possibilités de Vampir et de l'utilisation qui peut en être faite. Ces renseignements peuvent permettre par la suite d'optimiser l'application, de trouver le nombre de processus le plus intéressant pour un problème donné …

2.2.8 Les débogueurs

Il existe 2 types de débogueurs : les symboliques et les non symboliques. Un débogueur symbolique permet d'accéder aux symboles d'un programme :

- On peut accéder aux lignes du fichier source,
- On peut accéder aux variables du programme par leur nom, alors qu'avec un débogueur non symbolique, on n'aurait accès qu'aux lignes du programme machine (assembleur) et aux adresses physiques.

2.2.8.1 GDB

GDB est l'acronyme de Gnu DeBugger. C'est un débogueur puissant dont l'interface est totalement en ligne de commande, c'est à dire avec une invite en texte.
GDB est tellement apprécié qu'on le trouve aussi encapsulé dans des interfaces graphiques, comme XXGDB ou DDD. GDB est publié sous la licence GNU. Ce débogueur supporte les applications parallèles et les threads.

2.2.8.2 DBX

dbx est un débogueur symbolique textuel. Il travaille de manière optimale sur un code compilé avec l'option -g (option -g3 recommandée) et sans optimisation (option -O0 recommandée). Cela a pour effet de rajouter des informations de débogage dans le code objet.

2.2.8.3 TOTALVIEW

http://www.etnus.com/
TotalView™ est un débogueur symbolique fonctionnant par fenêtres, pour le C, le C++ et le Fortran(77, 90 et HPF). Il peut déboguer des application PVM ou MPI. TotalView™ supporte intégralement des programmes multi processus et fonctionne indifféremment sur les systèmes monoprocesseurs ou SMP, clusterisés, distribués et MPP.
TotalView™ acquiert des nouveaux processus ou des threads tels qu'ils ont été générés dans l'application, quelque soit le processeur sur lequel ils s'exécutent. Il est également possible de se raccorder à un processus démarré en dehors de TotalView™. Les tableaux de données peuvent être filtrés, affichés et visualisés afin de contrôler le comportement du programme. Enfin, vous pouvez plonger ("appeler les composants et détails de...") sur des objets et structures du programme.
L'aide contextuelle fournit à l'écran, sur un simple click, les informations de base du manuel. La documentation complète est disponible sur le site d'etnus :
http://www.etnus.com/Download/TV.html

2.2.9 Les outils de répartition de tâches

Le regroupement de plusieurs machines sur un réseau ne suffit pas à la constitution d'un véritable *cluster*. Un logiciel assurant la gestion de l'ensemble est nécessaire à la meilleure utilisation possible de la puissance de calcul de ces nœuds indépendants. Un système de gestion de tâches a justement pour but de gérer cette puissance de calcul afin d'avoir les meilleures performances possibles et de permettre à plusieurs utilisateurs de lancer des tâches simultanément.

Les systèmes de gestion de tâches disposent généralement des fonctionnalités suivantes :

- Une interface utilisateur
- Un ordonnanceur (*scheduler)*
- Un gestionnaire des ressources
- Rapatriement des fichiers de log et de configuration sur une même machine
- Un environnement sécurisé

L'interface utilisateur permet d'exécuter des tâches locales ou distantes. Celles-ci sont lancées par l'intermédiaire de files d'attente. L'utilisateur peut :

- Spécifier les ressources nécessaires aux tâches qu'il veut exécuter
- Supprimer l'exécution d'une tâche
- Faire de la suspension ou de la reprise d'exécution de tâche
- Connaître le statut d'une tâche en cours d'exécution

Pour ce faire, l'utilisateur dispose suivant les cas d'une ligne de commande, d'un environnement graphique ou bien des deux interfaces simultanément.

Le scheduler sert à définir une politique de scheduling, c'est à dire de lancement des tâches. En effet, celles-ci doivent être classées les une par rapport aux autres selon des priorités à définir. Ces priorités sont calculées généralement selon les critères suivants :

- Temps d'attente dans les files d'attente avant exécution
- Ressources nécessaires (nombre de nœuds, temps d'exécution, mémoire, disque, etc.)
- Type de la tâche (interactive, parallèle, batch, etc.)
- Identité de l'utilisateur

Le gestionnaire des ressources sert à allouer des ressources, à connaître l'état des ressources, à collecter toutes sortes d'informations relatives à l'exécution des tâches et à partir de celles-ci à appliquer la politique de scheduling.

OpenPBS :

http://www.openpbs.org
OpenPBS est un système de gestion de travaux (ou jobs) permettant de partager au mieux les ressources de la machine selon les besoins des utilisateurs (temps CPU, nombre de processeurs, espace mémoire, etc...). Il offre des outils permettant la construction, la soumission, le traitement et le contrôle de jobs séquentiels ou parallèles.

LSF(Platform) :

http://www.lerc.nasa.gov/WWW/LSF/lsf_homepage.html
Il s'agit d'un gestionnaire de batch propriétaire. C'est un ensemble d'outils dont le but est de répartir la charge CPU sur les machines du cluster et ce, d'une manière transparente pour l'utilisateur.
Voici quelques uns des principes de LSF :

- L'utilisation de files d'attentes
- Le choix de la machine la moins chargée
- La notion de jobs prioritaire
- …

Maui Scheduler :

http://mauischeduler.sourceforge.net/
Maui est un ordonnanceur qui permet entre autres de planifier le lancement des tâches dans le temps et dans l'espace. Il peut être utilisé avec un répartiteur de charge (ou gestionnaire de ressources) comme PBS.

Chapitre 3. Installation d'un cluster

3.1 Administration du matériel

3.1.1 Mise en route du nœud d'administration

Le nœud d'administration représentant la partie visible extérieurement du cluster, sa mise en route et son administration matérielle se fait normalement à l'aide de la console lui étant directement connectée.
L'opération du reste du cluster passe obligatoire par la disponibilité de ce nœud d'administration .
Il doit donc être le premier à être mis en route, installé avec son système d'exploitation et configuré.
L'administration matérielle des autres éléments se fait ensuite par l'intermédiaire de sa console ou de toute autre console lui étant connectée sur le réseau externe.

3.1.2 Mise en route et contrôle des autres nœuds

L'administration matérielle des autres nœuds du cluster s'effectue à travers le réseau d'interconnexion série.
Sur le nœud d'administration, un programme d'émulation de terminal minicom (ou xminicom dans une fenêtre X) doit être utilisé pour avoir accès à la console des autres nœuds.

Le lancement de ce programme se fait par :

$ minicom n0x0y &
(n0x0y identifie le nom du nœud auquel on veut accéder. Ce nom est utilisé comme suffixe du fichier /etc/minirc.n0x0y créé sur le nœud à administrer comme suit).
Pour « sortir » de minicom, il faut entrer les caractères « CTRL A » puis « X ».

Configuration de minicom :
minicom utilise un fichier de configuration propre à chaque nœud, situé dans le répertoire « /etc/ » et portant le nom « minirc.<nom du nœud >».
Chacun de ces fichiers positionne les valeurs des paramètres de travail de chaque ligne série.

Exemple pour un cluster 4 noeuds (admin, n0101, n0102, n0103):

```
$ cat /etc/minirc.n0101
# minicom default configuration file for node N0101
pr port             /dev/ttyD000
pu baudrate         115200
pu bits             8
pu parity           N
pu stopbits         1
pu mautobaud        Yes
pu statusline       disabled
pu hasdcd           No
pu minit
pu mreset
$

$ cat  /etc/minirc.n0102
# minicom default configuration file for node N0102
pr port             /dev/ttyD001
pu baudrate         115200
pu bits             8
pu parity           N
pu stopbits         1
pu mautobaud        Yes
pu statusline       disabled
pu hasdcd           No
pu minit
pu mreset
$

$ cat /etc/minirc.n0103
# minicom default configuration file for node N0103
pr port             /dev/ttyD002
pu baudrate         115200
pu bits             8
pu parity           N
pu stopbits         1
pu mautobaud        Yes
pu statusline       disabled
pu hasdcd           No
pu minit
pu mreset
$
```

3.2 SCALI SSP 3.0.1

Introduction :
Ce paragraphe décrit l'installation de SCALI SSP 3.0.1.
Vous devez consulter le site http://www.scali.com afin de vérifier l'existence de nouvelles mises à jour.

Pré-requis :

- Linux déjà installé sur chaque nœud
- Connexion des câbles comme indiqué dans le « SCALI system guide » .
- Bien que le logiciel SSP_3_0_1 soit fourni sur un CD ROM, il est aussi possible de le télécharger à partir de http://www.scali.com/Download/ssp.shtml.
  Si vous souhaitez télécharger le logiciel :

```
mkdir /home/scali
cd /home/scali
### Télécharger SSP_3_0_1_Linux2_ia64.tar.gz  ###
tar xvzf SSP_3_0_1_Linux2_ia64.tar.gz
```

- Le nom du nœud frontal doit être « admin » : Vérifier que la commande « uname –n » affiche « admin ».
- Configuration de l'Operating System comme indiqué dans le fichier OS (dans le sous-répertoire SSP_3_0_1/doc/SSP). **Sur tous les nœuds du cluster** :
  - Insérer tous les noms de nœuds avec le login *root* dans le fichier */root/.rhosts :*

```
admin root
n0101 root
n0102 root
n0103 root
etc …
```

  - Insérer tous les noms de nœuds dans le fichier */etc/hosts.equiv :*

```
admin
n0101
n0102
```

```
n0103
etc …
```

- Rajouter *rsh*, *rlogin* et *login* dans le fichier */etc/securetty*
- Exécuter les commandes :

```
chkconfig rsh on
chkconfig rlogin on
```

- Vérifier qu'il reste moins 200 Méga Octets de disponibles sur /opt/scali sur tous les nœuds.
- Avoir reçu une licence provisoire par email et l'avoir copiée sur le frontal (fichier /opt/license.dat)

Installation :

L'installation du logiciel d'interconnexion SSP de SCALI nécessite plusieurs passes :

1. Une première passe pour installer la licence, définir les nœuds (admin, n0101, n0102, n0103 etc …) et les fonctionnalités à installer (OpenPBS) ou à ne pas installer (Console Server, Power Switch Server, NIS/YP).
2. Une phase intermédiaire de mise à jour des rpm et du driver SCI
3. Une deuxième passe pour finir et vérifier l'installation

1) Installation : 1ère passe

- A partir du CD ROM (sur le noeud admin), faire la commande :

```
mount –t iso9660 /dev/cdrom /mnt/cdrom
cd /mnt/cdrom
./install
```

- Ou, à partir des fichiers issus du téléchargement, faire la commande :

```
/home/scali/install
```

- Suivre la procédure d'installation en répondant aux questions. La durée de certaines étapes peut être longue.
- A l'étape « **Expert mode** » faire le choix par défaut (« **n** »).
- A l'étape « **Specifying frontend** », valider « admin » (« **y** »).
- A l'étape « **Specifying node names** », énumérer les nœuds (dans l'ordre : admin, n0101, n0102, n0103 pour 4 nœuds).
- A l'étape « **Setup license(s)** » faire le choix « **2**) License file » et donner le chemin complet « /opt/license.dat ».

- A l'étape « **Determine node categories** », il est possible d'exclure certains nœuds qui ne sont pas des nœuds de calcul (par exemple des nœuds de stockage). Pour ce faire :
  - A la question « Do you accept the above configuration ? », il faut répondre « n ».
  - Sélectionner les nœuds à modifier
  - Ne sélectionner que la catégorie « eth_node » pour les nœuds à modifier
- A l'étape « **Configuring SCI network** », pour un cluster 4 nœuds, choisir « 2D torus », puis donner le nombre de nœuds (2) connectés sur les anneaux de l'interface L0, puis donner le nombre de nœuds (2) connectés sur les anneaux de l'interface L1. On a bien 2 x 2 = 4 (nombre total de nœuds). Puis choisir par défaut la méthode automatique (choix « **1** »).
- A l'étape « **install the console server** » répondre « **n** ».
- A l'étape « **install the power switch server** » répondre « **n** ».
- A l'étape « **install OpenPBS queue system** » répondre « **y** », puis renseigner le username « **linux** ».
- Les trois étapes suivantes de l'installation émettent des « WARNING ». Il ne faut pas en tenir compte et il faut répondre « i » (ignore). Il s'agit des étapes :
  - « Checking kernel versions »
  - « Checking HW support »
  - « Checking OS support »
- A l'étape « **Are you using NIS/YP** .. » répondre « **n** »
- Parmi les étapes suivantes de l'installation, certaines émettent des « WARNING ». Ceux-ci ne sont pas anormaux lors de la 1ère passe du logiciel d'installation et il faut répondre « i » (ignore). Il s'agit des étapes :
  - Problèmes d'installation ScaSCI, Checking SCI driver : Ces problèmes surviennent car le driver SCI ne se charge pas, ce qui n'est pas anormal à ce niveau d'avancement de l'installation.
  - « Checking SCI links »
  - « Testing SCI communication »

2) Phase intermédiaire : Mise à jour des « rpm » et du module SCI:

- Scali met à disposition des mises à jour de certaines parties de son logiciel. Il est conseillé de les installer et d'utiliser la procédure de « Mise à jour des « rpm » présentée ci-dessous.

- Scali installe tout ce qu'il faut sur le frontal et sur chacun des nœuds du cluster et teste si tout fonctionne (communication entre les nœuds). Cependant, le driver SCI fourni par défaut peut ne pas être adapté au noyau installé (voir les « WARNING » affichés lors de la 1ère phase d'installation) et ces tests ne peuvent pas fonctionner. Il faut mettre à jour ce driver par la procédure de « Mise à jour du module SCI » présentée ci-dessous.

Mise à jour des « rpm »:

- Télécharger dans le répertoire /opt/scali/repository/Linux2.ia64 les mises à jour accessibles sur le site www.scali.com (menu : «Download->Software->SSP3.0 updates).
  - ✓ ScaMPI
  - ✓ ScaSCI
  - ✓ ScaSCIddk
  - ✓ ScaSCIadap
  - ✓ ScaSISCI
  - ✓ Il faut noter que cette liste de rpm à mettre à jour peut évoluer.

```
cd /home/scali/repository/Linux2.ia64
### Télécharger les rpm (packages) ###
```

- Installer les packages sur les nœuds du cluster :

```
/opt/scali/sbin/scapkg –p ``<Package>''
```

(par exemple : « /opt/scali/sbin/scapkg –p ScaMPI », donc sans l'extension « Linux2.ia64-<version>.rpm »)

Mise à jour du module SCI:

Scali propose un outil (ScaSCIadap) pour adapter le driver SCI à un noyau différent du noyau supporté par défaut .

Pour utiliser cet outil, lire la documentation /opt/scali/doc/ScaSCIadap/README. Cependant voici une petite précision concernant cette documentation :

En ce qui concerne la partie de définition des variables d'environnement, si les sources du noyau sont sous /usr/src/linux, procéder comme suit :

```
export KERNEL_HEADER_BASE=/usr/src
export KERNEL_DIR=linux
mkdir –p /opt/scali/scascibuild/kernel
cd /opt/scali/scascibuild/kernel
/opt/scali/kernel/rebuild/configure
```

pour un fichier version.h situé dans : /usr/src/linux/include/linux

La suite ne diffère pas de la documentation originale :

```
make
make install
/opt/scali/sbin/scireload  (sur chaque nœud)
/opt/scali/sbin/SSPinstall
```

3) Installation : 2ème passe

Pour la deuxième passe de l'installation, relancer l'installation :

```
/opt/scali/sbin/SSPinstall
```

- A l'étape « **Do you want to upgarde** », répondre « **y** »
- Accepter ensuite les réponses par défaut, sauf pour l'installation de OpenPBS (répondre « **y** », puis fournir le username « linux »).
- A l'étape « **Checking kernel versions** » un « WARNING » est affiché, répondre « **i** » pour l'ignorer.
- A l'étape « **Checking HW support** » un « WARNING » est affiché, répondre « **i** » pour l'ignorer
- A l'étape « **Checking OS support** » un « WARNING » est affiché, répondre « **i** » pour l'ignorer

Lors de l'installation 2ème phase, seuls les trois « WARNING » ci-dessus doivent être affichés. Ils doivent être ignorés car sans conséquences sur le bon fonctionnement du système.

Contrôle de l'installation:

L'installation peut très facilement être testée par les commandes (plus aucune erreur ne doit être affichée) :

```
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
SSPinstall -v
SSPinstall -t
```

Les performances peuvent être évaluées par la commande :

```
export MPI_HOME=/opt/scali
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
```

```
mpirun –np 2 /opt/scali/examples/bin/bandwidth
```

- La latence dans le fichier devrait tourner aux alentours de 5 us pour un message de 0 octets et la bande passante au dessus de 320 MB/s pour de grands messages

IMPORTANT : à ce stade, on peut définir un fichier définissant les variables d'environnements permettant d'utiliser les outils suivants :

- compilateurs Intel v7.0
- librairies mathématiques Intel
- ScaMpi de SCALI

```
. /opt/envhpc/intel/compiler70/ia64/bin/efcvars.sh
. /opt/envhpc/intel/compiler70/ia64/bin/eccvars.sh
export MPI_HOME=/opt/scali
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
export
        LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:/opt/scali/lib:$LD
        _LIBRARY_PATH
export MANPATH=/opt/scali/man:$MANPATH
```

Nom proposé : ENV_VAR_SCALI
Un exemple de ce fichier est proposé sur ce CD « Bull ENH Open Source » dans le répertoire VARIABLES_ENVIRONNEMENT (voir ci-dessous le paragraphe « Installation Logicielle »)
Le copier dans /opt/envhpc

Procédure pour désinstaller:

- Sur le frontal : Faire la commande :

```
/opt/scali/sbin/SSPuninstall
```

- Sur tous les nœuds : Supprimer le répertoire /opt/scali (sur tous les noeuds) et certains fichiers sous /tmp :

```
rm -rf /opt/scali
rm /tmp/SSP_3_0_1*
```

3.3 SCALI SSP 3.1.0

Introduction :
Ce paragraphe décrit l'installation de SCALI SSP 3.1.0.
Vous devez consulter le site http://www.scali.com afin de vérifier l'existence de nouvelles mises à jour.

Pré-requis :

- Linux RedHat et le noyau déjà installés sur chaque nœud
- Connexion des câbles comme indiqué dans le « SCALI system guide » .
- Bien que le logiciel SSP_3_1_0 soit fourni sur un CD ROM, il est aussi possible de le télécharger à partir de http://www.scali.com/Download/ssp.shtml.
  Si vous souhaitez télécharger le logiciel :

```
mkdir /home/scali
cd /home/scali
### Télécharger SSP_3_1_0_Linux2_ia64.tar.gz  ###
tar xvzf SSP_3_1_0_Linux2_ia64.tar.gz
```

Le nom du nœud frontal doit être « admin » : Vérifier que la commande « uname –n » affiche « admin ».

Configuration de l'Operating System comme indiqué dans le fichier OS (dans le sous-répertoire SSP_3_1_0/doc/SSP). **Sur tous les nœuds du cluster** :

- Insérer tous les noms de nœuds avec leur adresse IP sur le réseau local dans le fichier */etc/hosts :*

```
<adresse IP> admin
<adresse IP> n0101
<adresse IP> n0102
<adresse IP> n0103
etc …
```

- Insérer tous les noms de nœuds avec le login *root* dans le fichier */root/.rhosts :*

```
admin root
n0101 root
n0102 root
n0103 root
etc …
```

- Insérer tous les noms de nœuds dans le fichier */etc/hosts.equiv :*

```
admin
n0101
n0102
n0103
etc …
```

- Rajouter *rsh*, *rlogin* et *login* dans le fichier */etc/securetty*
- Exécuter les commandes :

```
chkconfig rsh on
chkconfig rlogin on
```

- Vérifier qu'il reste moins 200 Méga Octets de disponibles sur /opt/scali sur tous les nœuds.
- Avoir reçu une licence provisoire par email et l'avoir copiée sur le frontal (fichier /opt/license.dat)

Installation :

L'installation du logiciel d'interconnexion SSP de SCALI nécessite plusieurs passes :

1. Une première passe pour installer la licence, définir les nœuds (admin, n0101, n0102, n0103 etc …) et les fonctionnalités à installer (OpenPBS) ou à ne pas installer (Console Server, Power Switch Server, NIS/YP).
2. Une phase intermédiaire de mise à jour du driver SCI
3. Une deuxième passe pour vérifier et tester l'installation

1) Installation : 1ère passe

- A partir du CD ROM (sur le noeud admin), faire la commande :

```
mount –t iso9660 /dev/cdrom /mnt/cdrom
cd /mnt/cdrom
./install
```

- Ou, à partir des fichiers issus du téléchargement, faire la commande :

```
/home/scali/SSP_3_1_0/install
```

- Suivre la procédure d'installation en répondant aux questions. La durée de certaines étapes peut être longue.
- A l'étape « **Expert mode** » faire le choix par défaut (« **n** »).
- A l'étape « **Specifying node names** », énumérer les nœuds (dans l'ordre : admin, n0101, n0102, n0103 pour 4 nœuds).
- A l'étape « **Setup license(s)** » faire le choix « **2**) License file » et donner le chemin complet « /opt/license.dat ».
- A l'étape « **Determine node categories** », si tous les nœuds sont des nœuds de calcul, faire le choix par défaut (« **y** »). Sinon, il est possible d'exclure certains nœuds qui ne sont pas des nœuds de calcul (par exemple des nœuds de stockage). Pour ce faire :
  - A la question « Do you accept the above configuration ? », il faut répondre « n ».
  - Sélectionner les nœuds à modifier
  - Ne sélectionner que la catégorie « eth_node » pour les nœuds à modifier
- A l'étape « **install the console server** » répondre « **n** ».
- A l'étape « **install the power switch server** » répondre « **n** ».
- A l'étape « **install OpenPBS queue system** » répondre « **y** », puis renseigner le username « **linux** ».
- Les trois étapes suivantes de l'installation peuvent afficher des « WARNING ». Il ne faut pas en tenir compte et il faut répondre « i » (ignore). Il s'agit des étapes :
  - « Checking kernel versions »
  - « Checking HW support »

    - « Checking OS support »
- A l'étape « **Are you using NIS/YP** .. » répondre « **n** »
- A l'étape « **Configuring SCI network** », pour un cluster 4 nœuds, choisir « 2D torus », puis donner le nombre de nœuds (2) connectés sur les anneaux de l'interface L0, puis donner le nombre de nœuds (2) connectés sur les anneaux de l'interface L1. On a bien 2 x 2 = 4 (nombre total de nœuds). Puis choisir par défaut la méthode automatique (choix « **1** »).
- Parmi les étapes suivantes de l'installation, certaines émettent des « WARNING ». Ceux-ci ne sont pas anormaux lors de la 1ère passe du logiciel d'installation et il faut répondre « i » (ignore). Ces problèmes surviennent car le driver SCI ne se charge pas, ce qui n'est pas anormal à ce niveau d'avancement de l'installation. Il s'agit des étapes :
    - « Checking SCI driver »
    - « Testing MPI communication »
    - « Testing OpenPBS »

2) Phase intermédiaire : Mise à jour du module SCI:

Scali installe tout ce qu'il faut sur le frontal et sur chacun des nœuds du cluster et teste si tout fonctionne (communication entre les nœuds). Cependant, le driver SCI fourni par défaut peut ne pas être adapté au noyau installé (voir les « WARNING » affichés lors de la 1ère phase d'installation) et ces tests ne peuvent pas fonctionner. Il faut mettre à jour ce driver par la procédure de « Mise à jour du module SCI » présentée ci-dessous.

Mise à jour du module SCI:

Scali propose un outil (ScaSCIadap) pour adapter le driver SCI à un noyau différent du noyau supporté par défaut .

Pour utiliser cet outil, lire la documentation /opt/scali/doc/ScaSCIadap/README. Cependant voici une petite précision concernant cette documentation :

En ce qui concerne la partie de définition des variables d'environnement, si les sources du noyau sont sous /usr/src/linux, procéder comme suit :

```
export KERNEL_HEADER_BASE=/usr/src
export KERNEL_DIR=linux
mkdir –p /opt/scali/scascibuild/kernel
cd /opt/scali/scascibuild/kernel
/opt/scali/kernel/rebuild/configure
```

pour un fichier version.h situé dans : /usr/src/linux/include/linux
La suite ne diffère pas de la documentation originale :

```
make
make install
/opt/scali/sbin/scireload  (sur chaque nœud)
```

3) Installation : 2ème passe

L'installation est très facilement vérifiée et testée par les commandes (plus aucune erreur ne doit être affichée) :

```
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
SSPinstall -v
SSPinstall -t
```

Les performances peuvent être évaluées par la commande :

```
export MPI_HOME=/opt/scali
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
cd /tmp
mpirun –np 2 /opt/scali/examples/bin/bandwidth
```

La latence dans le fichier devrait tourner aux alentours de 5 us pour un message de 0 octets et la bande passante au dessus de 320 MB/s pour de grands messages

Les vérifications et les tests peuvent être activés à tout moment pour vérifier le bon fonctionnement de l'interconnexion.

IMPORTANT : à ce stade, on peut définir un fichier définissant les variables d'environnements permettant d'utiliser les outils suivants :

- compilateurs Intel v7.0
- librairies mathématiques Intel
- ScaMpi de SCALI

```
. /opt/envhpc/intel/compiler70/ia64/bin/efcvars.sh
```

```
. /opt/envhpc/intel/compiler70/ia64/bin/eccvars.sh
export MPI_HOME=/opt/scali
export PATH=/opt/scali/bin:/opt/scali/sbin:$PATH
export
        LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:/opt/scali/lib:$LD
        _LIBRARY_PATH
export MANPATH=/opt/scali/man:$MANPATH
```

Nom proposé : ENV_VAR_SCALI
Un exemple de ce fichier est proposé sur ce CD « Bull Extension Pack For HPC Linux » dans le répertoire VARIABLES_ENVIRONNEMENT (voir ci-dessous le paragraphe « Installation Logicielle »)
Le copier dans /opt/envhpc

Procédure pour désinstaller:

- Sur le frontal : Faire la commande :

```
/opt/scali/sbin/SSPuninstall
```

- Sur tous les nœuds : Supprimer le répertoire /opt/scali (sur tous les noeuds) et certains fichiers sous /tmp :

```
rm -rf /opt/scali
rm /tmp/SSP_3_1_0*
```

3.4 Configuration du switch 3COM Gigabit

Pour que GANGLIA fonctionne correctement, il faut que le switch utilisé supporte le multicast IGMP sinon seuls les premiers paquets du multicast passent, ce qui donne l'impression que GANGLIA fonctionne correctement au démarrage, puis chaque nœud perd toute visibilité des autres nœuds.

Ce problème a été mis en évidence sur le switch 3COM Gigabit pour lequel il a fallu reconfigurer le multicast IGMP, en invalidant cette fonction puis en la revalidant pour que cela fonctionne normalement.

3.4.1 Ci dessous le mode opératoire utilisé pour accomplir cette manoeuvre d'après la documentation du switch

Pour configurer le switch, il faut se connecter sur la ligne série en face avant du boitier Gigabit. Une des lignes série "minicom" (par exemple celle de n0101) a été utilisée pour cette connexion.

Après avoir entré la commande "minicom n0101", la connexion a été configurée en 9600 bauds. En tapant sur "Return" on accède au login ("monitor", password "monitor").

Dans les menus proposés, trouver l'option qui autorise le multicast IGMP, l'invalider puis la revalider.

On peut en profiter pour configurer aussi l'adresse IP du switch (par exemple 172.16.12.254, masque 255.255.255.0) pour accéder plus tard à la configuration du switch par telnet ou à travers une interface web sur le port 80.

Chapitre 4. Installation logicielle, lancement

Suit une aide à l'installation de logiciels Open Source à partir du CD « **Bull Extension Pack for HPC Linux V2.0 June 2003 (ref : 76 741 190-001)** » ou directement à partir des « **CDs ou des sites fournisseurs** ».
L'installation de ces logiciels est identique sur NovaScale 4040, NovaScale 5080, NovaScale 5160 mono-noeud ou en mode cluster.
Les logiciels dont le nom est souligné ne sont utiles qu'en mode cluster.

			Open Source	Produit commercialisé
DEVELOPPEMENT D'APPLICATION	**Bibliothèques scientifiques**		Fftw, Petsc	Libmkl (Intel)
	Librairies parallèles	Mpi	Mpich, Lam-mpi	
		Autres	PVM	
	Compilateurs (C, C++, Fortran)		GNU	Intel
OUTILS	**Exploitation**	Répartition de tâches	OpenPBS, Maui,	
		Debogueurs	gdb	Intel idb TotalView(Etnus)
		Profiling	Cprof/vprof	Vampir (Pallas)
		Analyses de performances	Papi, pfmon, perfometer	
	Administration Cluster	Shell distribué	gexec	
		Contrôle et monitoring	Ganglia	
OS	**Administration Système**		Webmin, Nagios	
	Système d'exploitation		Linux	

L'environnement de compilation utilisé est celui des compilateurs Intel.

4.1 Installation du système d'exploitation

Cf. document « **Linux for HPC : Guide d'installation (ref : 86 F2 37EG 01)** »

4.2 Installation des composants GNU

La distribution de base Linux installe les composants GNU (gcc, g77, gdb, pthreads…)

4.3 Installation automatisée des composants logiciels qui sont sur le CD Bull Extension Pack For HPC Linux

Il faut dans un premier temps, monter le CD-ROM :

```
mount -t iso9660 /dev/cdrom /mnt/cdrom

cd /mnt/cdrom
```

Lancer le script d'installation automatique des composants Open Source :

```
./Bull_installer
```

Voici la liste des composants qui sont installés de manière automatique par le script Bull_installer :

- **MPICH / HPL**
- **FFTW**
- **LAM_MPI**
- **MAUI**
- **NAGIOS**
- **PETSC**
- **PROFILING**
- **PVM**

Il est à noter que cette installation automatisée s'applique à une machine « standalone ». Pour une installation en mode « cluster », se reporter au paragraphe « *Installation manuelle des composants logiciels qui sont sur le CD Bull* »

Pour les autres composants, se reporter au paragraphe « *Installation manuelle des composants logiciels qui sont sur le CD Bull* »

Utilisation :

1. Donner le chemin d'installation racine des compilateurs C et Fortran Intel.
   Par défaut, le script considère qu'ils sont installés dans /opt/envhpc/intel.

   Note : l'installation de ces packages Open Source ne pourra avoir lieu si les compilateurs C et Fortran Intel n'ont pas été installés

2. Donner le chemin d'installation racine de la librairie mathématique Intel.
   Par défaut, le script considère qu'elle est installée dans /opt/envhpc/intel/mkl

   Note : Si cette librairie mathématique Intel n'est pas installée, certains tests ne seront pas effectués. Par exemple, une application comme Linpack (HPL), ou encore la librairie PETSC ne seront pas installées et testées.

3. Donner le chemin où doivent être installés les packages HPC Open Source.
   Par défaut, le script considère qu'ils sont installés dans /opt/envhpc

   Note : la plupart des packages seront installés dans ce répertoire.

4. Ensuite, le script demande pour chacun des packages Open Source s'il doit les installer :
   répondre par "y" ou "n" pour chacun d'entre eux

   Do you want to install MPICH ? y/n
   Do you want to install FFTW ? y/n
   Do you want to install LAM_MPI ? y/n
   Do you want to install MAUI ? y/n
   Do you want to install NAGIOS ? y/n
   Do you want to install PETSC ? y/n
   Do you want to install PROFILING ? y/n
   Do you want to install PVM ? y/n

   Cas particuliers :

   - Si l'on souhaite installer MAUI, des questions complémentaires vont être posées :

     A release of PBS has to be installed in order to install MAUI
     Did you install PBS on your node ? y/n

Si PBS n'a pas déjà été installé, alors MAUI ne pourra pas être installé.

Where did you installed your PBS release ? [/usr/local]

Si PBS a été installé au préalable, donner le chemin d'installation racine.
Par défaut, le script considère qu'il est installé dans /usr/local.

- Pour l'installation de FFTW et PETSC :

Une version 1.2.5 de mpich est nécessaire pour l'installation de ces packages. Par conséquent, dans le cas où vous ne sélectionnez pas MPICH dans les composants à installer, des questions complémentaires vont être posés :

A release of MPICH has to be installed in order to install FFTW
Did you install MPICH on your node ? y/n

Si Mpich version 1.2.5 n'a pas déjà été installé, alors FFTW et PETSC ne pourront pas être installés.

Where did you installed your MPICH release ? [/opt/envhpc/mpich-1.2.5]

Si Mpich version 1.2.5 a été installé au préalable, donner le chemin d'installation racine.
Par défaut, le script considère qu'il est installé dans </opt/envhpc/mpich-1.2.5>.

5. Ensuite taper sur la touche "ENTER" pour lancer l'installation de chacun des packages selectionnés.

6. Pour chacun des packages, voici la démarche :

   Exemple : le package PVM

   - Un message indique que l'installation commence :

     pvm3 Installation is beginning ... It can take a while ...

   - Ensuite, le script affiche quelques instructions à suivre pour tester que l'installation s'est bien déroulée :

   **** IN ORDER TO TEST pvm3 :*
   Launch the following command :
   . /opt/envhpc/ENV_VAR_PVM
   Then, Launch : pvm
   If there is no error message, you can exit by :
   quit
   You should have the following message :
   Console: exit handler called
   pvmd still running.

- Un message indique que l'installation est terminée :

  ... pvm3 Installation is ending

- Un dernier message indique le fichier de log à vérifier pour suivre le déroulement de l'installation qui vient d'avoir lieu :

  You can read /tmp/bull/pvm3.log for more information

- Quand tous les packages ont été installés, un message indique que les installations sont terminés :

  Installation complete in /opt/envhpc !!

4.4 Préparation de l'environnement de développement HPC pour une installation manuelle

Dans le cas d'un cluster, il est nécessaire d'avoir effectué l'installation du cluster ,en l'occurrence SCALI, avant d'entamer la préparation de l'environnement HPC. Dès lors, les opérations d'installation et de lancement sont à effectuer sur le nœud choisi pour administrer le cluster. Dans ce document, les exemples sont donnés pour une configuration de 4 nœuds dont le noeud administrateur est appelé « admin » et les autres nœuds n0101 n0102 n0103.

Dans la suite de ce chapitre, quand non précisé, il n'y a pas de différence entre une installation sur une mono-machine et sur le nœud administrateur d'un cluster.

La première étape consiste à désinstaller le package rpm LAM-MPI s'il est installé par la distribution :

```
rpm –e lam-6.5.4-1
```

De façon à centraliser les logiciels de développements (compilateurs, librairies, outils) nous allons créer le répertoire /opt/envhpc sur la machine ou sur le serveur dans le cas d'un cluster (par défaut le nœud d'administration).

Configuration pour une seule machine :

Créer le répertoire

```
mkdir /opt/envhpc
```

Configuration pour un cluster :

L'accès au répertoire /opt/envhpc par les autres nœuds (clients) se fera par montage nfs. Un montage nfs de /home/packages_sources est aussi conseillé afin de pouvoir exécuter les différents tests d'installation

Pour cela :

Sur le serveur (admin) :

Créer le répertoire

```
mkdir /opt/envhpc
```

Mettre dans /etc/exports

```
/opt/envhpc n0101(rw,no_root_squash) n0102(rw,no_root_squash) n0103(rw,no_root_squash)
/home/packages_sources n0101(rw,no_root_squash) n0102(rw,no_root_squash) n0103(rw,no_root_squash)
```

Relancer le serveur NFS

```
service nfs stop
service nfs start
```

Afin de relancer le serveur nfs au démarrage de la machine

```
chkconfig --level 35 nfs on
```

Sur les machines clientes (n0101….) :

Créer le répertoire

```
mkdir /opt/envhpc
```

Mettre dans /etc/fstab

```
admin:/opt/envhpc /opt/envhpc nfs rsize=8192,wsize=8192,intr 0 0
admin:/home/packages_sources /home/packages_sources nfs
        rsize=8192,wsize=8192,intr 0 0
```

Faire le montage correspondant

```
mount -a
```

4.5 Installation manuelle des composants logiciels qui sont sur le CD Bull

Ces composants logiciels sont livrés via le CD-ROM **« Bull Extension Pack For HPC Linux »**

- Il faut donc dans un premier temps, monter le CD-ROM :

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

- Ensuite créer un répertoire qui contiendra les logiciels sources :

```
mkdir /home/packages_sources
```

- Copier les logiciels sources dans ce répertoire nouvellement créé :

```
cp -arf /mnt/cdrom/* /home/packages_sources
```

- Aller dans ce répertoire pour installer les différents packages :

```
cd /home/packages_sources
```

CERTAINS LOGICIELS SONT PRÉSENTS SOUS FORME DE SOURCES ET DOIVENT ÊTRE RECOMPILÉS AVEC DES OPTIONS ÉVENTUELLEMENT MODIFIÉES PAR L'INSTALLATEUR. IL EST DONC NÉCESSAIRE D'INSTALLER EN PREMIER :

- LES COMPILATEURS
- LES LIBRAIRIES NUMÉRIQUES
- LES LIBRAIRIES DE COMMUNICATIONS PARALLÈLES

4.6 Compilateurs Intel

Compilateur Fortran :

Exemple d'installation du compilateur FORTRAN 7.0.0.64 avec licence l_for_03387691.lic

- Créer un répertoire « licenses », s'il n'existe pas déjà, à l'endroit où on va installer le compilateur intel. Par exemple, si l'on compte installer le compilateur Intel dans /opt/envhpc/intel, créer un répertoire /opt/envhpc/intel/licenses :

```
mkdir -p /opt/envhpc/intel/licenses
```

- Installer les licences dans ce répertoire licenses . Par exemple :

```
cp l_for_03387691.lic /opt/envhpc/intel/licenses/
```

- Désarchiver le fichier .tar via la commande. Par exemple :

```
tar xvf l_fc_p_7.0.064.tar
```

- Installer le compilateur Fortran sur la distribution Linux installée via la commande install :

```
export INTEL_LICENSE_FILE=.:/opt/envhpc/intel/licenses
./install
```

- Choisir « 2 » pour sélectionner l'installation sur une plateforme « Itanium®-based system »
- Choisir « 1 » pour installer le compilateur fortran itanium
- Après le « license agreement », taper « accept »
- Choisir son chemin pour l'installation (par défaut : /opt/intel), par exemple, /opt/envhpc/intel
- Accepter les options d'installation
- Ainsi, un répertoire compiler70 est créé dans ce chemin
- X pour quitter

- Vérifier que <***chemin d'installation***>/compiler70/ia64/bin/efcvars.sh pointe bien vers les bons chemins puis l'exécuter :

```
. <chemin d'installation>/compiler70/ia64/bin/efcvars.sh
```

- Le compilateur fortran est désormais disponible sous le nom de efc

Test de l'installation :

- Exécuter la commande

```
efc
```

- Le résultat suivant doit s'afficher

```
/usr/lib/crt1.0 : In function `_start' :
/usr/lib/crt1.0(.text+0x41) : undefined reference to `main'
```

Compilateur C/C++ :

Exemple d'installation du compilateur C/C++ 7.0.0.65 avec licence l_cpp_40599793.lic

- Créer un répertoire « licenses », s'il n'existe pas déjà, à l'endroit où on va installer le compilateur intel. Par exemple, si l'on compte installer le compilateur Intel dans /opt/envhpc/intel, créer un répertoire /opt/envhpc/intel/licenses :

```
mkdir –p /opt/envhpc/intel/licenses
```

- Installer les licenses dans ce répertoire licenses . Par exemple :

```
cp l_cpp_40599793.lic /opt/envhpc/intel/licenses/
```

- Désarchiver le fichier tar via la commande. Par exemple :

```
tar xvf l_cc_p_7.0.065.tar
```

- Installer le compilateur C/C++ sur la distribution Linux installée via la commande install :

```
./install
```

- Choisir « 2 » pour sélectionner l'installation sur une plateforme « Itanium®-based system »
- Choisir « 1 » pour installer le compilateur C/C++ itanium
- Après le « license agreement », taper « accept »
- Choisir son chemin pour l'installation (par défaut : /opt/intel), par exemple, /opt/envhpc/intel
- Accepter les options d'installation
- Ainsi, un répertoire compiler70 est créé dans ce chemin
- X pour quitter
- Vérifier que <***chemin d'installation***>/compiler70/ia64/bin/eccvars.sh pointe bien vers les bons chemins puis l'exécuter :

```
. <chemin d'installation>/compiler70/ia64/bin/eccvars.sh
```

- Le compilateur C/C++ est désormais disponible sous le nom de ecc

Test de l'installation :

- Exécuter la commande

```
efc
```

- Le résultat suivant doit s'afficher

```
/usr/lib/crt1.0 : In function `_start' :
/usr/lib/crt1.0(.text+0x41) : undefined reference to `main'
```

Intel Debugger :

Première étape :

Le package permettant d'installer ce debugger est situé dans l'une ou l'autre des archives tar Fortran ou C

Seconde étape : Installation

Après installation du compilateur C, par exemple :

- Installer le debugger Intel sur la distribution Linux installée, via la commande install :

```
./install
```

- Choisir " 2 " pour sélectionner l'installation sur une plateforme " Itanium®-based system "
- Choisir " 2 " pour installer le debugger Intel
- Après le " license agreement ", taper " accept "
- Choisir son chemin pour l'installation (par défaut : /opt/intel), par exemple, /opt/envhpc/intel
- Accepter les options d'installation
- Ainsi, un répertoire compiler70 est créé dans ce chemin
- X pour quitter

- Vérifier que **<chemin d'installation>compiler70/ia64/bin/efcvars.sh** pointe bien vers les bons chemins puis l'exécuter pour avoir le debugger Intel idb dans le PATH :

```
. <chemin d'installation>/compiler70/ia64/bin/efcvars.sh
```

Le debugger Intel est désormais disponible sous le nom de idb

Test de l'installation :

Exécuter la commande :

```
idb
```

Le résultat suivant doit s'afficher

```
(idb)
```

Pour quitter faire

```
quit
```

4.7 MPICH 1.2.5

- Récupérer l'archive mpich.tar sur le site :

http://www-unix.mcs.anl.gov/mpi/mpich/download.html

ou encore :

```
cd /home/packages_sources/MPICH
```

- Désarchiver le fichier :

```
tar xvfz mpich.tar.gz
cd mpich-1.2.5
```

Pour l'installation proprement dite, suivre les instructions du README ainsi que les fichiers du répertoire doc situé dans le répertoire mpich-1.2.5.

Cependant, voici une petite aide pour compiler rapidement votre version de Mpich :

Première étape :

Note : les compilateurs **efc** et **ecc** devront pour cela être dans votre variable d'environnement PATH (cf. section « Compilateurs Intel V7.0 »)

Pour utiliser mpich sur une machine SMP seule, il faut compiler mpich avec le device ch_shmem :

```
./configure --prefix=<chemin d'installation> --with-device=ch_shmem
-cc=ecc -clinker=ecc -fc=efc -flinker=efc -f90=efc -f90linker=efc
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/mpich-1.2.5

En revanche, si l'on souhaite compiler mpich pour un cluster de machines SMP, on choisira le device ch_p4 :

```
./configure --prefix=<chemin d'installation> --with-device=ch_p4
-comm=shared -cc=ecc -clinker=ecc -fc=efc -flinker=efc -f90=efc
-f90linker=efc
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/mpich-1.2.5

Pour connaître la signification de chacun de ces flags, veuillez vous reporter à la documentation fournie.

Cependant, on peut résumer en disant que cela va permettre de construire des outils mpich (mpirun, mpicc, mpif90 …) utilisant la mémoire partagée et encapsulant les compilateurs ecc et efc de intel.

Deuxième étape :

```
make
```

(construction des outils qui utilisent la configuration précédente)

Dernière étape :

```
make install
```

Dans notre cas, cela installera les binaires mpich dans <chemin d'installation>/mpich-1.2.5, par exemple /opt/envhpc/mpich-1.2.5

- Pour utiliser ensuite les outils mpich, il faut au préalable effectuer les opérations suivantes :

```
export PATH=<chemin d'installation>/mpich-1.2.5/bin:$PATH
```

Note :
Lorsque l'installation se fait avec le device ch_p4, il faut modifier le fichier <*chemin d'installation*>/mpich-1.2.5/share/machines.LINUX et y indiquer le nom des machines ainsi que le nombre de processeurs sur chacune d'elles.

Par exemple :

```
admin:4
n0101:4
n0102:4
n0103:4
```

Tests :

On peut alors tester basiquement notre configuration :
1.

```
cd <chemin d'installation>/mpich-1.2.5/examples
make cpi
```

création du binaire cpi que l'on peut exécuter via la commande :

- pour une machine quatre processeurs

```
mpirun -np 4 ./cpi
```

- pour une machine 16 processeurs (ou un cluster 4x4)

```
mpirun -np 16 ./cpi
```

2.

```
make pi3f90
```

création du binaire pi3f90 que l'on peut executer via la commande :

- pour une machine 4 processeurs

```
mpirun -np 4 ./pi3f90
```

- pour une machine 16 processeurs (ou un cluster 4x4)

```
mpirun -np 16 ./pi3f90
```

Saisir le nombre d'intervalles, par exemple 10000, ou 0 pour sortir de l'application.

IMPORTANT : à ce stade, on peut créer un fichier définissant les variables d'environnements permettant d'utiliser les outils suivants :

- compilateurs Intel v7.0
- librairies mathématiques Intel
- mpich-1.2.5

Exemple :

```
. /opt/envhpc/intel/compiler70/ia64/bin/efcvars.sh
. /opt/envhpc/intel/compiler70/ia64/bin/eccvars.sh
export PATH=/opt/envhpc/mpich-1.2.5/bin:$PATH
export
        LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_
        PATH
export MANPATH=/opt/envhpc/mpich-1.2.5/man:$MANPATH
export P4_GLOBMEMSIZE=260000000
```

Nom proposé : ENV_VAR_MPICH
Un exemple de ce fichier est proposé dans
/home/packages_sources/VARIABLES_ENVIRONNEMENT

Le copier dans /opt/envhpc

On peut désormais tester les premiers packages HPC installés avec l'application HPL (cf. section suivante)

4.8 Librairies mathématiques

4.8.1 *Libmkl*

Exemple d'installation mkl 6.0

L'installation doit être lancée sous root.

Copier le fichier de licence l_mkl_xxxxxxxx.lic dans le répertoire /opt/envhpc/intel/licenses

Décompresser et désarchiver le fichier d'archive:

```
tar xvf l_mkl_p_6.0.011.tar
```

Procéder à l'installation en lançant :

```
cd  l_mkl_p_6.0.011
```

```
./install.sh
```

Le chemin d'accès à la licence est /opt/envhpc/intel/licenses.
Le chemin d'installation à préciser est /opt/envhpc/intel

Afin d'être indépendant du numéro de version de la bibliothèque, il est recommandé de faire le lien symbolique suivant :

```
ln –f –s /opt/envphc/intel/mkl60 /opt/envhpc/intel/mkl
```

Note : Pour pouvoir se servir des librairies dynamiques, il faut penser à inclure le répertoire dans la variable d'environnement LD_LIBRARY_PATH.

```
export LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_PATH
```

Note : Dans le cas d'un cluster, il est indispensable de positionner cette variable LD_LIBRARY_PATH dans l'environnement des login sur tous les nœuds sinon l'exécution d'un programme utilisant la libmkl sur plusieurs nœuds s'arrêtera avec le message suivant :

error while loading shared libraries: libmkl_itp.so: cannot open shared object file: No such file or directory

On peut, par exemple, insérer l'initialisation de LD_LIBRARY_PATH dans le fichier /etc/bashrc.

Test de l'installation :
L'exécution du benchmark HPL permet la vérification de l'installation.

4.8.2 FFTW

Récupérer l'archive sur le site :

http://www.fftw.org/

ou encore :

```
cd /home/packages_sources/FFTW
```

Désarchiver le fichier :

```
tar xvfz fftw-2.1.3.tar.gz
cd fftw-2.1.3
```

Première étape :

Note : les compilateurs **efc** et **ecc** devront pour cela être dans votre variable d'environnement PATH (cf. section « Compilateurs Intel V7.0 »)

Pour utiliser avec SCALI:

```
export MPICC='ecc -D_REENTRANT -I/opt/scali/include -L/opt/scali/lib'
export CC=ecc
export CFLAGS=-O2
export F77=efc
./configure --prefix==<chemin d'installation> -host=i786 --enable-mpi.
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/fftw-2.1.3

En revanche, si l'on souhaite compiler fftw-2.1.3 avec mpich :

Charger l'environnement MPICH via le fichier /opt/envhpc/ENV_VAR_MPICH

```
export CC=ecc
export CFLAGS=-O2
export F77=efc
```

```
./configure --prefix=<chemin d'installation>  -host=i786 --enable-mpi
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/ fftw-2.1.3

Seconde étape :

```
make
make install
```

(construction des outils qui utilisent la configuration précédente)

On peut alors tester basiquement notre configuration en mono processeur :

```
cd  tests
./fftw_test -s 100
```

On peut alors tester basiquement notre configuration avec mpi

```
cd ../mpi
mpirun -np 4 ./fftw_mpi_test -s 100
```

4.8.3 PETSC

Récupérer l'archive sur le site :

http://www-fp.mcs.anl.gov/petsc/index.html

ou encore :

```
cd /home/packages_sources/PETSC
```

Désarchiver le fichier :

```
tar xvfz petsc.tar.gz
cd petsc-2.1.5
```

Première étape :

Il faut initialiser les variables suivantes

```
export PETSC_ARCH=linux64_intel
export PETSC_DIR = <chemin ou se trouve les sources>/petsc-2.1.5
export LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_PATH
```

Modification du fichier

```
$PETSC_DIR/bmake/linux64_intel/packages
```

Pour utiliser avec MPICH:

Modification du fichier *$PETSC_DIR/bmake/linux64_intel/packages*

```
BLASLAPACK_LIB = -L/opt/envhpc/intel/mkl/lib/64 -lmkl_itp
            -lmkl_lapack /opt/envhpc/intel/mkl/lib/64/libguide.so
#
# Location of MPI (Message Passing Interface) software
#
#MPI_HOME       = /usr/local/vmi/mpich
#MPI_LIB        = -L${MPI_HOME}/lib/ecc -lfmpich -lmpich -lpmpich
#-lvmiquiet -lpthread -ldl
#MPI_INCLUDE    = -I${MPI_HOME}/include
#MPIRUN         =  /home/balay/bin/mpirun
#
MPI_HOME       = /opt/envhpc/mpich-1.2.5
MPI_LIB        = ${MPI_HOME}/lib/libmpich.a
MPI_INCLUDE    = -I${MPI_HOME}/include
MPIRUN         =  ${MPI_HOME}/bin/mpirun
#
```

```
# ----------------------------------------------------------------
#  Locations of OPTIONAL packages. Comment out those you do not have.
# ----------------------------------------------------------------
#
# Location of X-windows software
#
X11_INCLUDE    =
X11_LIB        = -L/usr/X11R6/lib -lX11
PETSC_HAVE_X11 = -DPETSC_HAVE_X11
```

Pour utiliser avec SCAMPI:
Modification du fichier *$PETSC_DIR/bmake/linux64_intel/packages*

```
BLASLAPACK_LIB = -L/opt/envhpc/intel/mkl/lib/64 -lmkl_itp
                -lmkl_lapack /opt/envhpc/intel/mkl/lib/64/libguide.so
#
# Location of MPI (Message Passing Interface) software
#
#MPI_HOME       = /usr/local/vmi/mpich
#MPI_LIB        = -L${MPI_HOME}/lib/ecc -lfmpich -lmpich -lpmpich
#-lvmiquiet -lpthread -ldl
#MPI_INCLUDE    = -I${MPI_HOME}/include
#MPIRUN         =  /home/balay/bin/mpirun
#
MPI_HOME       = /opt/scali
MPI_LIB        = ${MPI_HOME}/lib/libmpi.a
MPI_INCLUDE    = -I${MPI_HOME}/include
MPIRUN         =  ${MPI_HOME}/bin/mpirun
```

```
#
# ----------------------------------------------------------
#  Locations of OPTIONAL packages. Comment out those you do not have.
# ----------------------------------------------------------
#
# Location of X-windows software
#
X11_INCLUDE    =
X11_LIB        = -L/usr/X11R6/lib -lX11
PETSC_HAVE_X11 = -DPETSC_HAVE_X11
```

Modification du fichier *$PETSC_DIR/bmake/linux64_intel/variables*

```
C_CC            = ecc -D_REENTRANT -I/opt/scasli/include
C_CLINKER       = ecc  -L/opt/scali/lib          -lmpi -Wl,-R/opt/scali/lib
C_FLINKER       = efc   -L/opt/scali/lib -lfmpi  -lmpi -Wl,-R/opt/scali/lib
```

Seconde étape :

```
make BOPT=O allclean
make BOPT=O all | tee  make_log
```

(construction des outils qui utilisent la configuration précédente)

Pour l'installation elle se fait manuellement en transférant les fichiers se trouvant dans $PETSC_DIR/lib/libO/linux64_intel dans le répertoire <Chemin d'installation > où <Chemin d'installation> pourra être par exemple /opt/envhpc/petsc-2.1.5/lib

On peut alors tester basiquement notre configuration :

```
make BOPT=O testexamples |tee  examples_log
```

NOTE: Chaque résultat est comparé au résultat attendu par un diff. Avec SCAMPI, mpirun lance mpimon et affiche la commande mpimon lancée. Ainsi tous les tests comportent des différences avec le résultat attendu.
Par exemple :

```
######
> /opt/scali/bin/mpimon -stdin all  ex22  -da_grid_x 10 -nox --  n0101 1
Possible problem with ex22_1, diffs above
######
```

4.9 HPL

HPL est la version Linpack HPC.

Ce benchmark standard permet de mesurer de façon standardisée la scalabilité et les performances d'un cluster.
Il a pour but de résoudre un système linéaire de N équations à N inconnues (Ax=B). Les matrices utilisées sont des matrices denses et le code est constitué d'opérations en calculs flottants en double précision et de communications synchrones. Linpack est écrit en Fortran et utilise des fonctions des librairies Blas et MPI pour les communications.

Il retourne comme résultats :

- un temps d'exécution en secondes
- le nombre maximum d'opérations flottantes effectuées durant une seconde (Gflops)

Pour de plus amples informations :

http://www.netlib.org/benchmark/hpl

Et notamment :

http://www.netlib.org/benchmark/hpl/results.html
http://www.netlib.org/benchmark/hpl/tuning.html
http://www.netlib.org/benchmark/hpl/faqs.html

HPL a besoin :
- D'une implémentation MPI : ici, nous testons avec mpich-1.2.5 puis ScaMPI de Scali
- De la librairie mathématiques BLAS : nous utilisons ici la libmkl d'Intel

Pour récupérer le package :

```
cd /opt/envhpc
```

Décompresser et désarchiver le fichier par :

```
tar xvfz /home/packages_sources/HPL/hpl.tgz
cd hpl
```

Dans le fichier Makefile faire la modification suivante :

```
arch = Itanium
```

Récupérer un exemple de Make.Arch en faisant par exemple :

```
cp setup/Make.Linux_PII_CBLAS Make.Itanium
```

Premier cas : vous souhaitez compiler HPL avec mpich et la libmkl

Faire les modifications suivantes dans Make.Itanium :

```
ARCH = Itanium
Ajouter HOME = /opt/envhpc

CC = mpicc
CCFLAGS =

LINKER = mpif90
LINKFLAGS=$(CCFLAGS) /opt/envhpc/intel/mkl/lib/64/libguide.so

MPdir = /opt/envhpc/mpich-1.2.5
MPinc = $(MPdir)/include
MPlib = -L$(MPdir)/lib –lmpich –lpmpich

LAdir =
LAlib = -L/opt/envhpc/intel/mkl/lib/64 –lmkl_itp

HPL_OPTS =
```

Pour préparer la compilation, exécuter :

```
. /opt/envhpc/ENV_VAR_MPICH
```

Second cas : vous souhaitez compiler HPL avec ScaMPI de SCALI et la libmkl (en étant dans une configuration cluster)

Faire les modifications suivantes dans Make.Itanium :

```
ARCH = Itanium
Ajouter HOME = /opt/envhpc

CC = ecc –D_REENTRANT –I$(MPI_HOME)/include
CCFLAGS =

LINKER = efc
LINKFLAGS=$(CCFLAGS) /opt/envhpc/intel/mkl/lib/64/libguide.so

MPdir = /opt/scali
MPinc = $(MPdir)/include
MPlib = -L$(MPdir)/lib –lmpi

LAdir =
LAlib = -L/opt/envhpc/intel/mkl/lib/64 –lmkl_itp

HPL_OPTS =
```

Pour préparer la compilation, exécuter :

```
. /opt/envhpc/ENV_VAR_SCALI
```

Dans les 2 cas :

```
make clean_arch_all
make : création de l'exécutable xhpl dans ./bin/Itanium
cd bin/Itanium/
Modification de HPL.dat :
```

Rechercher le fichier de paramètres donnant la performance globale du cluster la meilleure : HPL.dat

Précisions :

N = taille du problème à résoudre – trouver le plus grand problème qui remplisse la mémoire physique sans swapper.

NB = taille de bloc – trouver la meilleure valeur demande de nombreuses expérimentations

P*Q : matrice pour résoudre le problème – découpe le problème en P*Q processus linpack. Ce nombre de processus linpack doit correspondre au nombre de nœuds pour être le plus efficient possible.

Remarque :

La librairie mathématique mkl étant parallélisée, il ne faut lancer qu'un seul processus MPI par nœud SMP pour être efficient. En effet, sur une machine SMP quadri-processeurs, quand on lance un processus MPI, la parallélisation de la librairie mathématique va créer 4 instances du programme hpl.

On peut inhiber cette parallélisation de la libmkl via la commande suivante :

```
export OMP_NUM_THREADS=1
```

Cependant, les exemples suivants ont été effectués avec OMP_NUM_THREADS=4 (valeur par défaut sur une machine quadri-processeurs)

Exemples :

MPICH : HPL.dat pour une exécution de HPL sur une machine SMP 4 processeurs :

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
10000  Ns
1            # of NBs
120      NBs
1            # of process grids (P x Q)
1        Ps
1        Qs
16.0         threshold
1            # of panel fact
2        PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
8          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
2        RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

Exécution :

```
mpirun -np 1 xhpl
```

Résultat:

```
“1 tests completed and passed residual checks”
```

MPICH : HPL.dat pour une exécution de HPL sur quatre machines SMP 4 processeurs :

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
10000  Ns
1            # of NBs
120      NBs
             1           # of process grids (P x Q)
4        Ps
1        Qs
16.0         threshold
1            # of panel fact
2        PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
8          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
2        RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

Configuration :

Modifier le .bashrc de l'utilisateur en question sur les nœuds distants en ajoutant la ligne suivante :

```
export LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_PATH
```

Sur le nœud d'administration, modifier **temporairement** le fichier /opt/envhpc/mpich-1.2.5/share/machines.LINUX pour avoir un processus MPI par nœud SMP, par exemple :

```
admin
n0101
n0102
n0103
```

Exécution :

```
mpirun -np 4 xhpl
```

Résultat de l'exécution :

```
"1 tests completed and passed residual checks"
```

Remettre le fichier /opt/envhpc/mpich-1.2.5/share/machines.LINUX dans son état avant la modification.

SCALI : HPL.dat pour une exécution de HPL sur quatre machines SMP 4 processeurs :

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
10000  Ns
1            # of NBs
120      NBs
1            # of process grids (P x Q)
4        Ps
1        Qs
16.0         threshold
1            # of panel fact
2        PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
8          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
```

```
2       RFACTs (0=left, 1=Crout, 2=Right)
1          # of broadcast
0          BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1          # of lookahead depth
1          DEPTHs (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
64          swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U  in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)
```

Exécution :

```
mpimon xhpl – admin 1 n0101 1 n0102 1 n0103 1
```

Résultat de l'exécution:

```
"1 tests completed and passed residual checks"
```

4.10 LAM_MPI 6.5.9

- Récupérer l'archive lam-6.5.9.tar.gz sur le site :

http://www.lam-mpi.org/download

ou encore :

```
cd /home/packages_sources/LAM_MPI
```

- Désarchiver le fichier :

```
tar xvfz lam-6.5.9.tar.gz
cd lam-6.5.9
```

- Pour l'installation proprement dite, suivre les instructions du README ainsi que le fichier INSTALL dans le répertoire lam-6.5.9

- Cependant, voici une petite aide pour compiler rapidement votre version de lam :

Première étape :

Note : les compilateurs efc et ecc devront pour cela être dans votre variable d'environnement PATH (cf. section « Compilateurs Intel V7.0 »)

Pour utiliser LAM sur une machine SMP seule, il faut compiler LAM avec l'option « usysv »

```
./configure --prefix=<chemin d'installation> --with-cc=ecc –-with-cflags=-O2 --with-cxx=ecc --with-cxxflags=-O2 --with-fc=efc –-with-fflags=-O2  --with-rpi=usysv –-with-select-yield
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/lam-6.5.9

Dans le cas de cluster de machines SMP, compiler avec l'option « tcp »

```
./configure --prefix=<chemin d'installation> --with-cc=ecc –-with-cflags=-O2 --with-cxx=ecc --with-cxxflags=-O2 --with-fc=efc  --with-fflags=-O2 --with-rpi=tcp
```

où <Chemin d'installation> pourra être par exemple /opt/envhpc/lam-6.5.9

Pour connaître la signification de chacun de ces flags, veuillez vous reporter à la documentation fournie dans l'archive lam-6.5.9.tar.gz (fichier INSTALL).

Cependant, on peut résumer en disant que cela va permettre de construire des outils LAM (mpirun, mpicc, mpif77 …) utilisant la mémoire partagée et encapsulant les compilateurs ecc et efc de intel.

Deuxième étape :

```
make
```

Troisième étape :

```
make install
```

Dans notre cas, cela installera les binaires lam-mpi dans <chemin d'installation>/lam-6.5.9, par exemple /opt/envhpc/lam-6.5.9

Pour utiliser ensuite les outils LAM-MPI, il faut au préalable effectuer les opérations suivantes :

```
export PATH=<chemin d'installation>/lam-6.5.9/bin:$PATH
```

Dernière étape :

On peut alors compiler les exemples en ajoutant dans le fichier Makefile se trouvant dans lam-6.5.9 l'option –Vaxlib au niveau du flag FFLAGS et en modifiant le fichier */home/packages_sources/LAM_MPI/lam-6.5.9/mpi2c++/contrib/test_suite/signal.cc* :

```
Remplacer la ligne :
        n.sa_mask= 0;
Par la ligne :
        sigemptyset(&n.sa_mask);
```

Enfin :

```
make examples
```

IMPORTANT : à ce stade, on peut créer un fichier définissant les variables d'environnement permettant d'utiliser les outils suivants :

- compilateurs Intel v7.0
- librairies mathématiques Intel
- lam-6.5.9

Exemple :

```
. /opt/envhpc/intel/compiler70/ia64/bin/efcvars.sh
. /opt/envhpc/intel/compiler70/ia64/bin/eccvars.sh
export PATH=/opt/envhpc/lam-6.5.9/bin:$PATH
export LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_PATH
export MANPATH=/opt/envhpc/lam-6.5.9/man:$MANPATH
```

Nom proposé : ENV_VAR_LAM

Un exemple de ce fichier est proposé dans
/home/packages_sources/VARIABLES_ENVIRONNEMENT

Le copier dans /opt/envhpc

Pour exécuter les exemples il faut avant tout être utilisateur « non root » puis exécuter le fichier ENV_VAR_LAM.

```
. /opt/envhpc/ENV_VAR_LAM
```

Pour un cluster, le fichier ENV_VAR_LAM doit en plus être chargé lors du login de l'utilisateur des machines distantes. Pour cela, sur tous les nœuds, modifier le .bashrc de l'utilisateur en question en ajoutant la ligne suivante :

```
. /opt/envhpc/ENV_VAR_LAM
```

Ensuite, lancer le démon LAM via la commande :

Si une seule machine :

```
lamboot
```

Un test simple est le programme fpi se trouvant dans
/home/packages_sources/LAM_MPI/lam-6.5.9/examples/pi sur la machine admin

```
cd /home/packages_sources/LAM_MPI/lam-6.5.9/examples/pi
mpirun –np 4 fpi
```

Si un cluster :

Par exemple, créer un fichier « hostfile » dans le répertoire
/home/packages_sources/LAM_MPI/lam-6.5.9/examples/pi :

```
cd /home/packages_sources/LAM_MPI/lam-6.5.9/examples/pi
#créer le fichier hostfile et y mettre :
admin cpu=4
n0101 cpu=4
n0102 cpu=4
```

```
n0103 cpu=4
```

```
lamboot hostfile
```

Rendre le programme fpi accessible sur les 4 nœuds et lancer l'exécution :

```
cp /home/packages_sources/LAM_MPI/lam-6.5.9/examples/pi/fpi /opt/envhpc/lam-6.5.9/bin
mpirun –np 16 /opt/envhpc/lam-6.5.9/bin/fpi
```

A la fin des tests on arrête le démon LAM par :

```
lamhalt
```

4.11 PVM 3.4.4

- Aller sur le site http://www.netlib.org/pvm3/index.html et télécharger pvm3.4.4.tgz ou le récupérer sur le CD Bull
- Dézipper et désarchiver le fichier par :

```
cd /opt/envhpc
tar xvfz /home/packages_sources/PVM/pvm3.4.4.tgz
```

- PVM s'est installé sous le répertoire pvm3.

- Création des variables d'environnement.
  Pour cela, il faut créer le script ENV_VAR_PVM (dans /opt/envhpc par exemple) :

```
. /opt/envhpc/intel/compiler70/ia64/bin/efcvars.sh
. /opt/envhpc/intel/compiler70/ia64/bin/eccvars.sh
export LD_LIBRARY_PATH=/opt/envhpc/intel/mkl/lib/64:$LD_LIBRARY_PATH

export PVM_ROOT=/opt/envhpc/pvm3
export PVM_ARCH=LINUX64
```

```
export PATH=$PVM_ROOT/lib/$PVM_ARCH:$PVM_ROOT/bin/$PVM_ARCH:$PATH
```

Un exemple de ce fichier est proposé dans /home/packages_sources/VARIABLES_ENVIRONNEMENT

- Exécution des variables d'environnement :

```
. /opt/envhpc/ENV_VAR_PVM
```

- Aller dans le répertoire /opt/envhpc/pvm3 :

```
cd /opt/envhpc/pvm3
```

Pour information : Les différentes architectures proposées sont dans le sous-répertoire conf. Celle qui correspond à notre machine est LINUX64.

- Faire les modifications suivantes dans le fichier Makefile.aimk :

```
CC = ecc
F77 = efc
```

- Ensuite, il suffit de taper :

```
make
```

- Pour lancer PVM et son démon, il faut lancer la commande :

```
pvm
```

Test de l'installation:

- Exécuter

```
pvm
```

- Le prompt suivant doit s'afficher

```
pvm>
```

- Pour quitter, taper :

```
quit
```

- Il s'affiche:

```
Console : exit handler called
pvmd still running
```

4.12 Outils d'analyse de performances et de profiling

Il y a deux types de livraison :

- des packages officiels sous forme de rpm
- des ensembles avec du source sous forme de tar zippes

Les packages :

libpfm-2.0-1.ia64.rpm
pfmon-2.0-1.ia64.rpm

se récupèrent dans l'ordre suivant, avec le droit root:

```
cd /home/packages_sources/PROFILING
rpm –U libpfm-2.0-1.ia64.rpm
rpm –U pfmon-2.0-1.ia64.rpm
```

Pour récupérer les tar,:

```
cd /opt/envhpc
```

Restaurer les archives .tar.gz par :

```
tar xvfz /home/packages_sources/PROFILING/papi-linux-ia64.tar.gz
```

```
tar xvfz  /home/packages_sources/PROFILING/vprof-linux-ia64.tar.gz
```

4.12.1 Pfmon

La documentation de cet outil se trouve sur le site :
http://www.hpl.hp.com/research/linux/perfmon/pfmon.php4

Exécute par exemple, sur la commande ls :

```
pfmon /bin/ls
```

et on obtient :

```
cycles.sh  pfdbg  pfmon

            1 559 994 CPU_CYCLES
```

Il est possible de capturer 4 évènements à la fois .

Il existe aussi une option qui permet de lancer pfmon system-wide pour observer l'ensemble de la machine.

4.12.2 PAPI

```
cd /opt/envhpc/ papi-linux-ia64
```

On trouve des libraries, des include toute une série de fichier man, des tests.

Pour lire les man associés aux fonctions PAPI :

```
cd man
export MANPATH=.:$MANPATH
man PAPI_read
```

Sinon la documentation se trouve sur le site :

http://icl.cs.utk.edu/projects/papi/documents

Pour effectuer un premier test de l'opérabilité de PAPI :

```
cd ../ctests
 ./zero
Test case 0: start, stop.
------------------------------------------------
Default domain is: 1 (PAPI_DOM_USER)
Default granularity is: 1 (PAPI_GRN_THR)
Using 10000000 iterations of c += a*b
-------------------------------------------------------------------------
Test type    :            1
PAPI_FP_INS  :     20000000
PAPI_TOT_CYC :    450008301
Real usec    :       452815
Real cycles  :    450550583
```

```
-------------------------------------------------------------------------
Verification: none
```

```
zero.c                          PASSED
```

Si on veut avoir une trace de debug, on peut positionner la variable PAPI_DEBUG

```
export PAPI_DEBUG= y
```

Si, au contraire, cette trace pollue :

```
unset PAPI_DEBUG
```

Une fois ces vérifications faites, la lecture, dans le directory tests, d'un fichier source aide à comprendre les modifications à faire dans le source de l'application. Il suffit donc de s'en inspirer pour faire les modifications nécessaires dans son programme.

Pour utiliser la bibliothèque PAPI de manière partagée, ne pas oublier d'initialiser la variable d'environnement LD_LIBRARY_PATH :

```
export LD_LIBRARY_PATH=/opt/envhpc / papi-linux-ia64/lib:$LD_LIBRARY_PATH
```

Remarque :
Attention à l'usage des threads pas encore bien compatible avec certaines fonctions de PAPI

4.12.3 VPROF/CPROF

```
cd /opt/envhpc/vprof-linux-ia64
```

La documentation de ces outils est un simple README livre:

vprof-linux-ia64.README

C'est la première chose qu'il faut lire pour comprendre ces outils.

Ensuite, sont livrés des binaires (les commandes) des librairies (pour faire l'édition de lien du programme à « profiler », et quelques exemples permettant de se faire la main.

Les sources de ces produits sont disponibles sur le site :

http://aros.ca.sandia.gov/~cljanss/perf/vprof/

Le Makefile ayant servi à la fabrication des binaires est fourni à titre d'information.

Il est possible, en particulier si on ne s'intéresse qu'a du profiling système, de fabriquer une version de cprof/vprof indépendant de PAPI. Si tel est le cas, repartir des sources du site et lancer la configuration :

```
configure
```

Avec les options désirées.

La version livrée ici nécessite PAPI.

Donc, avant d'utiliser les commandes cprof/vprof, spécifier :

```
export LD_LIBRARY_PATH=/opt/envhpc/papi-linux-ia64/lib:$LD_LIBRARY_PATH
export PATH= /opt/envhpc/vprof-linux-ia64/bin:$PATH
```

Voir au chapitre utilisation la mise en oeuvre d'un scénario de profiling.

4.12.4 VAMPIR

Si VAMPIR est fourni, il est disponible sous /home/packages_sources/VAMPIR, pour l'installer il faut procéder comme suit :

```
mkdir /opt/envhpc/VAMPIR
cd /opt/envhpc/VAMPIR
tar xvfz /home/packages_sources/VAMPIR/VA-IA64-LIN-PRODUCT.3.0.0.6.tar.gz
tar xvfz /home/packages_sources/VAMPIR/VT-IA64-LIN-72-PRODUCT.3.0.0.5.tar.gz
```

- Exécuter :

```
./install-Vampir
```

- Répondre « y » aux questions
- Sauvegarder la licence dans le sous-répertoire etc :

```
cp /home/packages_sources/VAMPIR/license.dat ./etc/
```

- Exécuter :

```
./install-Vampirtrace
```

- Répondre « y » aux questions
- Copier le fichier d'environnement ENV_VAR_VAMPIR dans /opt/envhpc :

```
cp /home/packages_sources/VARIABLES_ENVIRONNEMENT/ENV_VAR_VAMPIR /opt/envhpc
```

Ce fichier définit les variables d'environnement suivantes :

```
export PAL_ROOT=/opt/envhpc/VAMPIR
export PAL_LICENSEFILE=$PAL_ROOT/etc/license.dat
export VAMPIR_ROOT=$PAL_ROOT/bin
export PATH=$PATH:$VAMPIR_ROOT
export VAMPIR_LIB_DIR=$PAL_ROOT/lib
```

Comment obtenir la trace d'un programme?

Remarque: avant de lancer VAMPIR penser à mettre à jour l'environnement : « . /opt/envhpc/ENV_VAR_VAMPIR ».

Avec MPICH :
Dans le Makefile du programme à exécuter, faire la modification suivante à l'édition de liens :

```
-L$(VAMPIR_LIB_DIR) -lVT
```

Compiler puis exécuter le programme. Cela crée un fichier avec une extension bvt : « Vampirtrace INFO: Writing tracefile <programme>.bvt »

Exemple :

```
cd /opt/envhpc/mpich-1.2.5/examples
mpicc –o cpi cpi.c -L$VAMPIR_LIB_DIR -lVT
#Executer le programme cpi
mpirun –np 4 cpi
```

Utilisation de VAMPIR

Pour lancer VAMPIR , il faut exécuter la commande :

```
vampir
```

Une fois que VAMPIR est ouvert, il suffit d'ouvrir le fichier <programme>.bvt .

Pour l'utilisation, se référer au manuel utilisateur situé dans le répertoire $PAL_ROOT/doc

4.13 Outils d'administration système

4.13.1 Installation de Webmin et de Nagios

Webmin est installé avec le système HPC. Il est directement utilisable.

Webmin est un outil d'administration de système. Son interface est une interface Web.

Le module logiciel Nagios est présent sur le CD-ROM Bull.
Pour l'installer et le rendre utilisable, il est nécessaire que vous lanciez le script d'installation préparé par Bull.

Nagios est un logiciel qui vous permet de suivre et de surveiller des serveurs et des services (ex : http, pop3, charge cpu …) d'un système d'information, via le réseau.
Plus orienté prévention, il offre la possibilité à l'administrateur de définir des seuils critiques et de le prévenir quand ces seuils sont atteints ou quand un incident survient. L'administrateur est également informé quand la situation redevient normale sur un serveur ou un service.

Nagios est livré non compilé. Le script d'installation fourni par Bull.réalise les compilations nécessaires et prépare les fichiers de configuration.

Ce script est présent sur le CD-ROM Bull, sous le répertoire NAGIOS.

Pour le lancer :

Recopier éventuellement ce script sur un disque dur.
Se positionner dans le répertoire où il est présent.
Puis exécuter:

sh installBullnagios.sh *param1 param2* 2>&1 | tee param2/installBullnagios.log

où:
param1= arborescence où se trouve le répertoire NAGIOS (avec le logiciel associé)
(ex : /mnt/cdrom/bull_packages_sources)

param2= répertoire où seront écrits les fichiers de log
(ex : /tmp/bull)

Lorsque cette installation est terminée, l'utilisateur dispose d'un environnement prêt à être configuré en fonction de l'architecture de son système d'information.

L'utilisation, et donc la configuration de Nagios sont facultatives.

Le serveur sur lequel Nagios est installé sera appelé par commodité dans ce document : 'serveur Nagios'.

Si vous décidez d'utiliser Nagios, il vous faut dans une première phase :

- déterminer les autres serveurs (hôtes distants) que vous souhaitez surveiller avec Nagios, (munissez-vous de leur nom et leur adresse ip)
- déterminer les services que vous voulez surveiller.
  Ces services feront appel à des plugins (terminologie de Nagios).
  (cf fichiers de configuration /usr/local/nagios/etc/services.cfg et /usr/local/nagios/etc/checkcommands.cfg, pour voir des exemples de services et de plugins disponibles)

Ensuite, vous pourrez configurer Nagios sur le serveur Nagios.
Puis installer et configurer NRPE sur les autres serveurs.

Dans ce document, nous vous donnons quelques conseils sur la configuration de Nagios et de NRPE.

4.13.2 Utilisation et lancement de Webmin et de Nagios

4.13.2.1 Lancement de Webmin

Ouvrir une fenêtre de travail Linux et lancer successivement :

```
# service webmin start      (#: prompt unix)

# service httpd start

# netscape &
(ou mozilla&, …)

-------------------------
```

Sous le navigateur, se connecter sous **:**

- **https://localhost :10000**
- **ou : http://localhost:10000**

Remarque : lorsque vous effectuez des modifications, il peut être nécessaire d'exécuter les commandes suivantes :

```
service webmin start /restart  (mais aussi stop, reload, status)
```

service httpd start / restart (et aussi stop, …).

4.13.2.2 Exemple d'utilisation: configurer le réseau avec Webmin

Si vous n'avez pas encore configuré le réseau, vous pouvez le faire avec Webmin :

Sélectionner le folder 'Networking', puis :
'network configuration'
puis chacun des modules proposés
network interface, routing & gateways, DNS client, Host Addresses

Pour que vos déclarations soient prises en compte, lancez ensuite la commande suivante, dans une fenêtre de travail linux.

 # service network restart

4.13.2.3 Lancement de NAGIOS

Lors de la phase d'installation de Nagios, le script compile et installe les logiciels suivants :
Nagios, nagios-plugins, nagat, et nrpe
Lors de cette phase, les utilisateurs **'nagios'** et 'nagat' sont également créés.

Nagat, est l'interface d'administration vous permettant de **configurer nagios**. Après l'installation de Nagios, cette interface vous est déjà accessible.
Pour y accéder, entrer l'url:
..http://localhost/nagat/

De même **l'interface web de Nagios** (CGI scripts) est également déjà disponible. Pour pouvoir y accéder, entrer l'url:

..http://localhost/nagios/

Cette interface vous permet d'avoir accès à la **documentation de Nagios.**

Si Nagios est lancé, cette interface vous permet aussi de **visualiser le statut des serveurs et services surveillés avec Nagios.**

Remarque : si vous avez installé Nagios, avec le script d'installation de Bull, vous devez pouvoir lancer Nagios, dès l'installation terminée et visualiser la configuration fictive (cf menu : status map et service detail).

Pour lancer / arrêter nagios :

→ **etc/rc.d/init.d/nagios start** / restart / reload / status / stop

Remarque:
Si vous rencontrez des difficultés en appelant les urls, pensez à lancer ou relancer httpd et votre browser.

 # service httpd start / restart …
netscape & (ou mozilla&, …)

4.13.3 Configuration de Nagios (et nrpe)

4.13.3.1 Configurer Nagios sur le serveur Nagios

Nagios s'appuie sur 10 fichiers de configuration (.cfg) pour fonctionner.
Ceux-ci se trouvent tous sous /usr/local/nagios/etc.

Dans chacun de ces fichiers, des exemples sont fournis en standard avec Nagios.
Nous vous conseillons de vous inspirer de ces exemples pour créer votre propre configuration, qui dépendra notamment de :

- des serveurs à surveiller (y compris le serveur nagios)
- de la façon dont vous grouperez ces serveurs
- des services et commandes de surveillance que vous mettrez en œuvre
- de(s) administrateurs à informer, en cas d'alerte et/ou de changement d'etat.

Il vous faut donc au minimum configurer les données décrites dans les fichiers :
hosts.cfg
hostgroups.cfg
services.cfg
checkcommands.cfg.
contacts.cfg
contactgroups.cfg

Pour tester la cohérence de votre configuration, lancer la commande :
service nagios reload

Remarque :
Relancer et/ou recharger Nagios lors de toute modification des fichiers de configuration.
→ service nagios restart, ou reload

Une fois la configuration définie, visualiser les statuts des services et des serveurs, via l'interface Web : http://localhost/nagios

4.13.3.2 (Installer et) Configurer NRPE et check_nrpe pour un hôte distant

Remarque préliminaire : avec Nagios, vous pouvez surveiller un hôte distant via quelques services disponibles (plugins) de nagios : ping, http, …

Par contre, pour d'autres opérations, telle que la surveillance de la charge cpu sur un hôte distant, vous avez besoin d'installer :

- nrpe et des commandes nrpe sur cet hôte,
- les services nrpe correspondants sur le serveur nagios.

Conseil : faire déjà fonctionner Nagios sur le serveur nagios, avec au moins un service interrogeant cet hôte distant , avant d'ajouter nrpe et des services nrpe.

Vous pouvez bien sûr définir et gérer plusieurs hôtes distants de façon propre à chacun.

4.13.3.2.1 Opérations à réaliser sur un serveur (hôte) distant

Les informations données ci-dessous sont valables si votre hôte distant a le même environnement que le serveur nagios, cad s'il a le même système d'exploitation, même compilateur et libraires C et graphiques.
Dans le cas contraire, il faut installer ('downloader') et compiler nrpe sur l'hôte distant.

1) installer les fichiers nrpe sur le serveur distant

regarder si les répertoires suivants existent :
/usr/local/nagios/etc
/usr/local/nagios/libexec
(ces répertoires ne devraient exister que si Nagios a été pré-installé sur ce serveur).

Si ces répertoires n'existent pas, les créer et recopier sur ces répertoires tous les fichiers se trouvant sur ces mêmes répertoires sur votre serveur nagios.
Remarque : lors de ces opérations, vous devez notamment recopier un fichier nrpe.cfg et un exécutable nrpe.

De même, copier le fichier /etc/xinetd.d/nrpe du serveur nagios sur la même arborescence sur le(s) serveur(s) distant(s).

Créer l'utilisateur nagios :
adduser nagios

2) Editer le fichier /etc/services
enlever le caractère commentaire (cad '#' en début de ligne) de la ligne :
nrpe 5666/tcp #NRPE
pour la valider,
sinon créer la ligne

3) Editer /etc/xinetd.d/nrpe

supprimer tous les caractères commentaire des lignes comprises entre :
'service nrpe' '}' comprises.
Remarque : ces 2 dernières actions vous ont permis de définir le demon nrpe

4) Valider (activer) le démon nrpe
lancer la commande :
/etc/rc.d/init.d/xinetd.d restart

5) Visualiser le fichier /usr/local/nagios/nrpe.cfg
vérifier que plusieurs commandes sont définies, dont check-host.

Ceci termine l'installation et la mise-en-œuvre de nrpe sur le(s) serveur(s) linux distant(s) à surveiller., pour le service check-host.
Ce service est défini à titre d'exemple.

Il vous est tout-à-fait possible de définir (configurer) d'autres services. Voir pour cela les exemples donnés dans le fichier de configuration : /usr/local/nagios/etc/nrpe.cfg sur l'hôte distant.

Il est ensuite nécessaire de compléter ces installations :

- par l'installation et la validation de check_nrpe sur le serveur nagios
- par la creation d'un service qui demandera l'exécution du service adéquat sur l'hôte distant, via chek_nrpe.

4.13.3.2.2 Opérations à réaliser sur le serveur Nagios

1) Editer le fichier /usr/local/nagios/etc/checkcommands.cfg
Trouver l'exemple de commande check_nrpe
Valider cette commande, en enlevant les commentaires et en remplaçant par l'adresse d'un serveur distant ou en créer une ou plusieurs selon le même principe.

2) Editer le fichier /usr/local/nagios/etc/services.cfg
Trouver l'exemple de service avec nrpe fourni : service c-nrpe-dist1
Valider ce service, en enlevant les commentaires et en remplaçant par le nom de serveur distant
ou en créer un ou plusieurs selon le même principe.

Remarque : dans cet exemple, le service c-nrpe-dist1 fait appel à la commande check-host définie sur le(s) serveur(s) distant(s), cf paragraphe précédent et fichier nrpe.cfg.

4.13.3.2.3 Activation de nrpe et des services associés

Pour que nrpe et ces différents éléments soient pris en compte :

- recharger et relancer nagios sur le serveur nagios
  - service nagios reload
  - service nagios restart
- côté hôte distant : automatique (car par fait démon), si vous avez installé et configuré nrpe comme indiqué ci-avant, et si vous avez relancé xinetd.d :
  - /etc/rc.d/init.d/xinetd.d restart.

4.14 Outils d'administration cluster

4.14.1 Ganglia

Pré-requis

- Le nœud serveur doit être serveur web.

Installation de RRDTOOL 1.0.41

Ce composant est nécessaire pour l'utilisation de Ganglia (Il est téléchargeable sur le site : http://rrdtool.com/download.html).

- Installer RDDTOOL :

```
cd /home/packages_sources/GANGLIA
rrdtool-1.0.41-1.ia64.rpm
```

Installation de Ganglia

La documentation de Ganglia est disponible au format PDF sous le répertoire /home/packages_sources/GANGLIA.

Note : Les fichiers au format rpm sont issus du site http://ganglia.sourceforge.net/ (également disponibles sur le site http://prdownloads.sourceforge.net/ganglia/).

Pour rendre les informations des nœuds disponibles à l'affichage graphique, il faut installer *ganglia-monitor-core-gmond-2.5.3-1.ia64.rpm* sur tous les nœuds :

Sur tous les noeuds :

```
cd /home/packages_sources/GANGLIA
rpm –ivh ganglia-monitor-core-gmond-2.5.3-1.ia64.rpm
```

Pour l'affichage graphique, installer *ganglia-monitor-core-gmetad-2.5.3-3.ia64.rpm* sur le nœud serveur :

```
cd /home/packages_sources/GANGLIA
rpm –ivh ganglia-monitor-core-gmetad-2.5.3-3.ia64.rpm
```

Pour savoir si les fichiers ont bien été installés, utiliser la commande :

```
rpm –qa | grep ganglia
```

Normalement, si tout s'est bien passé, elle doit retourner :
ganglia-monitor-core-gmetad-2.5.3-1 (sur le nœud serveur uniquement)
ganglia-monitor-core-gmond-2.5.3-3

Pour savoir où ont été placés les différents fichiers de ganglia-monitor-core-gmetad-2.5.3-1 et de ganglia-monitor-core-gmond-2.5.3-3, il faut faire les commandes :

```
rpm –ql ganglia-monitor-core-gmetad-2.5.3-1
rpm –ql ganglia-monitor-core-gmond-2.5.3-3
```

Sur le serveur, modifier /etc/gmetad.conf et re-démarrer le démon gmetad

```
vi /etc/gmetad.conf
### modifier la ligne « data_source »
data_source "mon_cluster" <admin>
service gmetad restart
```

Sur tous les nœuds, modifier le fichier de configuration /etc/gmond.conf puis re-démarrer le démon.

```
vi /etc/gmond.conf
### modifier la ligne name
name "mon_cluster"
service gmond restart
```

Installation de *ganglia-web-frontend*

Installer *ganglia-webfrontend-2.5.3-1.ia64.rpm* sur le nœud serveur :

```
rpm –ivh /home/packages_sources/GANGLIA/ganglia-webfrontend-2.5.3-
        1.ia64.rpm
```

Ouvrir une page web à l'adresse web :
http:// <***adresse IP du nœud serveur***>/ganglia-webfrontend-2.5.3

Si la page ne s'ouvre pas, il est probable que le serveur Apache n'est pas lancé. Pour cela, il suffit de faire la commande :

```
service httpd start
```

Si la page s'affiche, mais que les informations concernant les nœuds sont manifestement erronées et que le switch utilisé est un switch 3COM Gigabit, vérifier la configuration au paragraphe : « Configuration du switch 3COM Gigabit ».

Utilisation de Ganglia

Listes des métriques et à quoi elles correspondent:

Métrique	Signification
boottime	Date du dernier reboot de la machine
cpu_idle	% CPU inoccupée
cpu_nice	% CPU utilisée pour traiter les priorités
cpu_num	Nombre de processeurs
cpu_speed	Fréquence des processeurs
cpu_System	% CPU utilisé par le noyau

cpu_user	% CPU utilisé par les processus
gexec	
load_fifteen	Charge du système par tranches de 15 min
load_five	Charge du système par tranches de 5 min
load_one	Charge du système par tranches de 1 min
machine_type	= ia64
mem_buffers	Quantité de mémoire attribuée au buffer
mem_cached	
mem_free	Quantité de mémoire libre
mem_shared	Segment de mémoire partagée
mem_total	Mémoire totale
os_name	= Linux
os_release	=2.4.9-18smp
proc_run	Nombre de processus en exécution (à vérifier car ce nombre = 0 si aucun processus important ne tourne)
proc_total	Nombre total de processus
swap_free	Quantité de swap libre
swap_total	Quantité totale de swap
sys_clock	Date à laquelle gmond a été lancé

4.14.2 Gexec

Pré-requis :

Pour utiliser **gexec** il faut d'abord installer l'outil d'authentification par clefs **authd**, après avoir créé le jeu de clefs privé/publique à diffuser sur tous les nœuds du cluster.

1) Génération des clés :

```
openssl genrsa -out /etc/auth_priv.pem
```

Dans le cas où vous obteniez un message indiquant d'utiliser l'option « -rand », procéder comme suit :

```
openssl genrsa –rand /etc/mime-magic –out /etc/auth_priv.pem
```

puis, pour la clef publique :

```
chmod 600 /etc/auth_priv.pem
openssl rsa –in /etc/auth_priv.pem –pubout –out /etc/auth_pub.pem
```

2) Distribution des clés :

Copier la clef publique sur tous les nœuds du cluster (ceux sur lesquels tournera le démon **authd**) :

```
scp /etc/auth_pub.pem nœud(i):/etc
```

Copier la clef privée sur tous les nœuds sur lesquels **gexec** sera utilisé (normalement, le nœud d'administration) :

```
scp /etc/auth_priv.pem nœud(i):/etc
```

3) Installation de **authd** :

Sur tous les nœuds du cluster, installation à partir de l'archive RPM

```
cd /home/packages_sources/GEXEC
rpm –ivh authd-0.2.1-1.ia64.rpm
service authd start
```

Installation:

Sur tous les nœuds du cluster, installation à partir de l'archive RPM

```
cd /home/packages_sources/GEXEC
rpm –ivh gexec-0.3.4-1.ia64.rpm
service xinetd restart
chkconfig --add authd
```

Utilisation:

Définir une variable d'environnement décrivant la liste des nœuds du cluster (IP ou DNS) :

```
export GEXEC_SVRS="noeud1 noeud2 noeud3 noeud4 noeud5"
```

Exécuter une commande sur les i premiers nœuds de la liste (0 pour tous) :

```
export PATH=/opt/envhpc/gexec-0.3.4/bin:$PATH
gexec –n i commande
```

Une description de **gexec** est disponible dans la documentation de **ganglia** version pre-2.5.0, disponible sous /home/packages_sources/GEXEC

Un document décrivant les procédures d'installation spécifique pour une adhérence entre **ganglia** et **gexec** est disponible sous /home/packages_sources/GEXEC

4.15 Outils de répartition de tâches

4.15.1 OpenPBS

Pré-requis :

- **Enregistrer ses coordonnées et obtenir l'autorisation de télécharger** Pour obtenir l'autorisation d'utiliser gratuitement le logiciel « open source » OpenPBS, il est nécessaire de s'enregistrer et d'accepter les conditions exposées sur le site : www.openpbs.org. Il faut compter quelques heures pour obtenir par mail son « user name » et son mot de passe.

Génération des binaires :

- **Récupérer les sources sur le site indiqué dans le mail**
  - Télécharger «OpenPBS Version 2.3.16 » par un clic droit de la souris. Le fichier récupéré est OpenPBS_2_3_16.tar.gz, le placer dans le répertoire /home/packages_sources/OPENPBS
  - La documentation peut également être téléchargée à ce niveau (en particulier le fichier v2.3_admin.pdf)
  - Décompresser le fichier récupéré

```
cd /home/packages_sources/OPENPBS
### Télécharger le fichier OpenPBS_2_3_16.tar.gz et la documentation ###
tar xvzf OpenPBS_2_3_16.tar.gz
```

- ***Compilation***
  - Modifier le fichier « config.sub » pour supporter ia64

```
vi OpenPBS_2_3_16/buildutils/config.sub
```

Après les lignes :

```
        pentiumpro | p6)
            basic_machine=i686-intel
            ;;
```

Rajouter :

```
        ia64-pc)
            basic_machine=i686-intel
            ;;
```

- Compiler les sources OpenPBS

```
cd OpenPBS_2_3_16
./configure
make
```

Installation :

```
make install
```

- Mettre le bit suid au programme pbs_iff

```
chmod +s /usr/local/sbin/pbs_iff
```

Configuration :

- Lancer les démons

```
pbs_mom
pbs_server –t create
### Attention, l'option –t create initialise la base de données, elle n'est à utiliser
        que la première fois.
qmgr -c "set server scheduling=true"
pbs_sched
```

- Déclarer les différents nœuds du cluster

```
vi /usr/spool/PBS/server_priv/nodes
### Mettre les noms de nœuds, par exemple :
noeud1 np=4
noeud2 np=4
noeud3 np=4
noeud4 np=4
vi /usr/spool/PBS/mom_priv/config
### insérer les lignes :
$logevent 0x1ff
```

```
$clienthost admin
```

- Déclarer les différentes files d'attente du cluster. Par exemple :

```
qmgr
c q dque
s q dque queue_type=Execution
s q dque enabled=true
s q dque started=true
```

- Configurer les services pbs_mom, pbs_server et pbs_sched.

```
cp /home/packages_sources/OPENPBS/pbs_mom /etc/init.d
chkconfig --add pbs_mom
chkconfig --level 35 pbs_mom on
cp /home/packages_sources/OPENPBS/pbs_server /etc/init.d
chkconfig --add pbs_server
chkconfig --level 35 pbs_server on
cp /home/packages_sources/OPENPBS/pbs_sched /etc/init.d
chkconfig --add pbs_sched
chkconfig --level 35 pbs_sched on
```

- Pour les éventuels autres nœuds de calcul
  - Créer les répertoires suivants sur tous les nœuds de calcul

```
mkdir /usr/spool
mkdir /usr/spool/PBS
mkdir /usr/spool/PBS/aux
mkdir /usr/spool/PBS/checkpoint
mkdir /usr/spool/PBS/mom_logs
mkdir /usr/spool/PBS/mom_priv
mkdir /usr/spool/PBS/mom_priv/jobs
```

```
mkdir /usr/spool/PBS/spool
mkdir /usr/spool/PBS/undelivered
```

- Recopier les fichiers suivants sur tous les nœuds de calcul

```
/usr/init.d/pbs_mom
/usr/init.d/pbs_server
/usr/init.d/pbs_sched
/usr/local/bin/chk_tree
/usr/local/bin/hostn
/usr/local/bin/nqs2pbs
/usr/local/bin/pbs_tclsh
/usr/local/bin/pbs_wish
/usr/local/bin/pbsdsh
/usr/local/bin/pbsnodes
/usr/local/bin/printjob
/usr/local/bin/qalter
/usr/local/bin/qdel
/usr/local/bin/qdisable
/usr/local/bin/qenable
/usr/local/bin/qhold
/usr/local/bin/qmgr
/usr/local/bin/qmove
/usr/local/bin/qmsg
/usr/local/bin/qorder
/usr/local/bin/qrerun
/usr/local/bin/qrls
/usr/local/bin/qrun
```

```
/usr/local/bin/qselect
/usr/local/bin/qsig
/usr/local/bin/qstart
/usr/local/bin/qstat
/usr/local/bin/qstop
/usr/local/bin/qsub
/usr/local/bin/qterm
/usr/local/bin/tracejob
/usr/local/include/pbs_error.h
/usr/local/include/pbs_ifl.h
/usr/local/lib/libattr.a
/usr/local/lib/libcmds.a
/usr/local/lib/liblog.a
/usr/local/lib/libnet.a
/usr/local/lib/libpbs.a
/usr/local/lib/libsite.a
/usr/local/lib/pbs_sched.a
/usr/local/sbin/pbs_demux
/usr/local/sbin/pbs_iff
/usr/local/sbin/pbs_mom
/usr/local/sbin/pbs_rcp
/usr/spool/PBS/pbs_environment
/usr/spool/PBS/server_name
/usr/spool/PBS/mom_priv/config
```

- Configurer le service pbs_mom sur chacun des nœuds de calcul et le démarrer

```
chkconfig --add pbs_mom
chkconfig --level 35 pbs_mom on
service pbs_mom start
```

- Relancer pbs_mom, pbs_server et pbs_sched sur le nœud d'administration

```
service pbs_mom restart
service pbs_server restart
service pbs_sched restart
```

Premier test :

- **Définir au minimum un login (« linux » par exemple**) présent sur tous les nœuds avec le même uid/gid. Ces logins, différents de « root » serviront à la soumission des travaux
- Créer un shell, le lancer par qsub et le voir dans la file d'attente

Sous un login différent de root (linux par exemple) créer le fichier exécutable SLEEP :

```
vi SLEEP
#!/bin/sh
hostname
sleep 30
date
```

***Puis faire les commandes** :*

```
chmod +x SLEEP
qsub –q dque SLEEP
qstat -a
```

Utilisation :

- L'utilisation d' OpenPBS implique l'acceptation des termes de la licence (voir le fichier PBS_License .text sous le répertoire home/packages_sources/OpenPBS/OpenPBS_2_3_16)

- La documentation d'OpenPBS (v2.3_admin.pdf) est disponible sur le site Web PBS (voir ci-dessus « Génération des binaires »). Elle explique plus en détail comment générer, configurer et utiliser OpenPBS.

4.15.2 OpenPBS de SCALI : ScaOPBS

Pré-requis :

- ***Définir au minimum un login (« linux » par exemple***) présent sur tous les nœuds avec le même uid/gid. Ces logins, différents de « root » serviront à l'exécution des travaux.

Installation :

- La procédure d'installation de la distribution d'OpenPBS faite par SCALI est décrite en détail au chapitre 8 du « Scali System Guide ».

Test de l'installation :

- La procédure d'installation lance automatiquement un test qui peut aussi être appelé par la commande :

```
/opt/scali/libexec/scaopbs.config -t linux
```

L'utilisation de l'implémentation OpenPBS de SCALI implique l'acceptation des termes de la licence (voir le fichier /opt/scali/contrib/pbs/doc/PBS_License.text).

La documentation d'OpenPBS (pbs_admin_guide.pdf) se trouve sous le répertoire /opt/scali/contrib/doc. Elle explique plus en détail comment générer, configurer et utiliser OpenPBS.

4.15.3 MAUI

Pré-requis

- Avoir installé OpenPBS
- Si MAUI est installé sur OpenPBS de SCALI, il faut aussi installer le rpm « ScaOPBSdk »

```
cd /opt/scali/repository/Linux2.ia64
rpm    -ivh ScaOPBSdk.Linux2.ia64-2.3.15-3.rpm
```

Installer des sources

- Le fichier « maui-3.0.7.tar.gz » contenant les sources de MAUI se trouve sous le répertoire /home/packages_sources/MAUI (ce fichier a été téléchargé à partir du site : http://supercluster.org/downloads/maui)
- Décompresser le fichier

```
cd /home/packages_sources/MAUI
tar xvzf maui-3.0.7.tar.gz
```

Compilation et installation

- Créer un répertoire /opt/envhpc/MAUI
- Configurer les makefile (utiliser le compilateur gcc)
- Générer et installer les binaires
- Exécuter maui

```
mkdir /opt/envhpc/MAUI
cd maui-3.0.7
./configure
### préciser :
### - répertoire d'installation : /opt/envhpc/MAUI
### - Home directory pour MAUI  : /opt/envhpc/MAUI
### - Compilateur               : gcc
### - Random number             : 10
### - Valider la configuration  : Y
### - Utilisation de PBS        : Y
### - Répertoire de PBS         : /opt/scali/contrib/pbs (SCALI)
###                               ou /usr/local si OpenPBS
make all
make install
# Dans le fichier /etc/init.d/maui, vérifier la valeur de la variable
            processname= "/opt/envhpc/MAUI/bin/$prog"
# Configurer le service maui:
```

```
cp /home/packages_sources/MAUI/maui /etc/init.d
chkconfig --add maui
chkconfig --level 35 maui on

# lancer maui (cette commande arrête aussi le service pbs_sched)
service maui start
/opt/envhpc/MAUI/bin/maui
```

Configuration et utilisation

- Toute la documentation de MAUI se trouve à l'adresse http://supercluster.org/documentation. Elle explique comment configurer et utiliser MAUI.
    - Sous le répertoire /home/packages_sources/MAUI/samples sont fournis des exemples de fichiers de configuration.

Chapitre 5. Désinstallation de la fourniture Intel

L'utilisation des logiciels Intel, compilateurs C/C++ et Fortran et bibliothèque mkl est soumise à licence; l'acceptation des règles de licence doit être faite directement par le client final. Afin de respecter ce processus, les logiciels Intel utilisés lors de la phase d'installation des logiciels HPC seront désinstallés avant livraison au client.

La désinstallation se déroule sous root. Lancer le script :

```
opt/envhpc/intel/compiler70/ia64/bin/uninstall
```

Apparaît une liste des logiciels Intel installés. Désinstaller l'un après l'autre le compilateur C/C++, le compilateur Fortran, le debugger et la librairie mkl..

Supprimer le répertoire contenant les licences Intel:

```
rm –rf /opt/envhpc/intel/licenses
```

Chapitre 6. FAQ

Problème de compilation et d'exécution

- Comment faire quand on a un message d'erreur du type : « **error while loading shared libraries** » lors de l'exécution d'un programme?
- Mon programme parallèle ne trouve pas le programme sur les autres machines.
- Comment optimiser les compilations avec **le compilateur fortran Intel** ?

Problème de compilation et d'exécution avec MPICH

- J'ai un problème avec des **allocations mémoire** quand j'utilise MPICH.

OpenMP

- Pour exécuter un programme parallélisé avec OpenMP, comment **définir le nombre de threads** (processeurs) utilisé ?

▪ *Comment faire quand on a un message d'erreur du type : « error while loading shared libraries » lors de l'exécution d'un programme?*

Il faut rajouter le chemin de cette bibliothèque dans la variable d'environnement LD_LIBRARY_PATH.

▪ *Mon programme parallèle ne trouve pas le programme sur les autres machines*

Il faut avoir les binaires sur toutes les machines sur lesquelles on tourne les benchs en respectant l'arborescence de la machine d'où est lancé le bench ou alors utiliser NFS.

▪ *Comment optimiser les compilations et déboguer avec le compilateur fortran Intel ?*

Pour les optimisations, ajouter les options de compilation suivantes :

- **implicitnone** : force la déclaration des variables : si une variable est utilisée sans avoir été déclarée, cela déclenche une erreur à la compilation

- **w95** : enlève les warnings pour les instructions non standard f95

- **mp** : respect de la double précision norme IEEE

- **unroll2** : fait de « l'unrolling » de boucle : cela favorise la vectorisation et le pipeline d'instructions

- **ip,ipo** : optimisation des appels à un sous-programme (gestion des paramètres).

- **auto** : alloue les variables dynamiquement dans la pile au lieu de les mettre de façon statique en mémoire

- **zero** : force l'initialisation des variables à 0

- **ftz :** flush-to-zero

- **i_dynamic :** évite de charger les bibliothèques en statique et dégraisse ainsi la taille de l'exécutable

- **parallel** : parallélise certaines séquences (elles sont fournies par l'option par_report)

- **par_report3** : logging sur le déroulement de la compilation (fournit par exemple les séquences parallélisées)

- **openmp :** prise en compte des directives OpenMP

- **restrict :** cette option est par défaut

Pour le débogage, ajouter les options de compilation suivantes :

- **g** : débogage

- **fpp** : préprocessing

▪ *Comment optimiser les compilations et déboguer avec le compilateur C / C++ Intel?*

- **O3** : optimisation de code la plus élevée

- **mp** : respect de la double précision norme IEEE

- **ip,ipo** : optimisation des appels à un sous-programme (gestion des paramètres).

- **unroll** : fait de « l'unrolling » de boucle : cela favorise la vectorisation et le pipeline d'instructions

prof_gen et prof_use : instrumentation

▪ *J'ai un problème avec des allocations mémoire quand j'utilise MPICH.*

Message d'erreur à l'exécution :

> p3_1858: (18446744073792.328125) xx_shmalloc: returning NULL; requested 65584 bytes
>
> p3_1858: (18446744073792.328125) p4_shmalloc returning NULL; request = 65584 bytes
>
> You can increase the amount of memory by setting the environment variable
>
> P4_GLOBMEMSIZE (in bytes)

La mémoire dont a besoin la communication ne peut pas être allouée correctement. Pour cela :

export P4_GLOBMEMSIZE=260000000

▪ *Pour exécuter un programme parallélisé avec OpenMP, comment définir le nombre de threads (processeurs) utilisé ?*

export OMP_NUM_THREADS=2 pour executer le programme sur 2 processeurs
export OMP_NUM_THREADS=4 pour executer le programme sur 4 processeurs

Glossaire

BANDE PASSANTE
La largeur de bande est l'intervalle de fréquences (« la bande ») transmises sans distorsions notables sur un support de transmission bien défini (atténuation...). Elle est mesurée en Hz. Pour les réseaux, et ensuite par extension pour tous les médias, cela représente la quantité de données transmise par unité de temps, c'est-à-dire le débit, aussi appelé (de façon peu orthodoxe mais fort commune) bande passante. Dans ce cas elle est mesurée en bit/s.

BEOWULF
Superordinateur composé de nombreuses stations de travail exécutant le même code en parallèle. Les Stations, aussi simples et idiotes que possible, sont appelées des « Noeuds », contrôlées via réseau par un ou plusieurs serveur(s). Un Beowulf n'utilise aucun matériel particulier, uniquement des machines et des équipements de réseau courants. Le logiciel est lui aussi courant, comme par exemple le système d'exploitation Linux, ainsi que PVM et MPI. La définition est parfois restreinte aux machines construites selon le premier modèle de Beowulf, conçu par la NASA. Les systèmes Beowulf sont répartis en deux classes, selon qu'ils sont entièrement construits à partir d'éléments trouvés à la mercerie du coin (Classe I), ou qu'ils comprennent des éléments spécifiquement conçus (Classe II).

CLUSTER (en français grappe)
Architecture de groupes d'ordinateurs, utilisée pour former de gros serveurs. Chaque machine est un noeud du cluster, l'ensemble est considéré comme une seule et unique machine. Utilisée pour le calcul scientifique, le décisionnel, le transactionnel et le datawarehouse.

DEBIT
Quantité d'information empruntant un canal de communication pendant un intervalle de temps. Mesuré en Mbit/s ou Mo/s.

FSF
La Free Software Foundation est une organisation américaine dédiée à l'abolition des restrictions sur la copie, la redistribution, la compréhension et la modification des programmes informatiques. Elle promeut le développement et l'utilisation du Logiciel Libre, et est l'initiatrice du projet GNU.

GNU
Un acronyme récursif "GNU Not Unix".

GNU/LINUX
Il s'agit de l'ensemble formé par le noyau Linux et du système GNU.

GPL
General Public License. Le statut juridique des logiciels distribués « librement », à l'origine utilisé pour le projet GNU de la FSF.

HPC
High Performance Computing. Informatique de haute performance.

Itanium™ Architecture
Architecture 64 bits des nouvelles puces d'Intel, destinées à remplacer les x86. Il s'agit d'une rupture totale avec la série x86, le jeu d'instructions n'ayant plus rien à voir, ni les éléments de l'architecture du processeur. Le résultat est quelque chose de globalement plus simple, donc de bien plus rapide, en donnant un contrôle plus fin au logiciel (en particulier les compilateurs) sur le matériel. La première version s'appelle Itanium (précédemment appelée Merced). Il aura fallu six ans de développement pour qu'il apparaisse sur le marché. Maintenant la deuxième génération, Itanium-2, a fait son apparition.

ITANIUM™ Processor
Processeur 64 bits d'Intel et destiné à remplacer les x86 (dont les Pentiums), tout en restant compatible. Itanium est le premier modèle de l'architecture Itanium et l'Itanium-2™ est le 2ième.

LATENCE
Temps minimal de propagation d'un signal. Par extension, temps minimal de transmission d'un ensemble de données à travers d'un réseau.

LINUX
Linux est un système d'exploitation de type UNIX, multi-tâches et multi-utilisateurs, disponible sur de nombreuses architectures matérielles, en particulier les machines à base de processeurs x86 et Itanium. Il intègre la plupart des technologies les plus récentes (SMP, clustering, RAID…).
Linux est un noyau. Pour l'utiliser, il faut des applications, c'est ce que proposent les distributions. Une distribution est un ensemble de programmes plus un noyau à installer sur une machine. Parmi les distributions Linux, on peut citer RedHat, Mandrake, Suse, TurboLinux …

MONITORING
Contrôle pas à pas, c'est-à-dire qu'on ne laisse aucune liberté au système contrôlé. Le monitoring permet aussi de contrôler et/ou surveiller un processus en temps réel.

MPI
Message Passing Interface. Bibliothèque portable utilisée pour des applications parallèles.
NFS
Network File System. C'est un système de gestion de fichiers de réseau, présenté par Sun en 1985 pour ses stations sans disque. Les versions principalement utilisées actuellement sont les versions 2 (utilisant UDP) et, depuis 1993, 3 (pouvant utiliser UDP ou TCP).

NOEUD
Ordinateur connecté à un réseau. Dans le monde du clustering, chaque machine du cluster est appelée nœud. Leurs fonctions peuvent ensuite être différentes : nœud de management, nœud de calcul, nœud de storage …

OPEN SOURCE
Définition particulière du logiciel libre, mise au point en 1998 par Eric Raymond, cherchant à adapter le principe à l'entreprise. Elle comporte neuf points (pour le moment): la libre redistribution, la mise à disposition du code source, la possibilité de distribuer ses travaux

dérivés, le respect du code source originel, l'absence de discrimination envers des personnes, l'absence de limitation sur le domaine d'application du logiciel, la distribution de la licence et sa non-spécificité à un produit, et enfin le fait qu'elle ne contamine pas le travail des autres.

PARALLELISATION
Transformer un programme de façon qu'il soit possible de l'exécuter efficacement sur plusieurs processeurs.

PROCESSUS
Programme en cours d'exécution, avec son environnement. Terme essentiellement utilisé dans le monde Unix à l'origine.

PVFS
Parallel Virtual File System. Projet se définissant comme l'exploration de la « conception, de l'implémentation et des utilisations potentielles des entrées/sorties parallèles ». Développé par l'université de Clemson et la NASA, PVFS est destiné aux clusters de Stations de Travail et aux machines Beowulf

PVM
Parallel Virtual Machine. API gérant la communication entre les nœuds d'un cluster de machines.

RAID
Installation de disques durs en batterie pour augmenter le stockage et la fiabilité.

RISC
Architecture de processeur.

SCALABILITE
La capacité d'un système à supporter une augmentation de ses contraintes dans un domaine particulier, ou dans tous les domaines. On pourrait proposer le nom " échelonnabilité" comme traduction. Par exemple un système est dit "scalable" en terme de nombre d'utilisateurs si ses performances de fonctionnement sont aussi bonnes avec une dizaine d'utilisateurs connectés qu'avec une centaine.

SERVEUR DE FICHIERS
Serveur qui met uniquement des fichiers à disposition du réseau, et pas ses autres ressources (comme sa puissance de calcul, ses liaisons...). En général, son point fort est son disque dur.

SMP
Symmetric MultiProcessing. Système multiprocesseur distribuant symétriquement les tâches entre différents processeurs partageant une mémoire commune, en s'assurant qu'ils ne vont pas se mettre à écrire tous à la même adresse en même temps.

SWAP
Le fait d'utiliser une partie d'un disque dur comme de la mémoire vive.

SYsTEME DE GESTION DE FICHIERS (SGF)

Il définit par exemple la structure interne de l'arborescence, les formats d'enregistrements, le découpage des disques, les métadonnées sur les fichiers...
Un SGF est constitué d'un service de gestion (pour l'organisation des fichiers entre eux) et d'un système de fichiers (pour les opérations sures et dans les fichiers).

TRACE
La Trace d'un programme est la succession des états de son environnement au cours de son exécution.

Références

Généralités sur les clusters

www.beowulf.org
www.tldp.org/HOWTO/Parallel-Processing-HOWTO.html
www.top500.org
www.phy.duke.edu/brahma/beowulf_online_book/
www.Linux-Consulting.com/Cluster/

Les architectures parallèles

www.lri.fr/~fci/support95.html
www.idris.fr

Les interconnects

www.scali.com
www.dolphin.com
www.essi.fr/~riveill/rapport01-these-cecchet.pdf
www.ens-lyon.fr/~rewestrel/thses.ps

HPC

www.epcc.ed.ac.uk/HPCinfo
L'ouvrage High Performance Computing de Dowd & Severance aux éditions O'Reilly.

Librairies scientifiques

www.lifl.fr/west/courses/cshp/bibsp.pdf
www.irisa.fr/orap/Publications/Forum6/petitet.ps
www.netlib.org

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull NovaScale HPC Linux Guide d'installation

Nº Reférence / Reference Nº : 86 F2 31EG 02

Daté / Dated : Juillet 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
B.P.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 / Télécopie +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web sites at: / Ou visitez nos sites web à:
http://www.logistics.bull.net/cedoc
http://www–frec.bull.com **http://www.bull.com**

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº 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.

PLACE BAR CODE IN LOWER LEFT CORNER

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 F2 31EG 02

Bull

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

NovaScale
Linux
HPC Linux
Guide d’installation

86 F2 31EG 02

NovaScale
Linux
HPC Linux
Guide d’installation

86 F2 31EG 02

NovaScale
Linux
HPC Linux
Guide d’installation

86 F2 31EG 02