Cluster de calcul Freeware en Océanographie Opérationnelle

Bertrand FERRET, Responsable du Service Informatique (*, **)
Carine CASTILLON, Ingénieure Systèmes et Réseaux (*)
Mondher CHEKKI, Ingénieur High Performance Computing (*, **)

(*) Mercator Océan
Parc Technologique du Canal
8-10 rue Hermès
31520 Ramonville Saint-Agne

(**) Observatoire Midi-Pyrénées
CNRS – UMS 831
14 avenue Edouard Belin
31400 Toulouse

Résumé

Dans cet article, nous présentons les choix effectués en vue de construire un cluster Free Home Made pour réaliser des prévisions de l'état physique tridimensionnel de l'océan, mission de la société civile Mercator Océan.

Pour des raisons de coût (budget maximum de 60 K€ HT), d'espace occupé dans la salle informatique, de consommation électrique, de bande passante CPU / RAM et de parallélisme à grains fins, nous avons opté pour des blades intégrant 4 cartes-mère. Ce calculateur est doté de 16 nœuds (128 cœurs de calcul) connectés en infiniband et de 768 Go de RAM.

Le cluster a été installé à l'aide de logiciels issus du monde libre : « Linux » pour le système d'exploitation, « SystemImager » pour le déploiement et la sauvegarde du système, « C3 tools » pour la gestion centralisée, « Torque / Maui » pour le gestionnaire de ressources et « Ganglia, Nagios / Centreon » pour la surveillance des nœuds.

Nous décrivons donc les solutions choisies tant au niveau hardware que software et nous essayons de répondre aux différentes questions que se pose un administrateur système lors de la construction d'un cluster de calcul.

Mots-clés

Cluster de calcul, High Performance Computing, SystemImager, C3 tools, Ganglia, Nagios / Centreon, Torque / Maui, OpenMPI, codes parallèles.

1 Puissance de calcul à petit prix

1.1 Le contexte

Depuis une dizaine d'années, Mercator Océan (Société Civile composée de 5 organismes, Météo France, SHOM, Ifremer, CNRS, IRD) utilise d'importants moyens de calcul pour effectuer des prévisions hautes résolutions en trois dimensions de l'état physique de l'océan jusqu'à 14 jours. Des paramètres comme la température, le courant, la salinité ... sont prévus sur tous les océans du globe grâce au modèle NEMO développé par le CNRS, avec assimilation des données altimétriques et des données de bouées (3000 bouées circulent dans les océans du monde entier et effectuent des mesures physiques). Pour information, les modèles que nous utilisons sont similaires en complexité à ceux utilisés par Météo France pour la prévision du temps.

Leader européen dans ce domaine, nous utilisons des moyens de calcul, externes (Météo

France, CEP, IDRIS ...) mais aussi internes. Pour ces derniers, nous possédons deux calculateurs, le premier, *Opale*, de marque Fujitsu / Siemens est un calculateur datant de 2005 composé de 27 nœuds bi-opteron, 252 Go de RAM, un réseau gigabit et 6 To d'espace disque pour une puissance théorique de 216 GFlops. Le deuxième, *Wallis & Futuna,* acquis en juin 2007 suite à un appel d'offre européen pour une somme de 800 K€ HT est de marque SGI, un Altix 4700 dont les caractéristiques sont 192 cœurs de calcul, 960 Go de RAM, un réseau numalink et enfin 40 To d'espace disque en cxfs pour une puissance théorique de 1,23 TFlops. Nous installons et maintenons l'ensemble des moyens informatiques de l'organisme dont les calculateurs précédemment cités.

Cet article a pour but de décrire l'ensemble des choix que nous avons effectués pour remplacer notre vieux cluster *Opale*.

1.2 Les contraintes

Les contraintes sont de trois types, tout d'abord *financière :* notre budget est de 60 K€ HT (avril 2011) ; ensuite *matérielle* : l'occupation dans notre salle informatique ne peut dépasser 42 U et la consommation électrique est limitée à 12 kWh (contraintes imposées par le vieux cluster *Opale*) ; et enfin *logicielle* : nos codes utilisent toute la bande passante CPU / RAM et possèdent un parallélisme à grains fins (échange de gros volumes de données en utilisant les bibliothèques MPI). De plus, ce cluster devra être accessible simultanément par de nombreux utilisateurs.

Spécialistes Linux et utilisateurs de produits du monde libre, nous voulons montrer que l'on peut construire un cluster répondant de façon optimale à nos contraintes, possédant une puissance théorique minimum de 1,5 TFlops avec un budget de 60 K€ HT.

2 Installation et administration du cluster

2.1 Matériel

L'architecture du cluster mis en place est la suivante (cf. Figure 1) :

- un nœud maître DELL R510 permet de compiler et de lancer le code (aucun calcul ne s'exécute sur ce nœud). Par souci d'économie, il sert aussi de nœud I/O distribuant 10 To par le protocole NFS ;
- 16 nœuds de calcul : 4 blades DELL C6100 de 4 cartes-mère bi-sockets intégrant chacune 2 CPU Intel Xeon de 4 cœurs et 48 Go RAM ;
- un réseau gigabit 10.202.206.* pour l'administration des nœuds et pour écrire sur les espaces disques centralisés (solution de secours) ;
- un réseau infiniband QDR 10.203.206.* pour les échanges de message MPI entre CPU de cartes-mère différentes et pour écrire sur les espaces disques centralisés.

Notre cluster de calcul possède donc une puissance théorique de 1,56 TFlops, 128 cœurs de calcul connectés en infiniband et 768 Go de RAM. Il consomme 6 KWh max et occupe 16 U.

Nous avons sélectionné des processeurs avec seulement 4 cœurs et un QPI élevé pour avoir des performances de bande passante CPU / RAM maximum, contrainte requise pour faire tourner nos codes de prévisions océanographiques. En effet, pour un prix sensiblement équivalent, nous aurions pu faire l'acquisition de processeurs avec, par exemple, 12 cœurs (passage de 90 W par CPU à 120 W), mais nous aurions perdu en performance car le cache du processeur aurait été partagé entre tous ses cœurs et nous n'aurions pu utiliser que 1/3 des cœurs de calcul. La puissance théorique, multipliée par 3, aurait été alors inutilisable.

Dans le calcul scientifique, il faut donc étudier finement les codes devant s'exécuter pour adapter au mieux les caractéristiques du matériel.

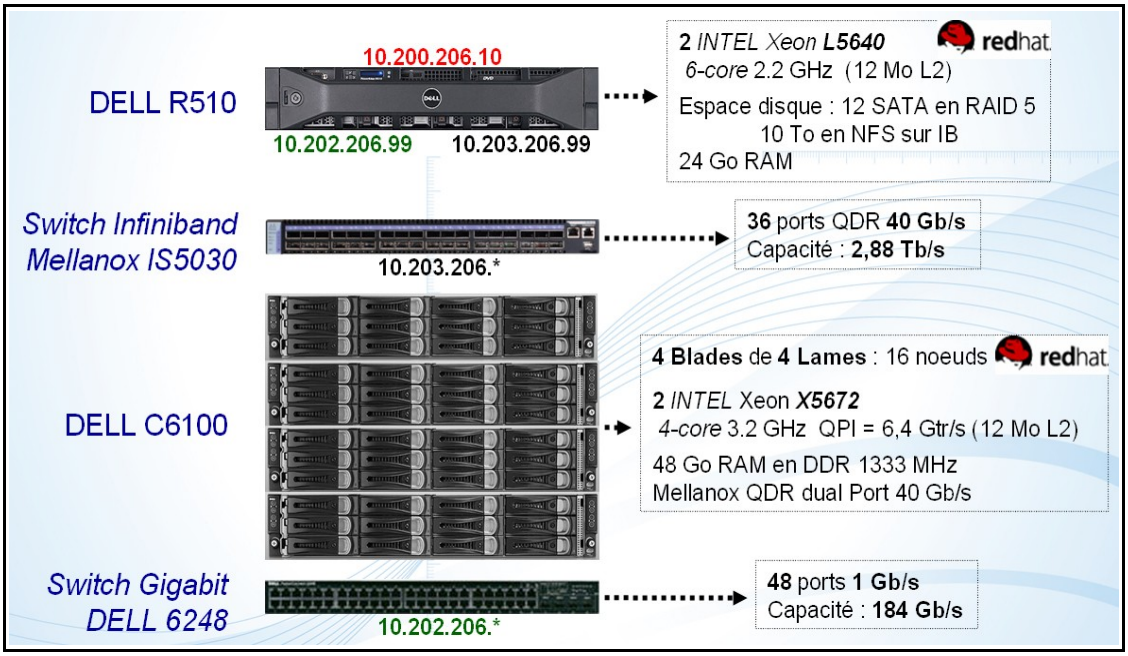

Figure 1 – Description du matériel

2.2 Installation

Nous décrivons dans ce paragraphe la partie administration système. Tout ce qui concerne la partie applicative est abordée dans le chapitre 4.

L'installation du cluster (nœud maître et nœuds de calcul) est basée sur le système d'exploitation *RedHat 5.6 (Tikanga)* certifié par le revendeur du matériel (drivers optimisés).

Une fois l'OS installé sur le serveur maître (R510), nous avons procédé au formatage des partitions qui seront utilisées par l'ensemble des nœuds de calcul. Au vu des différents tests de lecture/écriture réalisés (cf. Chapitre 5), nous avons opté pour le format xfs avec une exportation via le réseau infiniband.

Puis nous avons configuré l'OS du premier nœud de calcul (C6100) en nous concentrant sur l'accès aux partitions centralisées. Afin d'éviter des problèmes de montage au démarrage des nœuds, et pour minimiser la consommation de ressources, nous avons choisi le montage dynamique *automount* : les partitions seront montées à la demande et démontées après une période d'inactivité de 900 s. Ce type de montage a nécessité la modification du fichier `/etc/auto.master` ainsi que la création du fichier */etc/auto.cluster* sur ce même nœud.

Pour simplifier et homogénéiser l'administration du cluster, tous les nœuds de calcul doivent être installés de manière identique. Dans ce but, nous avons utilisé SIS[1] et plus particulièrement l'outil SystemImager[2] qui permet d'automatiser l'installation (cf. Figure 2). Son principe est simple : absorber l'image système d'un nœud déjà installé (*Golden Client*) et la déployer sur un ensemble de nœuds de configuration matérielle identique (nœuds de calcul).

---

[1]SIS : System Installation Suite : ensemble d'outils dédiés à l'installation et l'administration d'un groupe de machines au travers d'un réseau. (http://sourceforge.net/projects/sisuite)

[2] SystemImager : http://wiki.systemimager.org

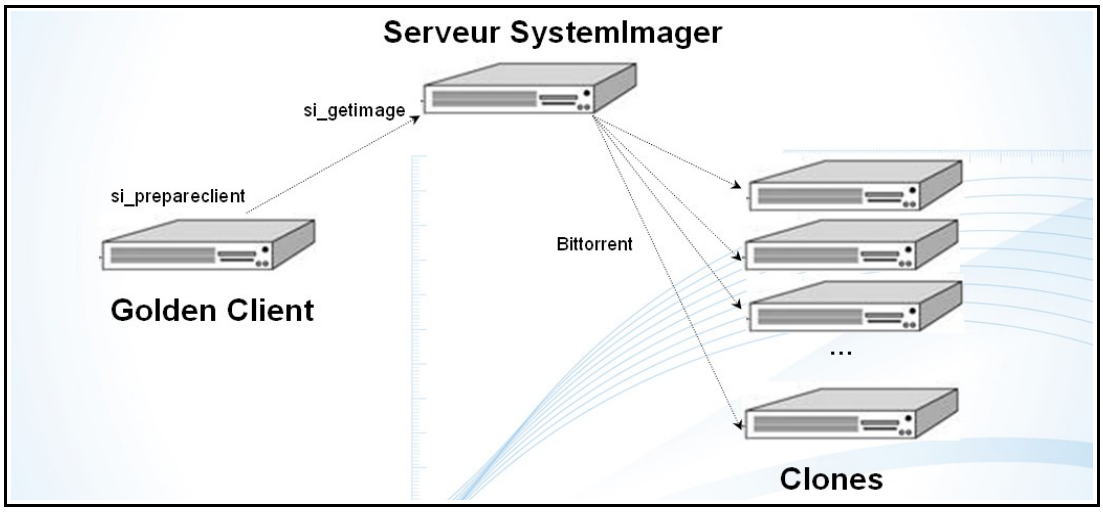

Figure 2 - Fonctionnement de SystemImager

La partie serveur de SystemImager est installée sur le serveur maître du cluster. Cet outil nécessite la présence d'un serveur DHCP qui, dans notre cas, est également présent sur le serveur maître. L'installation peut se résumer à l'ajout d'une série de packages RPM et à la modification de fichiers de configuration situés dans */etc/dhcp*, */etc/systemimager* et */tftpboot.*

Ensuite, nous absorbons l'image système du premier nœud de calcul. Voici la procédure :

1. démarrage du service *systemimager-server-rsyncd* sur le serveur maître ;
2. sur le nœud de calcul, préparation du *Golden Client* pour l'absorption de l'image :

```
# si_prepareclient –server <ip_serveur_maître>
```

3. sur le nœud maître, absorption de l'image :

```
# si_getimage --golden-client <ip_nœud_calcul>  --image RH56_calcul
```

L'image système du nœud de calcul est maintenant présente sur le serveur maître (par défaut dans */var/lib/systemimager*). En plus du déploiement, cette image nous servira de sauvegarde en cas de crash matériel ou logiciel. Il est donc nécessaire de la mettre à jour régulièrement.

Nous pouvons, dès à présent, déployer l'image simultanément sur plusieurs nœuds via l'outil *bittorrent* de SystemImager. Il suffit de configurer le serveur DHCP afin de sélectionner les nœuds à installer (*/etc/dhcp/dhcpd.conf*) et de lancer les services *systemimager-server-bittorrent* et *dhcpd*.

A partir du serveur maître, on démarre les nœuds cibles en mode PXE :

```
# /usr/bin/ipmitool -U <LOGIN> -P <PASS> -I lanplus -H <ip_nœud> chassis bootdev pxe
# /usr/bin/ipmitool -U <LOGIN> -P <PASS> -I lanplus -H <ip_nœud> chassis power reset
```

L'installation des nœuds de calcul terminée, nous reprogrammons le prochain boot sur disque :

```
# /usr/bin/ipmitool -U <LOGIN> -P <PASS> -I lanplus -H <ip_nœud> chassis bootdev disk
# /usr/bin/ipmitool -U <LOGIN> -P <PASS> -I lanplus -H <ip_nœud> chassis power reset
```

Pour garder une configuration logicielle identique des nœuds de calcul, nous utiliserons C3 tools[3] qui permettra , par exemple, de modifier les fichiers de configuration, de redémarrer les services, d'ajouter un package RPM ... simultanément sur l'ensemble des nœuds.

[3]C3 tools : Cluster Command Control (http://www.csm.ornl.gov/torc/C3/)

2.3 Monitoring

Pour rendre un service opérationnel, c'est à dire prévenir les pannes ou être prévenu au moindre dysfonctionnement matériel ou logiciel du cluster, il faut mettre en place un monitoring sur le système. Les différents services présents sur le nœud maitre et les nœuds de calcul sont observés par deux logiciels :

- Ganglia[4] : pour la surveillance de la charge CPU, de la mémoire et du trafic réseau gigabit. La partie serveur installée sur le nœud maître communique avec le client installé sur chaque nœud de calcul. Une interface Web visualise graphiquement la charge de chaque nœud ainsi que la charge totale du cluster ;
- Nagios / Centreon[5] : pour le monitoring des services vitaux et des occupations disques. Dans notre cas, peu de services sont installés sur les nœuds, nous surveillons donc l'accès (*ping*), ainsi que le service de gestion des ressources Torque (cf. Chapitre 3). Un email est envoyé à tous les membres de l'équipe *systèmes et réseaux* dès qu'un incident survient sur un des nœuds du cluster (maître ou calcul).

3 Configuration du gestionnaire de ressources

3.1 Théorie

Penser l'organisation des travaux comme un jeu. Si tout le monde peut exécuter des travaux à n'importe quel moment, occuper autant de nœuds, de processeurs et de mémoire qu'il le désire, c'est l'anarchie : le système est mal exploité.

Mais en utilisant un gestionnaire de ressources, il peut améliorer le rendement et décide : qui peut exécuter les travaux présents dans les queues et quand les travaux doivent être lancés. Les décisions sont basées sur une politique définie au préalable et sur des privilèges.

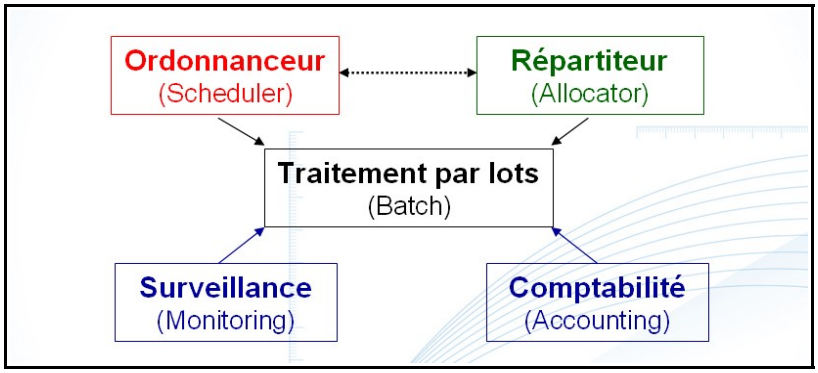

Figure 3 – Fonctionnement d'un gestionnaire de ressources

La figure 3 décrit une vue générale des outils d'exploitation d'un cluster en exposant leurs relations. Pour comprendre la gestion d'un cluster, il est nécessaire de définir quelques termes :

- *Traitements par lots* : un fichier rassemble un ensemble de commandes qui doivent être traitées séquentiellement (contraire à l'interactivité) ;
- *Ordonnanceur* : il planifie dans le temps une succession de travaux en attente (contraintes) et gère l'ordre dans lequel ils doivent s'exécuter ;

[4]Ganglia : http://ganglia.sourceforge.net/

[5] Nagios (http://ww.nagios.org) / Centreon (http://www.centreon.com)

- *Répartiteur* : il planifie dans l'espace, récupère les informations de l'*Ordonnanceur* et teste la disponibilité du matériel au moment de l'exécution des processus ;
- *Surveillance* : Contrôle temps réel ou post mortem du fonctionnement matériel et des travaux ;
- *Comptabilité* : Statistiques d'utilisation des ressources communes (facturation).

Un gestionnaire de ressources automatise le partage des moyens de calcul entre les différents utilisateurs.

3.2 Politique des queues

A Mercator Océan, la « cohabitation » de jobs sur le cluster provenant de plusieurs individus effectuant des réservations concurrentes est rendue possible grâce aux règles suivantes :

- un job ne pourra pas occuper plus de la moitié des nœuds (8 nœuds maximum) ;
- 3 queues d'exécution sont disponibles : *MONO* (1 seul nœud sur maximum 20 heures), *MULTI8* (entre 2 et 8 nœuds sur 20 heures maximum) et *ARCHIVES* (1 seul nœud sur 1 heure maximum) ;
- priorité des jobs de moins de 4 heures en journée, moins de 8 heures la nuit, et moins de 20 heures le weekend.

3.3 Implémentation sous Torque / Maui

Torque et Maui[6] sont les deux outils de gestion de ressources installés sur notre cluster de calcul. Torque gère la partie distribution des jobs sur les différents nœuds, alors que Maui s'occupe de l'ordre de lancement des jobs, suivant les priorités (nombre de nœuds, durée du job, mémoire nécessaire, ...).

Pour mettre en place notre politique de queues, nous avons modifié la configuration de Torque à l'aide de la commande *qmgr*. L'exemple ci-dessous montre la configuration de la queue d'exécution *MULTI8*.

```
create queue multi8
set queue multi8 queue_type = Execution
set queue multi8 resources_max.mem = 384000mb
set queue multi8 resources_max.nodect = 8
set queue multi8 resources_max.pmem = 6000mb
set queue multi8 resources_max.walltime = 20:00:00
set queue multi8 resources_min.nodect = 2
set queue multi8 resources_default.nodect = 1
set queue multi8 resources_default.nodes = 1
set queue multi8 resources_default.walltime = 00:30:00
set queue multi8 acl_group_enable = True
set queue multi8 acl_groups = mercator
set queue multi8 kill_delay = 2
set queue multi8 max_user_run = 1
set queue multi8 enabled = True
set queue multi8 started = True
```

Les jobs exécutés sur cette queue peuvent utiliser entre 2 et 8 nœuds (resources_min.nodect=2, resources_max.nodect=8) et pas plus de 384 Go de mémoire

[6] Torque / Maui : gestionnaire de ressources (http://www.clusterresources.com)

(resources_max.mem=384000mb). Ils pourront tourner jusqu'à 20 heures maximum (resources_max.walltime=20:00:00).

Maui gère l'ordre de passage des jobs. Un extrait du fichier */var/spool/maui/maui.cfg* montre que seuls les jobs de moins de 4 heures peuvent tourner en journée (entre 8h00 et 20h00) les lundi, mardi, mercredi, jeudi et vendredi.

```
SRCFG[jour] STARTTIME=08:00:00
SRCFG[jour] ENDTIME=20:00:00
SRCFG[jour] PERIOD=DAY
SRCFG[jour] DAYS=MON,TUE,WED,THU,FRI
SRCFG[jour] HOSTLIST=ALL
SRCFG[jour] TIMELIMIT=4:01:00
```

4 Mise en place de l'environnement utilisateur

4.1 Besoins

L'environnement logiciel de Mercator Océan est complexe, il est géré grâce aux modules qui offrent une grande flexibilité dans le choix des versions des bibliothèques informatiques.

Les ingénieurs et chercheurs disposent de machines aux architectures différentes et de codes mono / multiprocesseurs, il est donc nécessaire de fournir à l'ensemble des utilisateurs, un environnement de travail commun, performant et évolutif. Il doit être en mesure de charger un environnement précis pour une machine donnée de manière souple, transparente et doit permettre de reproduire les conditions dans lesquelles une simulation s'est déroulée.

4.2 Logiciels

Pour répondre aux besoins des utilisateurs, les logiciels suivants ont été installés :

- *Module / Environnement switcher*[7] : redéfinition automatique des variables d'environnement de l'utilisateur pour s'adapter aux différents logiciels ;
- Compilateurs (*INTEL, PGI, GNU*) : compilation et optimisation des codes séquentiels et parallèles ;
- Librairie parallèle (*OpenMPI*) : l'utilisation de la bibliothèque MPI (Message Passing Interface) pour les échanges de messages est indispensable pour les codes multiprocesseurs;
- Les bibliothèques scientifiques (*NetCDF, palm, Lapack, ...*) et de visualisation (*GMT, Ferret, IDL* ...) permettent, d'une part, de faire tourner une simulation et, d'autre part, de vérifier la cohérence et la validité des résultats numériques obtenus.

5 Benchmarks

Pour valider les performances de notre cluster (puissance de calcul, réseau et I/O), nous avons développé un bench (90 % calcul et 10 % I/O) basé sur le code NEMO[8] (configuration ORCA au 12eme de degré) permettant de tester la bande passante CPU / RAM ainsi que les I/O sur les espaces disques centralisés. Nous l'avons testé avec deux filesystem sur infiniband : *ext4* et *xfs*. Le filesystem *ext4* est plus rapide mais moins stable (lorsqu'il y a écriture, les accès disques d'un autre utilisateur sont fortement pénalisés), c'est pourquoi nous avons mis en

[7] Module : http://modules.sourceforge.net/

[8] NEMO : *Nucleus for European Modelling of the Ocean* (http://www.nemo-ocean.eu/)

opérationnel uniquement le *filesystem xfs*. Des tests de performances sur les espaces disques centralisés ont été aussi réalisés à l'aide de Iozone[9].

	#PROCS	TIME (s)	MEM (GB)
NEMO-ext4 (MO) Dell C6100	*128* 96	*13080* 18660	559 581
NEMO-xfs (MO) Dell C6100	128 96	14040 19260	559 581
SGI (MO) Altix 4700	128 96	18776 --	556 --
C1B (CEP) IBM Cluster 1600	*128* 96	18950 28780	*424* 443

Tableau 1 – Code NEMO : Perfs CPU et I/O

Ce même bench a été lancé sur de multiples plateformes, notamment notre SGI Altix 4700 mais aussi sur un IBM Cluster 1600 du CEP. Les résultats montrent que le bench est, par exemple, 25 % plus rapide sur DELL C6100 avec *xfs* que sur notre plateforme SGI Altix 4700 en *cxfs* ou IBM Cluster 1600 du CEP (cf. Tableau 1).

6 Bilan

Les performances sont donc au rendez-vous, le rapport puissance / prix est très intéressant si l'on compare des plateformes avec le même nombre de cœurs de calcul. De plus, une augmentation de puissance peut se réaliser aisément et ne demandera pas de modifications logicielles simplement un ajout de blades (seule limitation, le réseau avec un nombre de ports gigabit et infiniband fixé par les équipements).

Nous terminerons cet article par quelques conseils, fruit de notre expérience, pour élaborer un cluster *Free Home Made* :

- ne pas sous estimer la logistique (électricité, bâtiment, climatisation) ;
- préparer des benchmarks *Maisons* (mesure et vérification des performances) ;
- choisir une architecture matérielle cohérente (processeurs, mémoires, réseaux) ;
- installer des logiciels de surveillance matérielles et logicielles & de comptabilité des ressources en calcul (Torque, Maui, Ganglia, Nagios / Centreon, Qbank & Gold) ;
- ne pas négliger le temps à passer pour la gestion de l'exploitation (pannes, optimisations, gestionnaire de batch, politique d'occupation, gestion de la charge) ;
- choisir des logiciels cohérents correctement interfacés (Torque, Maui, OpenMPI, PGI / Intel, Netcdf) ;
- simplifier l'utilisation du cluster pour les développeurs (module, documentations, cours).

Un dernier conseil, attention aux *bottlenecks* ...

[9] Iozone : Filesystem benchmark tool (http://www.iozone.org/)