Cluster de calcul, machine Beowulf, ferme de PC
Principes, problématique et échanges d'expérience

29 mars 2002

Olivier BOEBION - Laboratoire de Mathématiques et de Physique Théorique - Tours

Principes d’un cluster de calcul

- Un ensemble de machines reliées en réseau pouvant effectuer des calculs parallèles ;
- cet ensemble se compose au moins d’une machine maître ;
- la machine maître dispose en général d’au moins une machine cliente (ou esclave) ;
- le matériel doit être le plus standard possible.

Classes de cluster de calcul

1. Beowulf Linux de classe I
   - matériel non spécifique à un constructeur basés sur des standards ;
   - driver intégrés couramment dans le noyau.
2. Beowulf Linux de classe II
   - utilisation de matériels et de standards réputés plus performants.

La terminologie de cluster

Elle est aussi utilisée par des constructeurs de stations de travail (*Hewlett Packard*, *Digital*...). Cette notion de cluster n'est pas la même que celle développée dans les clusters de calcul.

Au contraire d'un cluster de stations de travail "propriétaire", seule la station maître d'un cluster de calcul possède un clavier, une souris et un écran.

Un des objectifs est de pouvoir dupliquer les machines clientes facilement et rapidement ! ! !

Architectures Multi-Processeurs

Plusieurs types :

1. Machines partageant de la mémoire et communiquant à travers la mémoire (machines Symmetric Multi-Processing, programmation Multi-threads) ;
2. Machines avec des ressources mémoires locales et utilisant des messages.

Les messages

Ce sont les échanges entre les différentes CPU. Les messages nécessitent une recopie des données au contraire des threads qui utilisent des données localement disponibles.

Un message se compose :

1. des données à manipuler ;
2. d’une destination de CPU(s).

Comparatifs cluster de calcul/machine SMP

	Perf. Archi. SMP	**Perf. Archi. Cluster**	**Possibilités d'extension**
Messages	bonnes	les meilleures	les meilleures
Threads	les meilleures	mauvaises	mauvaises

Ecriture/portage d’un programme parallele (1/2)

Deux interfaces de programmation pour les messages :

– Parallel Virtual Machine (PVM)
– Message Passing Interface (LAM-MPI et MPICH)

Langages disponibles : C, C++, Fortran 77 et 90.

Ecriture/portage d’un programme parallèle (2/2)

– évaluer les parties d’un programme qui peuvent être traitées indépendamment ;
– se demander si ces parties peuvent être lancées en parallèle.

Les programmes automatiques de traduction de code ne fonctionnent pas bien. Il est nécessaire d’investir du temps pour :

1. juger si un cluster est adapté à votre besoin de calcul ;
2. apprendre à utiliser une nouvelle API.

Choix matériel et structurel

- Type de réseau : Ethernet 100Mb/s, Ethernet 1 Gb/s, Mirynet 2Gb/s ;
- Estimation de la mémoire (éviter la SWAP) ;
- Poste esclave avec ou sans disque ;
- Montage NFS pour les librairies, les répertoires utilisateurs ;
- Lancement des processus par rsh, ssh ;
- Sécurité pour l'accès au cluster (Filtrage de paquets, Ssh).

Les logiciels de développement

- compilateurs Gnu C, C++, Fortran 77 et VF-90 (Fortran 90)
- PGI Fortran et C/C++ ;
- Compilateurs Intel...

Un logiciel de gestion ?

Un cluster de calcul peut être réalisé avec un investissement logiciel minimal en utilisant des logiciels “OpenSource” **mais** la gestion du cluster est plus difficile pour :

– le déploiement et le clonage des postes esclaves ;
– le monitoring ;
– la soumission des jobs.

Etat du projet du laboratoire

Une plateforme d’essai est actuellement en place pour évaluer notre besoin et apprendre à utiliser l’API MPI.

– utilisation de PC de récupération ;
– déploiement du cluster avec les logiciels non-propriétaires.

Si l’intérêt pour cette technique de calcul se confirme, un véritable projet sera réalisé.

Références

- Beowulf HOWTO
- Beowulf Installation and administration HOWTO
- http ://www.beowulf-underground.org
- User's Guide for MPICH "A portable Implementation of MPI"
- http ://www.idris.fr/data/cours/parallel/mpi/choix_doc.html