Polytech'Lille - IMA3i - ARS **2004-2005**

TP4

Applications parallèles MPI

1 But du TP

Les applications logicielles sont aujourd'hui de plus en plus gourmandes en ressources de calcul, de mémorisation, d'entrées/sorties, etc. Pour améliorer les performances d'exécution de ces applications, il convient de rendre leur conception et exécution parallèles. Le but de ce TP est vous initier au développement de programmes parallèles et à l'analyse de leurs performances à travers MPI, une bibliothèque standard *de facto*, et le support d'exécution MPI/LAM.

2 Configuration de l'environnement MPI/LAM

Pour pouvoir compiler et exécuter des applications MPI, il est indispensable de passer par les étapes suivantes :

- Dans votre fichier ".bashrc", ajoutez la ligne suivante :
$$export\ PATH\ =\ \$PATH : /usr/local/lam-6.5.4/bin$$
- Dans votre répertoire ".ssh", générez une clé publique dans le fichier "identity.pub".
$$ssh-keygen\ -t\ rsa\ -f\ identity$$
Copiez cette clé dans le fichier "authorized_keys".
- Créez un répertoire que vous appelerez par exemple MPI pour acceuillir les applications que vous allez développer dans ce TP.
- Créez dans ce répertoire un fichier que vous appelerez par exemple "LAM_Hosts". Celui-ci doit contenir la liste des machines qui exécuteront vos applications MPI.
- Vérifiez que LAM est bootable sur toutes les machines de "LAM_Hosts".
$$recon\ -v\ LAM\_Hosts$$
- Démarrez une session LAM sur votre machine parallèle (LAM_Hosts).
$$lamboot\ -v\ LAM\_Hosts$$
- *Note importante* : Il est recommandé de terminer toute session MPI/LAM à la fin de chaque TP en exécutant "lamhalt".

3 Compilation et exécution

Il est conseillé de créer un nouveau réprtoire pour toute nouvelle application développée.

- Créez un répertoire "Simple" et copiez-y le fichier suivant :
  $$/home/imaEns/nmelab/TP/MPI/UnVersUn/simple.c$$
- Compilez ce programme.
  $$hcc\ simple.c\ -o\ simple$$
- Lancez son exécution sur votre machine parallèle de la manière suivante :.
  $$mpirun\ -c\ 2\ simple$$
- Réexécutez le programme en augmentant le nombre de processus "simple" jusqu'à ce qu'il soit supérieur au nombre de machines. Que constatez-vous ?
- Avant de passer à la partie suivante, éditez et analysez le programme "simple.c".

4 Communications un-vers-tous et tous-vers-un

- Modifiez le programme "simple.c" de manière à ce qu'un processus envoie un message à tous les autres processus du groupe.
- Modifiez ce programme afin qu'un processus reçoive un message de tous les autres processus du groupe.

5 Collecte de nombres et calcul de somme

Le but ici est de faire la somme de nombres détenus par un groupe de processus. Ecrivez un programme dans lequel chaque processus envoie un nombre (son identifiant i.e. rang par exemple) à un processus maître, ce dernier calcule et affiche la somme de tous les nombres qu'il a reçus.

6 Empaquettage de données

En utilisant le type de données dérivé MPI_PACKED et les primitives MPI_Pack et MPI_Unpack, réalisez un programme dans lequel chaque processus envoie un message accompagé de son rang à un processus maître.