

TD 5 Programmation PVM

5.1 Installation

1. Créer votre environnement :
 - Copier `germain/Pub/cshrc.pvm.stub` dans votre home directory. Modifier votre `.cshrc` en ajoutant à la fin source `$HOME/cshrc.pvm.stub`. Pour cette fois seulement, exécuter source `$HOME/cshrc.pvm.stub`. Vérifier que la variable `$PVM_ROOT` est bien positionnée à `/usr/share/pvm3`
 - Créer les répertoires
 - `$HOME/pvm3/bin/$PVM_ARCH`. Tous les exécutables utilisant pvm **doivent** y être placés.
 - `$HOME/pvm3/src/$PVM_ARCH`. Pour vos futurs programmes source.
2. Accéder aux exemples et manuels :
 - Les code d'exemples se trouvent dans `$PVM_ROOT/examples`.
 - Manuel en ligne : <http://www.csm.ornl.gov/pvm/man/manpages.html>.
 - Beaucoup d'autres informations : http://www.csm.ornl.gov/pvm/pvm_home.html
3. Lancer pvm par la commande `pvm`. Dans la console, `help`. Ajouter une machine, puis commande console `conf`. Comparer l'effet des commandes console `quit` et `halt`. Pour plus d'information sur la configuration, on peut lire `$PVM_ROOT/Readme`.

5.2 Syntaxe

1. Modifier le programme `hello_other.c` pour qu'il s'exécute obligatoirement sur un autre processeur que la console :
2. Comment le processus `hello_other` obtient-il le tid du processus auquel il doit envoyer un message ?
3. Modifier les programmes `hello.c` et `hello_other.c` pour qu'ils communiquent de façon continue jusqu'à ce que l'esclave décide d'arrêter l'échange.

5.3 Programmation SPMD : Anneau à jeton

Le programme `spmd.c` simule le passage d'un jeton dans un anneau de P tâches : la tâche 0 émet un jeton et attend qu'il lui revienne. Modifier le programme pour les tâches effectuent un calcul "utile" (incrémenter d'un compteur flottant) en attendant le jeton. Le processeur 0 récupèrera la valeur cumulée des compteurs.

5.4 Tableaux distribués

On considère un tableau $N \times N$, partitionné par lignes en bandes de taille $M = N/P$.

1. Que fait le programme `germain/Pub/decal.c`? On étudiera en particulier les opérations de compactage/décompactage.
2. Ecrire un programme maître (`master.c`) qui crée les processus et effectue l'affichage, et des programmes esclaves qui effectuent seulement les calculs.
3. Modifier le code des esclaves pour qu'ils effectuent un stencil 5 points (moyenne avec les 4 voisins).

5.5 Création dynamique de processus

La méthode du *crible d'Érathostène parallèle* calcule les nombres premiers de la façon suivante. Un ensemble dynamique de Q processus est étiqueté par la suite des Q nombres premiers; initialement $Q = 2$. Chaque processus (d'étiquette k) reçoit de son prédécesseur un élément de la suite des entiers naturels. Si ce nombre est multiple de k , il est éliminé; sinon il est soit transmis au processus successeur si celui-ci existe, soit envoyé au générateur et transmis comme étiquette à un successeur nouvellement créé.

Le prédécesseur du premier processus est un processus particulier, le générateur de la suite des entiers naturels.

1. Ecrire le code C + PVM de la recherche des nombres premiers inférieurs à N par cette méthode.
2. Comment faut-il modifier ce code pour qu'il calcule les N premiers nombres premiers?