

TD Calcul parallèle

MACS 2^{ième} année

Octobre 2009

1 Identification et interrogation de l'environnement

Écrire un programme parallèle où chaque processus devra écrire sur la sortie standard le rang qui lui a été attribué ainsi que le nombre de processus qui a été lancé.

Par exemple pour 3 processus :

Je suis le processus 1/3

Je suis le processus 0/3

Je suis le processus 2/3

2 Communication point à point dans un anneau

Un problème courant dans les applications distribuées est celui du contrôle : *qui peut ou doit faire quelque chose ?* Une solution consiste à utiliser un “jeton” qui peut circuler entre les nœuds de la machine. Le nœud qui possède le jeton a alors certains droits et peut travailler sans risque de conflit avec les autres processeurs.

2.1 Communications en anneau

Ici, nous essayerons de faire tourner le jeton (symbolisé par un entier) sur “un anneau” de processeurs. Le processus à décrire est le suivant : le nœud 0 lit une valeur, l'envoie au nœud 1 qui le transmet à 2, etc. jusqu'à ce qu'il revienne à 0 qui l'imprime et termine le processus.

Écrire le programme et tester le temps mis à l'aide de la fonction `wtime`

2.2 Communications simultanées

Cette fois ci, on veut que chaque processus envoie simultanément un entier (par exemple, `rang+1000`) au processus de rang directement supérieur au sien (avec toujours le processus de rang `n-1` qui envoie un entier au processus de rang 0) ...

2.2.1 Programme basique

Écrire le programme à l'aide des fonctions `send` et `recv`. Mesurer le temps pris par le programme à l'aide de la fonction `wtime`.

2.2.2 Programme amélioré

Écrire le programme à l'aide de la fonction `sendrecv`. Mesurer le temps pris par le programme à l'aide de la fonction `wtime`. Comparer avec le programme précédent.

3 diffusion d'un entier dans un cube de dimension n

3.1 Diffusion d'un entier dans un cube de dimension 1

Écrire un programme qui marchera que sur deux processus qui permet au processus 0 d'envoyer un entier choisi au hasard (on utilisera pour cela la fonction `rand` de la bibliothèque standard du C) au processus 1, de sorte qu'à la sortie du programme, l'entier est contenu dans le processus 0 et 1.

Exemple de sortie pour ce programme :

```
Processus 0 : J'envoie la valeur 100 au processus 1
Processus 1 : Je reçois un message du processus 0
Processus 1 : J'ai reçu la valeur 100 du processus 0
```

3.2 Diffusion d'un entier dans un cube de dimension 2 et 3

Écrire un programme semblable pour quatre processus pour que le processus 0 puisse diffuser son entier aux processus 1, 2 et 3.

Essayer de trouver un algorithme permettant de faire un maximum de communications en simultané. On imaginera pour cela que le réseau connectant les quatre processus est un carré.

De même, en imaginant que le réseau formé par huit processus forme un cube, écrire le programme permettant au processus 0 de diffuser son entier aux 7 autres processus.

Peut-on généraliser l'algorithme de diffusion d'un entier du processus 0 aux autres processus pour 2^n processus ($n \geq 1$) ? Si oui, comment ?

Essayer de généraliser la diffusion à 2^n processus.

Remarque : Pour pouvoir généraliser à 2^n processus, il faut concevoir un cube de dimension $n > 3$ (voir figure ci-dessous pour un cube de dimension 4).

