ISIMA — 2ème année F2 — le 9 Décembre 2005
Documents autorisés : Notes et Polycopiés du cours — durée : 2 heures

Examen de Programmation Parallèle

Exercice 1 : calcul de π par intégration (8 points)

On peut approcher la valeur de

$$\pi = \int_0^1 \frac{4}{1+x^2} dx$$

par la méthode de rectangle

$$\sum_{x=\frac{w}{2},1-\frac{w}{2},w} \frac{4}{1+x^2}$$

où w est la largeur des rectangles.

Le programme suivant calcule de cette manière une valeur approchée de π :

```
#include <stdio.h>
#define f(x) (4.0/(1.0+(x)*(x)))
const int n=1 << 24;
int main()
{
   int i;
   double w, x, sum=0.0, pi;
   w = 1.0 / n;
   for (i=0; i<n; i++) {
     x = w * ((double)i + 0.5);
     sum += f(x);
   }
   pi = w * sum;
   printf( "Valeur de pi calcule = %24.16\n", pi);
}
```

Q1. Proposez une parallélisation de ce programme en MPI avec N processus.

Q2. Analysez la performance théorique de votre programme en supposant que chaque opération (+,-,*,/, envoi et réception d'une donnée) s'exécute en une unité de temps ?

Q3. Comment mesurer la performance pratique d'un programme parallèle en MPI ?

Exercice 2 : **diffusion sur une grille de processus** (11 points)

Soit $N \times N$ processus communicants organisés en une gille. Chaque processus est numéroté par les indices (i, j), où, i est le numéro de ligne et j le numéro de colonne, avec $0 \leq i, j \leq N-1$. Chacun ne peut communiquer que avec ses voisins directs.

Supposons que l'on dispose de primitives suivants :

- `xrank()` et `yrank()` retournent l'indice de ligne et colonne du processus;
- `size()` retourne la valeur de N;
- `send(dest, buf, length)` envoie (de manière non bloquante) le buffer `buf` de taille `length` au processus `dest` qui est un des voisins du processus courant;
- `recv(src, buf, length)` reçoit du processus `src` des données de taille `length` et les range dans le buffer `buf`. La valeur de `src` peut être l'un des voisins du processus courant ou `any` pour une réception depuis un quelconque des processus voisins. Cette réception est bloquante.

Il s'agit de réaliser la diffusion d'une valeur détenue localement par l'un des processus à l'ensemble des autres processus.

Une solution élémentaire est de considérer au sein de la grille un chemin de communication d'un processus désigné vers chacun des processus; par exemple le processus (0,0). Cet ensemble de chemins forme un arbre de recouvrement dont le processus (0,0) est la racine. Chacun des processus étant à une distance d'au plus 2N-2 du processus (0,0), l'arbre est de hauteur 2N-2.

Cet arbre est utilisé pour diffuser la valeur depuis le processus (0,0). Nous avons un algorithme de diffusion en 2N-2 étapes de communications.

Le meilleur cas dans cet algorithme est de considérer le processus central de la grille. Supposons que $N = 2P + 1$, le processus central est (P, P).

Q1. Quelle est la hauteur H de l'arbre de recouvrement du processus (P, P) ?

Q2. A l'aide des fonctions ci-dessus, donner le code C de la diffusion à partir du porcessus (P, P) avec H étapes de communication.

On considère maintenant le cas de grille torique. Dans une grille torique, tout processus peut être considéré comme le centre. La diffusion peut être effectuée en 2 phases : la première phase consiste à diffuser la valeur sur la ligne du processus source, la deuxième phase sur les colonnes.

Q3. Donner le code C de cet algorithme.

Q4. Comparer les deux algorithmes.