

## Examen du module Programmation Parallèle

Documents autorisés : polycopiés de cours et de TD

*Durée 2 heures*

### Parallélisation de programme 1

Soit le programme suivant qui calcule l'intégrale de la fonction `fonc` dans l'intervalle  $[0, 1]$ .

```
#include <stdio.h>
#define NMAX 10000000
double fonc(double);

main()
{
    int i;
    double r=0, pas=1./NMAX, x1, x2;

    for(i=0;i<NMAX;i++) {
        x1=i*pas;
        x2=(i+1)*pas;
        r=r+(fonc(x1)+fonc(x2))/2;
    }
    r=r*pas;
    printf("r=%.15e\n",r);
}
```

**Question 1** Écrire une version parallèle de ce programme en utilisant la bibliothèque MPI

**Question 2** Écrire une version parallèle de ce programme en utilisant la bibliothèque OpenMP

### Parallélisation de programme 2

La fonction suivante effectue le produit d'une matrice triangulaire supérieure par un vecteur. La matrice `a` est carrée et remplie avec des 0 dans la matrices triangulaires inférieures.

Exemple de matrice triangulaire supérieure :

$$a = \begin{pmatrix} 2 & 1 & -2 & 0 & -5 \\ 0 & -1 & 4 & 6 & 7 \\ 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```

void prodmatvec(double a[], double vec[], double r[], int dim)
{
    int i,j;

    for (i=0;i<dim;i++){
        r[i]=0;
        for (j=i;j<dim;j++)
            r[i]+=a[i*dim+j]*vec[j];
    }
}

```

On cherche à écrire une implémentation de cette fonction permettant un bon équilibrage de la charge des calculs entre  $P$  processeurs. La matrice est de dimension  $N$  et  $P$  est un diviseur de  $N$ .

**Question 1** Est-il possible d'écrire facilement une version parallèle de ce fonction en utilisant OpenMP. Si oui écrire le programme en précisant le type de scheduling, sinon justifier votre réponse.

**Question 2** Est-il possible d'écrire facilement une version parallèle de cette fonction en utilisant MPI et en partageant les données sur  $P$  processeurs. Si oui réécrire le programme en, sinon justifier votre réponse.

**Question 3** Afin d'économiser de l'espace mémoire, un programmeur décide de changer la représentation de la matrice. Il propose de mettre dans un tableau 1D les valeurs non nulles de la première ligne, puis celles de la deuxième, ... Avec la matrice exemple ci-dessus, on obtient la représentation suivante:

$$a = \{2, 1, -2, 0, -5, -1, 4, 6, 7, 2, -1, 0, 4, 2, 1\}$$

La fonction devient alors

```

#define SOM(n) (n*(n+1)/2)

void prodmatvec1(double a[], double vec[], double r[], int dim)
{
    int i,j;
    for (i=0;i<dim;i++){
        r[i]=0;
        for (j=i;j<dim;j++){
            r[i]+=a[i*(dim)+j-SOM(i)]*vec[j];
        }
    }
}

```

Rappel :

$$1 + 2 + 3 + \dots + n = \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Est-il possible d'écrire une version parallèle de ce programme en utilisant OpenMP. Si oui écrire le programme en précisant le type de scheduling, sinon justifier votre réponse.

**Question 2** Avec la représentation de  $a$  sous forme de tableau 1D, est-il possible d'écrire une version parallèle de ce programme en utilisant MPI. Si oui écrire le programme, sinon justifier votre réponse.

## Parallélisation d'un algorithme séquentiel 1

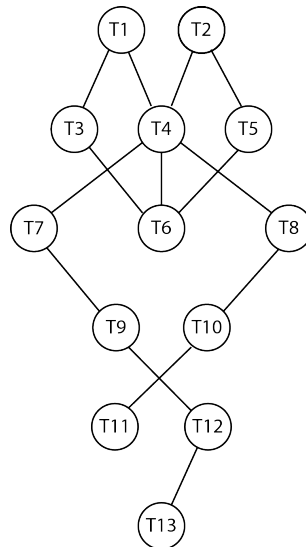
Considérons l'algorithme séquentiel suivant :

T1	$X1=A1+B1$
T2	$X2=A2+B2$
T3	$Z1=1+B1$
T4	$Z2=B2$
T5	$T=\max(\text{abs}(Z1),\text{abs}(Z2))$
T6	$S=\min(\text{abs}(Z1),\text{abs}(Z2))$
T7	$R=T*T*(1+S*S/T/T)$
T8	$T=X1*Z1+X2*Z2$
T9	$S=X2*Z1-X1*Z2$
T10	$C1=T/R$
T11	$C2=S/R$

- 1) Etablir le graphe de précedence en détaillant toutes les étapes.

## Parallélisation d'un algorithme séquentiel 2

Soit le graphe de précedence suivant :



- 1) Donner les décompositions du graphe de précedence sur une machine parallèle ayant suffisamment de processeurs
  - a) dans le cas d'une décomposition par prédécesseurs.
  - b) dans le cas d'une décomposition par successeurs.
  - c) Calculer l'accélération et l'efficacité de ces deux décomposition. Laquelle propose les meilleures performances ?
- 2) Quel est le temps d'exécution de l'algorithme parallèle sur une infinité de processeurs sachant que toutes les tâches ont le même temps d'exécution
- 3) Donnez une borne maximum du temps d'exécution du même algorithme sur une machine à 2 processeurs.
- 4) Pouvez-vous trouver une allocation sur 2 processeurs demandant un temps inférieur à cette borne ? Si oui laquelle ?