Université Gaston Berger de Saint-Louis UFR DES SCIENCES APPLIQUEES ET DE TECHNOLOGIE Section Informatique	Master Développement et de Systèmes d'Information	Année Universitaire 2017-2018 B. DIOP

Fiche TD[1] N°1 - Traitement de problèmes parallèles

Exercice 1

Traitement de tableau à deux dimensions

Cet exemple illustre des calculs sur des éléments d'un tableau à deux dimensions. Une fonction est évaluée sur chaque élément du tableau.
Le calcul sur chaque élément du tableau est indépendant des autres éléments du tableau.
Le problème est intensif en calcul.
Le programme série calcule un élément à la fois dans un ordre séquentiel.
Le code de série pourrait être de la forme :

```
do j = 1, n
     do i = 1, m
          a (i, j) = fcn (i, j)
     fin do
fin do
```

Questions :

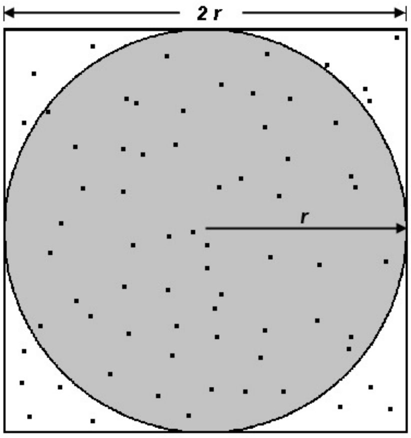
1. Est-ce que le problème peut être transformé en un problème parallèle ?
2. Comment pourrait-on diviser le problème ?
3. Est-ce que la communication est nécessaire ?
4. Existe-t-il une dépendance des données entre les différents sous problèmes ?
5. Y a-t-il un besoin de synchronisation ?
6. Y a-t-il un besoin d'équilibrage des charges ?

1 Introduction to Parallel Computing
Blaise Barney, Lawrence Livermore National Laboratory

Exercice 2

Estimation de PI

La valeur de PI peut être calculée de différentes manières. Considérons la méthode de Monte Carlo d'approximation de PI :



Estimation de Pi à travers un carré

Figure 1. Calcul de Pi à travers un carré

- Inscrire un cercle de rayon r dans un carré de longueur latérale égale à 2r
- L'aire du cercle est πr^2 et l'aire du carré est $4r^2$
- Le rapport entre la surface du cercle et la surface du carré est exprimé par :

$$\pi r^2 / 4r^2 = \frac{\pi}{4}$$

- Si l'on génère aléatoirement N points à l'intérieur du carré, environ $N * \frac{\pi}{4}$ de ces points auront la probabilité de tomber à l'intérieur du cercle.
- π est alors approximée comme:

$$N * \frac{\pi}{4} = M$$
$$\frac{\pi}{4} = \frac{M}{N}$$
$$\pi = 4 * \frac{M}{N}$$

- Notez que l'augmentation du nombre de points générés améliore l'approximation.

Pseudo-code séquentiel :

```
nbPoints = 10000
cercle_count = 0

Faire pour j = allant de 1 à nbPoints
   générer 2 nombres aléatoires représentant les coordonnées
   xcoordinate = random1
   ycoordinate = random2
   Si (xcoordinate, ycoordinate) se trouve dans le cercle alors

      cercle_count = cercle_count + 1
fin faire

PI = 4,0 * cercle_count / nbPoints
```

Questions

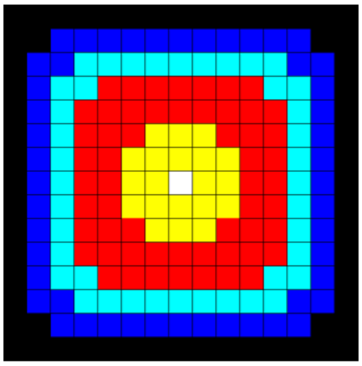
Le problème est d'une grande intensité de calcul - la majeure partie du temps est consacrée à l'exécution de la boucle

1. Ce problème peut-il être parallélisé ?
2. Comment le problème pourrait-il être partitionné ?
3. Est-ce que la communication est nécessaire ?
4. Y a-t-il des dépendances de données ?
5. Y a-t-il besoin de synchronisation ?
6. L'équilibrage de charge sera-t-il une préoccupation ?

Exercice 3

Équation de chaleur simple

La plupart des problèmes de calcul parallèle nécessitent une communication entre les tâches. Un certain nombre de problèmes communs nécessitent une communication avec les tâches "voisines". L'équation de la chaleur en 2D décrit le changement de température dans le temps, compte tenu de la distribution initiale de la température et des conditions aux limites.



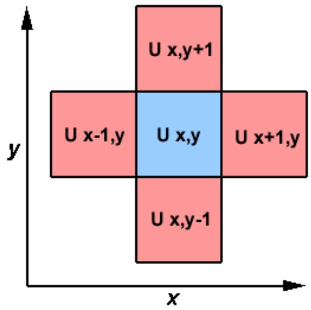


Figure 2. Modèle à automate cellulaire de propagation de la chaleur

Un schéma de différenciation finie est utilisé pour résoudre l'équation de la chaleur numériquement sur une région carrée.

- Les éléments d'un tableau à deux dimensions représentent la température aux points du carré.
- La température initiale est nulle sur les limites (frontière) et haute au milieu.
- La température limite est maintenue à zéro.
- Un algorithme itératif est utilisé.

Le calcul d'un élément dépend des valeurs de l'élément voisin à travers cette équation ci-dessous :

$$U_{x,y} = U_{x,y} + C_x * \left(U_{x+1,y} + U_{x-1,y} - 2 * U_{x,y}\right) + C_y * \left(U_{x,y+1} + U_{x,y-1} - 2 * U_{x,y}\right)$$

Un programme séquentiel contiendrait le code suivant :

```
Faire iy = 2, ny - 1
   Faire ix = 2, nx - 1
     u2 (ix, iy) = u1 (ix, iy) +
         cx * (u1 (ix + 1, iy) + u1 (ix-1, iy) - 2. * u1 (ix, iy)) +
         cy * (u1 (ix, iy + 1) + u1 (ix, iy-1) - 2. * u1 (ix, iy))
   Fin faire
Fin faire
```

Questions

1. Ce problème peut-il être parallélisé ?
2. Comment le problème serait-il partitionné ?
3. Les communications sont-elles nécessaires ?
4. Y a-t-il des dépendances de données ?
5. Y a-t-il des besoins de synchronisation ?
6. L'équilibrage de charge sera-t-il une préoccupation ?

Exercice 4

Équation d'onde 1-D

Dans cet exemple, l'amplitude le long d'une onde uniforme et vibrante est calculée après écoulement d'une durée spécifiée.
Le calcul implique :

- L'amplitude sur l'axe **y**
- ***i*** comme l'indice de position le long de l'axe **x**
- Points de nœud étalés le long de la chaîne
- Mise à jour de l'amplitude à des pas de temps discrets.

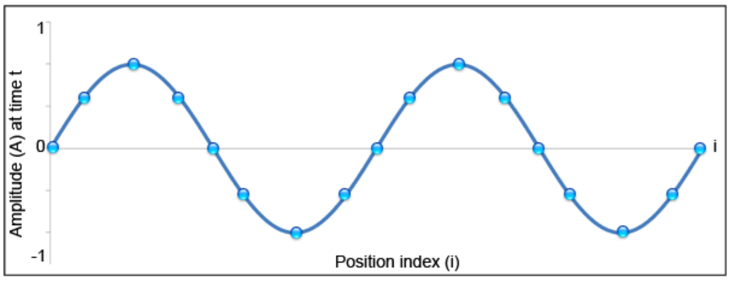


Figure 3. Oscillations d'ondes en 1-D

L'équation à résoudre est l'équation d'onde unidimensionnelle :

```
A(i,t+1) = (2.0 * A(i,t)) - A(i,t-1) + (c * (A(i-1,t) - (2.0 * A(i,t)) + A(i+1,t)))
```

Où C est une constante

Notez que l'amplitude dépend des temps précédents (t, t-1) et des points voisins (i-1, i + 1).

Questions :

1. Ce problème peut-il être parallélisé ?
2. Comment le problème serait-il partitionné ?
3. Les communications sont-elles nécessaires ?
4. Y a-t-il des dépendances de données ?
5. Y a-t-il des besoins de synchronisation ?
6. L'équilibrage de charge sera-t-il une préoccupation ?