
Calcul Parallèle Intensif
SERIE 3

Exercice 1 :

1. Compléter, exécuter et expliquer le fonctionnement du programme suivant :

```
int main()
{ ...
#pragma omp parallel private(i)
{   int tid=omp_get_thread_num();
    int Nt=omp_get_num_threads();
    for(i=p*tid;i<n;i=i+p*Nt)
    for(int j=i;j<i+p;j++)
    if(j<n)printf("i=%d,tid=%d\n",j,omp_get_thread_num());
}}
```

Les valeurs de n et p sont données au cours de l'exécution du programme.

2. Ecrire une autre procédure plus simple équivalente à celle de la question 1.

Exercice 2 :

Soit la section parallèle incomplète suivante :

```
#pragma omp parallel ...
{   int tid=omp_get_thread_num();
    int Nt=omp_get_num_threads();
    int is= (Nt-tid-1)*n/Nt;
    int ie=(Nt-tid)*n/Nt;
    for(i=n-ie;i<n-is;i=i+1)
printf("valeur de i=%d, son rang est:%d\n",i,tid);
}
```

1. Insérer cette section dans un programme et expliquer son action en précisant le sens des variables « is » et « ie » et les itérations affectées à chaque thread.
2. Ecrire une autre procédure plus simple équivalente à celle de la question 1.

Exercice 3 :

1. Ecrire un programme calculant $S = \sum_{i=1}^n (i^2 - 100 \times i)$. Insérer dans le programme une section parallèle utilisant la clause « reduction » pour exécuter les itérations de la boucle « for » simultanément. Mesurer les temps mis par la section parallèle pour $n=10^7, 10^8, 10^9$ et le nombre de threads égale à 2, 3 et 4.
2. Ajouter à ce programme une autre section parallèle pour calculer S en utilisant « omp critical ». Exécuter les mêmes instances de la question 1. Comparer les temps d'exécutions des deux sections parallèles.