

## **TD : Introduction aux architectures parallèles**

### **Performances**

#### ***Exercice 1***

On considère un problème d'une certaine taille. Un programme parallèle passe 6% de son temps d'exécution dans les opérations d'entrée/sortie exécutées sur un seul processeur.

**Q 1) Quel est le nombre minimal de processeurs pour que le programme parallèle produise une accélération de 10 par rapport au programme séquentiel ?**

#### ***Exercice 2***

Un programme a une accélération de 9 sur 10 processeurs.

**Q 2) Quelle est la fraction maximale du temps d'exécution séquentiel qui peut correspondre à des opérations purement séquentielles ?**

### **Equilibrage de charge**

On exécute le programme C suivant sur un monoprocesseur, puis sur quadriprocesseur. On suppose que le temps d'exécution du programme est proportionnel au nombre d'exécutions de l'itération de la boucle interne.

```
For (i=1 ; i <= 32 ; i++)  
{  
  sum (i) = 0 ;  
  for (j=1; j<=i ; j++)  
    sum(i) += y[i][j];  
}
```

**Q 3) En répartissant les boucles I sur les 4 processeurs (8 itérations de 0 à 7 sur le premier processeur, puis 8 à 15 sur le second, etc ...), quelle est l'accélération obtenue par rapport à l'exécution sur le monoprocesseur ?**

**Q 4) Modifier la parallélisation pour obtenir une exécution équilibrée (même temps d'exécution sur chaque processeur). Quelle est alors l'accélération obtenue ?**

### **OpenMP**

**Q 5) Sur un multiprocesseur à 4 processeurs, paralléliser le programme suivant à l'aide de directives OpenMP en minimisant les défauts de caches.**

```
Int N ;  
Float A[N], B[N], C[N] ;  
for (int =0 ; i<N ; i++)  
  C[i]= B[i]+A[i] ;
```

**Q 6) Sur un multiprocesseur à 4 processeurs, paralléliser le programme suivant avec des directives OpenMP en utilisant deux versions différentes pour la réduction.**

```
N=65636  
Double x[N] ;  
Double sum ;  
  
For (i=0 ; i<N; i++)  
  Sum += x[i];
```

**Q7) Ecrire la version parallèle OpenMP pour 4 processeurs du produit matrice – vecteur dont la version séquentielle est**

```
For (i=0 ;i<n ;i++) {  
    C[i]=0 ;  
    For (j=0 ;j<n-1 ;j++)  
        C[i]+=A[i][j]*B[j] ;  
}
```

**Q8) Ecrire la version parallèle OpenMP pour 4 processeurs du programme suivant**

```
For (i=1 ;i<n ;i++)  
    A[i] =(A[i]+A[i-1])/2 ;
```

**Q 9) Ecrire la version parallèle OpenMP pour 4 processeurs du programme suivant**

```
For (i=0 ;i<n ;i++)  
    For (j=1 ;j<n ;j++)  
        A[i][j]+=A[i][j-1] ;
```