ARCHITECTURE ET PROGRAMMATION PARALLELE

TD N°1

Exercice 1 :

Soit le système d'équations à 3 inconnues suivant :

$$\begin{cases} A_{0,0} * x_0 & = b_0 \\ A_{1,0} * x_0 + A_{1,1} * x_1 & = b_1 \\ A_{2,0} * x_0 + A_{2,1} * x_1 + A_{2,2} * x_2 & = b_2 \end{cases}$$

Où $a_{i,j}$ (i=0..2, j=0..2) et b_k (k=0..2) sont des constantes et les x_n sont les inconnus à déterminer. Les constantes $a_{i,j}$ et b_k sont stockées respectivement dans les tableaux A[ ][ ] et B[ ] et les valeurs des inconnus sont stockées dans un tableau x[ ].
On se propose de résoudre ce genre de système en utilisant un calcul parallèle en pipeline.

1. Déterminer l'équation de base nécessaire pour la résolution du système.
2. Déterminer les étapes (étages) du pipeline.
3. En déduire le type de ce pipeline.

On suppose que le temps nécessaire pour une opération d'addition est de **z** ms et que celui d'une opération de multiplication est de **t** ms. Le temps de chargement des données et le temps de communication entre les étages sont supposés de faibles valeurs.

4. Déterminer le start-up time de ce pipeline (i.e. temps d'amorce : temps durant lequel le pipeline est fonctionnel mais non productif).

Exercice 2 :

Nous considérons un hypercube de dimension **3**.

1. Déterminer la connectivité de ce réseau.
2. Calculer le coût de communication induit par un commérage dans l'hypercube de dimension d sachant que le message initial sur chaque nœud est de taille **L** octets, β est le temps d'initialisation de ce réseau et τ est le temps de transmission d'un octet sur ce réseau.
3. Calculer le coût de communication dans le cas de commérage séquentiel absolu
4. Evaluer les performances en ce qui concerne la communication

Exercice 3 :

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];
}
```

1. 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 ?
2. Modifier la parallélisation pour obtenir une exécution équilibrée (même temps d'exécution sur chaque processeur). Quelle est alors l'accélération obtenue ?