

TD Architecture Matérielle - Parallélisme

Exercice 1

On veut calculer la somme de n nombres. Il faut T_c unités de temps à une personne pour additionner deux nombres.

- a) Calculer le temps nécessaire à une personne pour calculer la somme de n nombres.
- b) Les n nombres sont répartis en 8 groupes. On demande à 8 personnes de calculer la somme des n nombres, sachant qu'il faut T_w unités de temps à une personne pour transmettre son résultat à une autre personne, calculer le temps nécessaire aux 8 personnes pour faire la somme n nombres dans les cas suivants :
- 8 personnes assises en cercle.
 - 8 personnes formant deux rangées de 4 personnes chacune.
 - 8 personnes dans une configuration d'arbre binaire.
 - 8 personnes dans une configuration d'arbre ternaire.

Exercice 2

Soit l'expression $E = (a + (b - c)/d) * (e - (((f - g)/h) * i))$.

- a) Dessiner l'arbre d'évaluation de cette expression et son graphe de dépendances. Paralléliser au maximum le calcul de l'expression.
- b) Calculer l'accélération et l'efficacité de la solution parallèle en supposant que chaque opération arithmétique prend une unité de temps.
- c) En utilisant la propriété d'associativité, peut-on modifier l'expression E pour en accélérer le calcul ? Quelles sont alors l'accélération et l'efficacité ?

Exercice 3

Soit T_1 le temps d'exécution d'un programme A sur une machine séquentielle à 1 processeur et le temps T_p le temps d'une solution parallèle du programme A sur une machine parallèle comprenant p processeurs. L'accélération de la solution parallèle, notée $S(p)$, est égale à $S(p) = T_1 / T_p$.

a) $S(p)$ peut-il être supérieur à p , c'est-à-dire peut-on obtenir une accélération surlinéaire ? Commenter et justifier votre réponse.

b) Déterminer une borne supérieure de $S(p)$ et montrer que $S(p)$ tend vers $1/f$ lorsque p tends vers l'infini - ($f = T_s/T_1$, $T_1 = T_s + T_{//}$ où T_s est égal au temps d'exécution de la partie séquentielle de A et $T_{//}$ est égal au temps d'exécution de la partie parallélisable de A).

Exercice 4

On désire effectuer une opération de normalisation d'une matrice triangulaire inférieure M. La matrice M contient $N * N$ éléments répartis en N lignes contenant 1 élément non nul sur la première ligne, 2 sur la deuxième, etc... jusqu'à contenir N éléments sur la Nième ligne.

L'opération de normalisation consiste à diviser chaque élément non nul de la matrice M par une constante. Cette division consomme T_{div} unités de temps par valeurs.

a) Pour une matrice de taille $N * N$, donner le temps d'exécution séquentielle de l'opération de normalisation.

On décide de paralléliser l'opération décrite ci-dessus sur une machine fournissant P processeurs tels que P divise N exactement.

b) Pour une matrice de taille $N * N$, donnez le temps d'exécution séquentiel de l'opération de normalisation si chaque processeur consomme N/P lignes. Donnez l'accélération et l'efficacité de cet algorithme.

c) Proposez un schéma de répartition des données permettant d'augmenter l'efficacité et l'accélération de cette algorithme parallèle.