TD Architecture Matérielle - Parallélisme

Exercice 1

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

$S(p)=\frac{T_1}{T_p}$. On néglige les effets de caches.

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

<u>Correction</u>

Non, loi d'Amdhal dit :

$$S(p)=\frac{T1}{Tp}=\frac{T1}{T1(f+\frac{1}{p}(1-f))}=\frac{1}{f+\frac{1}{p}(1-f)}$$

(p : le nombre de processus, f $(\in[0,1])$: le ratio purement séquentiel d'un algorithme)

Donc $\frac{1}{f+\frac{1}{p}(1+f)}>p$?

$\Leftrightarrow 1>pf+1-f \Leftrightarrow f>pf \Leftrightarrow 1>p$ $\rightarrow$ impossible

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

<u>Correction</u>

$S(p)=\frac{1}{f+\frac{1}{p}(1-f)}$ (Amdhal)

Or $\frac{1}{p}$ tend vers 0 quand p tend vers l'infini. Donc le total tend vers $\frac{1}{f}$. Borne : 1.

Exercice 2

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.

Correction

$$T_1 = \frac{(N+1)*N}{2} * T_{div}$$

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 $\frac{N}{P}$ lignes. Donnez l'accélération et l'efficacité de cet algorithme. Quelle est la limite de l'accélération lorsque P tend vers l'infini ?

Correction

On pose $S = \frac{N}{P}$, quantité de lignes par processeur

Le processeur avec le plus de ligne pleine (le dernier) est le plus lent

$$T1_P = \left(N*S - \frac{S*(S-1)}{2}\right) * T_{div}$$

$N*S$ = le nombre d'éléments au total sur les $\frac{N}{P}$ lignes.

$\frac{S*(S-1)}{2}$ = le nombre d'éléments nuls.

$$S(p) = \frac{T_1}{T1_P} = \frac{\frac{(N+1)*N}{2} * T_{div}}{\left(N*S - \frac{S*(S-1)}{2}\right) * T_{div}} = \frac{(N+1)*N}{2\mathrm{N}*S - S*(S-1)} = \frac{(N+1)*\frac{N}{S}}{2\mathrm{N} - S + 1} = \frac{P*(N+1)}{2\mathrm{N} - \frac{N}{P} + 1}$$

$$E_p = \frac{T_1}{P*T1_p} = \frac{\frac{(N+1)*N}{2}}{p*(N*S - \frac{S*(S-1)}{2})} = \frac{\frac{(N+1)*N}{2}}{p*(N*\frac{N}{P} - \frac{\frac{N}{P}*(\frac{N}{P}-1)}{2})}$$

$$E_p = \frac{(N+1)*N}{2*(N^2 - \frac{N*(\frac{N}{P}-1)}{2})} = \frac{N+1}{(2\mathrm{N} - \frac{N}{P}+1)}$$

$$S(p)_{inf} = \frac{P}{2}$$

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

Correction

Chaque processeur *i* travaille sur $\frac{N}{2\mathrm{P}}$ paires de lignes symétriques par rapport au milieu de la matrice. Tous les processeurs calculent la même quantité.

$$T2_p = \frac{N}{2\mathrm{P}}*(N+1)$$

$$S(p) = P$$

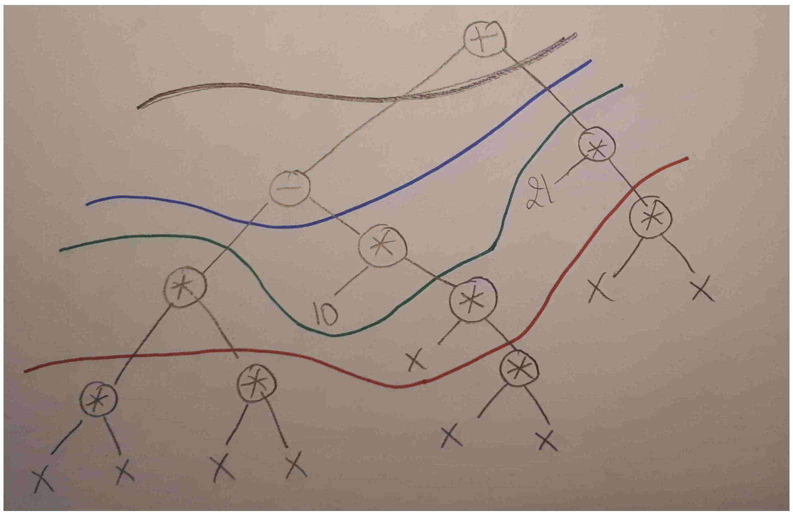
$$E_p = 1$$

Exercice 3

Soit l'expression $E = x^4 - 10x^3 + 21x^2$

a) Dessiner l'arbre d'évaluation de cette expression. Paralléliser au maximum le calcul de l'expression. En déduire le nombre de processeurs nécessaire pour cette parallélisation.

<u>Correction</u>
Une solution parmis d'autres :
Nombre de processus : 4 (lors de la première 'vague', on peut faire 4 calculs en parallèl).



b) Calculer l'accélération et l'efficacité de la solution parallèle en supposant que chaque opération arithmétique prend une unité de temps.

<u>Correction</u>
$T_1 = 10$
$P = 4,\ T_4 = 5$
$E_4 = 1/2,\ S(4) = 2$

D'autres solutions existent, par exemple avec 5 processus :
$P = 5,\ T_5 = 4$
$E_5 = 1/2,\ \ S(5) = 5/2$

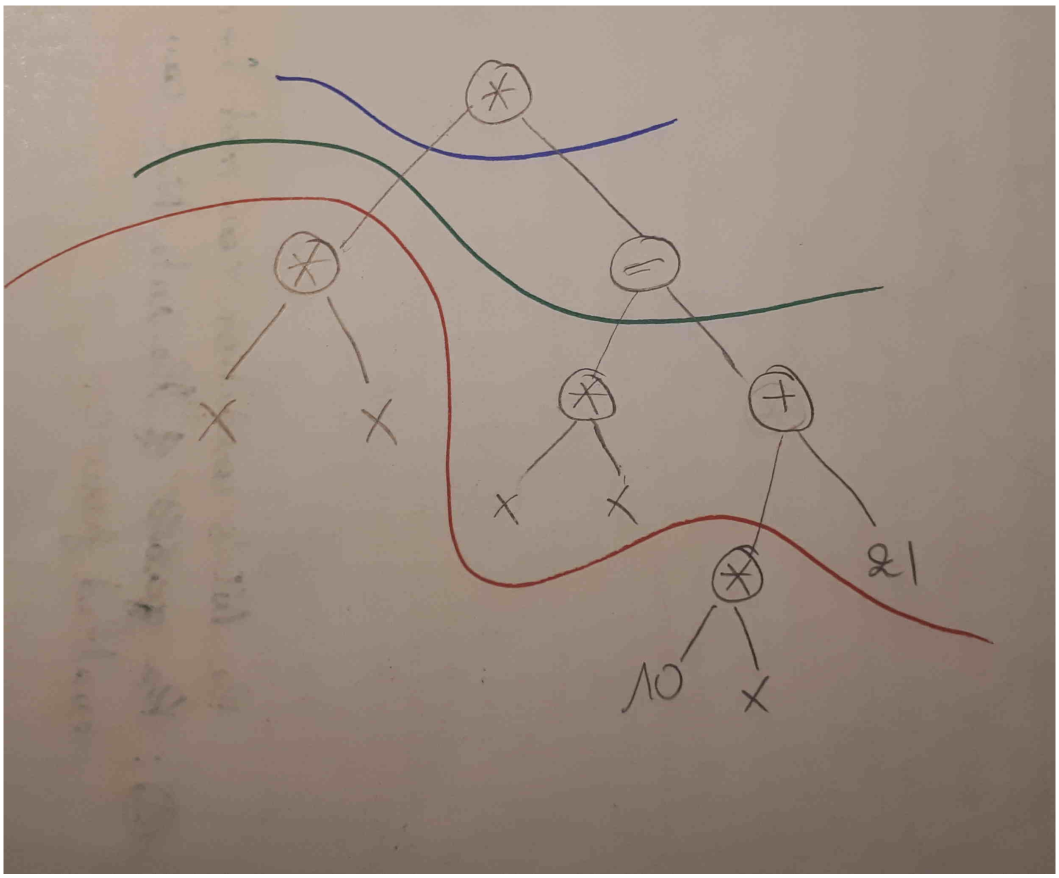
Ou avec 3 processus :
$P = 3,\ T_3 = 5$
$E_3 = 2/3,\ S(3) = 2$

Donc : tout dépend ce que l'on veut : aller le plus vite possible ou utiliser au maximum les processeurs qu'on a à disposition ?

c) Modifier l'expression en la factorisant par x^2 et recalculer l'accélération et l'efficacité obtenue. Quel est le nombre de processeurs nécessaires ?

Correction

$E' = x^2(x^2 - 10x + 21)$



$T_1 = 6$

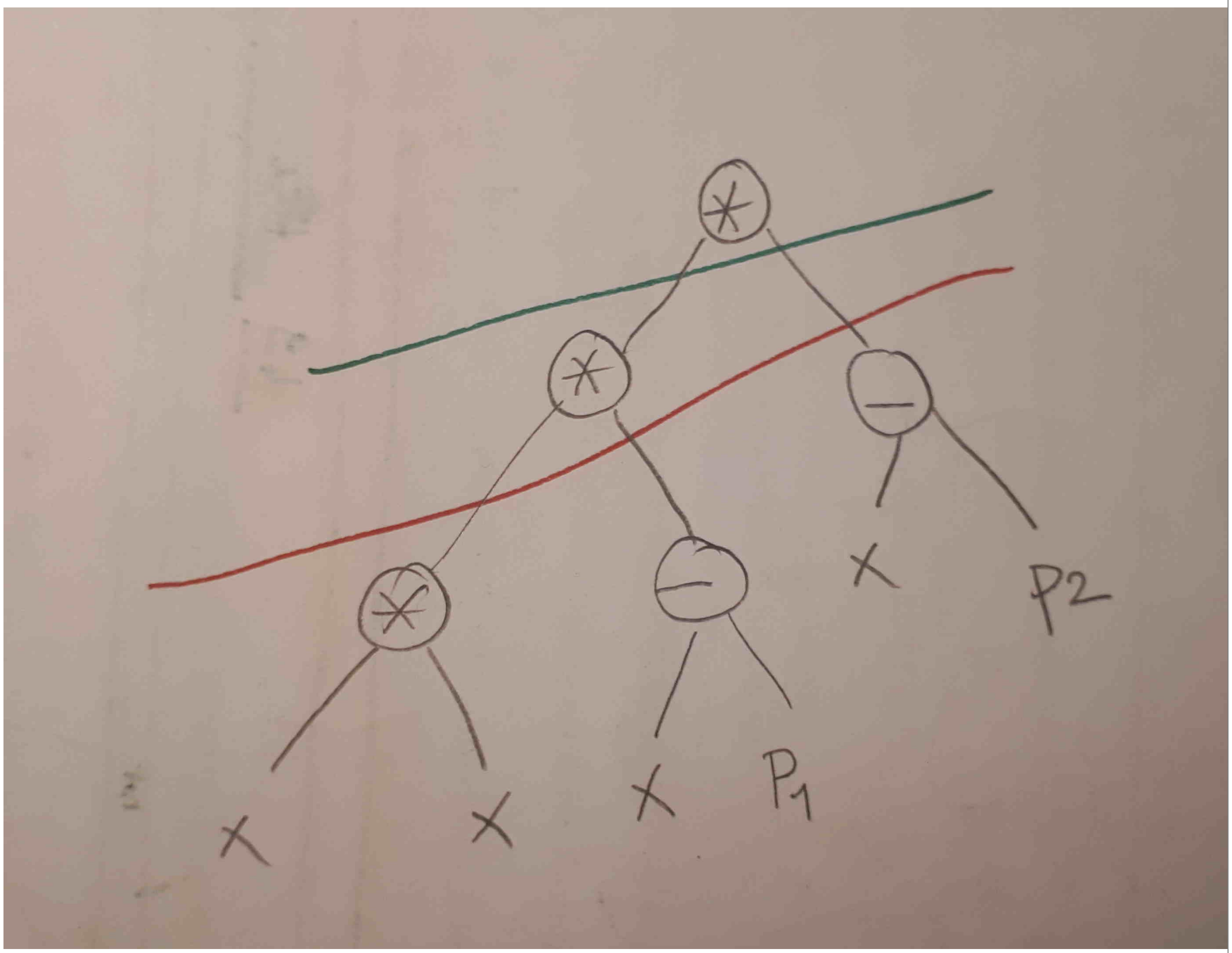
$P = 2$, $T_2 = 4$

$E_2 = 3/4$, $S(2) = 3/2$

d) Modifier l'expression en la factorisant sous la forme de $x^2(x - p1)(x - p2)$. Calculer à nouveau l'efficacité, l'accélération et le nombre de processeurs nécessaires.

Correction

$E''=x^2(x-7)(x-3)$



$T_1 = 5$

$P = 3,\ T_3 = 3$

$E_3 = 5/9,\ S(3) = 5/3$

e) Refaites les mêmes calculs en fixant le nombre de processeurs à 2. Quels changements pouvez-vous observer ?

Correction

La vague du bas ne prend que deux calculs (pas celui de droite, par exemple) et la deuxième fait 2 calculs)

$T_1 = 5$

$P = 2,\ T_2 = 3,$

$E_2 = 5/6,\ S(2) = 5/3$

Exercice 4

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.

Correction

$T_1 = (n-1) * T_c$

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 assise à portée, calculer le temps nécessaire aux 8 personnes pour faire la somme n nombres dans les cas suivants :

Correction

Pour toutes les configurations, on commence par calculer *(n/8 – 1)* sommes en même temps :

$(\frac{n}{8} - 1) T_c$

- 8 personnes assises en cercle.

Correction

$(\frac{n}{8} - 1) T_c + 4\mathrm{T}_c + 4\mathrm{T}_w$ car on parcourt en parallèle les deux côtés du cercle

- 8 personnes formant deux rangées de 4 personnes chacune.

Correction

$(\frac{n}{8} - 1) T_c + 3\mathrm{T}_c + 3\mathrm{T}_w$ car on envoie en face 4 semi sommes puis 2 puis 1

- 8 personnes dans une configuration d'arbre binaire équilibré.

Correction

$(\frac{n}{8} - 1) T_c + 4\mathrm{T}_c + 2\mathrm{T}_w$ car on remonte l'arbre du haut et du bas en même temps