

Semaine 11 : Mémoires hiérarchiques [Solutions]

1 Bande passante (des mémoires)

On a 2 heures – 1h45 = 15 minutes pour télécharger les vidéos.

- a) — L'ordinateur A : On a $\frac{800\text{Mo}}{4\text{octets/mot}} = 200 \times 10^6$ mots. Alors on a besoin de $5\mu\text{s} \times 200 \times 10^6$ mots = 1000 s = 16.67 minutes > 15 minutes. On ne peut pas utiliser cet ordinateur.
- L'ordinateur B : On a besoin de $\frac{800\text{Mo}}{400\text{Mo/s}} = 2\text{s} < 15$ minutes. On peut utiliser cet ordinateur.
- L'ordinateur C : On a besoin de $\frac{800\text{Mo}}{2\text{Mo/s}} = 400\text{s} = 6.67$ minutes < 15 minutes. On peut utiliser cet ordinateur.
- b) — L'ordinateur A : On a $\frac{2000\text{Mo}}{4\text{octets/mot}} = 500 \times 10^6$ mots. Alors on a besoin de $5\mu\text{s} \times 500 \times 10^6$ mots = 25000 s = 416.67 minutes > 15 minutes. On ne peut pas utiliser cet ordinateur.
- L'ordinateur B : On a besoin de $\frac{2000\text{Mo}}{400\text{Mo/s}} = 5\text{s} < 15$ minutes. On peut utiliser cet ordinateur.
- L'ordinateur C : On a besoin de $\frac{2000\text{Mo}}{2\text{Mo/s}} = 1000\text{s} = 16.67$ minutes > 15 minutes. On ne peut pas utiliser cet ordinateur.

2 Acheter un ordinateur

- a) Sachant qu'une plus grande bande passante permet une lecture plus rapide, on fait la comparaison entre les bandes passantes des ordinateurs. Pour l'ordinateur A, c'est donné : 400 Mo/s. Pour l'ordinateur B, on fait le calcul : 12.5×10^6 bloc/s \times 4 mots/bloc \times 4 octets/mot = 200 Mo/s. Alors on choisit l'ordinateur A parce que 400 Mo/s > 200 Mo/s.
- b) Même si le processeur de B est plus rapide, avec moitié moins de cache on ne risque pas de profiter de son gain en vitesse.
- En effet, avec moitié moins de cache on a (en gros) deux fois plus de chance de faire des défauts de cache. Or chaque défaut de cache nous «coûte» de l'ordre de 100ns alors que 1 GHz de gain sur la vitesse du processeur ne fait gagner «que» 1ns par opération. Il faudrait donc avoir effectivement plus de 200 opérations réellement utiles du processeur entre deux défauts de cache (de A) pour pouvoir profiter du gain de vitesse de B en dépit de son cache plus petit, ce qui est peu réaliste.

3 Nombre de défauts de cache

Il y a 11 défauts au total :

Adresse	Cache	Cas
1		défaut de cache
3	Block(0-3)	en cache
8	Block(0-3)	défaut de cache
5	Block(8-11)	défaut de cache
20	Block(8-11) Block(4-7)	défaut de cache
18	Block(8-11) Block(20-23)	défaut de cache
19	Block(16-19) Block(20-23)	en cache
53	Block(16-19) Block(20-23)	défaut de cache
9	Block(16-19) Block(52-55)	défaut de cache
11	Block(8-11) Block(52-55)	en cache
4	Block(8-11) Block(52-55)	défaut de cache
43	Block(8-11) Block(4-7)	défaut de cache
5	Block(40-43) Block(4-7)	en cache
6	Block(40-43) Block(4-7)	en cache
9	Block(40-43) Block(4-7)	défaut de cache
18	Block(8-11) Block(4-7)	défaut de cache

4 Localité spatiale et temporelle

La localité spatiale est plus importante pour cette application, car chaque élément du vecteur est accédé seulement une fois (pas de localité temporelle) ; alors que des éléments successifs du vecteur (stockés les uns après les autres) sont utilisés presque en même temps. Ça n'aide pas d'avoir de petits blocs dans ce cas ; il vaut mieux de grands blocs pour lire plusieurs éléments d'un vecteur à la fois.

Pour aller plus loin

5 Hiérarchie de données : exemple du CERN

- a) Quel est le débit de données à stocker résultant ?
 $10^{15} \times 10^{-3} \times 10^{-3} = 10^9$, c.-à-d. **1 Go/s**
- b) Combien de disques à 100 Mo/s en parallèle sont nécessaires pour un tel débit ?
100 Mo/s = 0.1 Go/s, donc **10 disques**
- c) Combien d'heures faut-il pour remplir ces disques de 1 To (= 10^{12} octets) chacun ?
1 Go/s nécessite $\simeq 10^4$ secondes pour remplir 10 To; $10^4 \text{ s} \simeq$ **2h45**
- d) Quel volume de données cela représente-t-il en 1 an ?
1 an $\simeq 3.16 \cdot 10^7 \text{ s}$; donc \simeq **31.6 Po** ($3.16 \cdot 10^7 \text{ Go}$).

6 Mémoire cache idéale

L'étudiant 1 s'est trompé car son résultat n'est pas optimal ($6000 > 5000$).

Les résultats de l'étudiant 2 semblent corrects.

L'étudiant 3 s'est trompé car il trouve plus de défauts pour 6 Mo que pour 4 Mo ce qui n'est pas possible avec OPT.

L'étudiant 4 s'est trompé car 0 défauts pour OPT impliquerait que la cache est assez grande pour contenir toutes les données et donc aucun algorithme (LRU ou autre) ne devrait constater de défaut de cache (il n'y a rien à remplacer dans le cache, donc la façon de remplacer n'importe pas ici); or LRU en a fait 3000.

7 Une analyse plus informative

- a) Le programme vérifie si x est premier.
 b) Les accès de mémoire sont dans le tableau suivant :

Pas	Instruction	Accès Mémoire
1	$y \leftarrow 1$	lire @1 (défaut de cache)
2	$y \leftarrow 1$	lire bloc @0 en mémoire
3	$y \leftarrow 1$	placer bloc @0 en cache
4	$y \leftarrow 1$	lire @1 (en cache)
5	$x = 0?$	lire @0 (en cache)
6	$x = 1?$	utilise registre (x déjà lu)
7	$s \leftarrow 0$	lire @13 (défaut de cache)
8	$s \leftarrow 0$	lire bloc @12 en mémoire
9	$s \leftarrow 0$	placer bloc @12 en cache
10	$s \leftarrow 0$	lire @13 (en cache)
11	$i \leftarrow 2$	lire @12 (en cache)
12	$s \leq x?$	utilise registre (s déjà lu)
13	$s \leq x?$	utilise registre (x déjà lu)
14	$y = 1?$	utilise registre (y déjà lu)
...	...	n'utilise plus que des registres

- c) On a 4 accès en cache en total et 2 défauts de cache.
 d) Pour exécuter le programme, le processeur accède la mémoire cache chaque fois qu'il veut lire ou écrire une variable. On compte 7 accès en tout, 4 en lecture comme nous l'avons vu et 3 en écriture à la fin (pour i , y et s ; x , lui, n'a pas été modifié).

On calcule donc le temps nécessaire pour chaque ordinateur de finir ce programme. Pour l'ordinateur A, on a besoin de

$$1 \text{ ns} \times 7 + 120 \text{ ns} \times 2 = 247 \text{ ns.}$$

Pour l'ordinateur B, on a besoin de

$$1.2 \text{ ns} \times 7 + 100 \text{ ns} \times 2 \simeq 208 \text{ ns.}$$

- e) Si on utilise cet exemple pour décider, on choisit l'ordinateur B; ce qui est normal ici, le programme utilisé ne nécessitant pas beaucoup de variables/mémoire, il y a peu d'accès à celle-ci mais «beaucoup» de défauts de cache (relativement au ratio d'environ 100 du temps d'accès mémoire par rapport à l'accès cache).