Semaine 11 : Mémoires hiérarchiques

1 Bande passante (des mémoires)

Vous voudriez suivre une leçon vidéo d'un cours en ligne et vous souhaitez la sauvegarder sur le dique d'un ordinateur *avant* de la regarder. Mais vous avez un autre rendez-vous pour lequel vous devez partir dans 2 heures. La durée de la leçon est de 1 heure 45 minutes (vous n'avez donc que 15 min. pour effectuer le téléchargement). Vous avez les ordinateurs suivants à disposition :

- l'ordinateur A a un disque dur qui peut transférer un mot [1] chaque 5 μs ;
- l'ordinateur B a un disque dur avec un débit [2] de 400 Mo/s ;
- l'ordinateur C a un disque dur avec un débit de 2 Mo/s.

On suppose ici que la bande passante de l'accès Internet est largement supérieure à celle de l'accès aux disques.

Quel(s) ordinateur(s) vous permettra/permettront de télécharger la leçon, de la regarder et de partir à temps pour votre rendez-vous si

a) la taille du fichier de la leçon est de 800 Mo ?

b) la taille du fichier de la leçon est de 2 Go ?

2 Acheter un ordinateur

Vous êtes dans un magasin d'électronique et vous voudriez acheter un ordinateur. Vous hésitez entre deux possibilités :

- l'ordinateur A a un processeur à 2 GHz et un disque dur ayant une bande passante de 400 Mo/s. Il a 8 Mo de cache ;
- l'ordinateur B a un processeur à 3 GHz et et un disque dur ayant une bande passante de 12.5 MegaBlocs/s (chaque bloc se compose de 4 mots). Il a 4 Mo de cache.

Que choisissez vous si vous désirez minimiser le temps nécessaire pour :

a) lire un fichier ?

b) exécuter un programme qui accède souvent à la mémoire ?
(Remarquer que les caches de A et B sont du même modèle, mais il y a deux fois plus de blocs dans celui de A que dans celui de B.)

1. En informatique, un mot est l'unité de base manipulée par un microprocesseur. La taille d'un mot s'exprime en bits ou en octets, et est souvent utilisée pour classer les microprocesseurs (32 bits, 64 bits, ...). Dans cet exercice, on considère qu'un mot est égal à 4 octets. [Référence : wikipedia.org]
2. On utilise aussi souvent le mot « bande passante » pour parler de débit numérique.

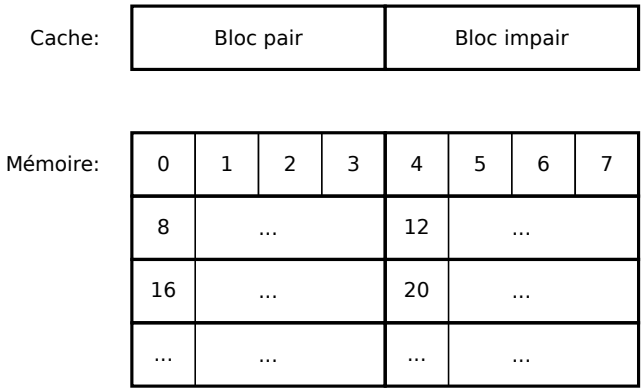

FIGURE 1 – Cache à accès direct.

3 Nombre de défauts de cache

Vous avez une mémoire cache ayant une taille de 8 mots. La taille d'un bloc de mémoire cache est de 4 mots. La mémoire cache utilise un accès direct. Les blocs de mémoire ayant une adresse paire sont mis en correspondance avec le premier bloc dans la mémoire cache et les blocs de mémoire ayant adresse impaire sont mis en correspondance avec le second bloc dans la mémoire cache (voir Fig. 1). Par exemple, les blocs de mémoire contenant les mots d'adresses 0–3 (addresse de bloc 0), 8–11 (addresse de bloc 2) et 16–19 (addresse de bloc 4) sont mis en correspondance avec le premier bloc dans la cache. Les blocs de mémoire contenant les mots d'adresses 4–7 (addresse de bloc 1), 12–15 (addresse de bloc 3) et 20–23 (addresse de bloc 5) sont mis en correspondance avec le second bloc dans la cache.

En supposant que la mémoire cache est vide au départ, combien de défauts de cache se produisent lorsque le processeur accède à la séquence de mots d'adresses suivante :

1, 3, 8, 5, 20, 18, 19, 53, 9, 11, 4, 43, 5, 6, 9, 18 ?

4 Localité spatiale et temporelle

Supposons que vous vouliez calculer la somme de deux grands vecteurs dont la taille est beaucoup plus grande que la taille de la mémoire cache. Quel type de localité est la plus pertinente pour cette application ? Est-ce qu'il vaut mieux avoir des petits blocs de cache pour cette application ?

Pour aller plus loin

5 Hiérachie de données : exemple du CERN

L'accélérateur de particules LHC du CERN génère 1 Po/s ($= 10^{15}$ octets/s), soit l'équivalent de 12 ans de vidéo en HD par seconde ! Il n'y a simplement aucune technologie au monde qui puisse enregistrer toutes ces données en temps réel. Même en mettant beaucoup de disques en parallèle, on n'arriverait pas à construire une connexion assez rapide pour y enregistrer toutes ces données en temps réel sans en perdre.

Une batterie de circuits logiques (hardware) analysent ces données en temps réel et les filtrent par un facteur de l'ordre de 1000 avant de les injecter directement dans les mémoires centrales d'une batterie de processeurs spécialisés qui les réduisent encore par un facteur de l'ordre de 1000.

En arrondissant aux ordres de grandeur :

a) Quel est le débit de données à stocker résultant ?

b) Combien de disques à 100 Mo/s en parallèle sont nécessaires pour un tel débit ?

c) Combien d'heures faut-il pour remplir ces disques de 1 To ($= 10^{12}$ octets) chacun ?

d) Quel volume de données cela représente-t-il en 1 an ?

6 Mémoire cache idéale

L'algorithme de remplacement de blocs en mémoire cache vu en cours s'appelle « LRU », pour « *Least Recently Used* », car il consiste à toujours remplacer en cache le bloc utilisé le plus loin dans le passé (c.-à-d. le bloc non utilisé depuis le plus longtemp).

Le meilleur de tous les algorithmes de remplacement serait celui qui donne toujours le nombre de défauts de cache minimal. Appelons « OPT » (pour optimal) cet algorithme théorique qui consiste à toujours remplacer en cache le bloc qui serait (ré)utilisé le plus loin dans le futur. Cet algorithme est irréalisable en pratique vu que la mémoire cache ne peut pas prédire le futur.

LRU et OPT possèdent tous les deux la bonne propriété qu'un accroissement de la taille de la mémoire cache ne conduit jamais à un accroissement du nombre de défauts de cache [3].

Après remplissage initial de la mémoire cache, [4] l'exécution d'un programme donné a causé 3000 défauts de cache pour une mémoire cache de taille 6 Mo gérée par l'algorithme LRU.
Pour une cache de seulement 4 Mo gérée par le même LRU, le même programme a causé 5000 défauts de cache.

3. Ce n'est pas le cas de tous les algorithmes. Par exemple, un algorithme de gestion de file « FIFO (First-In-First-Out) » ne vérifie pas de cette propriété.

4. On ne compte donc pas ici les défauts de cache liés au remplissage initial.

Pour ce même programme et ces mêmes tailles de cache, quatre étudiants ont simulé le comportement qu'aurait eu l'algorithme OPT et on trouvé les résultats suivants :

	6 Mo	4 Mo
1er étudiant	3000	6000
2e étudiant	2000	3000
3eétudiant	2000	1000
4eétudiant	0	3000

Plusieurs de ces étudiants ont fait des erreurs. Sauriez-vous dire lesquels et pourquoi ?

7 Achat : une analyse plus informative

Vous êtes dans un magasin d'électronique et hésitez entre deux choix. Vous demandez plus de détails et découvrez que pour l'ordinateur A, chaque accès au cache prend 1 ns (=1 nanoseconde) et chaque défaut de cache prend 120 ns, tandis que pour l'ordinateur B, chaque accès au cache prend 1.2 ns et chaque défault de cache prend 100 ns. Vous décidez de prendre un exemple d'un programme réel pour comparer les deux ordinateurs. L'algorithme du programme envisagé est le suivant :

Algorithme pour programme de test

entrée : x *entier naturel*
sortie : y *entier naturel*

```
y ← 1
Si x = 0 ou x = 1
    y ← 0
s ← 0, i ← 2
Tant que s ≤ x et y = 1
    Si i divise x
        y ← 0
    Sinon
        i ← i + 1
        s ← i × i
Sortir : y
```

Supposons que pour chaque ordinateur la mémoire cache contient 2 blocs et la mémoire en contient 4. Chaque bloc contient 4 mots. L'adresse d'un mot en mémoire est dans l'intervalle $[0, 15]$.

Dans cet exercise on ne compte pas les accès mémoire necessaires pour lire les instructions. On suppose par ailleurs quele processeur a plus de 4 registres de sorte qu'avec le programme utilisé ici, une fois qu'une variable a été lue dans le cache il n'est pas nécessaire de la relre.

Supposons enfin que x est stocké à l'adresse mémoire 0, y en 1, i en 12, et s à l'adresse 13 (toutes les variables sont initialement en mémoire), et que la valeur de x est initialement 4.

a) Que fait le programme ci-dessus ?

b) Explicitez les accès à la mémoire que le programme fait pour lire les 4 variables.

c) Combien d'accès au cache a-t-on ? Combien de défauts de cache a-t-on ?

d) Si vous devez choisir l'ordinateur le plus rapide, lequel prendriez-vous sur la base de cet algorithme ?