

TD 4 – Architecture des Ordinateurs

Mémoire cache

Licence Informatique 3^{ème} Année – UPPA

Les 2 premiers exercices traitent de l'étude de hiérarchies mémoire incluant plusieurs niveaux de cache. Les temps d'accès des mémoires y sont exprimés en nombre de cycles d'horloge processeur. Le temps d'accès en cas d'échec correspond au temps qu'il faut pour rechercher la donnée dans le niveau de cache. Le temps d'accès en cas de succès est le temps total pour accéder à la donnée en cas de succès (temps de recherche de la donnée plus temps de lecture/écriture de la donnée). Le temps d'accès moyen t_i d'un niveau i est donné par la formule suivante :

$$t_i = \text{temps}_{\text{accès-succès}} * \text{taux}_{\text{succès}} + (\text{temps}_{\text{accès-échec}} + t_{i+1}) * \text{taux}_{\text{échec}}$$

Exercice 1 : étude d'une hiérarchie mémoire à 3 niveaux

Niveau	Temps accès (succès)	Temps accès (échec)	Taux de succès	Taille
Cache L1	3	1	80%	128 Ko
Cache L2	5	2	90%	512 Ko
Cache L3	12	4	95%	2 Mo
Mémoire	40			1 Go

- À partir des performances du tableau ci-dessus, calculer le temps d'accès moyen global pour les 4 hiérarchies mémoires suivantes :
 - Mémoire centrale seule
 - Mémoire centrale avec cache L1
 - Mémoire centrale avec cache L1 et L2
 - Mémoire centrale avec cache L1, L2 et L3
- Conclure sur l'apport de chaque niveau de cache dans cette hiérarchie

Exercice 2 : choix d'une hiérarchie mémoire

Type de mémoire	Temps accès succès	Temps accès échec	Taux de succès en L1	Taux de succès en L2	Taille
A	2	1	50%	X	32 Ko
B	7	5	80%	50%	64 Ko
B	10	7	90%	70% si L1=32 Ko 50% si L1=64 Ko	128 Ko
RAM	50				512 Mo

Le tableau ci-dessus décrit le temps d'accès pour une mémoire centrale ainsi que pour deux types de mémoire A et B en différentes tailles pouvant servir à construire des mémoires cache à 2 niveaux. La mémoire cache fonctionne en mode inclusif. La quatrième colonne du tableau donne le taux d'accès si la

mémoire est utilisée au niveau L1. La cinquième colonne du tableau donne le temps d'accès d'un type de mémoire si elle est utilisée au niveau L2 en fonction de la taille du niveau L1.

Questions :

1. Pourquoi le tableau ne donne pas de temps d'accès en cas d'échec ou de taux de succès d'accès pour la mémoire centrale ?
2. Expliquer d'où pourrait venir les différences de performances, à la fois en terme de temps d'accès et de taux de succès des 2 types de mémoire, et pourquoi cela peut dépendre du niveau précédent.
3. Déterminer et justifier la meilleure combinaison (parmi tous les types et tailles proposés) possible de ces mémoires A et B pour construire une mémoire cache à au plus 2 niveaux.

Exercice 3 : gestion en cache des données d'un programme

Nous allons dans cet exercice étudier comment les données d'un programme sont gérées en mémoire cache. Ce programme est le suivant :

```
int A[1000], B[1000], C[1000];
int sum, i, moy;
int NB = 1000;
...
sum = 0;
for(i = 0; i < NB; i++)
{
    C[i] = A[i] + B[i];
    sum += A[i];
}
moy = sum / NB;
```

Pour simplifier cette étude, nous allons considérer une hiérarchie mémoire avec un seul niveau de cache, dont voici les caractéristiques :

- Une mémoire centrale de 64 Mo
- Une mémoire cache d'une taille de 8 Ko
- Les lignes ont une taille de 32 octets

Les entiers ont une taille de 32 bits. Une adresse mémoire référence en mémoire centrale un mot mémoire d'un octet. Les données du programme sont stockées en mémoire de manière contigüe dans l'ordre de leurs déclarations. La première donnée est donc le tableau A dont le premier élément en mémoire se trouve à l'adresse 4000.

Questions :

1. Préciser où sont placées les données du programme en mémoire et les lignes auxquelles elles appartiennent.
2. La mémoire cache fonctionne en mode correspondance directe. Déterminer les principales évolutions du contenu de la mémoire cache. Déterminer les problèmes (défaut de cache, conflit de ligne ...) avec l'utilisation de ce mode.
3. La mémoire cache fonctionne maintenant en mode correspondance associative par ensemble de 2. Déterminer les principales évolutions du contenu de la mémoire cache.
4. Comparer l'efficacité des deux modes pour ce programme et estimer le taux de succès de chaque mode pour ces 2 méthodes.
5. Pour le mode associatif par ensemble, montrer qu'un *pre-fetching* efficace peut augmenter considérablement le taux de succès.