Caches

1 Étiquettes Et Index De Cache

Un processeur a 2 Gio de mémoire principale. Pour les différents caches ci-dessous :

1. cache de 2 Mio à correspondance directe et écriture simultanée avec des blocs de 16 octects
2. cache de 4 Mio à correspondance directe, réécriture et blocs de 32 octects
3. cache de 4 Mio associatif 4 voies (4 blocs par ensemble), réécriture et blocs de 32 octects

Adresse mémoire :

Étiquette	Index	Adresse dans bloc

Ligne de cache :

Étiquette	Contrôle	Instructions ou Données

1.1 Quelle est la décomposition d'une adresse mémoire (nombre de bits des différentes parties) ?

1.2 Donner les différentes parties d'une ligne (bloc) de cache (nombre de bits des différentes parties). Quel est le surcoût lié aux bits de contrôle et d'étiquette (par rapport à la partie « données » du cache) ?

2 Cache Données

On considère une architecture possédant un cache de données de 8 Kio octects organisé en blocs de 32 octects. On considère deux cas : correspondance directe et associativité par ensembles de 2 blocs, avec le LRU (*Least Recently Used*) comme algorithme de remplacement (on remplace la ligne la moins récemment utilisée). Les tableaux suivants de 4096 `floats` (la taille d'un `float` est de 32 bits) sont aux adresses suivantes :

X	Y	Z	X1	Y1	X2	Y2
0x0001 0000	0x0001 4000	0x0001 8000	0x0001 C000	0x0002 0000	0x0002 4000	0x0002 8000

2.1 Quels sont les éléments des tableaux `X` et `Y` qui peuvent occuper le premier mot du premier bloc du cache ?

Soit les quatre kernels de calcul suivants :

```
for (i = 0; i < N; ++i)
{
    s += X[i] * Y[i]
}
```

```
for (i = 0; i < N; ++i)
{
    s1 += X1[i] + Y1[i]

    s2 += X2[i] + Y2[i]
}
```

```
for (i = 0; i < N; ++i)
{
    s1 += X1[i] + Y1[i]
}
for (i = 0; i < N; ++i)
{
    s2 += X2[i] + Y2[i]
}
```

```
for (i = 0; i < N; ++i)
{
    s1 += X[i] + Y[i]
    s2 += X[i] + Z[i]
}
```

2.2 Pour chaque boucle, quel est le nombre de défauts de cache données par itération (on suppose que les variables scalaires sont toujours en registre) ?

3 Cache Données Et Algorithme De Remplacement

Un ordinateur a une mémoire principale constituée de 1 Mio. Il a aussi un cache de 4 Kio associatif par ensemble, avec 4 blocs par ensemble et 64 octects par bloc.

3.1 Calculer le nombre de bits pour l'étiquette, l'index et l'adresse dans le bloc de l'adresse d'un mot de la mémoire principale.

3.2 Le cache est initialement vide. Le processeur lit 4352 octects à partir des adresses 0, 1, 2, ..., 4351 (dans cet ordre) 10 fois de suite. L'accès à un bloc dans le cache est de 1 unité de temps auquel on ajoute 10 si le bloc n'est pas dans le cache (doit être chargé) ; estimer l'accélération résultant de l'utilisation du cache en supposant que l'algorithme LRU est utilisé pour le remplacement du bloc.

3.3 Même question en supposant que l'algorithme de remplacement remplace maintenant le bloc le plus récemment utilisé (MRU : *Most Recently Used*).

4 Cache Données (Bonus)

Soit le programme suivant, qui effectue la normalisation des colonnes d'une matrice `X[8][8]` : chaque élément de la colonne est divisé par la moyenne des valeurs de cette colonne.

```
float X[8][8]; float sum = 0.f;
for (size_t j = 0; j < 8; ++j)
{
    for (size_t i = 0; i < 8; ++i)
    {
        sum += X[i][j];
    }
    float average = sum / 8.f;
    for (size_t k = 8; k >= 1; -k)
    {
        X[k - 1][j] /= average;
    }
}
```

On suppose que l'on a un cache de 128 octects avec des blocs de 16 octects (soit 8 blocs pour le cache). L'adresse de X[0][0] est `0xF000` (sur 16 bits).

4.1 En supposant la correspondance directe, définir dans quelles lignes du cache vont chaque élément de la matrice. En déduire le nombre de défauts de cache pour l'exécution du programme. Quel serait le nombre de défauts de cache en écrivant la seconde boucle interne sous la forme : `for (size_t k = 0; k < 8; ++k)` ?

4.2 Avec un cache totalement associatif, quel est le nombre de défauts de cache pour le programme initial en utilisant le LRU comme algorithme de remplacement ?

4.3 Pour un cache associatif par ensemble 2 voies, définir dans quels ensembles vont les éléments de la matrice. Quel est le nombre de défauts de cache pour le programme initial en utilisant le LRU comme algorithme de remplacement ?