TD 1: 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 Do | nnées |

| .1 | Quelle est la décomposition d'une adresse mémoire (nombre de bits des différent parties)? |
|----|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

| 1.2 | Donne différe rappor | rt à la par | tie « don | nces " | , | | | | |
|--|---|---|---|---|---|---|--|--|-------------|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | Cache | Donnée | | | | | | | |
| On consi | considère un idère deux) comme al | cas : corresp lgorithme de | re possédant ondance dire remplacemen | ecte et as nt (on rei | ssociativité pa mplace la lign | de 8 Kio octects r ensembles de 2 e la moins récem l'resses suivantes | 2 blocs, avec le ament utilisée). I | LRU (Least Re | cen |
| on consi onsi Jsed e 40 | considère un idère deux) comme al | ne architectu: cas : corresp lgorithme de | re possédant ondance dire remplacemen | ecte et as nt (on red de 32 bits | ssociativité pa mplace la lign | r ensembles de 2 | 2 blocs, avec le ament utilisée). I | LRU (Least Re | cen |
| On consi Used e 40 | considère un idère deux) comme al | ne architectu cas : corresp lgorithme de (la taille d'u | re possédant ondance dire remplacemen n float est o | ecte et as nt (on rei de 32 bits | ssociativité pa mplace la lign s) sont aux ac | r ensembles de 2 e la moins récem lresses suivantes | 2 blocs, avec le ment utilisée). I : | LRU (<i>Least Re</i> Les tableaux su | cen |
| onsi Used Le 40 X | considère un dère deux) comme al 096 floats 0001 0000 Quels | ne architectur cas : corresp lgorithme de (la taille d'ur Y 0x0001 400 | re possédant ondance dire remplacemen n float est o Z 00 0x0001 | ecte et as nt (on rer de 32 bits 8000 c | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 | r ensembles de 2 e la moins récem lresses suivantes Y1 | 2 blocs, avec le ment utilisée). I : X2 | LRU (Least Re Les tableaux su Y2 0x0002 8000 | ivan |
| On consi | considère un dère deux) comme al 096 floats 0001 0000 Quels du pre | ne architectur cas : corresp lgorithme de (la taille d'un Y 0x0001 400 sont les é emier bloc | re possédant ondance dire remplacemen float est o Z 00 0x0001 léments o du cache | ecte et as nt (on rer de 32 bits 8000 0 des tab | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 | r ensembles de 2 e la moins récem lresses suivantes Y1 0x0002 0000 | 2 blocs, avec le ment utilisée). I : X2 | LRU (Least Re Les tableaux su Y2 0x0002 8000 | cen ivan |
| On consideration of the consid | considère un idère deux) comme al 1996 floats 19001 0000 Quels du pre | ne architectur cas : corresp lgorithme de (la taille d'ur y 0x0001 400 sont les é emier bloc | re possédant ondance dire remplacemen float est o Z 00 0x0001 léments o du cache | ecte et as nt (on rer de 32 bits 8000 0 des tab | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 oleaux X et | r ensembles de 2 e la moins récem lresses suivantes Y1 0x0002 0000 | 2 blocs, avec le ment utilisée). I : X2 | LRU (Least Re Les tableaux su Y2 0x0002 8000 | cen ivan |
| On consideration of the consid | considère un idère deux) comme al 1996 floats 19001 0000 Quels du pre | ne architectur cas : corresp lgorithme de (la taille d'ur y 0x0001 400 sont les é emier bloc kernels de cal < N; ++i) | re possédant ondance dire remplacement float est o Z 00 0x0001 léments o du cache cul suivants for (i = 0; { | ecte et as nt (on rer de 32 bits 8000 0 des tab | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 pleaux X et ++i) for (i { 11[i] s1 | r ensembles de 2 e la moins récem lresses suivantes Y1 0x0002 0000 Y qui peuve | 2 blocs, avec le ment utilisée). I | Y2 | centivan |
| en consideration of the consid | considère un dère deux) comme al 1996 floats 1996 floats 1996 floats 1996 du pre | ne architectur cas : corresp lgorithme de (la taille d'ur y 0x0001 400 sont les é emier bloc kernels de cal < N; ++i) | re possédant ondance dire remplacement float est o Z 00 0x0001 léments o du cache cul suivants for (i = 0; { | ecte et as nt (on rer de 32 bits 8000 0 des tab e? | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 pleaux X et ++i) for (i { 1[i] s1 } | r ensembles de 2 e la moins récem lresses suivantes Y1 0x0002 0000 Y qui peuve = 0; i < N; ++i | 2 blocs, avec le ment utilisée). I X2 | LRU (Least Re Les tableaux su Y2 0x0002 8000 le premier | ivan |
| on consisted with the consistence of the consistenc | considère un dère deux) comme al 1996 floats 1996 floats 1996 floats 1996 du pre | ne architectur cas : corresp lgorithme de (la taille d'ur y 0x0001 400 sont les é emier bloc kernels de cal < N; ++i) | re possédant ondance dire remplacemen n float est o Z 00 0x0001 léments o du cache cul suivants for (i = 0; { s1 += X | ecte et as nt (on rer de 32 bits 8000 0 des tab e? | ssociativité pa mplace la lign s) sont aux ac X1 0x0001 C000 pleaux X et ++i) for (i { 11[i] s1 } for (i { | r ensembles de 2 e la moins récem lresses suivantes Y1 0x0002 0000 Y qui peuve = 0; i < N; ++i += X1[i] + Y1[i] | 2 blocs, avec le ment utilisée). I X2 | Y2 | cen ivar |

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 |
|-----|--|
| | ordinateur a une mémoire principale constituée de 1 Mio. Il a aussi un cache de 4 Kio associatif par ensemble, avec ocs 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. |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

| nant le bloc le plus récemment utilisé (MRU : $Most\ Recently\ Used$). |
|---|
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

Même question en supposant que l'algorithme de remplacement remplace mainte-

4 Cache Données (Bonus)

3.3

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?