TP n°1 : Simulation d'un cache de données

1. Introduction

Ce TD machine est destiné à étudier les performances d'un cache données de premier niveau.
Ce cache sera simulé «au dessus» de programmes C. On utilise le fait de pouvoir en C obtenir l'adresse d'une *variable* avec l'opérateur &*variable*. Avant chaque utilisation d'une variable, on appelle une fonction « accès cache » qui comptabilise l'accès mémoire et détermine si l'accès au cache est un succès ou un échec.

2. Caractéristique du cache de données.

Le cache de données est caractérisé par les paramètres suivants :

- Taille du cache : 8 Ko ou 16 Ko ou 32 Ko
- Taille de la ligne : 16 octets ou 32 octets ou 64 octets
- Associativité : 1 ligne/ensemble (correspondance directe) ou 2 lignes/ensemble ou 4 lignes par ensemble

L'écriture est allouée (sur un défaut en écriture, la ligne est chargée dans le cache avant l'écriture).
Politique de remplacement : pour les caches associatifs par ensemble, la politique de remplacement est le LRU.
La performance du cache est définie par le taux d'échec, qui est le rapport du nombre d'échecs (défauts) de cache sur le nombre d'accès mémoire.
Tous les programmes fournis permettent d'obtenir le taux d'échec pour les 27 configurations (taille de cache – taille de ligne – associativité). La taille des vecteurs ou des matrices est donnée par #define N. Pour utiliser une autre taille de vecteurs ou de matrices, il faut recompiler le programme.

3. Produit scalaire

Prog31.c correspond au produit scalaire.
Pour les 27 configurations de cache données, quel est le taux d'échecs pour les valeurs suivantes de N : 1000, 64, 512, 1024, 2048
Expliquez les résultats.

4. Produit matrice – vecteur

Prog32.c correspond au produit matrice - vecteur.
Pour les 27 configurations de cache données, quel est le taux d'échecs pour les valeurs suivantes de N : 64, 100, 512, 1024
Expliquez les résultats.

5. Produit de matrices ijk

Prog33.c correspond au produit de matrices carrées $Z = X * Y$ dans l'ordre ijk.
Pour les 27 configurations de cache données, quel est le taux d'échecs et le nombre d'échecs par itération de la boucle interne pour les valeurs suivantes de N : 16, 64, 100
Expliquez les résultats.

6. Produit de matrices ijk après transposition.

Prog33T.c correspond au produit de matrices carrées Z = X * Y dans l'ordre ijk après transposition de la matrice Y.
Pour les 27 configurations de cache données, quel est le taux d'échecs et le nombre d'échecs par itération de la boucle interne pour les valeurs suivantes de N : 16, 64, 100
Expliquez les résultats.

7. Produit de matrices ikj

Prog33ikj.c correspond au produit de matrices carrées Z = X * Y dans l'ordre ikj.
Pour les 27 configurations de cache données, quel est le taux d'échecs et le nombre d'échecs par itération de la boucle interne pour les valeurs suivantes de N : 16, 64, 100
Expliquez les résultats.

8. Produit de matrices avec blocage

On utilise l'algorithme suivant :

```
for (jj=0;jj<N;jj+=B)
for (kk=0;kk<N;kk+=B)
for (i=0;i<N;i++)
      {
      for (j=jj;j<min(jj+B-1,N);j++)
            {
            s=0;
            for (k=kk;k<min(kk+B-1,N);k++)
                  {
                  s = s + x[i][k]*y[k][j];
                  }
            z[i][j]=s+z[i][j];
            }
      }
```

Avec le cache 16 ko, le programme Prog34.c donne le taux d'échecs et le nombre de défauts par itération de la boucle interne en fonction de la taille du ligne et de l'associativité pour différentes valeurs du facteur de blocage B.
Quels sont les taux d'échec pour N=100 et N=64 ?
Expliquer les résultats.

9. Annexes

Les programmes sont disponibles à l'adresse suivante :

http://www.lri.fr/~de/ArchiL3-1112.htm

Compte rendu de TP à faire par binôme. Voir site pour les dates limite.