

Architecture matérielle et parallèle

Simulation d'un cache de données

1. Introduction

Ce TP 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 & Détails techniques

Le cache

Le cache de données est caractérisé par les paramètres suivants :

Taille du cache : 4 Ko ou 8 Ko ou 16 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.

Exécuter le programme

Copiez dans <https://godbolt.org/> le contenu de chaque fichier (un fichier par exercice).

Dans la fonction *main*, la variable *N* détermine la taille. Changez et ré-exécutez pour voir les changements d'affichage. L'affichage permet, pour une taille donnée *N* de fournir les stats de défaut de cache en fonction de la taille du cache, des lignes et de l'associativité utilisée (soit 27 configurations). **Note : vous pouvez avoir un affichage plus complet de ce qui se passe en changeant `detailed_log` à `true`.**

3. Produit scalaire

La fonction `calc3` dans le fichier `exo3.cpp` calcule le produit scalaire de deux vecteurs. Pour les 27 configurations de cache données, quel est le taux d'échecs pour les valeurs suivantes de N : 64, 512, 1000, 1024, 2048
Expliquez les résultats.

4. Produit matrice – vecteur

La fonction `calc4` dans le fichier `exo4.cpp` calcule le produit d'une matrice et d'un 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

La fonction `calc5` dans le fichier `exo5.cpp` calcule le 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.

La fonction `calc6` dans le fichier `exo6.cpp` calcule le 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.