

TD-TP10 : Optimisations cache et SIMD.

Introduction

Les objectifs de ce TD machine sont les suivants.

- Observer l'impact des caches et des instructions SIMD sur les temps d'exécution de programmes en fonction des accès mémoire d'une part et des options de compilation d'autre part
- Observer la possibilité de « vectorisation » (utilisation des instructions SIMD par le compilateur)

Les programmes à utiliser sont disponibles dans la page Web du L313

<http://www.lri.fr/~de/ArchiL3-1314.htm>

Pour chaque exercice, on donnera les temps demandés pour N=100, N=400, et avec les options de compilation suivantes :

- Avec l'option de compilation **-O1 et -msse2**
- Avec l'option de compilation **-O2 et -msse2**
- Avec l'option de compilation **-O3 et -msse2**

1. Somme de deux vecteurs

Le programme TP9SV.c calcule la somme de deux vecteurs de deux manières différentes :

- Version scalaire simple
- Version utilisant des intrinsics d'instructions SIMD

Donner le temps d'exécution par élément des vecteurs pour les deux versions et les deux options de compilations.

2. Produit de matrices

Le programme TP9MM.c calcule le produit de deux matrices [N][N] de plusieurs manières

- Produit ijk
- Produit ijkT (ijk après transposition)
- Produit ikj
- Produit ikj_simd avec intrinsics
- Produit ijk4 (avec déroulage de la boucle interne)
- Produit ijk_simd avec intrinsics

Donner le temps d'exécution par élément des matrices pour les différentes versions et les deux options de compilations.

3. Produit matrice-vecteur

Le programme TPMV.c calcule un produit matrice-vecteur.

Mêmes questions.

4. Annexe : Mesures de temps

Outils de mesure

Les mesures de temps sous Linux utilisent les fonctions suivantes, qui donnent le nombre de cycles d'horloge.

```
unsigned long long read_cycles()
{
    unsigned int hi = 0, lo = 0;
    __asm__ __volatile__ ("rdtsc" : "=a"(lo), "=d"(hi) : :
        "memory" );
    unsigned long long that = (unsigned long long)((lo)|
        (unsigned long long)(hi)<<32 );
    return that;
}

double dtime()
{
    return (double) read_cycles();
}
```

Mesure du temps d'exécution

Pour obtenir les temps d'exécution, on exécute les programmes plusieurs fois. On peut faire la moyenne des temps obtenus en enlevant les valeurs « aberrantes » (très supérieures aux autres). On peut aussi utiliser la plus petite valeur. On peut aussi utiliser la médiane des valeurs.

```
double t1,t2 ;//déclaration des variables
t1 = dtime();
//Partie du programme dont on mesure le temps d'exécution.
t2 = dtime();
dt = t2-t1; // Nombre de cycles d'horloge processeur
```

Documentation du compilateur

man gcc