TP n° 1 : Mesures de temps d'exécution.

0. Introduction

Les objectifs de ce TP sont les suivants :

- se familiariser avec l'utilisation du compilateur C d'Intel (ICC) sous Linux, notamment pour mesurer de manière relativement précise les temps d'exécution de programmes.
- Observer l'impact de certaines options de compilation
- Mettre en évidence l'influence des caches sur les temps d'exécution

Ce TP est à effectuer sur des PC équipés de processeurs Intel

On demande un compte rendu de TP par binôme à rendre le **01-10-2012** par courrier électronique à de@lri.fr .

Pour toutes les mesures demandées, on utilisera successivement les options de compilation

- ***- O2***
- ***- O2 avec –msse2***

1. Copie de tableaux

Accès ligne ou accès colonne

Soit un programme C qui effectue la copie d'un tableau d'entiers 32 bits N x N dans un autre en utilisant successivement les deux algorithmes suivants :

Version ij

```
for (i=0;i<N;i++)
      for (j=0;j<N;j++)
            Y[i][j]=X[i][j];
```

Version ji

```
for (j=0;j<N;j++)
      for (i=0;i<N;i++)
            Y[i][j]=X[i][j];
```

Donner le temps d'exécution/élément (c'est-à-dire le temps d'exécution total divisé par N*N) pour chacune des versions pour N=100, N=500, N=1000..

Copie d'octets, copie floats et doubles (flottants SP et DP)

Soient deux programmes C qui effectuent la copie d'un tableau N x N octets, de N x N « floats » et NxN doubles dans un autre tableau

Donner le temps d'exécution/élément (c'est-à-dire le temps d'exécution total divisé par N*N) pour chacune des versions pour N=100, N=500, N=1000

2. Produit scalaire

En utilisant successivement des données entières (int) et des données flottantes (float), donner le temps d'exécution/élément (c'est-à-dire le temps d'exécution total divisé par N) pour chacune des versions pour N=100, N=500, N=1000.

3. Produit de matrices avec des données flottantes (double)

Produit de matrices ijk

Donner le temps d'exécution/nombre d'exécutions de la boucle interne (c'est-à-dire le temps d'exécution total divisé par N*N*N) pour chacune des versions pour N=1000.

Produit de matrices ijk après transposition d'une des matrices

Donner le temps d'exécution/ nombre d'exécutions de la boucle interne (c'est-à-dire le temps d'exécution total divisé par N*N*N) pour chacune des versions pour N=1000.

Produit de matrices ikj

Donner le temps d'exécution/nombre d'exécutions de la boucle interne (c'est-à-dire le temps d'exécution total divisé par N*N*N) pour chacune des versions pour N=1000.

Produit de matrices ijk avec blocage

Donner le temps d'exécution/nombre d'exécutions de la boucle interne (c'est-à-dire le temps d'exécution total divisé par N*N*N) pour chacune des versions pour N=1000.

4. Produits de matrices avec des données entières (int)

Refaire la partie 3 en utilisant des matrices « entières » au lieu de matrices « flottantes ».

5. Annexe : Mesures de temps

Les mesures de temps sous Linux sont données par les fonctions suivantes, qui donnent le nombre de cycles d'horloge.
Pour obtenir les temps d'exécution, on exécute les programmes un certain nombre de fois, et on fait la moyenne des temps obtenus en enlevant les valeurs « aberrantes » (très supérieures aux autres).

```
double dtime();
long long readTSC ();

long long readTSC ()
{
  long long t;
  asm volatile (".byte 0x0f,0x31" : "=A" (t));

  return t;
}
double dtime()
{
  return (double) readTSC();
}
```

Mesure du temps d'exécution

```
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
```

6. Annexe : utilisation de ICC

Documentation : http://www.lri.fr/~de/DOC-ICC.pdf

Les programmes à utiliser et modifier sont disponibles dans la page Web du M1-Archi
http://www.lri.fr/~de/ArchiM1-1213.htm