

Travaux Dirigés Structure des ordinateurs: Feuille 3

Télécom 1ère année. ENSEIRB-MATMECA 2014/2015

—Denis Barthou—

► Exercice 1. Mémoire cache

On rappelle que la mémoire cache est une mémoire plus rapide que la mémoire principale, dans laquelle on copie des données que l'on a utilisées et que l'on va réutiliser. Elle agit comme un proxy.

On considère le code suivant

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 8192

int main() {
    int i,j,k;
    float (*A)[SIZE]=(float (*)[SIZE])malloc(sizeof(float)*SIZE*SIZE);
    float (*B)[SIZE]=(float (*)[SIZE])malloc(sizeof(float)*SIZE*SIZE);

    // Mise a 0 des tableaux
    memset(A,0,SIZE*SIZE*sizeof(float));
    memset(B,0,SIZE*SIZE*sizeof(float));
    // Le code a modifier
    for (k=0; k<10; k++) {
        for (i=0; i<SIZE;i++) {
            for (j=0; j<SIZE; j++) {
                A[j][i]+=B[j][i]+i-j;
            }
        }
    }
    free(A);
    free(B);
    return EXIT_SUCCESS;
}
```

1. Mesurer le temps d'exécution du code en le compilant puis en le lançant avec :

```
> time ./monexecutable
```

2. Calculer la taille en octets des tableaux A et B La comparer avec la taille du cache mémoire de votre machine (linux) indiqué par "cache size" en faisant :

```
> more /proc/cpuinfo
```

3. Combien de fois chaque élément de A est-il réutilisé dans les boucles ? Est-ce qu'il y a une bonne localité temporelle ?

4. Combien d'octets séparent deux éléments de A accédés successivement dans la boucle j, comme $A[0][0]$ et $A[1][0]$?

5. Quelle quantité mémoire est accédée entre un accès à $A[0][0]$ et $A[0][1]$?

6. On peut permuter les boucles dans l'ordre qu'on veut sur ce code sans changer le sens du calcul. Proposer un ordre d'imbrication des boucles qui favorise la localité spaciale et temporelle et mesurer le temps d'exécution.