

Pruebas sobre el comportamiento de la memoria caché

Correa Ochoa Wilber *Departamento de Ciencia de la Computación*
Universidad Católica San Pablo
Arequipa, Perú
wilber.correa@ucsp.edu.pe

March 25, 2021

1 Implementación

Primero se generó números aleatorios a la matriz A y el arreglo x de la siguiente forma:

```
//random
static std::uniform_real_distribution <double>
    sample_01_(10000.0, 50000.0);
std::random_device rd;
std::mt19937 rng(rd());

for (int i=0; i<MAX; i++){
    x[i]=sample_01_(rng);
    for (int j=0; j<MAX; j++){
        A[i][j]=sample_01_(rng);
    }
}
```

Luego se realizó el primer bucle anidado según el libro, antes se le colocó al arreglo "y" todos sus elementos 0:

```
//First Loop
memset(y, 0.0, sizeof(y));
chrono::duration<float, micro> duration;
auto start = chrono::system_clock::now();
for (int i=0; i<MAX; i++){
    for (int j=0; j<MAX; j++){
        y[i]+= A[i][j]*x[j];
    }
}
auto end = chrono::system_clock::now();
```

```

duration = end - start;
cout<<"The time 1st loop  "<<duration.count()<<endl;

```

Al final se realizó el segundo bucle anidado según el libro, antes se le colocó al arreglo "y" todos sus elementos 0:

```

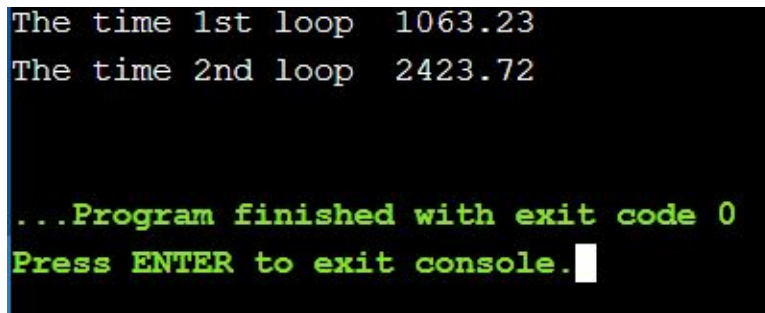
//SECOND LOOP
memset(y, 0.0, sizeof(y));
start = chrono::system_clock::now();
for(int j=0; j<MAX; j++){
    for(int i=0; i<MAX; i++){
        y[i] += A[i][j]*x[j];
    }
}
end = chrono::system_clock::now();
duration = end - start;

cout<<"The time 2nd loop  "<<duration.count()<<endl;

```

2 Resultados

Los resultados en tiempo mostrados en microsegundos son los siguientes:



```

The time 1st loop  1063.23
The time 2nd loop  2423.72

...Program finished with exit code 0
Press ENTER to exit console.

```

Figure 1: Resultado 1 de la ejecución del programa.

```
The time 1st loop  852.381
The time 2nd loop  77057.4

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 2: Resultado 2 de la ejecución del programa.

```
The time 1st loop  1079.86
The time 2nd loop  1971.57

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 3: Resultado 3 de la ejecución del programa.

3 Análisis de Resultados

Los resultados arrojan que el 1er bucle es más rápido. Lo que ocurre es que el primer bucle va recorriendo el bloque 0 de cache hasta llegar al final recorriéndolo por completo, luego acaba y recién pasa al bloque 1. Suponiendo que cada fila es capaz de llenar un bloque. Lo que pasa en el 2do bloque es que no está recorriendo el mismo bloque de manera consecutiva, sino que en la 1ra iteración va al bloque 0, en la 2da iteración pasa al bloque 1, luego al bloque 2 y así hasta llegar al final, luego regresa al bloque 0 de nuevo hasta el bloque final, así hasta recorrer por completo la matriz.

Vemos la diferencia, es que el 1er bucle realiza menos cambios de líneas de cache (la cantidad de filas), mientras que el 2do bucle va cambiando las líneas de cache a cada iteración (Max^2 veces).

4 Repositorio

<https://github.com/wilberever100/Paralelo/tree/main/bucles>