

Laboratorio 01

Comparación de desempeño - bucles anidados

Autor: Juan Manuel Soto Begazo
CUI: 20192267
Docente: Dr. Alvaro Henry Mamani Aliaga
Computación Paralela y Distribuida
Grupo A
Arequipa, Peru

1. Desarrollo

En el presente trabajo vamos a presentar la implementación, resultados y análisis de la ejecución para el siguiente problema:

- Implementar y comparar los 2-bucles anidados FOR presentados en el cap. 2 del libro, pag 22.

La implementación se encuentra en el siguiente repositorio de Git Hub.

1.1. Primer bucle for

En el primer bucle se recorre el el arreglo A por filas y luego por columnas, esta es la forma clasica de recorrer un arreglo de dos dimensiones. Esto significa que se está accediendo a los elementos de A de manera secuencial por filas, lo que generalmente resulta en un acceso de memoria más eficiente debido a la localidad de los datos. Esto se debe a que las matrices están almacenadas en la memoria de manera contigua por filas en la mayoría de los casos.

```
1      for (int i = 0; i < temp; i++){
2          for (int j = 0; j < temp; j++){
3              y[i] += A[i][j]*x[j];
4          }
5      }
```

1.2. Segundo bucle for

En el primer bucle se recorre el el arreglo A por columnas y luego por filas, lo que puede resultar en un acceso de memoria menos eficiente. Esto se debe a que, como mencionamos anteriormente, en la mayoría de los casos las matrices se almacenan en la memoria de manera contigua por filas, por lo que acceder a elementos de manera secuencial por columnas puede dar lugar a más caché misses y, por lo tanto, un mayor tiempo de acceso a memoria.

```
1      for (int j = 0; j < temp; j++){
2          for (int i = 0; i < temp; i++){
3              y[i] += A[i][j]*x[j];
4          }
5      }
```

2. Comparación y Análisis

Para realizar las comparaciones entre ambos bucles tratados en este trabajo realizamos simulaciones con diferentes cantidad de elementos en el array A .

Arrays de menos de 2000 filas y 2000 columnas ($A[2000][2000]$ o menos) Como podemos observar en la figura 2.2 el comportamiento para estos tamaños es similar, esto se produce debido a que en estas instancias muy probablemente estos arreglos estén cargados en la memoria cache.

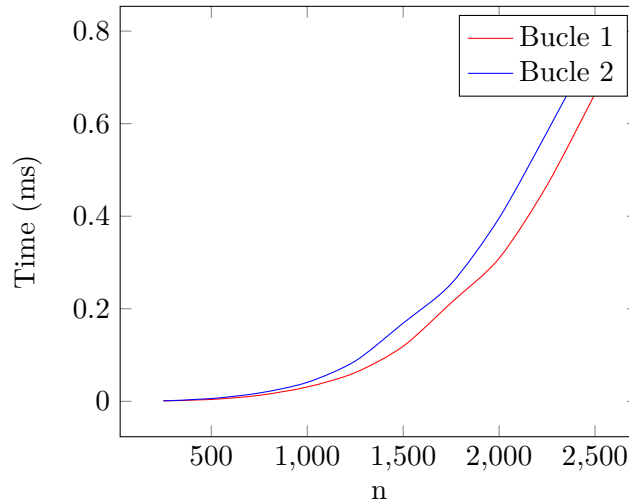


Figura 2.1: Tamaños pequeños hasta 2500*2500

Las diferencias en tiempos entre ambos bucles son menores a 0.2 s en el peor de los casos.

Arrays con más de 2000 filas y 2000 columnas ($A[2000][2000]$ o mas) Como podemos observar en la figura ?? el comportamiento para estos tamaños deja de ser similar, esto se produce debido a que

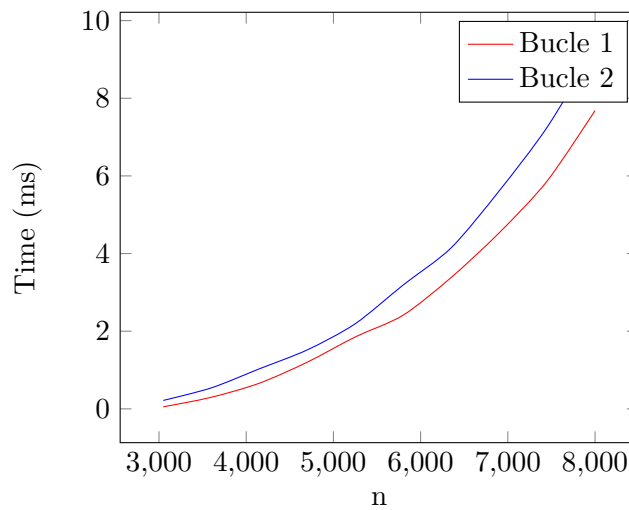


Figura 2.2: Tamaños mayores a 2500*2500

Las diferencias en tiempos entre ambos bucles en los casos con más elementos son de 2 a 3 segundos en tiempo de ejecución.

3. Conclusiones

- Bucle 1 es más eficiente en términos de acceso a la memoria y tiende a tener un mejor rendimiento en la mayoría de los casos.
- Bucle 2 puede experimentar más caché misses y, por lo tanto, tiempos de acceso a la memoria más largos.
- La elección entre Bucle 1 y Bucle 2 puede depender de la implementación específica y de las características del hardware, pero en la mayoría de los casos, Bucle 1 es preferible debido a su eficiencia en el acceso a la memoria.