**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod 202021368

Estudiante 2 Cod 201914771

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz | 1,3 GHz Intel Core i5 |
| Memoria RAM (GB) | 16.0 GB (15.8 GB usable) | 8 GB |
| Sistema Operativo | Windows 10 pro | macOS |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 0.30 | 314811.37 | 55639.84 |
| 0.50 | 312635.03 | 67265.13 |
| 0.80 | 311306.49 | 94820.11 |

Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 1.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 350201.80 | 63454.91 |
| 4.00 | 349580.46 | 69608.26 |
| 6.00 | 349350.68 | 83223.30 |

Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 1.**

* Comparación de memoria y tiempo de ejecución para PROBING y CHAINING

# **Maquina 2**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga** (PROBING) | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 0.30 | 1315531.890 | 90532.049 |
| 0.50 | 1314889.595 | 81507.917 |
| 0.80 | 1314883.704 | 87005.795 |

Tabla 4. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 2.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 1315532.532 | 85309.591 |
| 4.00 | 1315528.286 | 86267.497 |
| 6.00 | 1314889.188 | 92821.675 |

Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 2.**

* Comparación de memoria y tiempo de ejecución para PROBING y CHAINING

# **Preguntas de análisis**

1. ¿Por qué en la función **getTime()** se utiliza **time.perf\_counter()** en vez de la previamente conocida **time.process\_time()**?

El módulo time permite trabajar con medidas del tiempo de forma amplia. Las funciones perf\_counter() y process\_time() permiten obtener unidades de tiempo del código en cuestión. No obstante, estas tienen ciertas diferencias entre sí. perf\_counter()proporciona resultados más precisos mientras que process\_time() retorna la suma del sistema y el CPU. Así, la primera se acomoda más a lo que se necesita en este reto para tomar las mejores decisiones en cuanto a optimización y, así mismo, se acomoda mejor al TDA usado (maps).

1. ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?

El módulo tracemalloc permite medir el uso de memoria. La función start() inicia el recorrido de las líneas de código y stop() lo detiene. Así, permite medir el uso de memoria ocupada por una función o segmento de código en particular.

1. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de videos?

Dentro del esquema de colisiones chaining, se observa una relación casi directamente proporcional entre estas (a mayor factor de carga, mayor tiempo de ejecución). Mientras que en probing, 0.5, a pesar de estar en la mitad de 0.4 y 0.6, presentó mejores resultados que estos dos valores.

1. ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para

cargar el catálogo de videos?

Este no varía mucho, pero su comportamiento de asemeja al del tiempo de ejecución en menores rangos de dispersión.

1. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

Se pudieron obtener mejores resultados para probing que para chaining, aunque no por mucho. Los tiempos de ejecución fueron cercanos entre sí. No obstante, en chaining, hubo una relación directamente proporcional entre tiempo de ejecución y factor de carga, mientras que en probing no.

1. ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

Se pudieron obtener mejores resultados para probing que para chaining, aunque no por mucho. Los tiempos de ejecución fueron mucho más cercanos entre sí que los de ejecución, aunque, como se expuso, estos últimos también lo fueron.

En conclusión, se evidenció que los mejores resultados se obtuvieron para probing 0.5, aunque no por mucho. Esto se puede explicar por el funcionamiento de cada uno. Probing, gracias a su espacio disponible, permite una rápida solución de las colisiones. Así mismo, podemos entender que el chaining, al conformarse de varias listas enlazadas, aumenta la complejidad y, por lo tanto, el uso de tiempo y memoria.