**OBSERVACIONES DEL LA PRACTICA**

Gregorio Salazar 202022085

Valentina Uribe Salcedo 201817485

|  |  |  |
| --- | --- | --- |
|  | **Máquina 1** | **Máquina 2** |
| **Procesadores** | Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz | Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz |
| **Memoria RAM (GB)** | 8 | 8 |
| **Sistema Operativo** | Windows 10 | Windows 10 |

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 | 1626778.10 | 31102.10 | | 0.50 | 1626778.10 | 31314.50 | | 0.80 | 1626778.10 | 33388.20 | | | |

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.

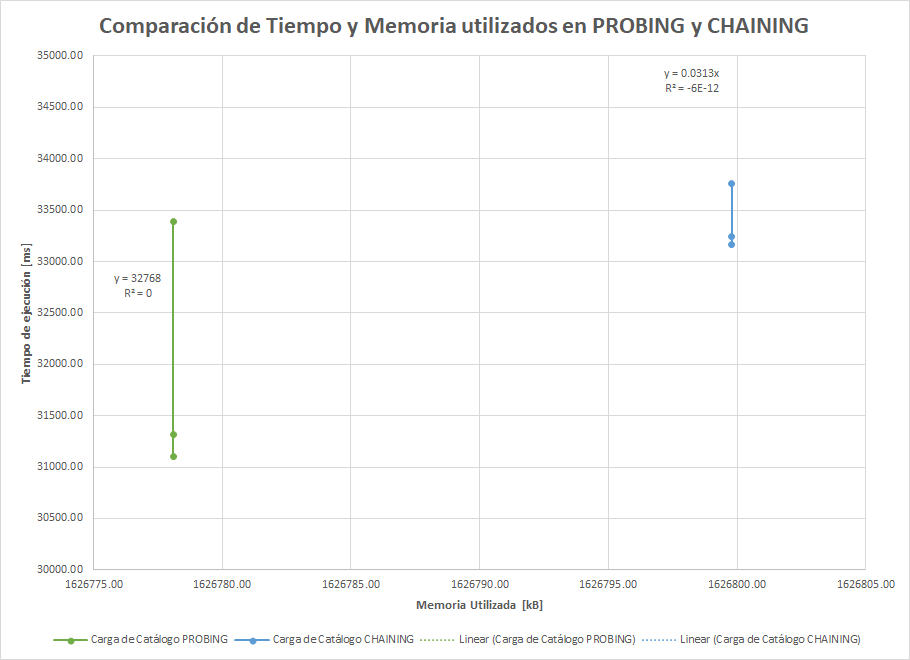
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Carga de Catálogo CHAINING***   |  |  |  | | --- | --- | --- | | **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** | | 2.00 | 1626799.80 | 33245.20 | | 4.00 | 1626799.80 | 33167.30 | | 6.00 | 1626799.80 | 33756.10 | | | |

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 | 1626778.10 | 50194.40 |
| 0.50 | 1626778.10 | 53601.81 |
| 0.80 | 1626778.10 | 53581.02 |

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.

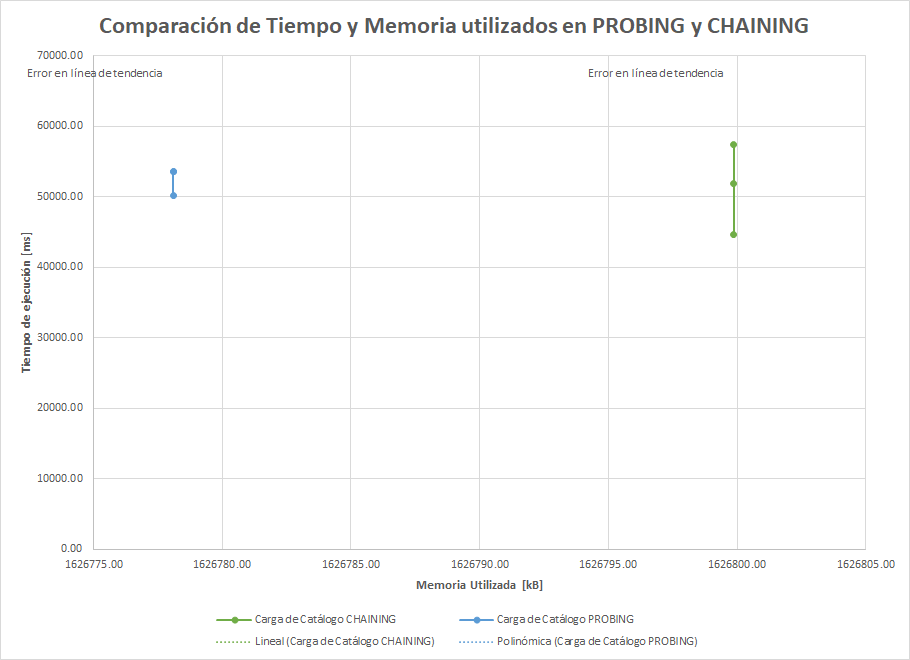
|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 1626799.87 | 51853.46 |
| 4.00 | 1626799.87 | 44690.31 |
| 6.00 | 1626799.87 | 57402.68 |

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 ves de la previamente conocida **time.process\_time()**?

En la función getTime() se utiliza time.perf\_counter() en ves de time.process\_time() ya que perf\_counter () cuenta la cantidad real de tiempo que tarda un proceso, como si fuera un cronómetro, mientras que process\_time () mide únicamente el tiempo empleado por la computadora para el proceso actual, ya que un operativo generalmente no dedica el 100% del tiempo a un proceso dado. Este contador no contaría el tiempo que la CPU está ejecutando cualquier otra cosa. Adicional a esto, time.perf\_counter() es más precisa.

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

La función start() es importante ya que da inicio al rastreo de las asignaciones de memoria y la función stop() captura los datos del rastreo de las asignaciones de memoria antes de que estos sean limpiados.

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

En el Probing el mejor factor de carga es 0.3, ya que minimiza las colisiones y reduce en gran medida las operaciones en el sistema. Esto puede usar mucha memoria porque usa un mapa más grande, pero como en este caso el mapa es bastante pequeño (40 elementos), la memoria no es un problema. Despúes los factores de 0.5 y 0.8 son ligeramente más lentos ya que hay más colisiones. En Chaining el mejor factor de carga e 4, ya que este optimiza las operaciones que se hacen con los buckets y con los espacios, y el peor es 6, ya que cuando se empieza a aumentar el factor de carga hay muchas colisiones y empieza a haber muchas operaciones.

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

Al cambiar el factor de carga no se ve absolutamente ningún cambio en el uso de memoria. Vemos que los valores son idénticos. Esto es porque la memoria usada es principalmente el peso del archivo a cargar y el mapa apenas espera alrededor de 40 elementos, que al cambiar el factor de carga, la cantidad de espacios aumenta muy poco que para el computador es insignificante.

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.

En promedio, los tiempos de Probing fueron mejores. Esto se debe que al tener buckets en los espacios del mapa, el chaining los debe referenciar cuando se hace el recorrido y este recorrido en la memoria es menos eficiente que en el linear probing. En la maquina dos vemos que el menor tiempo se da en separate chaining y no sabemos explicar esta situación.

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.

Al usar Chaining se usa un poco más de memoria al comparar con el Probing. Aunque la diferencia es minúscula, debido a que la mayoría de memoria viene por el tamaño del archivo, la pequeña diferencia en memoria se puede deber a que el Chaining usa estructuras dentro el mapa (los buckets) y estas referencias pueden usar un poco de espacio.