**OBSERVACIONES DE LA PRACTICA**

Juan Camilo Falla C.E. 201922219

Nicolas Klopstock C.E. 202021352

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | i7 10700 @2.9GHz- 4.8GHz | i5 6200U CPU @2.30GHz 2.40GHz |
| Memoria RAM (GB) | 16 gb | 4 gb |
| Sistema Operativo | Windows 10 Home | Windows 10 Home |

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 | 647919.55 | 25661.10 |
| 0.50 | 647818.65 | 25881.37 |
| 0.80 | 647815.00 | 26519.75 |

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 | 648744.15 | 29818.97 |
| 4.00 | 648670.16 | 30327.82 |
| 6.00 | 648521.11 | 31225.55 |

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**

NOTA: Es importante aclarar que, por temas de máquina, estos datos son con el archivo “videos\_small.csv”. La razón es porque el procesador y la memoria de la máquina 2 no son tan capaces (ahora) de cargar tantas veces el archivo mediano ni el grande sin esperar problemas luego de procesador y memoria.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING (videos\_small.csv)*** | | |
| **Factor de Carga** (PROBING) | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 0.30 | 1966,02 | 162,00 |
| 0.50 | 1962,79 | 164,18 |
| 0.80 | 1959,36 | 168,53 |

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 (videos\_small.csv)*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 2021,86 | 209,36 |
| 4.00 | 2014,38 | 211,64 |
| 6.00 | 2023,98 | 215,73 |

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

Para las pruebas de rendimiento, la función time.perf\_counter() da acceso al reloj con la mayor resolución disponible para unas mediciones cortas más exactas. Por otro lado, getTime() devuelve el tiempo del sistema y el tiempo del procesador combinados.

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

Con start(), empieza el proceso de medición de memoria. Sin esta función, no se “abriría” la opción para que la función getMemory() fuera efectiva, porque no podría revolver ningún valor de memoria usada.

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

En el caso de CHAINING, entre mayor sea el factor de carga, el tiempo de ejecución también aumenta. En el caso de PROBING, entre mayor sea el factor de carga, también mayor es el tiempo de ejecución.

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

A medida que se va aumentando el factor de carga máximo para poder subir la información de los videos al catalogo, se va reduciendo el consumo de memoria. En el caso de probing, se reducen los espacios vacíos en cadda tabla de Hash y en chaining, se puede agrupar una mayor cantidad de información bajo una misma llave.

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.

Después de realizar las pruebas con los dos tipos de solución de colisiones, *separate chaininig* y *linear probing*, se puede observar que el segundo método de solución de colisiones es más rápido para la carga de los datos. Esto puede ser porque hay más espacios vacíos en la tabla de hash y que cada llave va a tener un único espacio en dicha tabla. Por la naturaleza de este método, es poco probable que haya la misma cantidad, o más, colisiones que deban ser resueltas como lo sería para *separate chaining*.

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 puede observar que, en ambas máquinas, al utilizar el esquema *linear probing* se da un consumo menor de memoria. Esto puede ocurrir debido a que las listas encadenadas que se dan en *separate chaining* cuando ocurren colisiones, empiezan a ocupar un mayor espacio a comparación que los valores únicos o None que se pueden ver en *linear probing.*