**OBSERVACIONES DEL LA PRACTICA**

Felipe Garzón 202021161

Pablo Ortega 202021700

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core (TM) i5-8265 CPU @  1.60 GHz 1.80GHz | 1,6 GHz Dual-Core Intel Core i5 |
| Memoria RAM (GB) | 12.0 GB | 8.0GB |
| Sistema Operativo | Windows 10 Pro | Mac 10.15.6 |

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 | 1403130.873 | 25782.536 |
| 0.50 | 1403130.873 | 27519.940 |
| 0.80 | 1403130.819 | 24093.443 |

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 | 1403143.326 | 24288.766 |
| 4.00 | 1403143.326 | 24659.134 |
| 6.00 | 1403143.326 | 24745.641 |

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 | 1403130.08 | 30117.54 |
| 0.50 | 1403130.08 | 28394.46 |
| 0.80 | 1403130.03 | 28588.07 |

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 | 1403142.76 | 31228.53 |
| 4.00 | 1403142.76 | 26987.59 |
| 6.00 | 1403142.70 | 29791.390 |

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()**?

* Esto se debe a que time.process\_time() retorna un valor que hace referencia a fracciones de segundo de la suma del tiempo del usuario del proceso actual y del CPU del sistema con la comunicación entre Python y el procesador del equipo (kernel/nucleo kernel). Mientras que time.perf\_counter() un valor en float que hace referencia al contador de rendimiento y mide una duración corta.

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

* Con la función start() se da inicio a la toma de datos previo a ejecutar la función del model, mientras que con el stop() se detiene esta toma de datos y se finaliza el proceso.

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

* Al modificar el factor de carga máximo para cargar el catálogo se puede evidenciar que si se usa CHAINING el aumento de su respectivo factor de carga implica mayor tiempo de ejecución. Al ejecutar PROBING se evidencia que también si se aumenta su respectivo factor de carga, este también aumenta 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?

* Para este caso se mantendría constante dependiendo del factor de carga, ya que PROBING mantiene un consumo constante un poco diferente a CHAINING. Esto se debe a que el subset de datos y la cantidad de elementos con los que se crean los maps se mantienen constantes. Para este caso se da un poco mayor en CHAINING.

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 demuestra que el tiempo de ejecución es mayor en PROBING ya que se tiene un tiempo menor que en el resto para el caso de que se use un factor en PROBING de 0.8 en la maquina 1 y 0.5 en la 2. Esto por supuesto teniendo en cuenta la relación de aumento entre los factores de carga y el tiempo de ejecución donde si aumenta el factor, aumenta el tiempo. Esto se ve demostrado en la toma de datos ya que el menor tiempo se obtuvo en PROBING con un factor de carga de 0.3. Por último, se evidencian diferencias entre los tiempos de ejecución de PROBING y CHAINING ya que se muestra una mayor dispersión y notorias diferencias en estos resultados. Siendo CHAINING la que más consume memoria, pero acaba un poco más rápido.

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.

* La verdad se denota una mínima diferencia de consumo para el caso de PROBING y CHAINING. Para la maquina 1 hay una diferencia de 12kB y para la maquina 2 hay una de 13kB. Esto se puede deber a que se escogió un tamaño razonable del hash table, siendo 50 entonces en PROBING había muchos espacios vacíos y esto hace que no se necesite tanta memoria para buscar espacios vacíos, mientras que en CHAINING se pudieron haber hecho más recorridos y usar un poco más de memoria.