**OBSERVACIONES DEL LA PRACTICA**

Javier Cerino Cod 202020873

Marco Zuliani Cod 202022412

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core (TM)i5-1035G1 CPU @ 1.00GHz | AMD Ryzen 3 3300U with Radeon Vega Mobile Gfx 2.10 GHz |
| Memoria RAM (GB) | 8.0 GB (7.76 GB Utilizable) | 12.0 GB (9.92 GB Utilizable) |
| Sistema Operativo | Windows 10 Home Single Language | Windows 10 Home Single Language |

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 | 1057147,475 | 27257,422 |
| 0.50 | 1057134,063 | 27634,441 |
| 0.80 | 1057144,584 | 28776,889 |

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 | 1057157,837 | 26732,481 |
| 4.00 | 1057157,179 | 27409,479 |
| 6.00 | 1057157,873 | 28444,162 |

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 | 1644167,035 | 55618,765 |
| 0.50 | 1644154,410 | 52741,635 |
| 0.80 | 1644167,035 | 54743,266 |

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 | 1644184,152 | 57810,052 |
| 4.00 | 1644183,599 | 61048,581 |
| 6.00 | 1644181,327 | 61901,388 |

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()**?

A decir verdad, aún no tenemos claro por qué en este código se prefiere utilizar **time.perf\_counter()** no obstante, la diferencia entre estos dos procesos radica en el tiempo de toma de datos. En **time.perf\_counter()** se considera todo el tiempo utilizado por el CPU durante la toma de los datos, esto quiere decir que en caso de que el procesador tenga que priorizar un proceso por encima de la ejecución de este código se tomará en cuenta el tiempo en el que este nuevo proceso se desarrolla, en otras palabras, esta función toma en consideración el 100% del tiempo que la CPU utiliza para realizar esta operación sin importarle la cantidad de subprocesos existentes. Por otro lado, la función **time.process\_time()** toma en consideración solo el tiempo de ejecución del proceso de interés. Si se desea tener una toma de datos 100% verídica es recomendable utilizar la función **time.perf\_counter()**.

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

Son importantes el Start y el Stop para poder empezar a obtener la asignación de memoria de Python que empieza en el proceso especifico que se quiere medir. Por otro lado, la función stop indica cuando debe dejar de considerar la asignación de memoria utilizada por el programa para que la medición de memoria del método de interés sea precisa.

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

No percibimos un cambio significativo en el tiempo de ejecución al momento de cambiar el factor de carga. El único cambio que se nota es en los diferentes equipos, pero más allá de eso los tiempos entre prueba y prueba son prácticamente idénticos.

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

No percibimos un cambio significativo en el gasto de memoria de ejecución al momento de cambiar el factor de carga. El único cambio que se nota es en los diferentes equipos, pero más allá de eso los cambios en memoria son similares por decir iguales en cada una de las ejecuciones.

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 la máquina 1 el tiempo de ejecución entre cada uno de los esquemas es prácticamente el mismo con una diferencia de 1 segundo como máximo en el factor más bajo. Sin embargo, este mínimo cambio no lo consideramos algo significativo para dar una respuesta contundente entre la solución más eficiente.

Por otro lado, en la máquina dos hubo un cambio un tanto superior entre los dos tipos de esquema el cambio más drástico se dio en el factor de carga mas alto que fue una diferencia de 4 segundos, elemento que nuevamente no consideramos relevante para concluir cual de los dos esquemas es superior al otro.

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.

En ambos casos percibimos un leve cambio por no decir casi inexistente en el cambio de memoria. El esquema de colisiones “Chaining” mostró un gasto de memoria un poco superior al mostrado en esquema “Probing”. No obstante, consideramos este mínimo cambio irrelevante. Por ende, nuestra conclusión es que no hay una mayor diferencia entre los esquemas.