**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1, Tomas Angel, Codigo: 202020366

Estudiante 2, Daniel Osorio, Codigo: 202022996

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 2,3 GHz Intel Core i7 de cuatro núcleos | AMD Ryzen 5 4600H 3.0 GHz |
| Memoria RAM (GB) | 32 GB | 8 GB |
| Sistema Operativo | macOs Catalina Version 10.15.7 | Windows 10 Home 64 bits |

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 | 615801.425 | 17700.328 |
| 0.50 | 615801.425 | 17489.276 |
| 0.80 | 615801.425 | 18044.236 |

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 | 615821.243 | 18932.420 |
| 4.00 | 615821.243 | 19604.988 |
| 6.00 | 615821.180 | 19057.513 |

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

Gráfico

Descripción generada automáticamente

# **Maquina 2**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga** (PROBING) | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 0.30 | 615947.89 | 21289.24 |
| 0.50 | 615971.16 | 21522.48 |
| 0.80 | 615992.94 | 21933.93 |

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 | 616023.45 | 22825.72 |
| 4.00 | 616023.43 | 22893.71 |
| 6.00 | 616023.65 | 22797.82 |

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

Gráfico, Gráfico de líneas

Descripción generada automáticamente

# **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 este caso, se utiliza la funcion de *perf\_counter()* porque esta funcion se utiliza para realizar

pruebas de rendimiento con un reloj con mayor resolucion para hacer mediciones a corto plazo

con mayor presicion. Por el otro lado, la funcion *time.process\_time()* accede al tiempo del

procesador y al tiempo del sistema simultaneamente, haciendo un estimado de ambos tiempos.

Al comparar una funcion con la otra, es mejor usar la funcion *time.perf\_counter()* en este caso ya

que nos ayuda a tomar muestras con mas presicion y saber si los algoritmos se estan ejecutando

de la manera mas eficiente.

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

*Tracemalloc* es una herramienta que ayuda a rastrear la asignacion de memoria por Python, por lo que las funciones de *start()*  y *stop()* son importantes porque hacen que el proceso para rastrer el uso de memoria sea mas preciso siempre y cuando estas mismas se pongan al inicio y al final de las funciones en el codigo. La funcion *start()* deberia estar al inicio de la funcion para que se empiece a rastrear el uso de memoria lo antes posible, y la funcion *stop()* para el rastreo de memoria, dando un valor exacto de la cantidad de memoria asignada por python cuando el codigo fue ejecutado.

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 maximo para cargar el catalogo de videos, se puede evidenciar que el tiempo de ejecucion va aumentando del mismo modo en el que el factor de carga aumenta, pero sin embargo, la maquina uno demuestra ser mas rapida cuando carga la informacion del catalogo con respecto a la maquina dos, pero en ambos casos, el tipo de mapa linear probing demuestra ser un tipo de mapa mucho mas veloz que el tipo de mapa chaining.

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

Al modificar el factor de carga maximo para cargar el catalogo de videos, se puede evidenciar que la maquina uno no tuvo cambios significativos en el consumo de memoria, puesto que en todos los casos, el consumo fue casi igual para cada uno de los tipos de mapas con los que se realizaron las pruebas. El unico cambio evidente, es al cargar los datos con el map chaining pero aun asi este demuestra ser muy pequeño en las pruebas realizadas en ambas maquinas. Sin embargo, al cargar los datos con un tipo de mapa linear probing, se puede evidenciar que la maquina dos si tiene una diferencia grande en cada uno de los factores de carga en el consumo de memoria. A medida que va aumentando el factor de carga en este tipo de mapa, el consumo de memoria tambien va a aumentando.

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.

Al modificar el esquema de colisiones, se puede evidenciar que el tiempo de ejecucion es muy diferente entre uno y el otro. Cargar los datos con un tipo de mapa linear probing, hace que los datos carguen de manera mas rapida a diferencia de un mapa de tipo 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.

Es evidente que el consumo de memoria al modificar el esquema de colisiones es mucho mayor cuando se cambia el esquema de colisiones. Ademas de que aumente el tiempo, la cantidad de memoria utilizada tambien presenta una diferencia significativa cuando se comparan ambos esquemas.