**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1: Santiago Díaz Moreno Cod 201912247

Estudiante 2: Juana Mejía B Cod 202021512

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-de 11va generation | 2,4 GHz Quad-Core Intel Core i5 |
| Memoria RAM (GB) | 4GB | 8 GB |
| Sistema Operativo | Windows 10 Pro 64-bits | MacOS Big Sur (Version 11.1) |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

Nota: Me toco hacer la medicion con el archivo que contiene el 80% de los datos porque mi computador moria cuando intentaba cargar el archivo completo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 0.30 | 978299.748 | 141559.300 |
| 0.50 | 953846.248 | 261456.841 |
| 0.80 | 935050.658 | 154687.602 |

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 | 968229.040 | 119374.540 |
| 4.00 | 958203.989 | 178306.631 |
| 6.00 | 956706.829 | 128604.893 |

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 | 1744408,70 | 57723,35 |
| 0,50 | 1656457,30 | 57977,56 |
| 0,80 | 1588050,37 | 71508,86 |

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 | 1611998,63 | 51762,41 |
| 4,00 | 1593555,66 | 52687,10 |
| 6,00 | 1590799,15 | 58832,54 |

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 pref\_counter() incluye el tiempo en el que el programa entra en sleep mientras se ejecuta mientras que process\_time() no, esto nos ayuda a tener una medida mas precisa del tiempo que queremos calcular.

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

Estos son importsntes debido a que para poder saber cuanto tiempo se recorre se debe medir lo transcurrido entre el inicio de la funcion y el final de esta, por medio de estas funciones se encuentra el tiempo de inicio (start) y el del final (stop) y se puede hacer la resta stop- start.

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

Al momento de modificar el factor de carga maximo se pudo ver un incremento en el tiempo de carga del catalogo.

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

Contrario a lo que ocurre cuando se incrementa el factor de carga maximo el consumo de memoria se vuelve menor.

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.

Cuando se cambia el esquema de colisiones existe un cambio en el tiempo de ejecucion, con CHAINING los tiempos son menores, asi que en treminos de tiempo es mucho mas favorable utilizar este esquema de colisiones. Esto es debido a que en Linear probing se debe recorrer cada posicion del hash map para poder encontrar uno que este disponible y meter la informacion, mientras que en Chaining añade en la posicion que se arroja con la funcion de hash ()

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.

Cuando hay un cambio en el esquema de colisiones el consumo de memoria es menor en CHAINING. Esto se debe a que en Linear probing los espacios de la memoria son lineares, mientras que en Chaning los espacios utilizados para guardar los datos son los que se encuentren disponibles y no linearmente, lo cual gasta menor memoria.