**OBSERVACIONES DEL LA PRACTICA**

Juan José Osorio (202017020)

Thais Tamaio Ramírez (202022213)

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 3,1 GHz Intel Core i5 de dos núcleos | Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz 2.50 GHz |
| Memoria RAM (GB) | 8 GB 2133 MHz LPDDR3 | 4.00 GB (3.90 GB usable) |
| Sistema Operativo | macOS Big Sur versión: 11.2.1 | 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 | 550840,064 | 42517,477 |
| 0.50 | 550840,064 | 43451,868 |
| 0.80 | 550197,729 | 43483,396 |

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 | 550210,799 | 42479,668 |
| 4.00 | 550208,291 | 42329,905 |
| 6.00 | 550855,932 | 42847,353 |

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 | 541049,338 | 64624,054 |
| 0.50 | 541049,338 | 62826,344 |
| 0.80 | 541049,272 | 63101,728 |

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 | 541064,705 | 67797,867 |
| 4.00 | 541064,650 | 63628,854 |
| 6.00 | 541064,646 | 63037,975 |

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()**?

En la función getTime() se utiliza time.perf\_counter() ya que la función time.process\_time() retorna un valor de la suma del tiempo de CPU del sistema y del usuario del proceso actual, por lo que el resultado de esta funnción puede llegar a depender del Hardware de la máquina en la que se ejecute el programa. Por otro lado, se utiliza time.perf\_counter() debido a que esta función utiliza un contador de rendimiento, el cual tiene una resolución que es más alta que la resolución de time.process\_time(). Esta resolución se refiere a el número de tics por unidad de tiempo, por lo que time.perf\_counter() llega a ser más preciso y también, es independiente de la máquina en la que se ejecute el programa.

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

Las funciones start() y stop() de la librería son esenciales para calcular la memoria utilizada a lo largo de la ejecución de una función del programa. Primero, la función start() inicializa el proceso para medir la cantidad de memoria utilizada durante la ejecución de un proceso. Por otro lado, la función stop() finaliza la medición de la memoria. Es por medio de estas dos funciones que se puede obtener la diferencia entre la memoria final y la inicial (es decir, la memoria total utilizada) durante el proceso escogido, desde el momento exacto en el que la función empieza a correr, hasta que este finaliza por completo su ejecución.

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 de videos no se evidenciaron cambios significativos en los tiempos de ejecución. Esto significa que el tiempo de ejecución siempre tiene un valor semejante sin importar el factor de carga y el tipo de mapa/tabla de hash.

Por ejemplo, para la máquina 1 se puede apreciar que al utilizar un factor de carga de 0.3 con un mapa de tipo probing se obtuvo un tiempo de ejecución de 42517 milisegundos. Similarmente, al utilizar un factor de carga de 0.8 con este mismo tipo de mapa se obtuvo un tiempo 43483 milisegundos. Finalmente, con un mapa de tipo chaining con un factor de mapa de 6 se obtuvo un tiempo de ejecución de 42847 milisegundos. Por lo tanto, se puede afirmar que el tiempo de ejecución siempre será muy similar, sin importar el factor de carga.

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 máximo para cargar el catálogo de videos no se evidenciaron cambios significativos enel consumo de memoria. Esto significa que el consumo de memoria siempre tiene un valor semejante sin importar el factor de carga y el tipo de mapa/tabla de hash. Por ejemplo, para la máquina 1 se puede apreciar que al utilizar un factor de carga de 0.3 con un mapa de tipo probing se consumieron 550850 kilobytes. Similarmente, al utilizar un factor de carga de 0.5 con este mismo tipo de mapa se se consumieron 550850 kilobytes. Finalmente, con un mapa de tipo chaining con un factor de mapa de 6 se se consumieron 550855 kilobytes. Por lo tanto, se puede afirmar que el consumo de memoria siempre será muy similar, sin importar el factor de carga.

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 para cargar el catálogo de videos no se evidenciaron cambios significativos en los tiempos de ejecución. Esto significa que el tiempo de ejecución siempre tiene un valor semejante sin importar el factor de carga y el esquema de colisiones (Probing o Chaining). Por ejemplo, para la máquina 2 se puede apreciar que al utilizar un factor de carga de 0.3 con un mapa de tipo probing se obtuvo un tiempo de ejecución de 64524 milisegundos. Similarmente, al utilizar un factor de carga de 0.8 con este mismo tipo de mapa se obtuvo un tiempo 63101 milisegundos. Finalmente, con un mapa de tipo chaining con un factor de mapa de 6 se obtuvo un tiempo de ejecución de 63037 milisegundos. Por lo tanto, se puede afirmar que, al cargar el catálogo de videos, el tiempo de ejecución siempre será muy similar, sin importar el esquema de colisiones.

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.

Al modificar el consumo de memoria para cargar el catálogo de videos no se evidenciaron cambios significativos en el consumo de memoria. Esto significa que el consumo de memoria siempre tiene un valor semejante sin importar el factor de carga y el esquema de colisiones (Probing o Chaining). Por ejemplo, para la máquina 2 se puede apreciar que al utilizar un factor de carga de 0.3 con un mapa de tipo probing se consumieron 541049 kilobytes. Similarmente, al utilizar un factor de carga de 0.8 con este mismo tipo de mapa se consumieron 541049 kilobytes. Finalmente, con un mapa de tipo chaining con un factor de mapa de 6 se consumieron 541064 kilobytes. Por lo tanto, se puede afirmar que, al cargar el catálogo de videos, el consumo de memoria siempre será muy similar, sin importar el esquema de colisiones.

**Gráfica 1**

**Gráfica 2**