**OBSERVACIONES DEL LA PRACTICA**

Santiago Pardo Morales - 202013025

Juan Jose Ramirez- 202013228

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 2,4 GHz Intel Core i5 de cuatro núcleos | Intel Core i3-8145U @ 2.30 GHz |
| Memoria RAM (GB) | 8 GB | 4 GB |
| Sistema Operativo | MacOS | Windows |

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 | 1327329.413 | 45145.759 |
| 0.50 | 1327329.437 | 48539.574 |
| 0.80 | 1327329.437 | 44607.850 |

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.

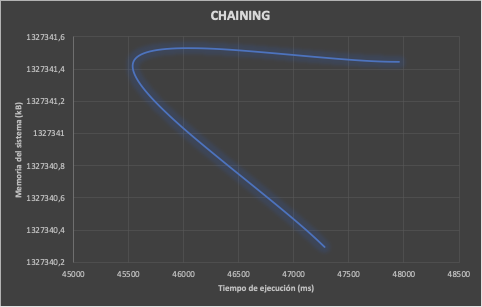
|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 1327340.295 | 47278.075 |
| 4.00 | 1327341.443 | 45541.419 |
| 6.00 | 1327341.443 | 47953.782 |

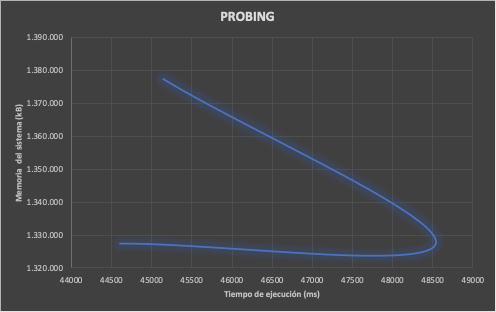
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

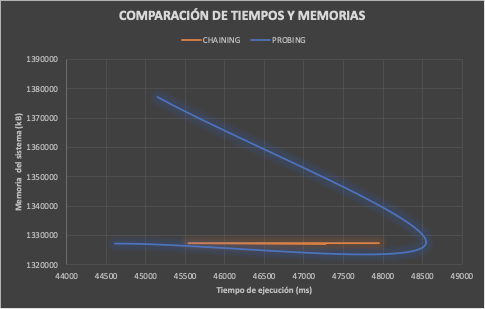
## **Graficas**

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 | 488157.020 | 26693.005 |
| 0.50 | 488156.949 | 51192.848 |
| 0.80 | 488156.926 | 44448.603 |

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 | 488156.926 | 35399.652 |
| 4.00 | 488152.863 | 32583.040 |
| 6.00 | 488152.863 | 38839.831 |

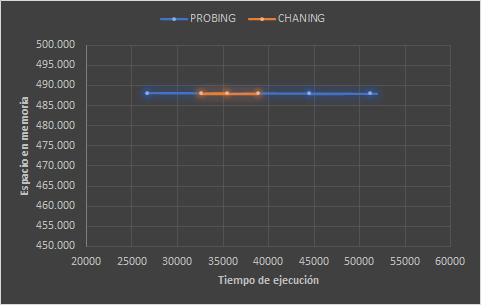
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

## **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

Maquina 2



# **Preguntas de análisis**

1. ¿Por qué en la función **getTime()** se utiliza **time.perf\_counter()** en ves de la previamente conocida **time.process\_time()**?

La función perf\_counter de la librería time de Python calcula con más precisión el tiempo transcurrido, debido a que utiliza un “performance counter”, por lo que no requiere de ningún cálculo matemático para obtener el valor, a comparación de la función process\_time, la cual también retorna el tiempo transcurrido, pero para obtenerlo tiene que hacer manualmente cálculos como la suma del sistema y el CPU del usuario durante el proceso. La escogencia de esta función podría mejorar la complejidad de los sistemas debido a que se retorna el tiempo de una manera más rápida y fácil que con la función process\_time.

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

La función start permite inicializar el conteo del tiempo, para poder obtener el valor del tiempo inicial, mientras que la función stop permite detener el conteo, para obtener el tiempo final y posteriormente para poder calcular la diferencia de tiempo.

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

El tiempo de ejecución tiene una especie de comportamiento cuadrático con respecto al factor de carga, debido a que en un factor de carga determinado, el tiempo de ejecución alcanzará su valor máximo o su valor mínimo (dependiendo del esquema de colisiones), y para los otros factores de carga podría ser aproximadamente el mismo valor, sin importar que sea un factor de carga menor o mayor al factor en el que se alcanzó el tiempo de ejecución máximo o mínimo.

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

Como se puede evidencia en los datos registrados en el caso del esquema probing entre mayor sea el factor de carga menos memoria consume esto debido que la lista creada tendra un numero menor de posiciones, para el caso del esquema chaning entre sucede de la misma manera ya que en una sola posicion se podran guardar varias llaves.

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 utiliza el sistema de coliosenes probing con un factor de carga menor o igual a 0.3 aunque aunmenta un poco la memoria usada el tiempo de carga es bastante menor comparado al esquea de colision chaning el cual entre mayor sea el factor de carga menor memoria consume pero su tiempo de carga se mantiene estable.

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.

Basado en los resultados no podríamos concluir que el sistema de colisiones influye en el consumo de memoria, ya que por parte de la máquina 1 se observó un aumento al cambiar de probing a chaining, mientras que haciendo este mismo procedimiento en la máquina 2 podemos observar que la memoria de almacenamiento disminuye. Además de lo visto anteriormente, también podemos ver que el cambio entre los consumos de memoria es pequeño, por lo que esto puede depender de otros motivos como el factor de carga.