Análisis de la velocidad de memoria en diferentes dispositivos





Universidad Industrial de Santander

Resumen

Este artículo presenta un programa desarrollado en C utilizando MS Visual Studio para evaluar el comportamiento de sistemas de memoria. El programa se enfoca en lograr una sincronización precisa y luego realizar accesos a memoria que abarcan diferentes niveles de jerarquía de caché. La metodología incluye la medición precisa del tiempo de CPU utilizando utilidades estándar, seguido de bucles anidados para lectura y escritura en memoria con diferentes tamaños y pasos de caché. Los resultados se calculan en términos de tiempo de acceso y se generan en formato .csv para su análisis en hojas de cálculo.

Introducción

En el libro "Computer Architecture: A Quantitative Approach" de Patterson y Hennessy, el Caso de Estudio 2, "Poniendo todo junto: sistemas de memoria altamente paralelos", analiza cómo evaluar el comportamiento del sistema de memoria mediante un programa en C. Este programa, diseñado para MS Visual Studio, se enfoca en:

- 1. Sincronización Precisa: Usa una utilidad para medir con exactitud el tiempo de CPU.
- 2. Bucle Anidado para Accesos a Memoria: Realiza lecturas y escrituras en memoria variando tamaños de pasos y cachés para evaluar el rendimiento.
- 3. Cálculo de Tiempos de Sobrecarga: Calcula los tiempos reales de acceso a la memoria excluyendo la sobrecarga del bucle.

Objetivo

Este programa permite a estudiantes y profesionales obtener una visión detallada de cómo las configuraciones de memoria afectan el rendimiento, proporcionando datos valiosos para optimizar tanto hardware como software, logrando sistemas más eficientes y de alto rendimiento.

Código 1

El programa mide el rendimiento de la memoria accediendo a un array de diferentes tamaños en C. Aquí están los pasos clave:

Inicialización:

Define el tamaño del array y las cachés a evaluar.

Función de Tiempo:

Implementa una función para obtener el tiempo actual del sistema en segundos.

Generación de Etiquetas:

Crea etiquetas para diferentes tamaños de memoria.

Bucle Principal:

Evalúa el rendimiento de la memoria para distintos tamaños de caché y pasos (stride). Accede secuencialmente a la memoria para simular cargas de trabajo reales.

Medición de Tiempos:

Mide el tiempo de acceso a la memoria durante 20 segundos. Calcula la sobrecarga del bucle vacío y ajusta el tiempo total.

Cálculo del Rendimiento:

Calcula el tiempo de carga en nanosegundos por acceso a memoria y lo imprime en formato CSV.

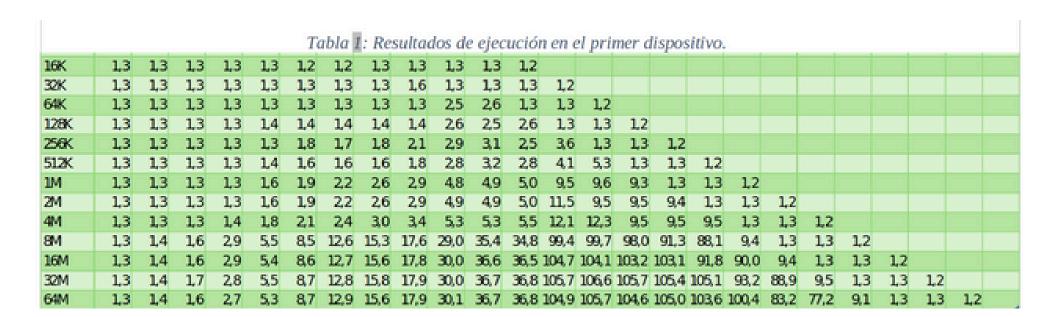
Este programa ayuda a entender cómo diferentes configuraciones de caché afectan el rendimiento de la memoria.

Resultados de Memoria en un Sistema con 16 GB de RAM y Ryzen 3 3200g

El código mide tiempos de procesamiento en nanosegundos para datos desde 4 bits hasta 64 MB, con cachés de 4 KB a 64 MB. Los resultados en CSV se presentan en la Tabla 1.

Observaciones Clave

- **Campos Vacíos:** Ocurren cuando la memoria es insuficiente para ciertos tamaños de datos.
- **Principio de Localidad:** Los tiempos de ejecución son más rápidos con cachés más cercanos al procesador.



Campos Vacíos: Ocurren cuando la memoria es insuficiente para ciertos tamaños de datos.

Gráfica de Tiempo vs. Información: Muestra que los tiempos de ejecución son más rápidos con cachés más cercanos al procesador, siguiendo el principio de localidad.

Ilustración 1: Grafico de la Tabla 1.

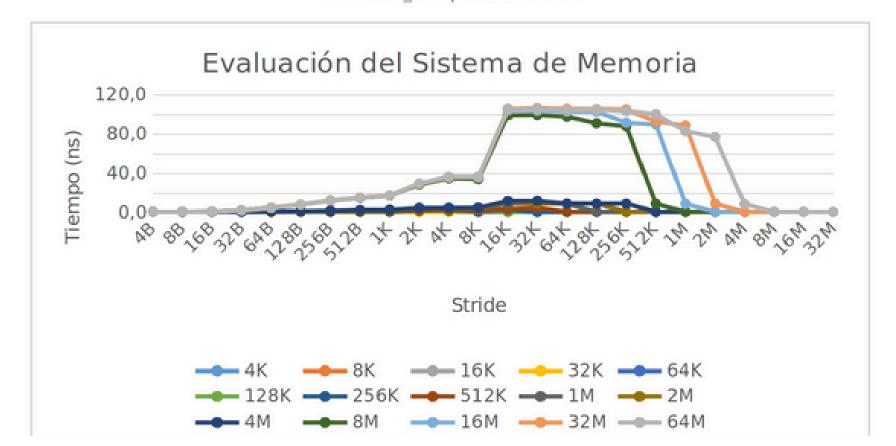
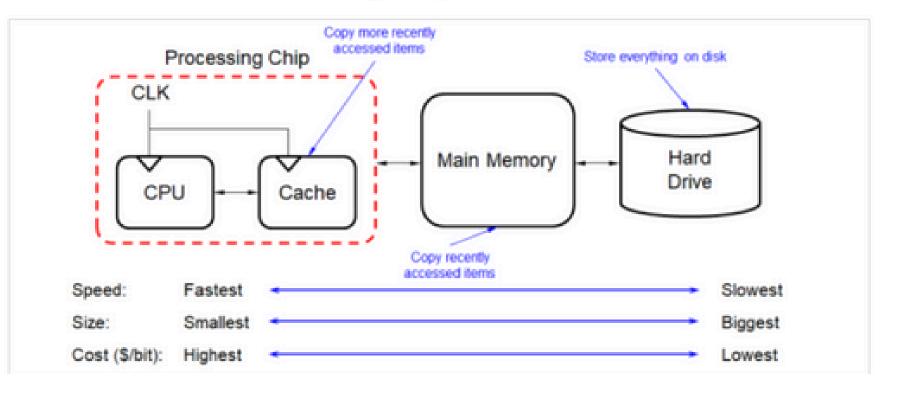


Ilustración ᠌: Principio de localidad.[1]



Campos Vacíos: Ocurren cuando la memoria es insuficiente para ciertos tamaños de datos.

Gráfica de Tiempo vs. Información: Muestra que los tiempos de ejecución son más rápidos con cachés más cercanos al procesador, siguiendo el principio de localidad. **Rendimiento del Procesador:** El manejo de datos en caché es rápido. El mayor tiempo registrado fue de 106.6 ns al procesar 32 KB con una memoria de 32 MB.

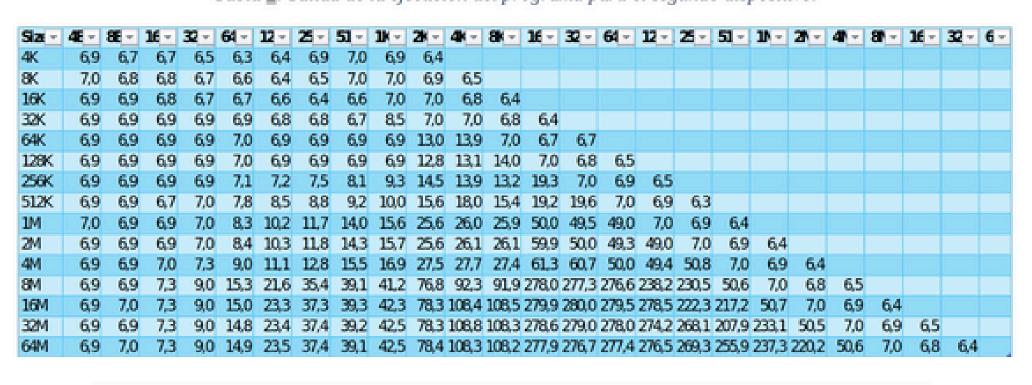
SCAN PROPERTY OF THE PROPERTY

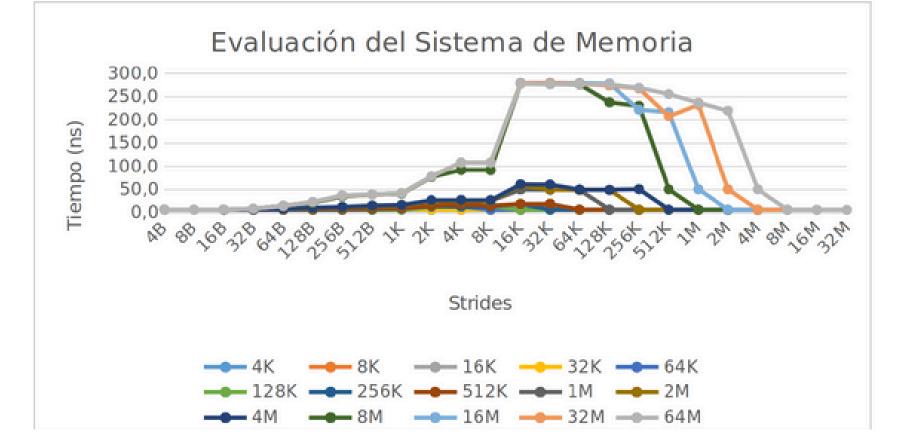
Resumen de Resultados de Memoria en un Sistema con 8 GB de RAM y AMD 3020e

Resultados en Tabla 2: El código se ejecutó en un dispositivo AMD 3020e con componentes más antiguos.

Tiempo Máximo: El mayor tiempo registrado fue de 280 ns al procesar 32 KB con una memoria de 16 MB, más del doble que en el Ryzen 3 3200g.

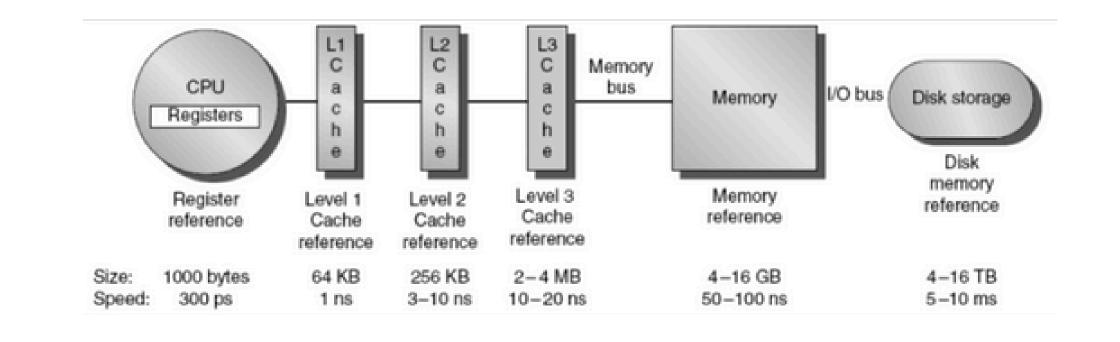
Tabla 2: Salida de la ejecución del programa para el segundo dispositivo.





Intervalo Crítico: El procesamiento de datos entre 16 KB y 2 MB muestra los mayores retardos.

Jerarquía de Memoria: Los resultados reflejan la influencia significativa de la jerarquía de memoria (ver Ilustración 4).



Observaciones sobre el Comportamiento de las Gráficas

Demora en Nivel 3 de Caché: Se observa una notable demora en el nivel 3 de caché.

Impacto en Velocidad: Dispositivos sin nivel 3 de caché envían información directamente a la RAM, resultando en menor velocidad.

Eficiencia con Grandes Cantidades de Información: A pesar de lo anterior, el procesamiento de grandes cantidades de información muestra tiempos aceptables, indicando una eficiencia razonable en estos casos.

Resumen de Características de Caché y Memoria Principal

Tamaño del Bloque de Caché de Segundo Nivel (L2)

Ryzen 3 3200g y AMD 3020e: Ambos procesadores muestran un aumento significativo en los tiempos de acceso a memoria cuando el tamaño del bloque supera los 64 KB, sugiriendo que el tamaño del bloque de caché L2 es aproximadamente 64 KB.

Significado

Caché L2 de 64 KB: La caché de segundo nivel puede almacenar hasta 64 KB de datos eficientemente. Acceder a datos dentro de este tamaño es rápido, mientras que acceder a datos mayores implica tiempos más largos al acceder a la memoria principal.

Tamaño de la Memoria Principal

Ryzen 3 3200g: Tiene 16 GB de RAM DDR4

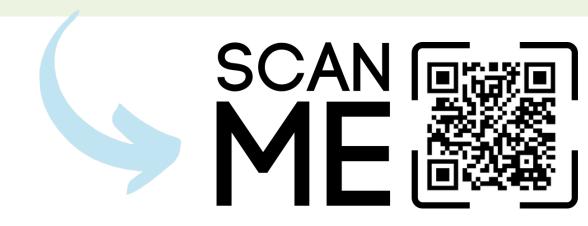
AMD 3020e: Tiene 8 GB de RAM.

Modificaciones al Código para Optimizar la Caché de Instrucciones:

- **Separación de Funciones:** Funciones separadas para medir tiempos de carga de datos e instrucciones.
- Modificación del Bucle Principal: Bucle principal ajustado para medir el rendimiento de la caché de instrucciones.
- **Medición de Tiempos de Instrucciones:** Evaluación del rendimiento de bloques de instrucciones para diferentes tamaños de caché.

Importancia de Conocer las Características de la Caché:

- Estas modificaciones permiten entender mejor el comportamiento y optimizar el código según las características específicas del hardware.
- Es esencial para mejorar el rendimiento y la eficiencia en el diseño de sistemas informáticos.



Conclusiones

Impacto del Tamaño de Caché y Stride en el Rendimiento:

- En el Ryzen 3 3200g con 16 GB de RAM DDR4, los tiempos de acceso son bajos hasta cierto tamaño de caché, a partir del cual aumentan significativamente.
- En el AMD 3020e con 8 GB de RAM, se observa un comportamiento similar pero con tiempos de acceso más altos, reflejando diferencias en la arquitectura y capacidad de los sistemas.

Optimización del Hardware y Software:

- Los resultados ayudan a identificar configuraciones de memoria óptimas, guiando decisiones para estructurar niveles de caché.
- Medir tiempos precisos de acceso a la caché es crucial para optimizar hardware y software, mejorando la eficiencia y el rendimiento del sistema.

Importancia en la Relación entre Caché y Rendimiento:

- Comparar resultados destaca la importancia de ajustar características de la caché de instrucciones para optimizar el rendimiento.
- Este estudio es esencial para ingenieros de sistemas y computación, facilitando decisiones informadas en el diseño y optimización de arquitecturas computacionales eficientes.

Referencias

[1] Barrios, C. J. (s/f). Computer Architecture A Quantitative Approach, Fifth Edition. http://wiki.sc3.uis.edu.co. Recuperado el 22 di junio de 2024, de http://wiki.sc3.uis.edu.co/images/d/d4/MemoryH.pd Pagina 4.

[2] Barrios, C. J. (s/f). Computer Architecture A Quantitative Approach, Fifth Edition. http://wiki.sc3.uis.edu.co. Recuperado el 22 de junio de 2024, de http://wiki.sc3.uis.edu.co/images/d/d4/MemoryH.pdf. Pagina 10.

Pagina 10.
[3] Casos de Estudio y Ejercicios por Norma P. Jouppi, Naveen Muralimanohar, y Sheng LI en el libro Patterson and Hennesy, Computer Architecture; A Quantitative Approach. Página 133 de la Quinta Edición

Autores

Diego Alejandro García Barajas Manuela Alejandra García Valbuena Santiago Gomez Villarreal Cristian Manuel Hernández Delgado Jefferson Holguin Ferro