

Estructura de Computadores

Práctica 6: Caché

Gustavo Romero López

Updated: 12 de diciembre de 2022

Arquitectura y Tecnología de Computadores

1. Índice
2. Objetivos
3. Tamaño de línea
4. Tamaño de caché
5. Enlaces

Objetivos

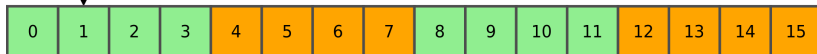
- ⊙ Comprender la importancia de la memoria caché mediante el estudio de la misma.
- ⊙ Nos centraremos en dos de sus parámetros más importantes:
 - Tamaño de línea o **bloque**: mínima unidad de transferencia.
 - **Tamaño** de la caché: tamaño de cada nivel.
- ⊙ Intentaremos calcularlos para nuestro procesador.
- ⊙ En Linux podemos consultar todos los parámetros de la caché mediante la orden `lscpu` o examinando el directorio `/sys/devices/system/cpu/cpu0/cache`.
- ⊙ “`make info`” muestra por pantalla información relevante sobre la caché extraída del directorio anterior.

Tamaño de línea

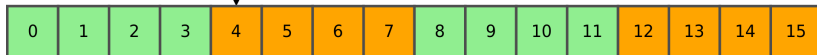
primer acceso: supongamos fallo... trae 0, 1, 2 y 3



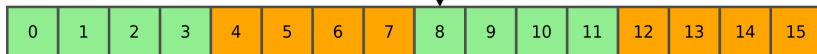
segundo acceso: acierto, 1 está dentro de la misma línea que 0



quinto acceso: fallo, 4 pertenece a otra línea... trae 4, 5, 6 y 7



noveno acceso: fallo



Tamaño de línea

- ⦿ Una línea o bloque de caché es la **cantidad de información** que viaja entre los niveles de caché y la memoria principal.
- ⦿ Es tan importante que a veces **prevalece** el tiempo de acceso a los datos frente al del tratamiento de los mismos.
- ⦿ ¿Cuánto debería tardar cada bucle?

```
char bytes[1 << 24]; // 16MB
```

```
char c = 0;
```

```
for (unsigned i = 0; i < sizeof(bytes); i += 1)  
    c ^= bytes[i];
```

```
for (unsigned i = 0; i < sizeof(bytes); i += 2)  
    c ^= bytes[i];
```

```
for (unsigned i = 0; i < sizeof(bytes); i += 4)  
    c ^= bytes[i];
```

Indicaciones para calcular el tamaño de línea

- ⊙ Mida el tamaño de línea **generalizando** el proceso anterior para cada posible tamaño de línea.
- ⊙ Cuánto más **ligero** sea el bucle interno mejor se evidenciará la diferencia de tiempo entre la operación realizada y el tiempo de acceso a memoria.
- ⊙ Mida tiempos y compare... ¿Obtiene los resultados esperados?
- ⊙ El resultado en mi ordenador puede verse en la figura 1.
- ⊙ En vez de partir de cero, complete el esqueleto: line.cc.
- ⊙ makefile genera un gráfico de forma automática.
- ⊙ Puede ayudarse con las ordenes “lscpu” y “make info”.

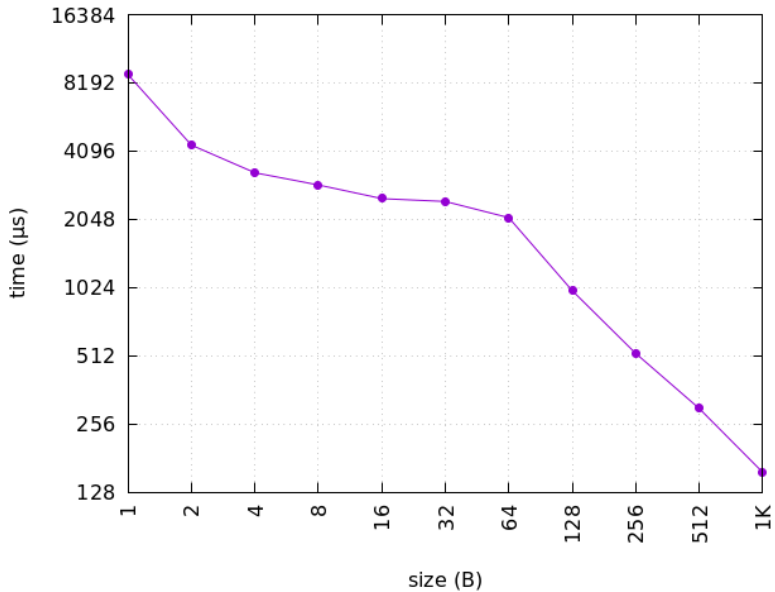


Figura 1: Tamaño de línea.

Tamaño de caché

- ⊙ Mejor cuanto más grande si no fuese por precio, calor, superficie, consumo,...
- ⊙ Para medir el tamaño de caché debemos:
 - Para cada tamaño de caché
 - Crear un vector de dicho tamaño
 - Repetir muchas veces
 - Leer un elemento de cada línea de caché
- ⊙ Cuánto más **ligero** sea el bucle mejor se evidenciará la diferencia de tiempo entre cálculo y acceso a memoria.
- ⊙ Medir tiempos y comparar.
- ⊙ El resultado en mi ordenador puede verse en la figura 2.
- ⊙ En vez de partir de cero, complete el esqueleto: `size.cc`.
- ⊙ `makefile` genera un gráfico de forma automática.
- ⊙ ¿Cuántos niveles tiene tu caché y de qué tamaño son?

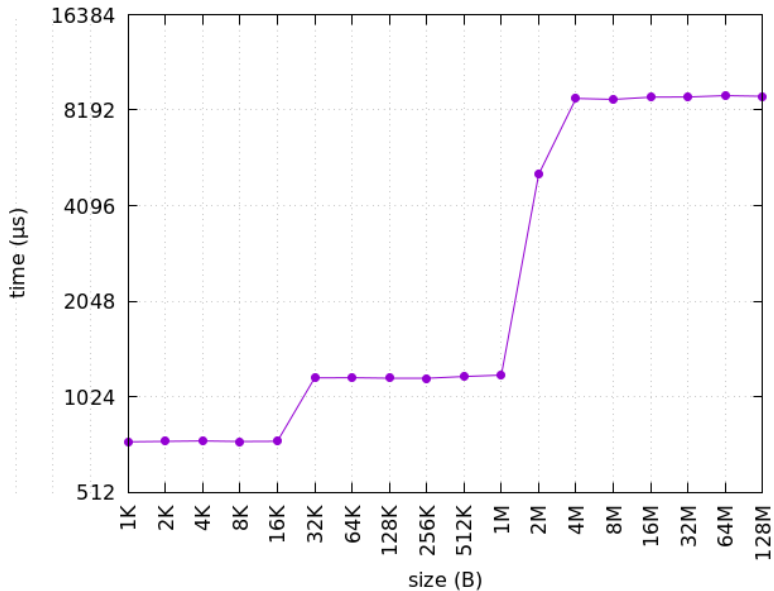


Figura 2: Tamaño de cache.

```
[gustavo@casa 6s]$ lscpu
Arquitectura:                x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:          Little Endian
Tamaños de las direcciones:   48 bits physical, 48 bits virtual
CPU(s):                       4
Lista de la(s) CPU(s) en línea: 0-3
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»:      2
«Socket(s)»:                  1
Modo(s) NUMA:                 1
ID de fabricante:             AuthenticAMD
Familia de CPU:                21
Modelo:                        19
Nombre del modelo:             AMD A10-6800K APU with Radeon(tm) HD Graph
Revisión:                      1
CPU MHz:                       1995.692
CPU MHz máx.:                  4100,0000
CPU MHz mín.:                  2000,0000
BogoMIPS:                       8184.86
Virtualización:                 AMD-V
Caché L1d:                      32 KiB
Caché L1i:                      128 KiB
Caché L2:                        4 MiB
CPU(s) del nodo NUMA 0:        0-3
```

make info

```
[gustavo@casa 6s]$ make info
```

level	type	line	size
1	Data	64	16K
1	Instruction	64	64K
2	Unified	64	2048K

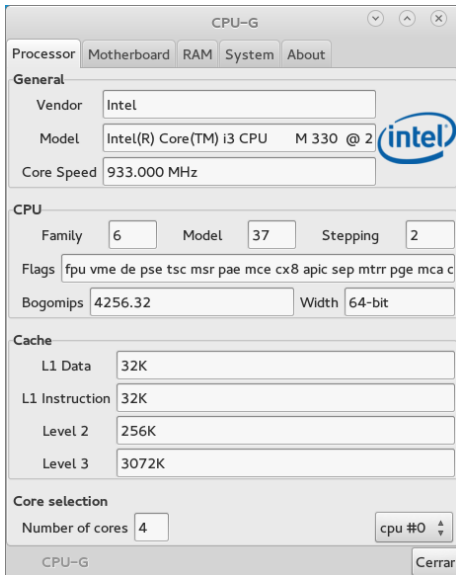


Figura 3: La CPU de mi portatil vista con CPUG

Enlaces de interés

- ⦿ https://en.wikipedia.org/wiki/CPU_cache
- ⦿ <http://igoro.com/archive/gallery-of-processor-cache-effects/>
- ⦿ <http://cpug.sourceforge.net/>