

Práctica 5: Caché

Estructura de Computadores

Gustavo Romero López

Updated: 18 de diciembre de 2017

Arquitectura y Tecnología de Computadores

1. Índice
2. Objetivos
3. Tamaño de linea
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**.
 - **Tamaño de la caché**.
- ⊙ Intentaremos calcularlos para el procesador que utilizamos.
- ⊙ 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`” extraerá la información más importante de la caché del directorio `/sys/devices/system/cpu/cpu0/cache` y la mostrará por pantalla.

Tamaño de línea

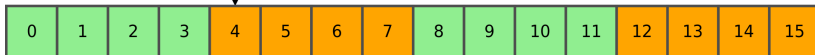
primer acceso: supongamos fallo... trae 0, 1, 2 y 3



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



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.
- ⦿ Para ello mida cuánto tardan en ejecutarse los siguientes bucles con diferentes **incrementos**:

```
std::array<char, 1 << 24> bytes; // 16MB
```

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

```
for (unsigned i = 0; i < bytes.size(); i += 2)  
    bytes[i] ^= 1;
```

```
for (unsigned i = 0; i < bytes.size(); i += 4)  
    bytes[i] ^= 1;
```

- ⦿ ¿Cuánto debería tardar cada bucle?

Indicaciones para calcular el tamaño de linea

- ⊙ Para medir el tamaño de linea se **generaliza** proceso anterior para los tamaños de linea posibles.
- ⊙ **bf** Introducir el bucle anterior dentro de otro que recorra todos los tamaños de línea posibles.
- ⊙ 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.
- ⊙ Razone que tamaño de linea utiliza su procesador.
- ⊙ Puede ayudarse con las ordenes `"lscpu"` y `"make info"`.

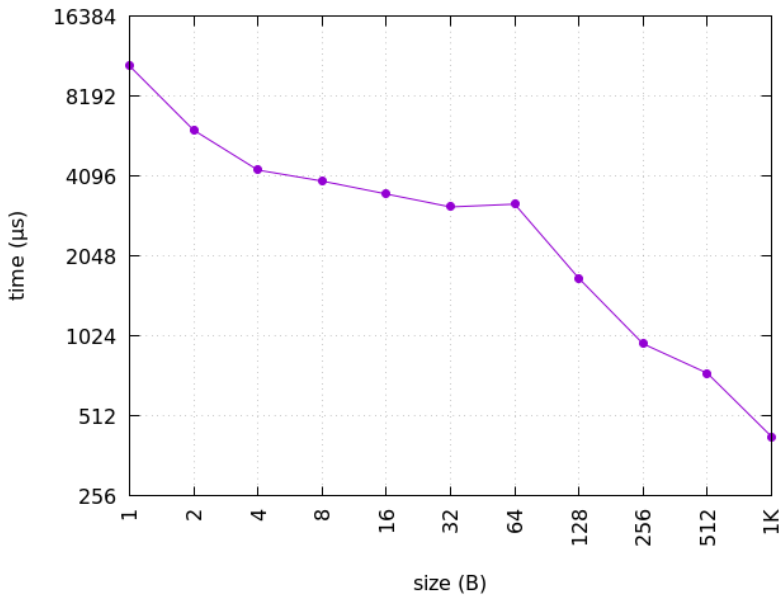


Figura: Tamaño de línea

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 1000000 veces.
Realizar una pequeña alteración en cada linea.
- ⊙ 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.
- ⊙ ¿Cómo es tu caché (niveles/tamaño)?

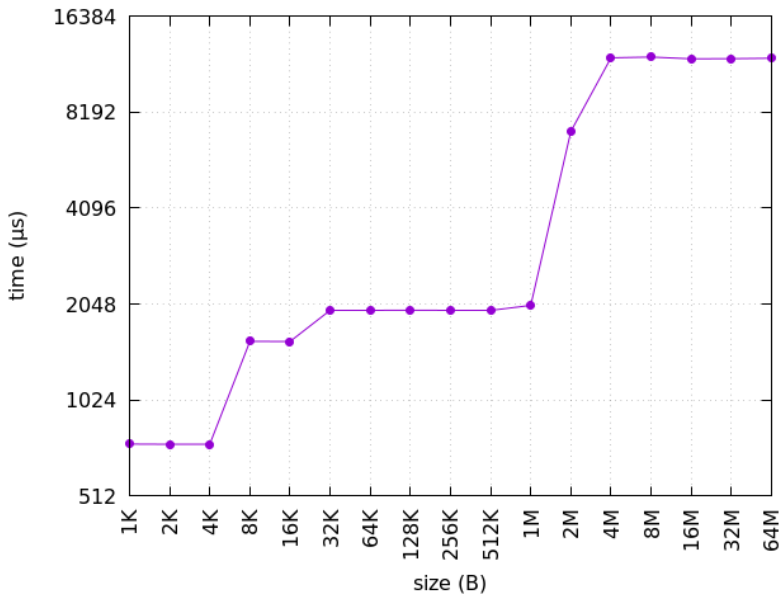


Figura: Tamaño de cache.

lscpu

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):              1
NUMA node(s):          1
Vendor ID:              AuthenticAMD
CPU family:             21
Model:                  48
Model name:             AMD A10-7700K APU with Radeon(TM) R7 Graphics
Stepping:               1
CPU MHz:                2000.000
CPU max MHz:            3400,0000
CPU min MHz:            2000,0000
BogoMIPS:               6787.77
Virtualización:         AMD-V
L1d cache:              16K
L1i cache:              96K
L2 cache:               2048K
NUMA node0 CPU(s):     0-3
```

make info

```
[gustavo@pccito 5]$ make info  
line size = 64B  
cache size = 16K/96K/2048K/  
cache level = 1/1/2/  
cache type = Data/Instruction/Unified/
```

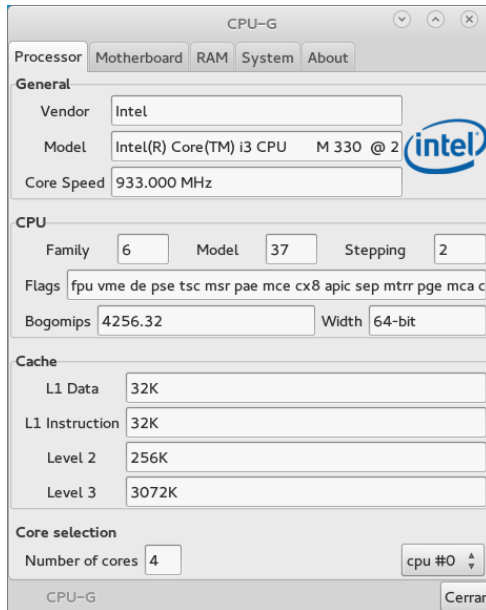


Figura: La CPU de mi portatil vista con CPUG

Enlaces de interés

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