

Organización de Computadores Prácticas

Rendimiento I

Felix.Villanueva@uclm.es
ManuelJose.Abaldea@uclm.es
Santiago.GTalegon@uclm.es

Contenido

- **Monitorización del rendimiento en tiempo de ejecución:**
 - **Herramienta perf**
 - **Análisis de fallos de cache**
 - **Caso I: estudio del tamaño de línea**
 - **Caso II: recorrido en estructuras 2D**

Performance Events for Linux: herramienta estándar en Linux para análisis de rendimiento (profiling).

- Perf accede a los contadores hardware de rendimiento.
 - Son registros de la CPU que cuentan eventos como instrucciones ejecutadas, fallos de caché o bifurcaciones mal predichas, etc.
- Las capacidades de perf dependen de las propiedades de la CPU

Perf

- Para ver los comandos que ofrece perf, ejecutar:
- `$ perf`
 - ⇒ Algunos comandos pueden no estar disponibles.
- Para obtener la lista de opciones para un comando, ejecutar:
`$ perf comando -h`
- Ejemplo: `$ perf stat -h`
- Más información sobre un comando:
 - ⇒ `$ perf help comando,`
- Ejemplo: `$ perf help stat`
- Para listar el tipo de eventos capturables: `$ perf list`
`perf list | grep Hardware`

- En la lista de eventos podemos apreciar siete grupos diferentes:
 - ⇒ **Software event.** Entre otros permite contar cambios de contexto o fallos de página.
 - ⇒ **Hardware event.** Este evento se refiere a las PMU de los procesadores modernos, permite contar ciclos, instrucciones ejecutadas, fallos de cache etc.
 - ⇒ **Hardware cache event.** Estadísticas de memorias de caché de primer nivel de último nivel o del TLB.
 - ⇒ **Hardware breakpoint.** Los puntos de ruptura hardware permiten detener la ejecución del programa cuando el procesador intenta leer, escribir, o ejecutar el contenido de una determinada posición de memoria.
 - ⇒ **Tracerpoint event.** Se trata de una infraestructura flexible para trazar todo tipo de eventos en el kernel o en cualquier modulo del kernel. Incluye eventos de GPU, sistemas de archivos o del propio scheduler.
 - ⇒ **Raw hardware event.** En el caso de que perf no incluya todavía un nombre simbólico para un contador concreto de una PMU actual, se puede emplear el código hexadecimal correspondiente de acuerdo al manual del fabricante.

- Herramienta para el comportamiento del sistema, basada en el uso de contadores hardware

Modo captura de estadísticas

Eventos capturados por defecto

```
$ perf stat wget http://www.google.es
```

```
Performance counter stats for 'wget http://www.google.es':
```

| | | | | |
|------------|-------------------------|---|------------------------------|----------|
| 12,862369 | task-clock (msec) | # | 0,118 CPUs utilized | |
| 44 | context-switches | # | 0,003 M/sec | |
| 4 | cpu-migrations | # | 0,311 K/sec | |
| 717 | page-faults | # | 0,056 M/sec | |
| 13.967.013 | cycles | # | 1,086 GHz | [69,54%] |
| 3.549.130 | stalled-cycles-frontend | # | 25,41% frontend cycles idle | [56,82%] |
| 5.253.620 | stalled-cycles-backend | # | 37,61% backend cycles idle | |
| 16.853.395 | instructions | # | 1,21 insns per cycle | |
| | | # | 0,31 stalled cycles per insn | |
| 1.236.267 | branches | # | 96,115 M/sec | |
| 52.662 | branch-misses | # | 4,26% of all branches | [75,82%] |

```
0,108753070 seconds time elapsed
```

- Para listar el tipo de eventos capturables:

- `$ perf list`

- Selección de evento con `-e`

- `-e instructions, cycles ..`

- `<evento:u>` para filtrar código en espacio de usuario

Listado explícito de eventos a capturar

```
$ perf stat -e instructions, cycles, cache-references, cache-misses wget http://www.google.es
```

```
Performance counter stats for 'wget http://www.google.es':
```

| | | | |
|------------|------------------|---|----------------------------|
| 12.040.538 | cycles | | |
| 16.398.561 | instructions | # | 1,36 insns per cycle |
| 114.464 | cache-references | | |
| 46.024 | cache-misses | # | 40,208 % of all cache refs |

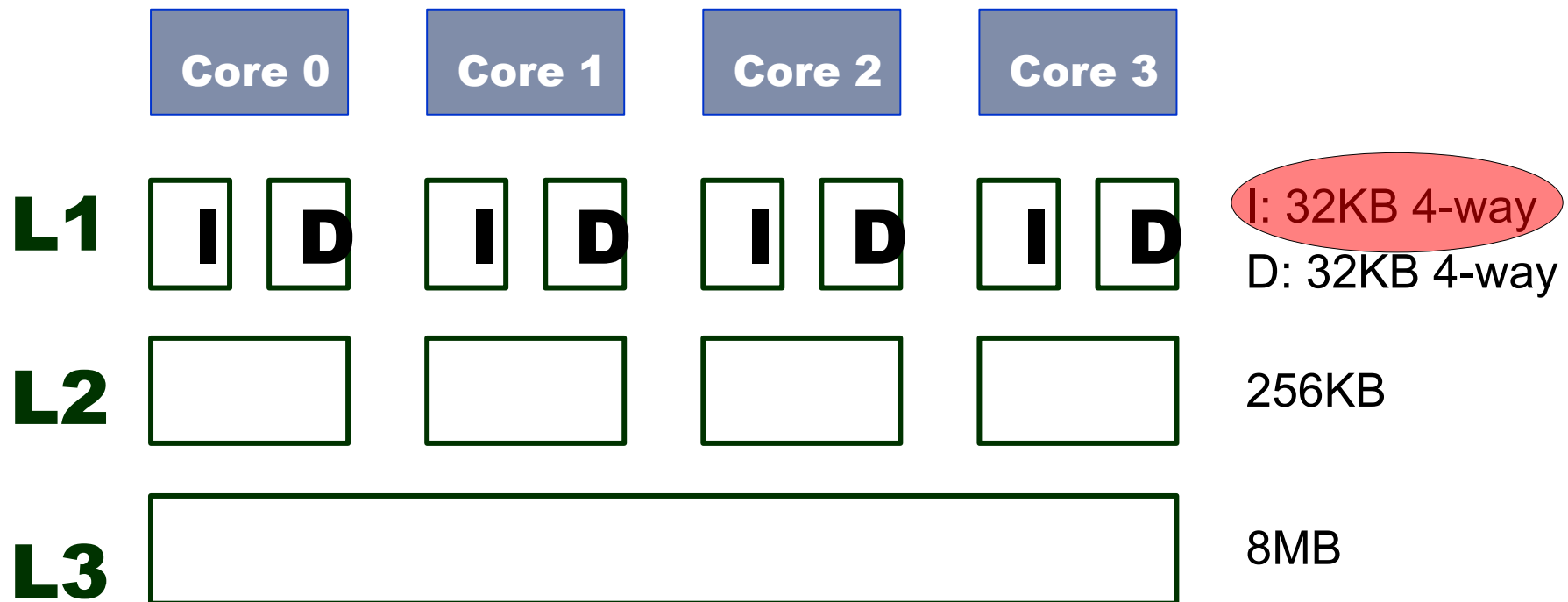
```
0,132842854 seconds time elapsed
```

Perf

- Ejemplo de uso utilizando un proceso de nuestra computadora.
- `perf stat -p PID sleep 5`
- Muestra el contador de estadísticas de un proceso determinado, por ejemplo Xorg, durante 10 segundos.

Jerarquía de memoria

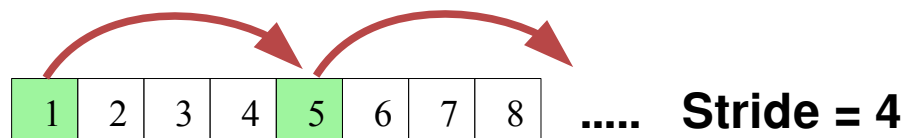
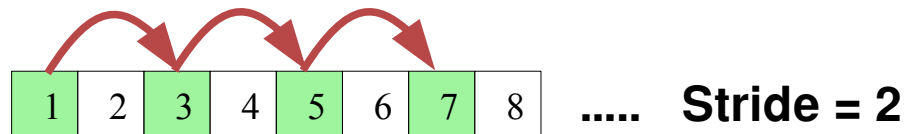
- Core i7 de tercera generación



Caso de estudio I

- Análisis de acceso a la cache: tamaño de la línea
- Recorrido de un vector de forma salteada

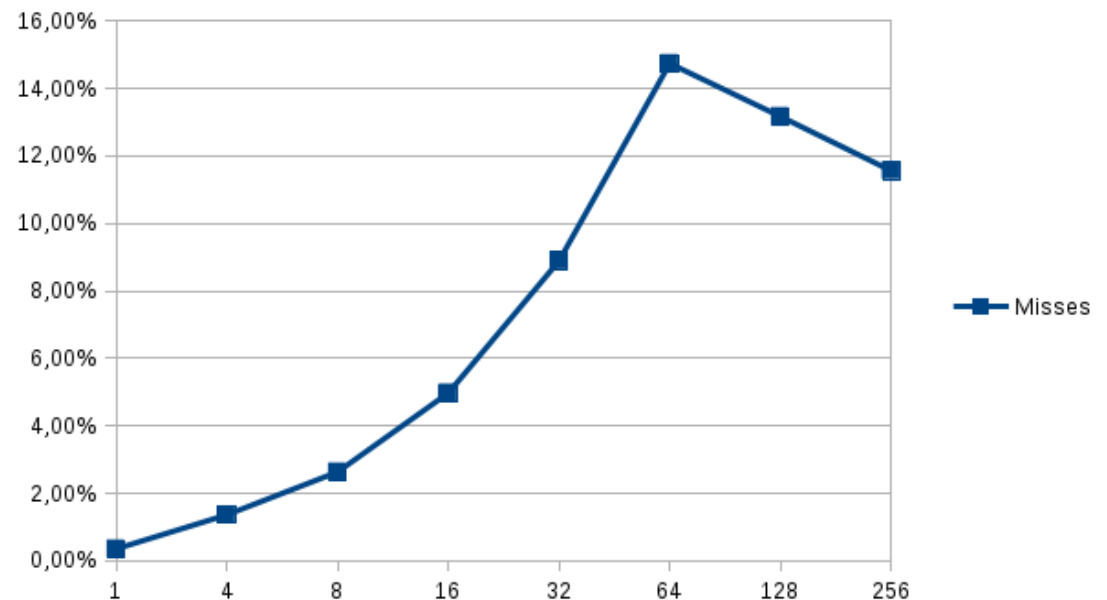
```
volatile unsigned char A[BUFFER_SIZE];           cache_size.c  
...  
for (i = 0; i < BUFFER_SIZE; i += STRIDE)  
    A[i]++;
```



Caso de estudio I

- El tamaño del buffer excede al de la cache L1
- Variando el stride (salto) debería observarse una variación anormal en la tasa de fallos
- ¿Por qué razón?

Ejecución del algoritmo sobre una arquitectura Core i5, con buffer de tamaño 64KB, y stride Entre (1 y 256)



Caso de estudio I

- Recompila el código de ejemplo variando el valor de la macro STRIDE:

```
gcc -O0 -DSTRIDE=<valor> -o cache_size cache_size.c
```

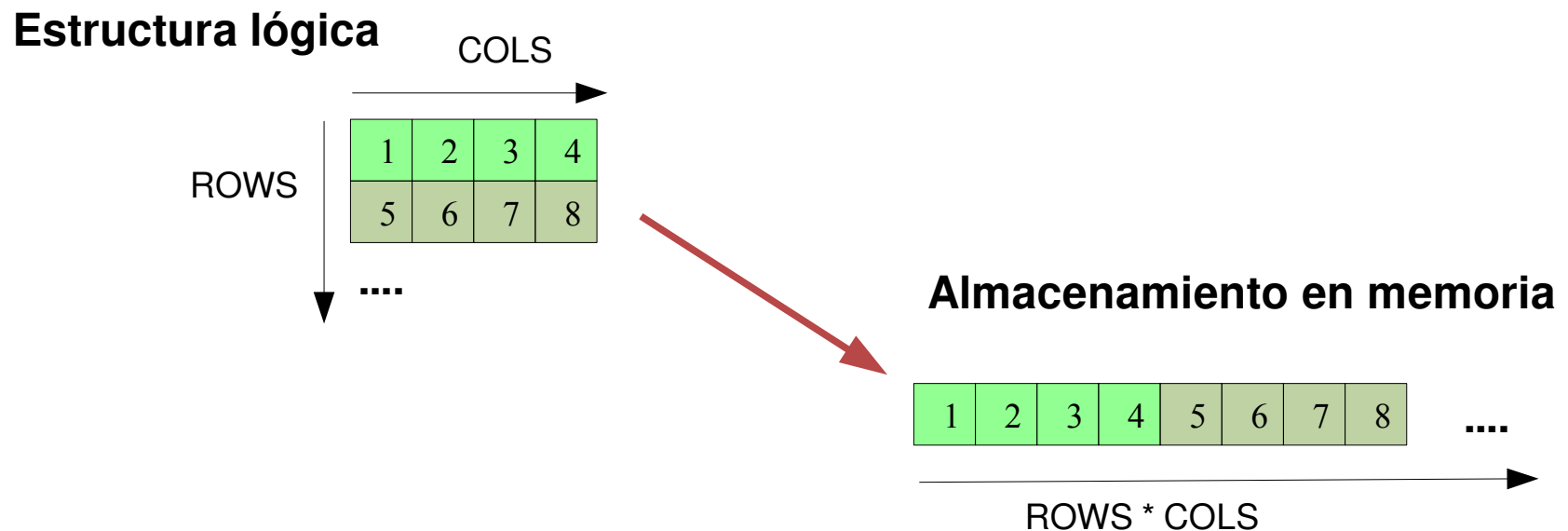
- Mide el efecto sobre la cache L1

```
perf stat -r 10 -e instructions:u,L1-dcache-loads-misses:u,L1-dcache-loads:u ./cache_size
```

- `Perf stat -help`, para conocer el significado del switch “-r”
- Analiza el resultado, especialmente el de los fallos de cache

Caso de estudio II

- Análisis de acceso a la cache: recorrido de matrices
- Las matrices en C se almacenan en memoria por filas (row_major)



Caso de estudio II

- Dos tipos de pruebas
 - Recorrido por filas (tal y como almacena la cache)

```
for ( i = 1; i < ROWS; i++)  
    for (j = 1; j < COLS; j++)  
        A[i][j]= A[i-1][j-1];
```

cache_row_major.c

- Recorrido por columnas
- cache_column_major.c

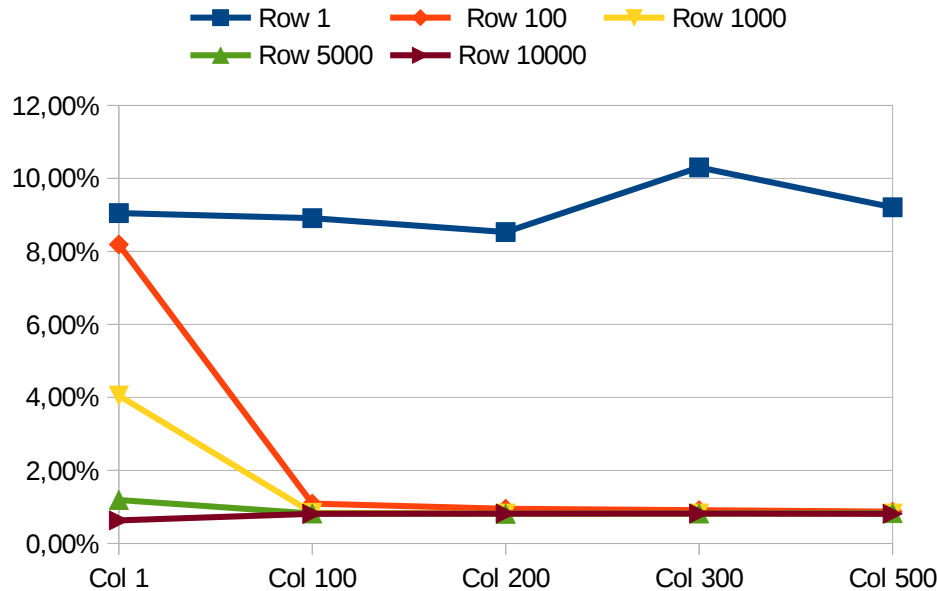
```
for ( j= 1; j < COLS; j++)  
    for (i = 1; i < rows; i++)  
        A[i][j]= A[i-1][j-1];
```

- Recompila el código de ejemplo variando el valor de las macros COLS y ROWS:

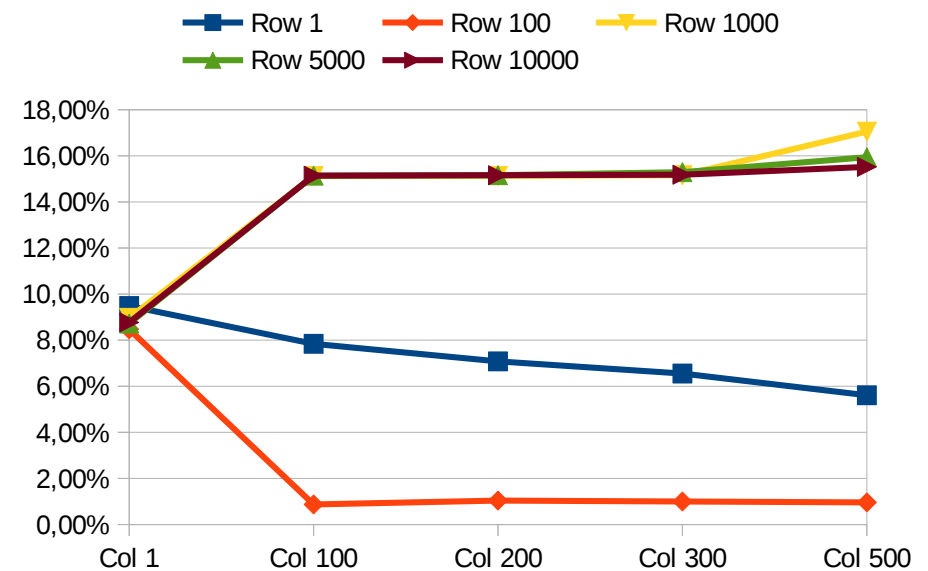
```
gcc -O0 -DROWS=<valor> -DCOLSS=<valor> -o cache_row_major cache_row_major.c
```

Caso de estudio II

- Representa gráficamente el recorrido sobre la matriz en ambos casos.
 - ¿Cuál se adapta mejor en la cache?
- Analiza con perf la tasa de fallos



Recorrido por filas



Recorrido por columnas