

Análisis memoria Ejercicio 10

Antes de nada hay que obtener los parámetros de memoria del equipo que estoy usando para realizar los test. Ejecutando en un terminal “\$ getconf -a | grep CACHE” se obtienen.

```
pablo@pablo-CatedraBQ ~ $ getconf -a | grep CACHE
LEVEL1_ICACHE_SIZE          32768
LEVEL1_ICACHE_ASSOC         8
LEVEL1_ICACHE_LINESIZE      64
LEVEL1_DCACHE_SIZE          32768
LEVEL1_DCACHE_ASSOC         8
LEVEL1_DCACHE_LINESIZE      64
LEVEL2_CACHE_SIZE           262144
LEVEL2_CACHE_ASSOC          8
LEVEL2_CACHE_LINESIZE       64
LEVEL3_CACHE_SIZE           3145728
LEVEL3_CACHE_ASSOC          12
LEVEL3_CACHE_LINESIZE       64
LEVEL4_CACHE_SIZE            0
LEVEL4_CACHE_ASSOC           0
LEVEL4_CACHE_LINESIZE       0
pablo@pablo-CatedraBQ ~ $
```

Por lo que L1 = 32768, 8,64 ; D1 = 32768,8,64; LL = 3145728,12,64

Ejecutando cachegrind sobre el programa “process_pgm” con los parámetros anteriores, obtenemos:

```
Closing output file vertical_edges.pgm
==15388==
==15388== I   refs:      1,705,984,739
==15388== I1 misses:      1,516
==15388== L1i misses:      1,284
==15388== I1 miss rate:      0.00%
==15388== L1i miss rate:      0.00%
==15388==
==15388== D   refs:      695,593,407 (423,370,855 rd + 272,222,552 wr)
==15388== D1 misses:      254,104 ( 169,976 rd + 84,128 wr)
==15388== L1d misses:      23,418 ( 2,081 rd + 21,337 wr)
==15388== D1 miss rate:      0.0% ( 0.0% + 0.0% )
==15388== L1d miss rate:      0.0% ( 0.0% + 0.0% )
==15388==
==15388== LL refs:      255,620 ( 171,492 rd + 84,128 wr)
==15388== LL misses:      24,702 ( 3,365 rd + 21,337 wr)
==15388== LL miss rate:      0.0% ( 0.0% + 0.0% )
pablo@pablo-CatedraBQ ~/MISE/SEMA/sema2018/semana2/10/images $
```

[illegible]

										y = 0;
1,030	1	1	2	0	0	0	0	512	4	while (y < h-1) {
										x = 0;
657,657	1	1	511	0	0	0	0	327,551	543	while (x < w) {
			Texas - Wallis							// Find the difference in colour between
			Thomson							// this pixel and the one below it
984,186	0	0	654,080	20,735	0	0	0	0	0	difff = p[y+1][x] - p[y][x];
			Father Paul's room							
			Helen Motilal Bhambhani							// Make sure difference is positive
										if (difff < 0) difff = -difff;
										// If the difference exceeds the threshold
										// make the pixel white
2,943,366	1	1	0	0	0	327,041	0	0	0	if (difff > min_difff) p[y][x] = 255;
										else p[y][x] = 0;
										x = x + 1;
										}
										y = y + 1;
										}
										printf("OK\n");
7	0	0	6	1	0	0	0	0	0	}

	Ir	Ilmr	Ilmr	Or	Olmr	Olmr	Dw	Dlwm	Dlwm	Bc	Bcm	Bi	Bin	
1,705,795,939	1,586	1,277	423,368,969	169,881	2,085	272,221,719	84,130	21,354	294,380,292	3,498,526	10,493,758	172	PROGRAM TOTALS	
	Ir	Ilmr	Ilmr	Or	Olmr	Olmr	Dw	Dlwm	Dlwm	Bc	Bcm	Bi	Bin	file: function
594,313,692	122	75	155,656,929	106	70	91,802,348	82,963	20,737	111,983,152	1,318,649	2,621,480	3	/build/glibc-GL567W/glibc-2.23/stdio-common/vfsconf.c: IO_vfsconf	
380,781,844	73	50	122,369,716	81	7	83,437,627	98	0	60,275,016	91,897	5,243,024	21	/build/glibc-GL567W/glibc-2.23/stdio-common/vfprintf.c: vfprintf	
150,165,426	1	1	684,266	4	0	17,639,466	37	0	1,536,862	0	0	0	/build/glibc-GL567W/glibc-2.23/stdlib/./stdlib/strotl.c: _strotl_internal	
128,468,583	8	8	27,529,740	18	1	20,972,898	12	0	14,420,618	829	844	2	/build/glibc-GL567W/glibc-2.23/libio/printf.c: file_xpunct@GLIBC_2.2.5	
64,225,476	11	5	10,39,412	0	0	14,417,964	0	0	9,175,968	39	0	0	/build/glibc-GL567W/glibc-2.23/debug/printf.c: printf_chk	
55,050,912	18	6	14,418,096	0	0	14,418,096	0	0	1,553,688	727	0	0	/build/glibc-GL567W/glibc-2.23/stdio-common/isc99_fscanf.c: _isc99_fscanf	
52,431,008	0	0	5,243,126	40	0	5,243,856	25	0	5,243,856	25	0	0	/build/glibc-GL567W/glibc-2.23/string/./sysdeps/x86_64/stchr.S: stchrnul	
48,778,131	3	3	5,642,229	0	3	8,021,316	284	20	15,725,529	432,953	0	0	/build/glibc-GL567W/glibc-2.23/string/./sysdeps/x86_64/memcpy.S: G1_memcpy	
39,320,362	2	2	15,728,984	0	0	5,242,968	0	0	5,242,968	25	0	0	/build/glibc-GL567W/glibc-2.23/libio/genops.c: IO_sputbackc	
38,252,630	0	0	1,133,612	0	0	1,133,612	0	0	1,448,676	186,264	0	0	/build/glibc-GL567W/glibc-2.23/string/./sysdeps/x86_64/stpncpy.S: stpncpy	
19,550,332	1	1	3,932,212	0	0	5,242,948	5	0	1,310,736	6	0	0	/build/glibc-GL567W/glibc-2.23/stdio-common/./include/scratch_buffer.h: IO_vfsconf	
15,728,688	1	1	6,553,620	0	0	2,621,448	0	0	5,242,896	16	0	0	/build/glibc-GL567W/glibc-2.23/debug/./libio/libioP.h: _printf_chk	
14,438,664	12	5	1,314,860	0	0	1,314,860	0	0	2,625,560	2,059	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: load_image_from_file	
13,107,560	0	0	7,864,416	0	0	1,310,736	0	0	5,242,844	6	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: _isc99_fscanf	
13,107,560	0	0	0	0	0	6,553,780	5	0	0	0	0	0	/build/glibc-GL567W/glibc-2.23/stdio-common/vfprintf.c: file_vfprintf	
7,872,680	3	3	1,310,740	82,948	0	1,312,800	0	0	0	0	0	0	/usr/include/x86_64-linux-gnu/bits/stdio2.h: save_image_to_file	
5,261,428	3	3	1,314,648	0	0	36	0	0	1,314,620	2,061	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: save_image_to_file	
1,505,256	2	2	654,599	20,736	1	327,610	0	0	328,063	547	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: vertical_edge_detect	
1,392,184	1	1	2,621,456	0	0	0	0	0	328,193	535	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: horizontal_edge_detect	
2,626,147	32	20	2,623,755	74	0	43	1	1	11	5	6,263,716	33	???:???	
2,287,970	3	3	327,684	20,736	0	327,668	0	0	328,765	547	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: apply_threshold	
1,967,120	3	3	327,171	20,736	0	327,170	0	0	328,193	540	0	0	/home/pablo/MISE/SEMA/sem2018/sema2018/10/images/process/pnp.c: invert_colours	

Proponemos reducir la caché de primer nivel. Una primera aproximación conservadora sería reducirla a la mitad.

Caché a 16384

```
Closing output file vertical_edges.pgm
==15586==
==15586== I   refs:      1,705,985,509
==15586== I1 misses:      2,896
==15586== L1i misses:      1,284
==15586== I1 miss rate:      0.00%
==15586== L1i miss rate:      0.00%
==15586==
==15586== D   refs:      695,593,598 (423,371,046 rd + 272,222,552 wr)
==15586== D1 misses:      263,487 ( 173,382 rd + 90,105 wr)
==15586== L1d misses:      23,418 ( 2,081 rd + 21,337 wr)
==15586== D1 miss rate:      0.0% ( 0.0% + 0.0% )
==15586== L1d miss rate:      0.0% ( 0.0% + 0.0% )
==15586==
==15586== LL refs:      266,383 ( 176,278 rd + 90,105 wr)
==15586== LL misses:      24,702 ( 3,365 rd + 21,337 wr)
==15586== LL miss rate:      0.0% ( 0.0% + 0.0% )
pablo@pablo-CatedraBQ ~/MISE/SEMA/sema2018/semana2/10/images $
```

Caché a 8192

```
Printing pixel data to file vertical_edges.pgm
Closing output file vertical_edges.pgm
==15674==
==15674== I   refs:      1,705,986,844
==15674== I1 misses:      35,206
==15674== L1i misses:      1,284
==15674== I1 miss rate:      0.00%
==15674== L1i miss rate:      0.00%
==15674==
==15674== D   refs:      695,593,927 (423,371,375 rd + 272,222,552 wr)
==15674== D1 misses:      388,508 ( 248,960 rd + 139,548 wr)
==15674== L1d misses:      23,419 ( 2,082 rd + 21,337 wr)
==15674== D1 miss rate:      0.1% ( 0.1% + 0.1% )
==15674== L1d miss rate:      0.0% ( 0.0% + 0.0% )
==15674==
==15674== LL refs:      423,714 ( 284,166 rd + 139,548 wr)
==15674== LL misses:      24,703 ( 3,366 rd + 21,337 wr)
==15674== LL miss rate:      0.0% ( 0.0% + 0.0% )
pablo@pablo-CatedraBQ ~/MISE/SEMA/sema2018/semana2/10/images $
```

Aquí ya observamos que la tasa de fallos se ha incrementado , por lo que lo dejamos en 8192.

Comparación de consumos

32kB

```
pablo@pablo-CatedraBQ ~ $ ecacti 32768 64 8 0.13 1
Command line: ecacti 32768 64 8 0.13 1
-----
eCACTI 1.0
-----
Cache Subarray Parameters (C, B, A): (32768 B, 64 B, 8)
Number of Subarrays: 1
Ports (RW, R, W): (1, 0, 0)
Technology: 0.13 um, Vdd: 1.1 V

#Cache configurations expressed in (Ndwl, Nspd, Ndbl, Ntl, Ntspd, Ntbl) format
Optimal Power-AccessTime-Area Config: (4, 1, 1, 1, 1, 8)

Power Stats:
-----
Read hit power: 249.4904 mw
Write hit power: 118.5853 mw

Read miss power: 345.5423 mw
Write miss power: 123.7422 mw

Timing Stats:
-----
Access time: 1.74779 ns
Cycle time (wave pipelined): 0.932986 ns

Area Stats:
-----
Aspect ratio (height/width): 6.828946
Total area one subarray : 0.028348 cm^2
```

8kB

```
pablo@pablo-CatedraBQ ~ $ ecacti 8192 64 8 0.13 1
Command line: ecacti 8192 64 8 0.13 1
-----
eCACTI 1.0
-----
Cache Subarray Parameters (C, B, A): (8192 B, 64 B, 8)
Number of Subarrays: 1
Ports (RW, R, W): (1, 0, 0)
Technology: 0.13 um, Vdd: 1.1 V

#Cache configurations expressed in (Ndwl, Nspd, Ndbl, Ntl, Ntspd, Ntbl) format
Optimal Power-AccessTime-Area Config: (8, 1, 1, 4, 1, 2)

Power Stats:
-----
Read hit power: 147.5782 mw
Write hit power: 86.3801 mw

Read miss power: 217.7139 mw
Write miss power: 89.4268 mw

Timing Stats:
-----
Access time: 1.03046 ns
Cycle time (wave pipelined): 0.344086 ns

Area Stats:
-----
Aspect ratio (height/width): 13.251415
Total area one subarray : 0.015430 cm^2
```

Observando los resultados obtenidos podemos apreciar que el consumo es aproximadamente un 40 por ciento menor en el caso de la memoria de 8kB. Dado que la tasa de fallos es insignificante en ambos casos, se pasa de un consumo medio de 250/119 mW a 148/86 mW, lo que supone una mejora del 35 por ciento aproximadamente. Otra manera de mejorar el consumo sería optimizando el procesamiento de la imagen, reduciendo el número de accesos a memoria mediante la reordenación o desenrollado del bucle.