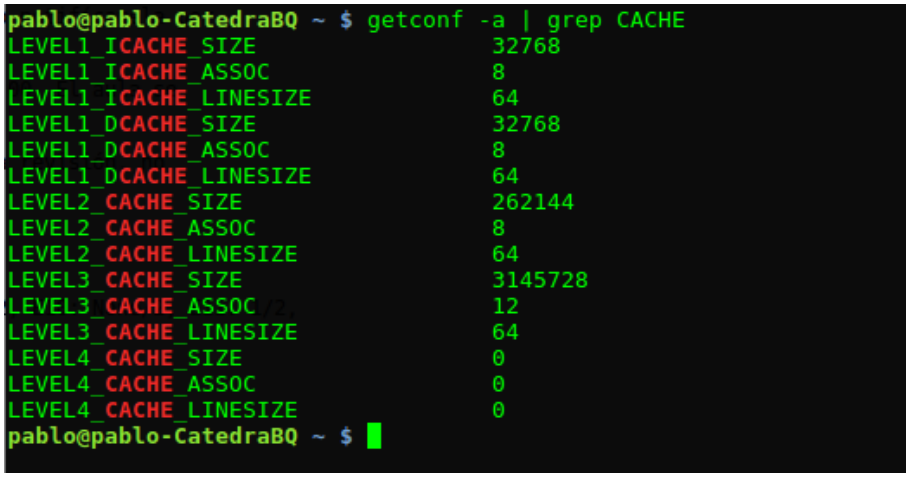
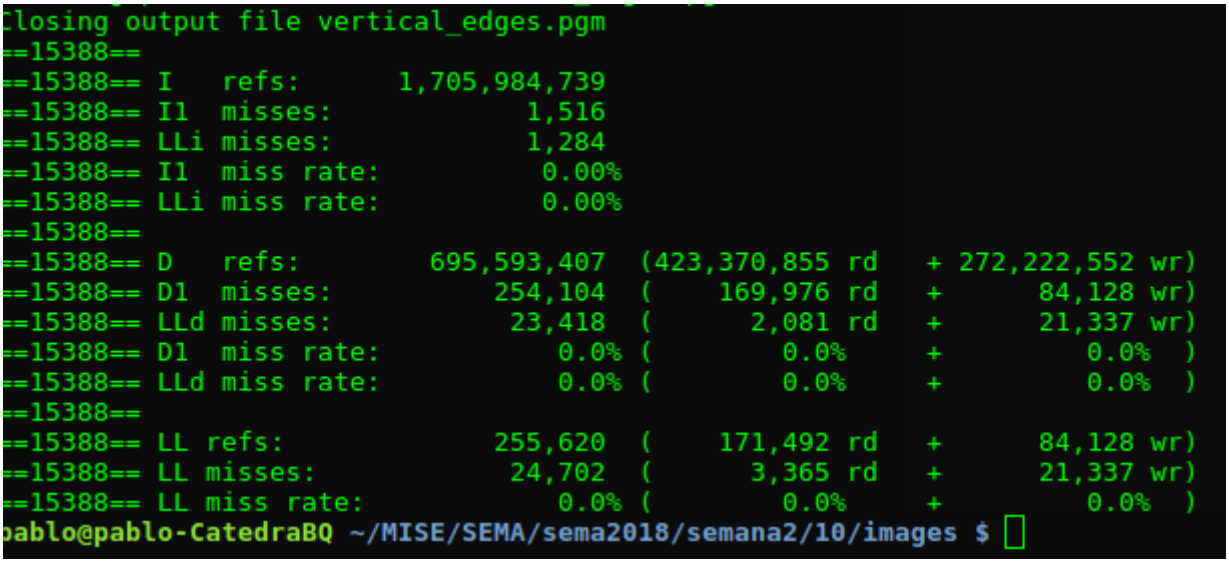
Análisis memoria Ejercicio 10

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.



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:



De compilar el programa con make, obtenemos que donde se realizan la mayoría de los accesos a memoria es en los bucles de procesamiento y guardado de la imagen:

```
      .    .    .          .    .    .          .    .    .          .    .    .    printf("Printing pixel data to file %s\n", filename);
      .    .    .          .    .    .          .    .    .          .    .    .    y = 0;
  4,120    1    1      2,052    0    0          0    0    0      2,052    7    0    0  while (y < h) {
      .    .    .          .    .    .          .    .    .          .    .    .      x = 0;
2,633,728    0    0 1,312,768    0    0          0    0    0 1,312,768 2,054    0    0  while (x < w) {
      .    .    .          .    .    .          .    .    .          .    .    .        fprintf(f, "%d ", p[y][x]);
2,621,440    0    0          0    0    0          0    0    0          0    0    0        x = x + 1;
      .    .    .          .    .    .          .    .    .          .    .    .      }
      .    .    .          .    .    .          .    .    .          .    .    .      fprintf(f, "\n");
  2,048    0    0          0    0    0          0    0    0          0    0    0      y = y + 1;
      .    .    .          .    .    .          .    .    .          .    .    .    }
```

```
      .    .    .          .    .    .          .    .    .          .    .    .    y = 0;
  1,030    1    1          2    0    0          0    0    0        512    4    0    0  while (y < h-1) {
      .    .    .          .    .    .          .    .    .          .    .    .      x = 0;
657,657    1    1        511    0    0          0    0    0    327,551  543    0    0    while (x < w) {
      .    .    .          .    .    .          .    .    .          .    .    .      // Find the difference in colour between
      .    .    .          .    .    .          .    .    .          .    .    .      // this pixel and the one below it
984,186    0    0    654,080 20,735    0          0    0    0          0    0    0      diff = p[y+1][x] - p[y][x];
      .    .    .          .    .    .          .    .    .          .    .    .
      .    .    .          .    .    .          .    .    .          .    .    .      // Make sure difference is positive
      .    .    .          .    .    .          .    .    .          .    .    .      if (diff < 0) diff = -diff;
      .    .    .          .    .    .          .    .    .          .    .    .
      .    .    .          .    .    .          .    .    .          .    .    .      // If the difference exceeds the threshold
      .    .    .          .    .    .          .    .    .          .    .    .      // make the pixel white
2,943,366    1    1          0    0    0    327,041    0    0          0    0    0    if (diff > min_diff) 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    0    0  }
```

Y como resumen de los accesos a memoria de las funciones de librería de C, tenemos que :

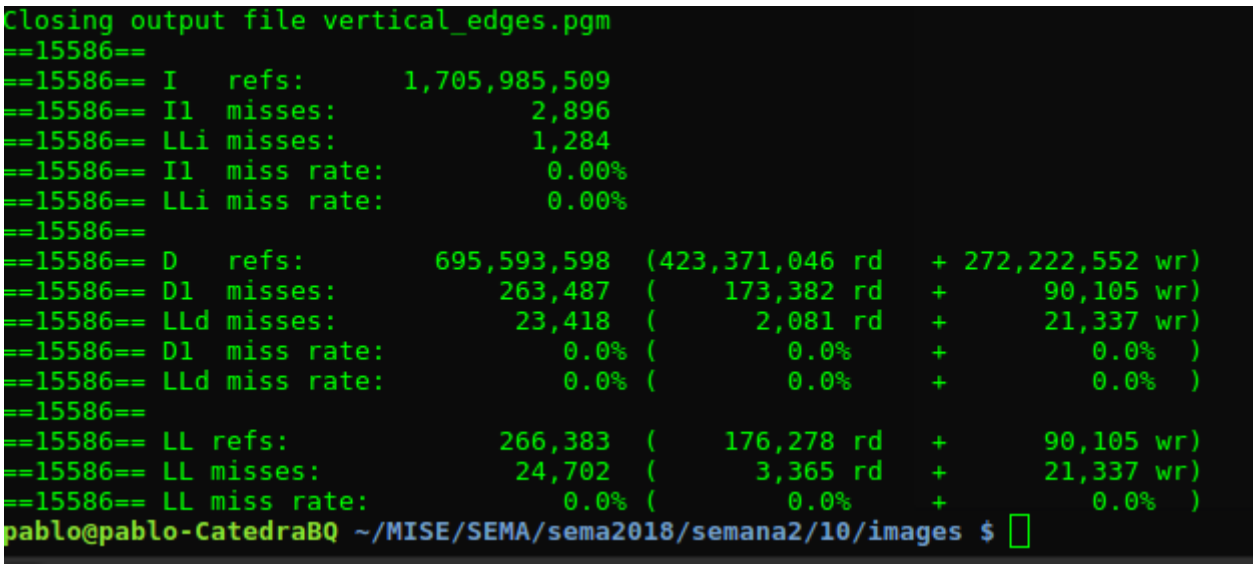
```
--------------------------------------------------------------------------------
          Ir  I1mr  ILmr          Dr    D1mr  DLmr          Dw    D1mw  DLmw          Bc          Bcm         Bi Bim
--------------------------------------------------------------------------------
1,705,975,939 1,506 1,277 423,368,969 169,881 2,085 272,221,719 84,130 21,354 294,380,202 3,498,526 10,493,758 172  PROGRAM TOTALS

--------------------------------------------------------------------------------
          Ir I1mr ILmr          Dr   D1mr DLmr          Dw    D1mw   DLmw          Bc       Bcm        Bi Bim  file:function
--------------------------------------------------------------------------------
594,313,692  122   75 155,656,920    106    70 91,802,348 82,963 20,737 111,983,152 1,318,649 2,621,480   3  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/vfscanf.c:_IO_vfscanf
388,781,844   73   50 122,309,716     81     7 83,437,627    58      0  60,275,016    91,897 5,243,024  21  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/vfprintf.c:vfprintf
186,165,616   32   17  26,684,200      4     0 17,039,464     0      0  45,167,592 1,536,062         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdlib/../stdlib/strtol_l.c:___strtol_l_internal
128,468,583    8    8  27,529,740     18     1 20,972,898    12      0  14,420,618       829       844   2  /build/glibc-Cl5G7W/glibc-2.23/libio/fileops.c:_IO_file_xsputn@@GLIBC_2.2.5
 64,225,476   11    5  17,039,412      0     0 14,417,964     0      0   9,175,068        39         0   0  /build/glibc-Cl5G7W/glibc-2.23/debug/fprintf_chk.c:__fprintf_chk
 55,050,912   10    6  14,418,096      0     0 14,418,096     0      0   6,553,680        47         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/isoc99_fscanf.c:__isoc99_fscanf
 52,431,008    8    5   5,243,120     46     8          0     0      0   5,243,056        25         0   0  /build/glibc-Cl5G7W/glibc-2.23/string/../sysdeps/x86_64/strchr.S:strchrnul
 48,778,131    3    3   5,642,229      0     0  3,021,316   284     20  15,725,529   432,053         0   0  /build/glibc-Cl5G7W/glibc-2.23/string/../sysdeps/x86_64/memcpy.S:__GI_mempcpy
 39,322,260    2    2  15,728,904      0     0  5,242,968     0      0   5,242,968        25         0   0  /build/glibc-Cl5G7W/glibc-2.23/libio/genops.c:_IO_sputbackc
 38,713,812    6    3   3,444,539      4     1  2,133,811     0      0   3,444,539   106,264         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/_itoa.c:_itoa_word
 18,350,332    4    1   3,932,212      0     0  5,242,948     3      0   1,310,736         6         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/../include/scratch_buffer.h:_IO_vfscanf
 15,728,688    1    1   6,553,620      0     0  2,621,448     0      0   5,242,896        16         0   0  /build/glibc-Cl5G7W/glibc-2.23/debug/../libio/libioP.h:__fprintf_chk
 14,438,664   12    5   1,314,860      8     0  1,310,780     0      0   2,625,560     2,059         0   0  /home/pablo/MISE/SEMA/sema2018/semana2/10/images/process_pgm.c:load_image_from_file
 14,418,096    4    1   7,864,416      0     0  1,310,736     0      0   5,242,944         6         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/../libio/libioP.h:__isoc99_fscanf
 13,107,560    0    0           0      0     0  6,553,780     5      0           0         0         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdio-common/printf-parse.h: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   1,314,848      4     0         36     0      0   1,314,820     2,061         0   0  /home/pablo/MISE/SEMA/sema2018/semana2/10/images/process_pgm.c:save_image_to_file
  4,586,256    3    3     654,599 20,736     0    327,048     0      0     328,063       547         0   0  /home/pablo/MISE/SEMA/sema2018/semana2/10/images/process_pgm.c:vertical_edge_detect
  4,584,469    2    2     654,340 20,736     0    327,171     0      0     328,193       535         0   0  /home/pablo/MISE/SEMA/sema2018/semana2/10/images/process_pgm.c:horizontal_edge_detect
  3,932,184    1    1   2,621,456      0     0          0     0      0           0         0         0   0  /build/glibc-Cl5G7W/glibc-2.23/stdlib/../stdlib/strtol.c:__strtol_internal
  2,626,147   32   20   2,623,755     34     0         43     1      1          11         6 2,623,716  33  ???:???
  2,297,879    2    2     327,684 20,736     0    327,683     0      0     328,705       542         0   0  /home/pablo/MISE/SEMA/sema2018/semana2/10/images/process_pgm.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/sema2018/semana2/10/images/process_pgm.c:invert_colours
--------------------------------------------------------------------------------
```

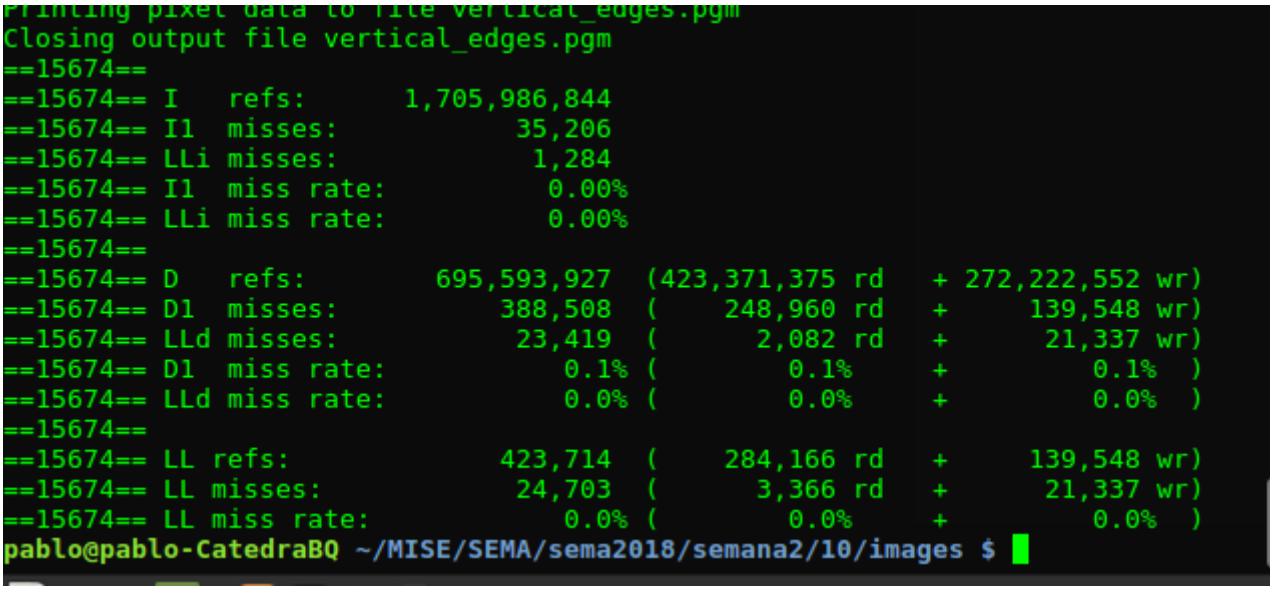
La mayoría de los accesos a la caché vienen por los bucles de procesamiento de los pixels de las imágenes. Se puede observar que la tasa de fallos de esta caché es muy baja. No obstante para los accesos a memoria principal para salvar las imágenes la tasa de fallo es mayor aunque siga siendo baja. Gran parte de estos fallos se deben a predicciones de salto erróneas por parte del planificador.

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

Caché a 16384



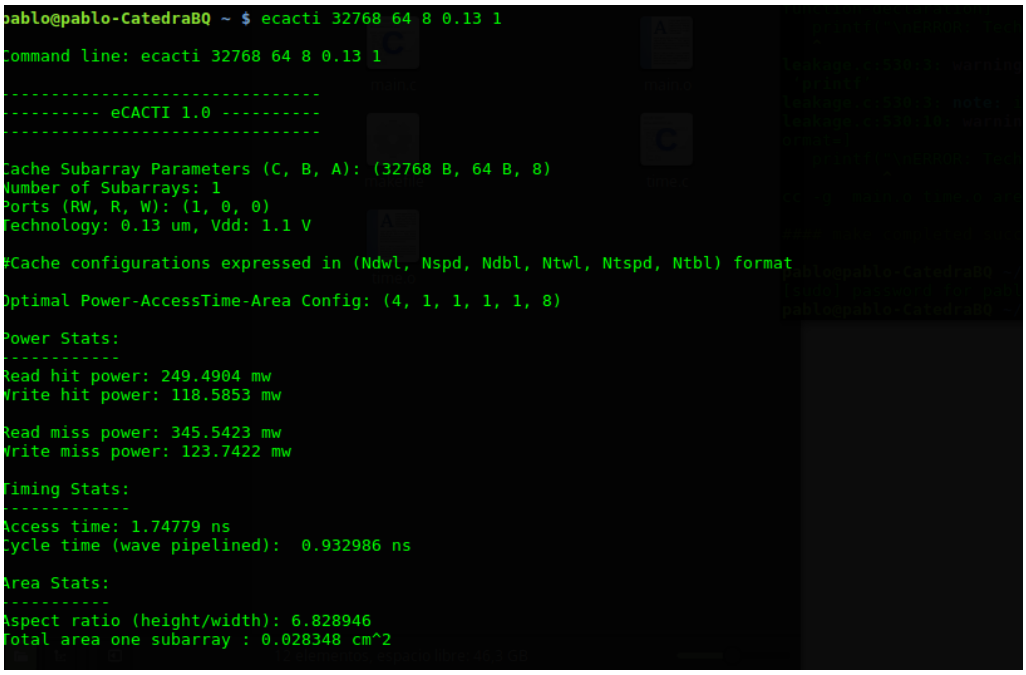
Caché a 8192



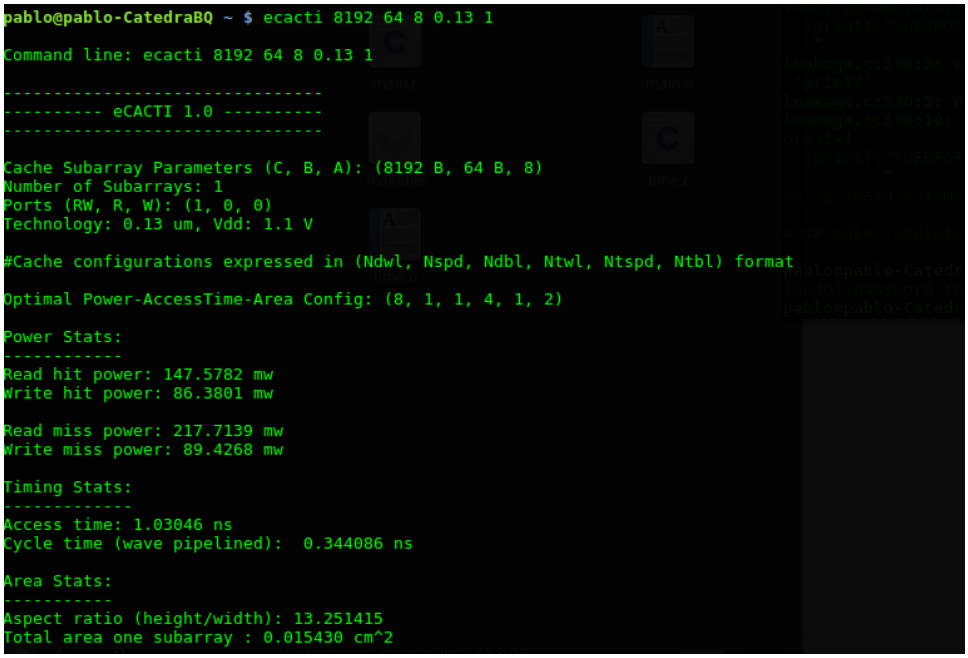
Aqui ya observamos que la tasa de fallos se ha incrementado , por lo que lo dejamos en 8192.

Comparación de consumos

32kB



8kB



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 procesado de la imagen, reduciendo el numero de accesos a memoria mediante la reordenación o desenrollado del bucle.