

Memoria de la práctica 3

Manel Lurbe Sempere

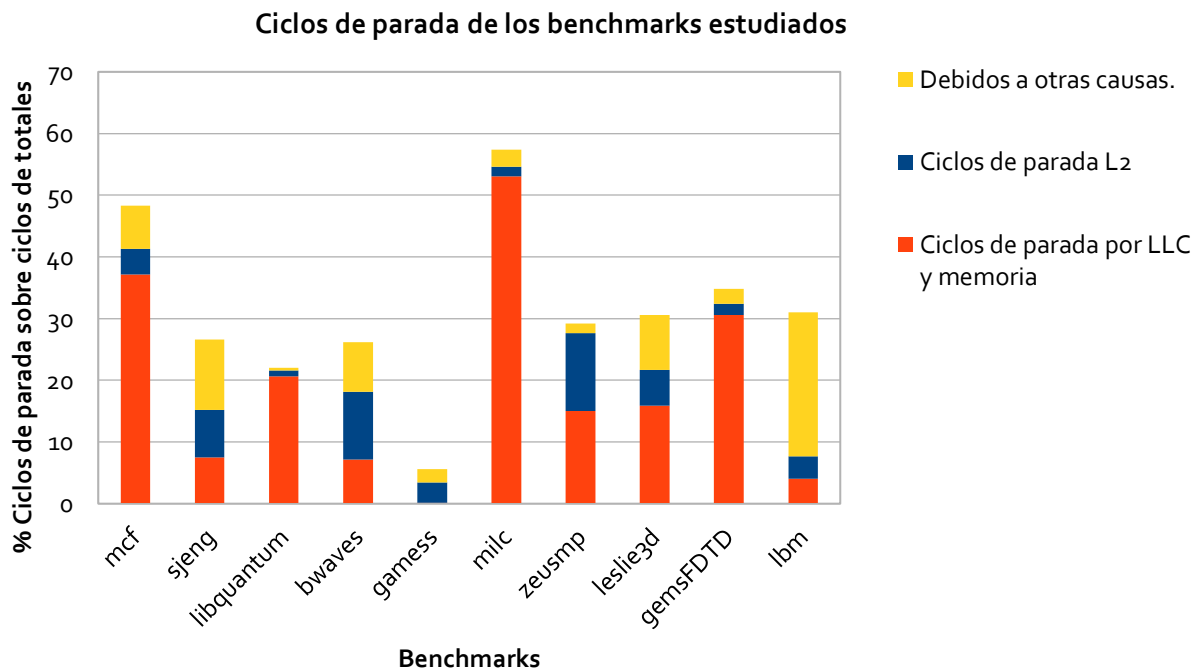
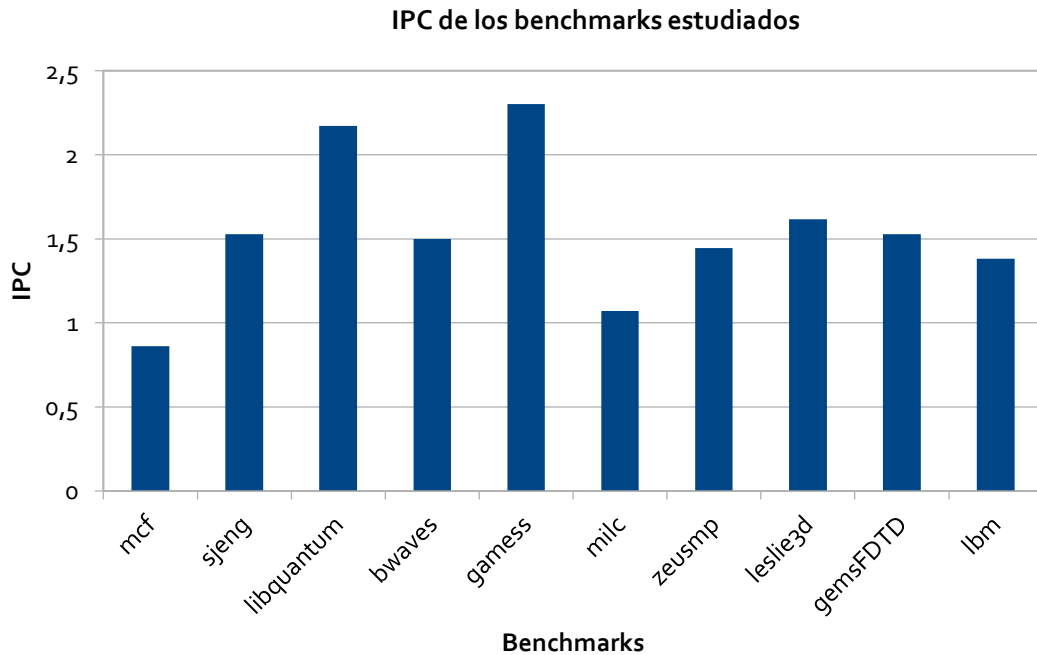
Manuel José Martínez Baños

Contenido

Actividad 1	2
Monitorización IPC y ciclos de parada en la jerarquía de memoria	2
Actividad 2	3
Activación y desactivación de la prebúsqueda en tiempo de ejecución.....	3
Actividad 3	5
Análisis de las características de las aplicaciones que más se le benefician de los mecanismos de prebúsqueda	5

Actividad 1

Monitorización IPC y ciclos de parada en la jerarquía de memoria

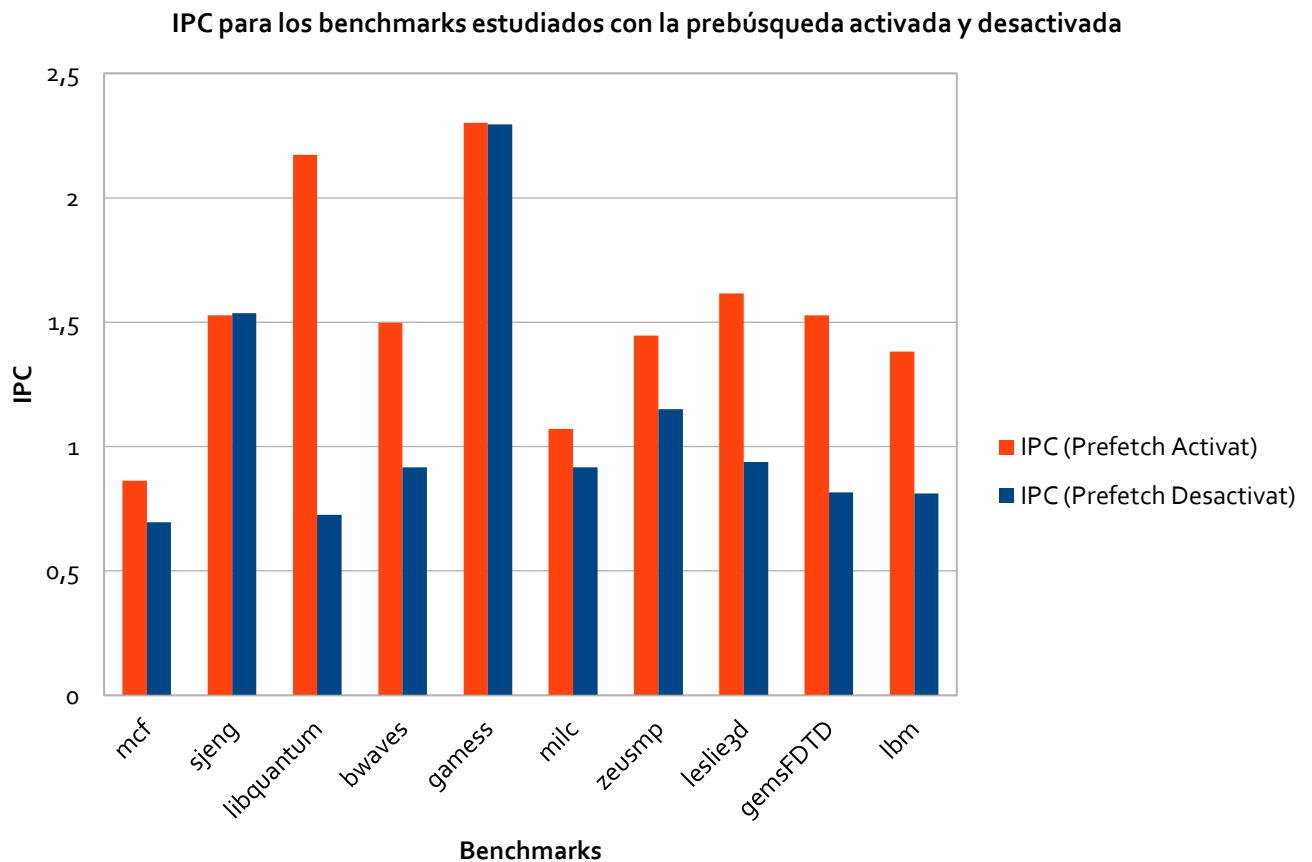


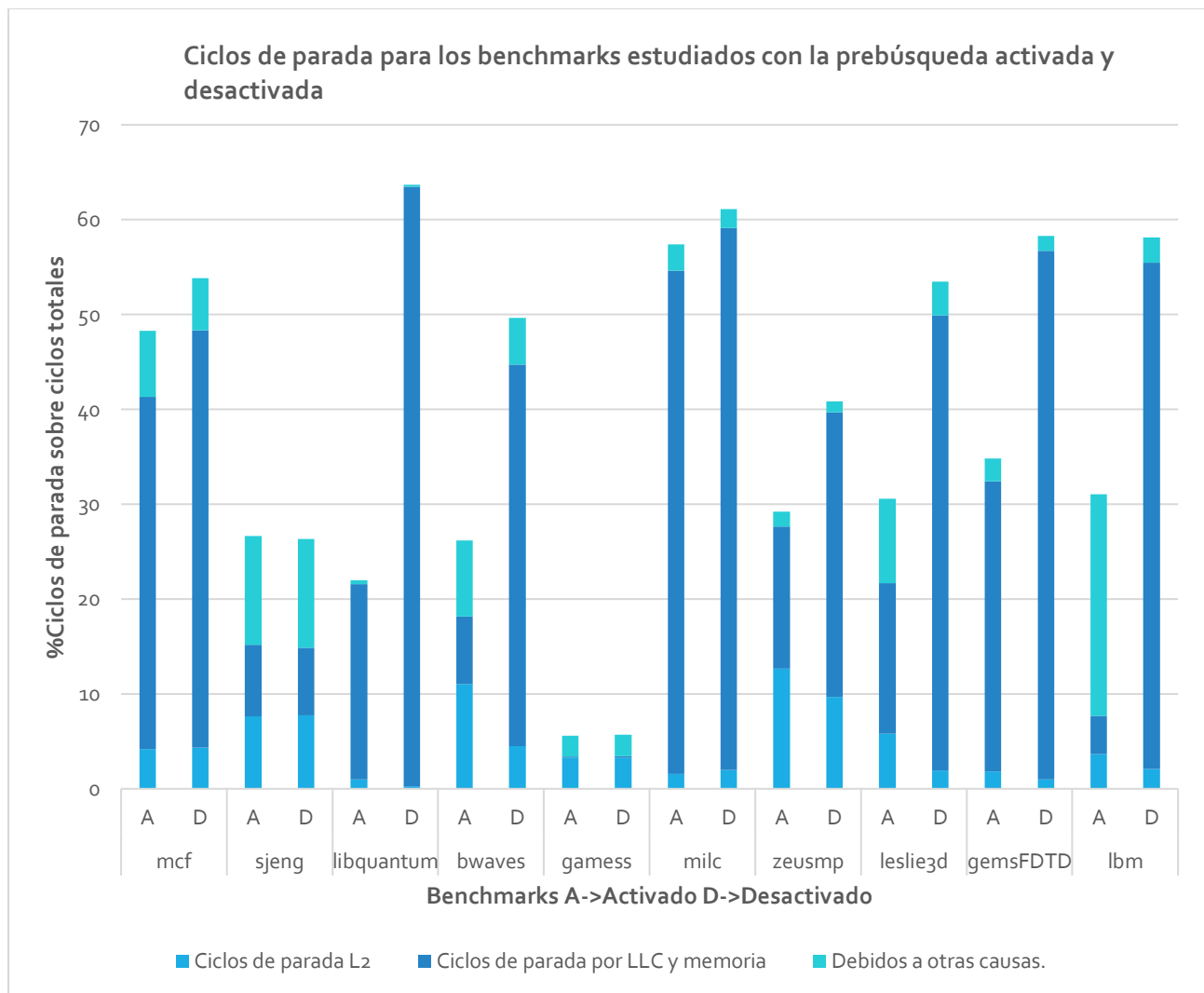
¿Existe alguna relación entre el IPC y los ciclos de parada de las aplicaciones?

En los gráficos se observa que a menor IPC tenemos más ciclos de parada, y a mayor IPC menos ciclos de parada. Por ejemplo, comparando los casos más significativos, tenemos “gameess” con el mayor IPC, sobrepasando el 2, y pocos ciclos de parada, inferior al 10%. Por otro lado, tenemos “milc” que tiene un IPC cercano al 1 y con muchos más ciclos de parada.

Actividad 2

Activación y desactivación de la prebúsqueda en tiempo de ejecución





¿Cuál es el componente principal de los ciclos de parada que se reduce? ¿Cómo contribuye la prebúsqueda a la reducción de este componente?

En el gráfico vemos como se reduce significativamente en la mayoría de los casos los ciclos de parada por LLC y memoria, porque gracias a la prebúsqueda los ciclos de parada que se evitan son los que aciertan en la L2. Con la mejora de la prebúsqueda reducimos los ciclos de parada y por tanto mejoramos el IPC. La prebúsqueda contribuye a reducir este componente ya que anticipa con la búsqueda de los datos en paralelo con la ejecución de instrucciones, cosa que permite que los datos lleguen antes reduciendo así los ciclos de ejecución de las instrucciones.

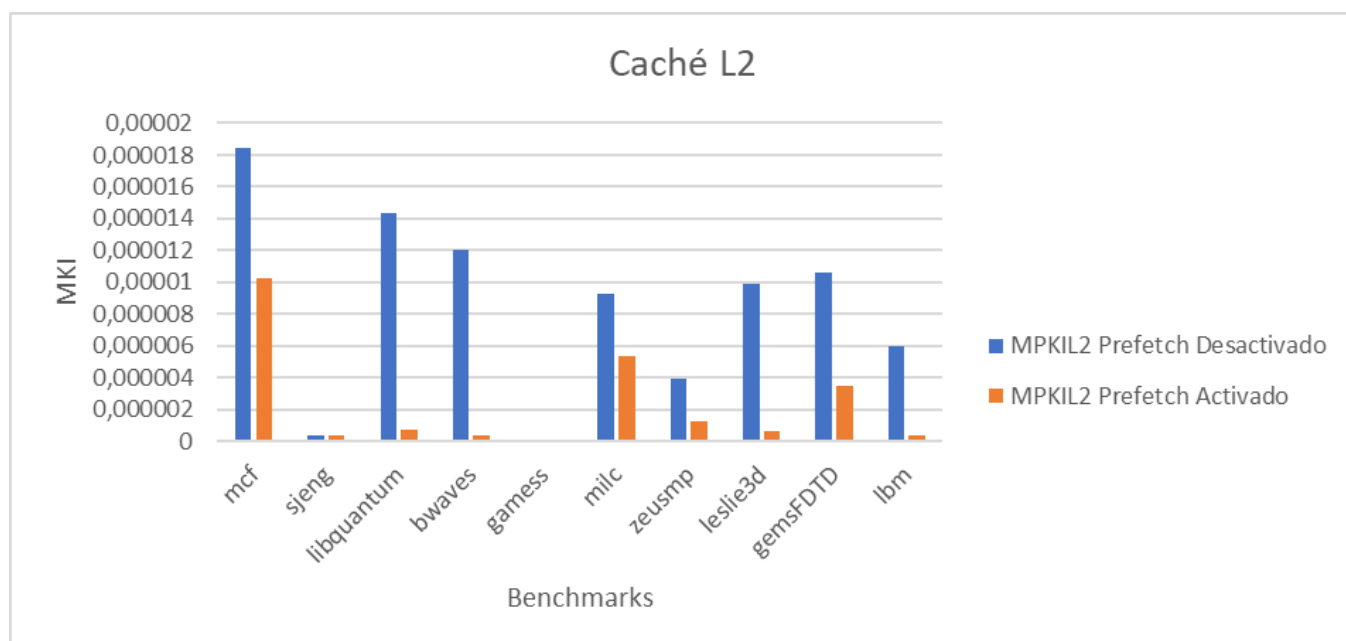
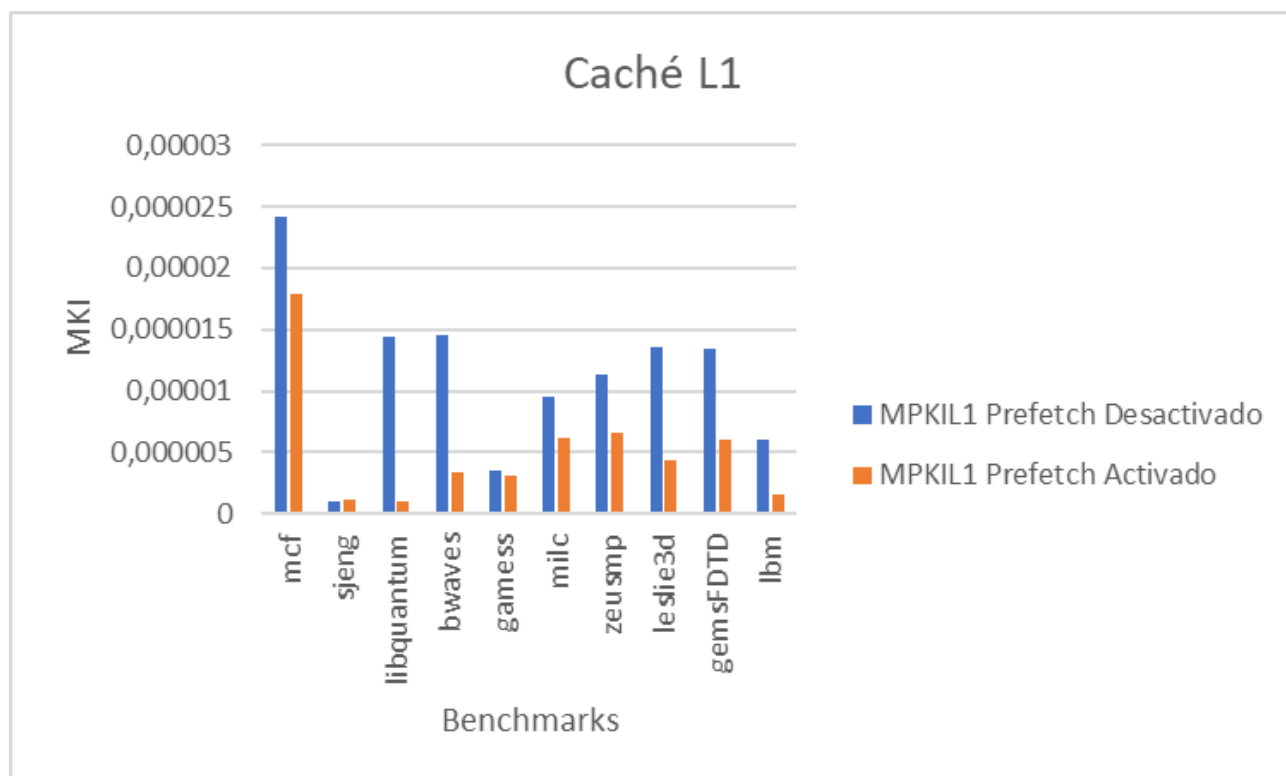
Con el prefetch activado, se nota más la diferencia con el valor de ciclos de parada de L2 que en memoria. Esto se debe a que gracias a este prefetch cuando no se produce un fallo en L2, no se produce un ciclo de parada, por el contrario, en memoria, mínimo se ha producido un ciclo de parada por L2.

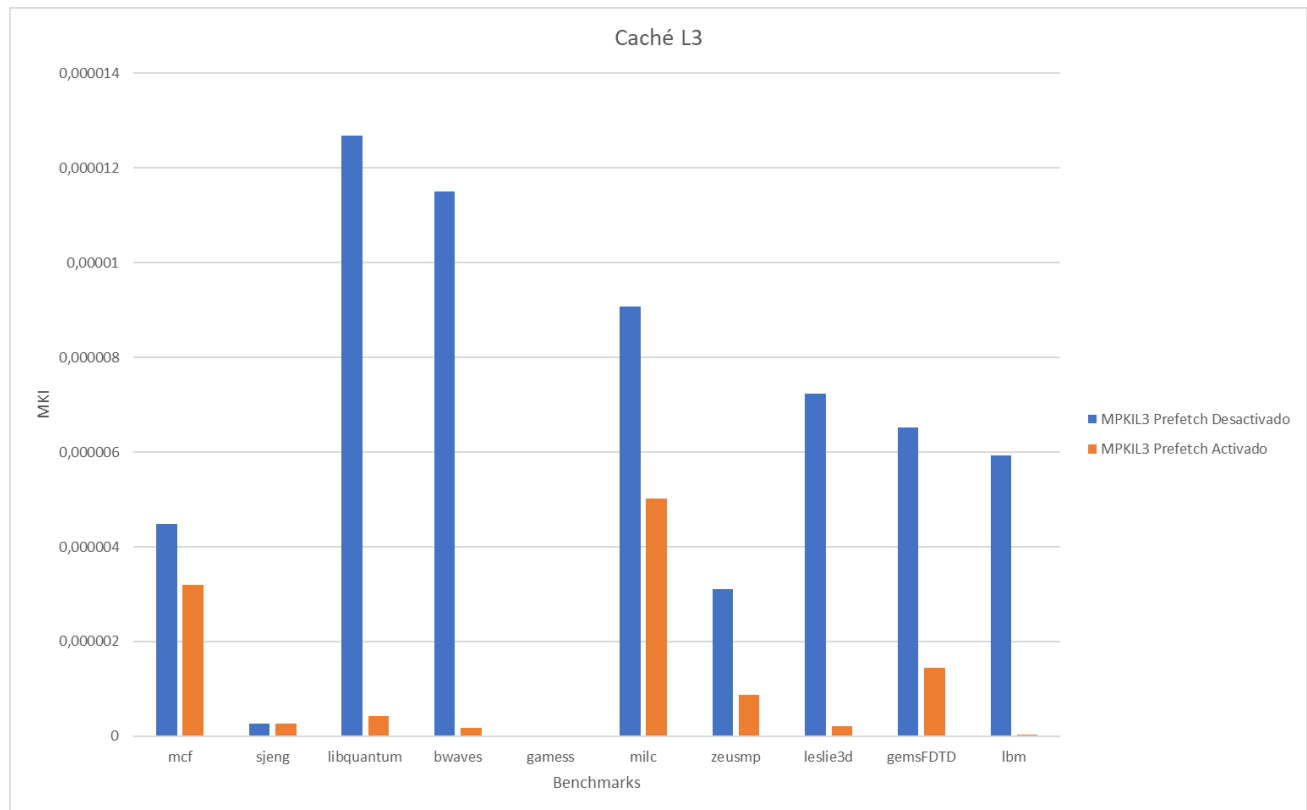
Además, realizamos más accesos L1, si el prefetch funciona correctamente, nos dará muchos más aciertos. Mucho más acierto significa, mayor diferencia en el número de paradas que antes.

Actividad 3

Análisis de las características de las aplicaciones que más se le benefician de los mecanismos de prebúsqueda

```
1  #!/bin/bash
2
3  ##### Variable a editar #####
4
5  EVENTS="mem_load_uops_retired:l1_miss,mem_load_uops_retired:l2_miss,mem_load_uops_retired:l3_miss"
6
7  #####
8
9
10 Benchmark_list=('perlbench' 'gcc' 'bzip2' 'mcf' 'gobmk' 'hmmer' 'sjeng' 'libquantum' 'h264ref' 'omnetpp' 'astar' 'xalancbmk'
11 'bwaves' 'gameess' 'milc' 'zeusmp' 'gromacs' 'cactusADM' 'leslie3d' 'namd' 'dealII' 'soplex' 'povray' 'gemsFDTD' 'lbm')
12
13 sal=$(mktemp)
14 sal2=$(mktemp)
15
16
17 events_string=$(echo $EVENTS | sed 's/,/ /g')
18
19 echo "Benchmark Cycles Instructions" $events_string
20
21 for bench in 3 6 7 12 13 14 15 18 23 24; do
22
23     ./monitoriza_aplicacion -W $bench -C 0 -e $EVENTS > $sal 2> $sal2
24
25     cyc=$(cat $sal2 | grep "Final_count" | awk '{print $5}')
26     ins=$(cat $sal2 | grep "Final_count" | awk '{print $7}')
27
28     ev1=$(cat $sal2 | grep "Final_count" | awk '{print $9}')
29     ev2=$(cat $sal2 | grep "Final_count" | awk '{print $11}')
30     ev3=$(cat $sal2 | grep "Final_count" | awk '{print $13}')
31     ev4=$(cat $sal2 | grep "Final_count" | awk '{print $15}')
32
33     echo ${Benchmark_list[$bench]} $cyc $ins $ev1 $ev2 $ev3 $ev4
34 done
```





A partir de esta observación, ¿qué comportamiento, desde el punto de vista de la jerarquía de cache, crees que debe presentar una aplicación para beneficiarse de la prebúsqueda?

Para mejorar el rendimiento de la memoria en estos programas es aplicar la técnica de prefetch en los accesos a objetos de grandes dimensiones. Para que esta técnica sea efectiva, es necesario anticipar las próximas referencias a memoria, para cargar esas páginas con antelación y de forma paralela con el cálculo y, de esta manera reducir el número de fallos de página de los programas.

¿Qué características presentan los MPKIs (misses per kilo instructions) de las aplicaciones que se ven más y menos beneficiadas por los mecanismos de prebúsqueda?

En las aplicaciones más beneficiadas por los mecanismos de prebúsqueda reducimos considerablemente los fallos en las caches de todos los niveles, además el IPC aumenta con la prebúsqueda activada, y podemos decir que son las aplicaciones que más accesos a memoria realizan ya que en la actividad 2 hemos visto como se veían los ciclos debidos a memoria y LLC que eran los que más se aumentaban. En las que se ven menos beneficiadas vemos que ocurre el caso contrario.