CS422 Assignment-4 Report

Ketan Chaturvedi (190428)

November 19, 2021

This report consists analysis of various cache replacement policies on the **SPEC 2006** benchmark application suite. PIN tool is used to collect a trace of addresses touched by each application. For each application, memory accesses in 1 billion instructions are analysed after fast-forwarding them by a specified amount. Following benchmark applications are analysed:

- Perlbench
- GCC
- Soplex
- Omnetpp
- BZip2
- MCF
- Hmmer
- Xalancbmk

Following cache replacement policies are evaluated:

1. **Least Recently Used(LRU)**
   When a eviction is to be done from a set, the least recently used cache block in the set is freed. Drawback of this policy is that before getting evicted after a hit, the cache block will have to climb down from *Most Recently Used (MRU)* position to *LRU* which can take a significant time.
2. **Static Re-Reference Interval Prediction(SRRIP)**
   In this policy, each cache block is associated with an age. When the block is filled in cache, it is assigned an age of 2. Age is updated to 0 on a hit. Block gets evicted at age 3. If there is no such block in the set, the ages of all blocks are incremented until there is at least one block with age 3. If there are multiple blocks with age 3, the block with lowest way id is evicted. The age is referred as *Re-Reference Prediction Value (RRPV).*
3. **Not Recently Used(NRU)**
   The NRU policy requires only 1 bit per cache block known as *reference (REF) bit.* When block is filled, REF is set to one. On a hit, REF is updated to 1. If in any of the aforementioned cases, all blocks in the cache set have REF bits set, the REF bits of all blocks in that set except the most recently accessed block are reset to zero. The replacement algorithm evicts a block with REF bit zero. If there are multiple blocks with REF bit zero, the block with the lowest way id among these is evicted.

Two level inclusive L1-L2 cache hierarchy is used to simulate these cache policies.

- **L1 Cache**
  Size: 64KB
  Block size: 64 bytes
  Associativity: 8
- **L2 Cache**
  Size: 1MB
  Block size: 64 bytes
  Associativity: 16

The result tables consists of 4 rows. First two are cache accesses and misses respectively. Third one contains dead-on-fill % calculated as:

$$\frac{Dead\ on\ fill\ blocks}{Total\ blocks\ filled\ in\ cache} \times 100$$

Last row contains following:

$$\frac{Blocks\ with\ atleast\ 2\ hits\ before\ eviction}{Blocks\ with\ atleast\ 1\ hit\ before\ eviction} \times 100$$

Perl benchmark

Fast-forward count = 207000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	594417671	945669
Cache misses	945669	16708
Dead on fill %	5.29	7.73
Atleast 2 hits %	89.49	44.19

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	594417671	945669
Cache misses	945669	16707
Dead on fill %	5.29	7.98
Atleast 2 hits %	89.49	0.00

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	594417671	945671
Cache misses	945671	16740
Dead on fill %	5.29	7.57
Atleast 2 hits %	89.49	57.14

Observations:

1. All the 3 caches have almost comparable performance. The huge drop in percentage for $L1(LRU) + L2(SRRIP)$ is misleading. If we look at actual numbers, 44.19 for $L1(LRU) + L2(LRU)$ is calculated as $\frac{19}{43} \times 100$. So the difference in actual numbers is not that much.

BZip2 benchmark

Fast-forward count = 301000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	705156390	8745705
Cache misses	8745705	4543891
Dead on fill %	48.26	67.21
Atleast 2 hits %	79.47	52.16

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	705156390	8745751
Cache misses	8745751	4594175
Dead on fill %	48.26	74.70
Atleast 2 hits %	79.47	62.14

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	705156390	8745773
Cache misses	8745773	4565068
Dead on fill %	48.26	67.11
Atleast 2 hits %	79.47	51.37

Observations:

1. L2 cache with SRRIP has very high misses. As dead on fill is not very large for LRU, it means some blocks are required in the near future. In SRRIP we are assigning the new blocks an age of 2, so they are quickly evacuated resulting in high misses.

GCC benchmark

Fast-forward count = 107000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	524752647	23197728
Cache misses	23197728	5067194
Dead on fill %	2.65	94.82
Atleast 2 hits %	96.70	49.85

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	524752647	23197678
Cache misses	23197678	5057109
Dead on fill %	2.65	95.15
Atleast 2 hits %	96.70	54.60

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	524752647	23197726
Cache misses	23197726	5071403
Dead on fill %	2.65	94.79
Atleast 2 hits %	96.70	50.04

Observations:

1. As dead on fill is very high for LRU, very few cache blocks are required again in near future and LRU fails because climbing down from MRU to LRU takes time. But in SRRIP, we assign the age 2 to new blocks. So they are removed quickly preventing the eviction of necessary blocks.

MCF benchmark

Fast-forward count = 377000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	539805102	68833401
Cache misses	68833401	33574062
Dead on fill %	44.99	60.98
Atleast 2 hits %	91.63	18.18

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	539805102	68833774
Cache misses	68833774	34477694
Dead on fill %	44.99	66.45
Atleast 2 hits %	91.63	18.39

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	539805102	68833655
Cache misses	68833655	33800264
Dead on fill %	44.99	60.84
Atleast 2 hits %	91.63	18.54

Observations:

1. SRRIP misses are very high. As the blocks which gets hits twice before evicting are quite less, and dead on fill is not very high, SRRIP fails. On a hit, SRRIP changes the age to 0. But the chances of hit again are quite less and it ends up evicting blocks which haven't received a hit but going to receive a hit in near future.

Soplex benchmark

Fast-forward count = 364000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	534666860	19416704
Cache misses	19416704	18679822
Dead on fill %	1.67	97.89
Atleast 2 hits %	93.42	22.39

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	534666860	19416718
Cache misses	19416718	18625941
Dead on fill %	1.67	98.18
Atleast 2 hits %	93.42	33.01

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	534666860	19416680
Cache misses	19416680	18683245
Dead on fill %	1.67	97.88
Atleast 2 hits %	93.42	22.76

Observations:

1. In this case SRRIP has fewer misses. The dead on fill are very large and the explanation for SRRIP few miss is same as that for GCC benchmark application.

Hmmer benchmark

Fast-forward count = 264000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	658129704	3265978
Cache misses	3265978	1608113
Dead on fill %	0.45	95.55
Atleast 2 hits %	99.27	30.48

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	658129704	3265983
Cache misses	3265983	1550968
Dead on fill %	0.45	96.47
Atleast 2 hits %	99.27	46.60

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	658129704	3266046
Cache misses	3266046	1591149
Dead on fill %	0.45	94.58
Atleast 2 hits %	99.27	28.28

Observations:

1. In this case SRRIP has fewer misses. The dead on fill are very large and the explanation for SRRIP few miss is same as that for GCC benchmark application.

Omnetpp benchmark

Fast-forward count = 43000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	588617541	14142620
Cache misses	14142620	10524591
Dead on fill %	19.29	81.10
Atleast 2 hits %	81.56	36.65

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	588617541	14141832
Cache misses	14141832	10508774
Dead on fill %	19.29	85.28
Atleast 2 hits %	81.56	49.40

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	588617541	14142806
Cache misses	14142806	10536317
Dead on fill %	19.29	80.84
Atleast 2 hits %	81.56	35.74

Observations:

1. SRRIP has fewer misses. The dead on fill are very large and the explanation is same as that for GCC benchmark application. As dead on fill is not that high as compared to GCC, hit count difference(b/w LRU and SRRIP L2 cache) is not that high in this case as compared to GCC.

Xalancbmk benchmark

Fast-forward count = 1331000000000

L1 (LRU) + L2 (LRU)

	L1 Cache	L2 Cache
Cache accesses	568615118	16172039
Cache misses	16172039	2423120
Dead on fill %	14.54	49.58
Atleast 2 hits %	37.23	62.84

L1 (LRU) + L2 (SRRIP)

	L1 Cache	L2 Cache
Cache accesses	568615118	16172156
Cache misses	16172156	2329596
Dead on fill %	14.54	56.14
Atleast 2 hits %	37.23	65.75

L1 (LRU) + L2 (NRU)

	L1 Cache	L2 Cache
Cache accesses	568615118	16172042
Cache misses	16172042	2429066
Dead on fill %	14.54	49.94
Atleast 2 hits %	37.23	62.42

Observations:

1. In this case SRRIP has fewer misses. The dead on fill is not very large and still SRRIP wins. The reason might be, as the blocks with atleast 2 hits is high, blocks are required again in near future. And in SSRIP, blocks might be getting a hit before getting evicted making their age come down to 0.