December 11, 2021

The University of Texas at Austin

EVALUATION OF LRU v/s SRRIP CACHE REPLACEMENT POLICY

GRAD LAB PART - 2

Nithyashree Srinivasan (ns32928)
Shreyas Ravishankar (sr48925)
Abhijjith Venkkateshraj (av36677)

Cache Replacement Policies

- Cache Replacement decides which cache line are to be replaced in set associative and fully associative caches.
- Commonly used policies-
    - LRU (Least Recently Used)
    - NRU (Not Recently Used)
    - FIFO
    - Random
    - RRIP

Types of Access patterns and Why LRU doesn’t perform well

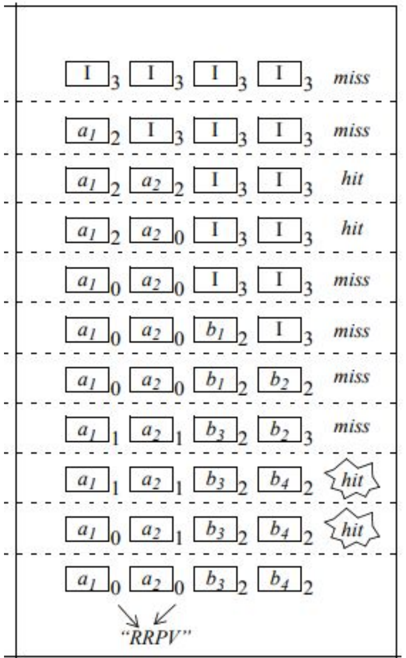
- **Recency-friendly access patterns**- Access pattern that repeats N times. LRU works well in this case.
- **Thrashing access patterns**- Cyclic access pattern of length K which repeats N times. K> no. of cache blocks → LRU performs very badly (0 hit rate).
- **Streaming access patterns**- Patterns with infinite re-reference interval. No hits in any replacement policy, so LRU doesn’t matter.
- **Mixed access patterns**- A mix of scan and frequently referenced working set. LRU fails to preserve the working set after a scan.

RRIP

- Based on re-reference interval prediction.
- LRU-MRU block will be re-referenced sooner, and the block at LRU position will be referenced last.
- **Optimal algorithm**- Predicts distance re-reference interval for scan type data and near re-reference interval for working set/frequently used data.
- RRIP prevents scan blocks from replacing the working set
- SRRIP - scan resistant
- DRRIP- scan resistant and thrash resistant

SRRIP (Static RRIP)

- Overhead- M bit register holding RRPV bits (M bit saturating counter) for each cache line.
- Two types
  - HP- Change RRPV = 0 on a hit
  - FP- Change RRPV = RRPV - 1 on a hit
- The figure on the right shows RRIP-HP with a 2 bit saturating counter (M=2)


I 3 I 3 I 3 I 3 miss
a_1 2 I 3 I 3 I 3 miss
a_1 2 a_2 2 I 3 I 3 hit
a_1 2 a_2 0 I 3 I 3 hit
a_1 0 a_2 0 I 3 I 3 miss
a_1 0 a_2 0 b_1 2 I 3 miss
a_1 0 a_2 0 b_1 2 b_2 2 miss
a_1 1 a_2 1 b_3 2 b_2 3 miss
a_1 1 a_2 1 b_3 2 b_4 2 hit
a_1 0 a_2 1 b_3 2 b_4 2 hit
a_1 0 a_2 0 b_3 2 b_4 2
"RRPV"

(c) 2-bit SRRIP with Hit Promotion

Cache Hit:

(i) set RRPV of block to '0'

Cache Miss:

(i) search for first '3' from left
(ii) if '3' found go to step (v)
(iii) increment all RRPVs
(iv) goto step (i)
(v) replace block and set RRPV to '2'

Methodology and Configuration

- Base configuration
  - **1st level**
    - I-Cache 32KB, Associativity 4
    - D-Cache 32KB, Associativity 4
  - **2nd level**
    - LLC 1MB, Associativity 8
  - Prefetcher on
- Evaluation- SRRIP and LRU are compared across various LLC configurations-
  - Fixed Associativity (8) with varying cache sizes 256KB, 512KB, 1MB, 2MB, 4MB, 8MB.
  - Fixed Cache Size (1M) with varying associativity 2,4,8,16.
  - Prefetch vs No-Prefetch with Associativity (16) and Cache Size (1M).

Code Changes

Cache_lib.h

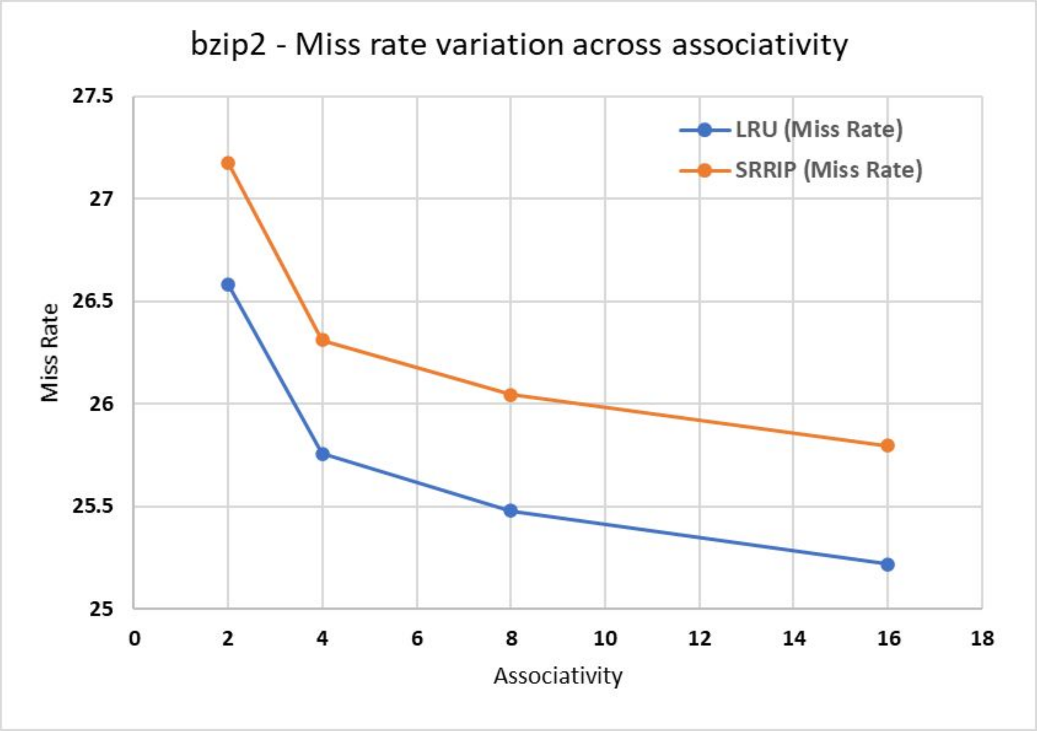
- Cache_Entry_struct- Added rrpv bits
- Added SRRIP replacement to
  - Repl_Policy_enum
  - Cache_Insert_Repl_enum

Cache_lib.c

- Added hit logic for SRRIP to cache_access function.
- Added miss logic for SRRIP find_repl_entry function to find the entry to be replaced.
- Insertion logic is placed in cache_insert_replpos function.

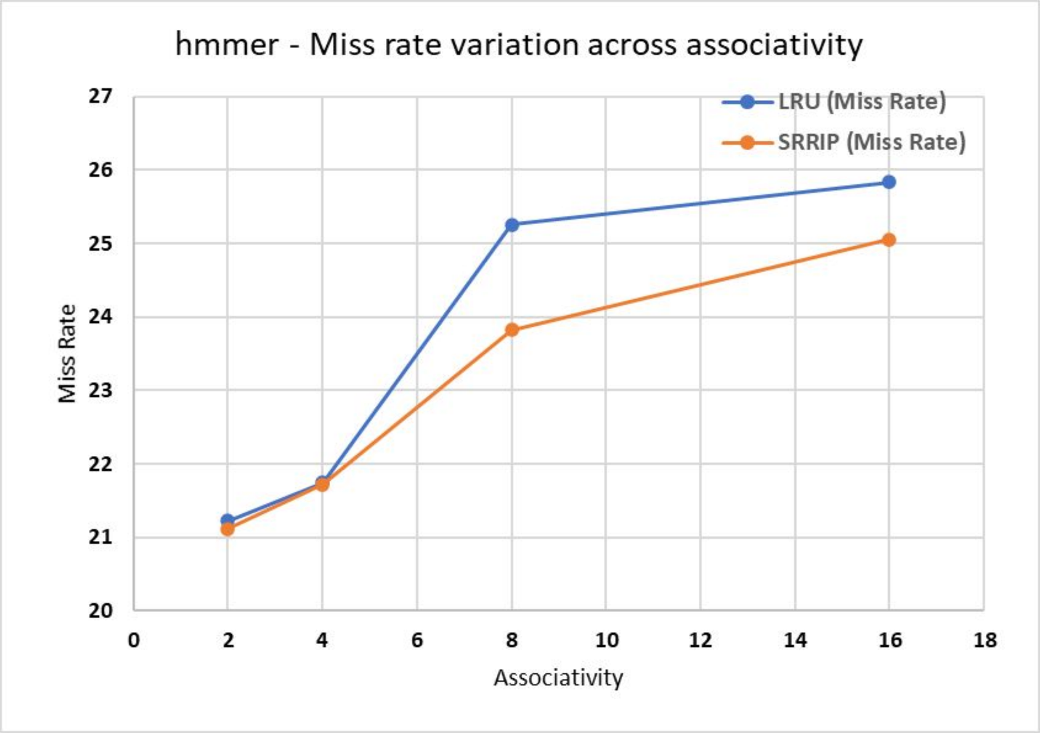
RESULTS

BZIP2- Miss rate variation across associativity



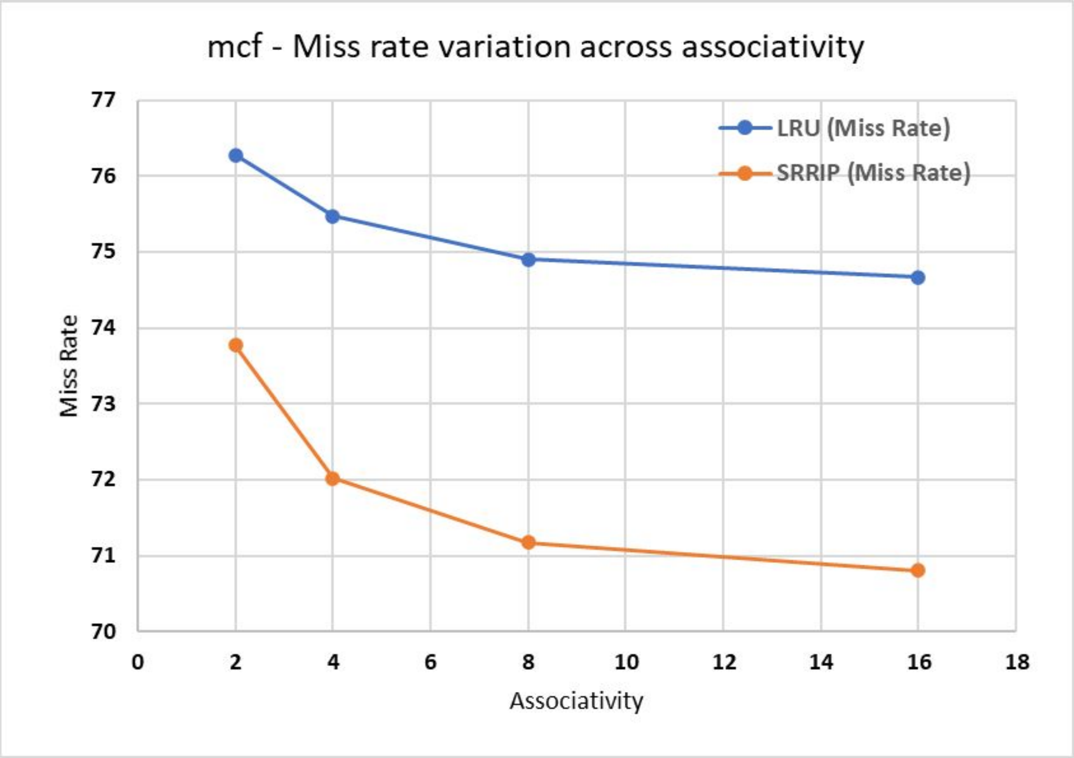
- Miss rate increased from LRU to SRRIP.

HMMR- Miss rate variation across associativity



- Miss rate decreased from LRU to SRRIP.

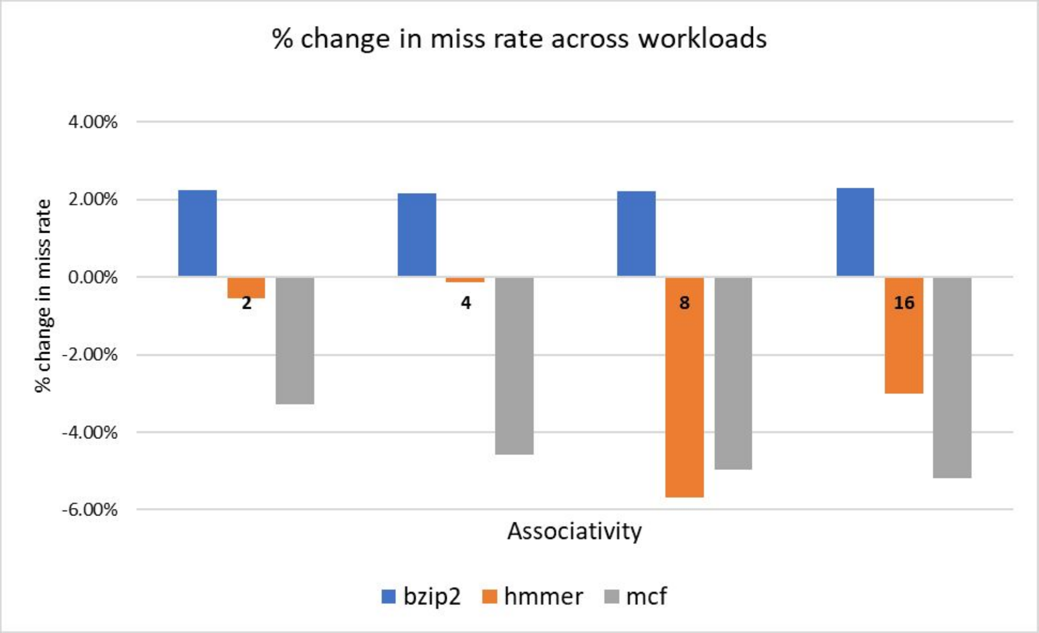
MCF- Miss rate variation across associativity



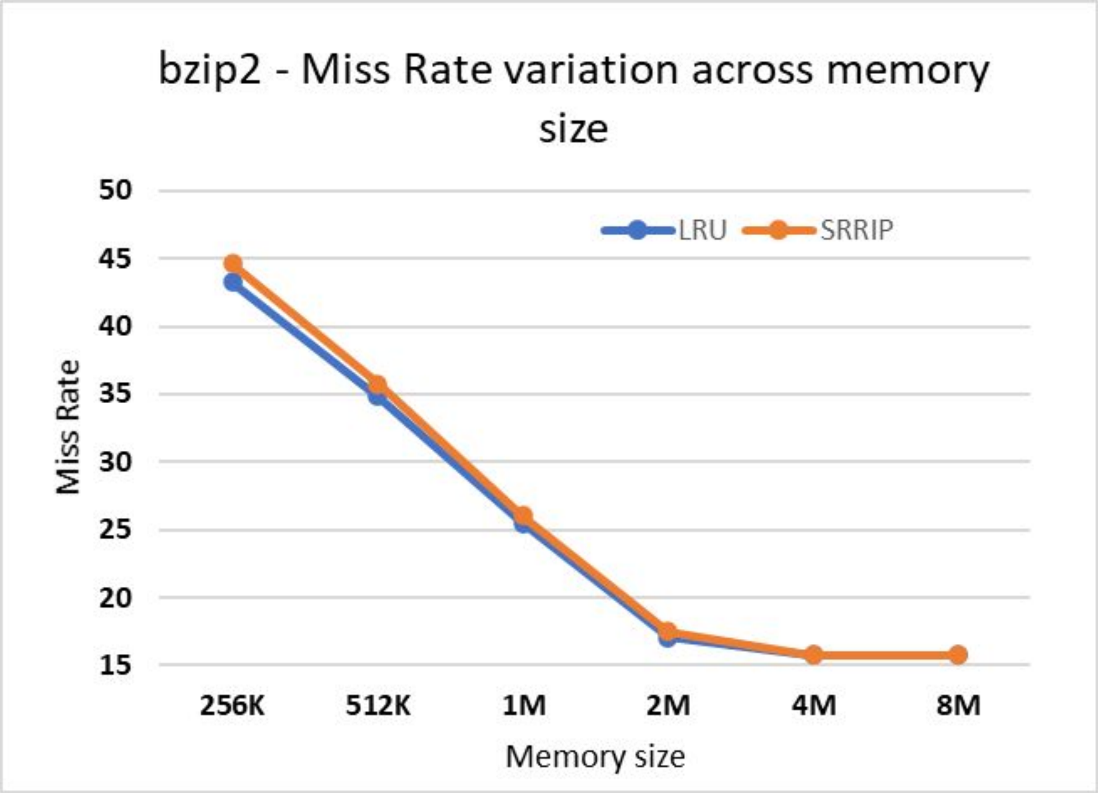
- Significant reduction in the miss rate across all associativities.

Variation of % change in miss rate with associativity

- The graph shows that RRIP doesn’t show any benefits for BZIP benchmark.
- The HMMR benchmark shows some benefit in miss rate, which increases with associativity.
- The MCF workload shows the maximum benefits for SRRIP

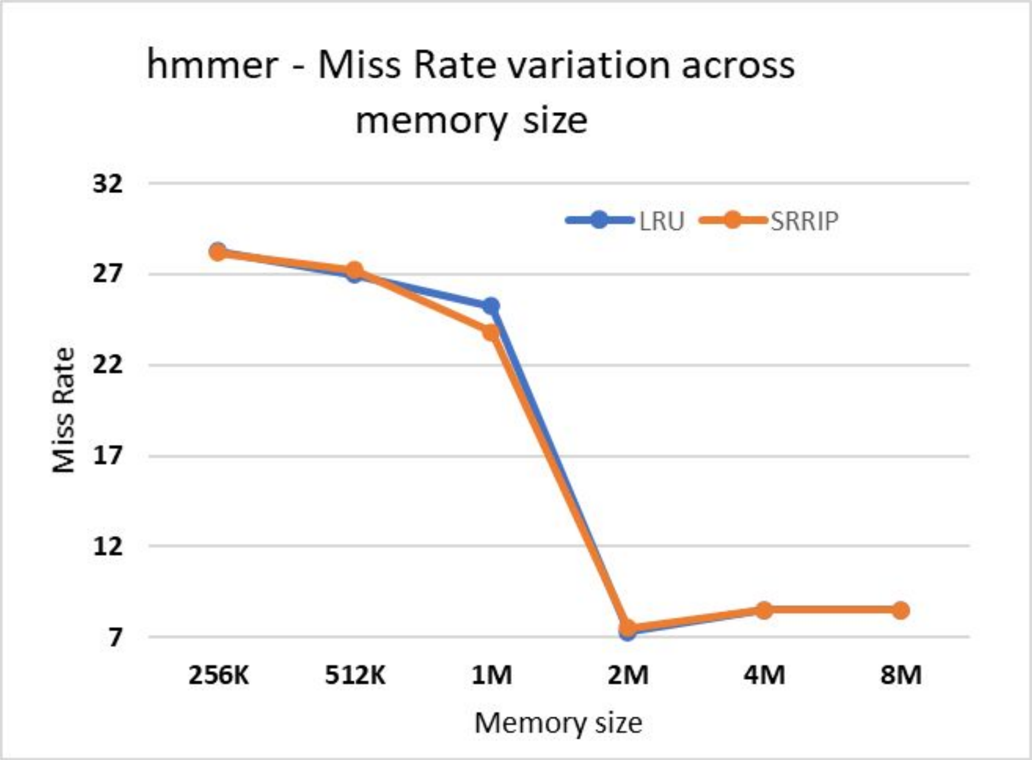


BZIP2- Miss rate variation across memory size



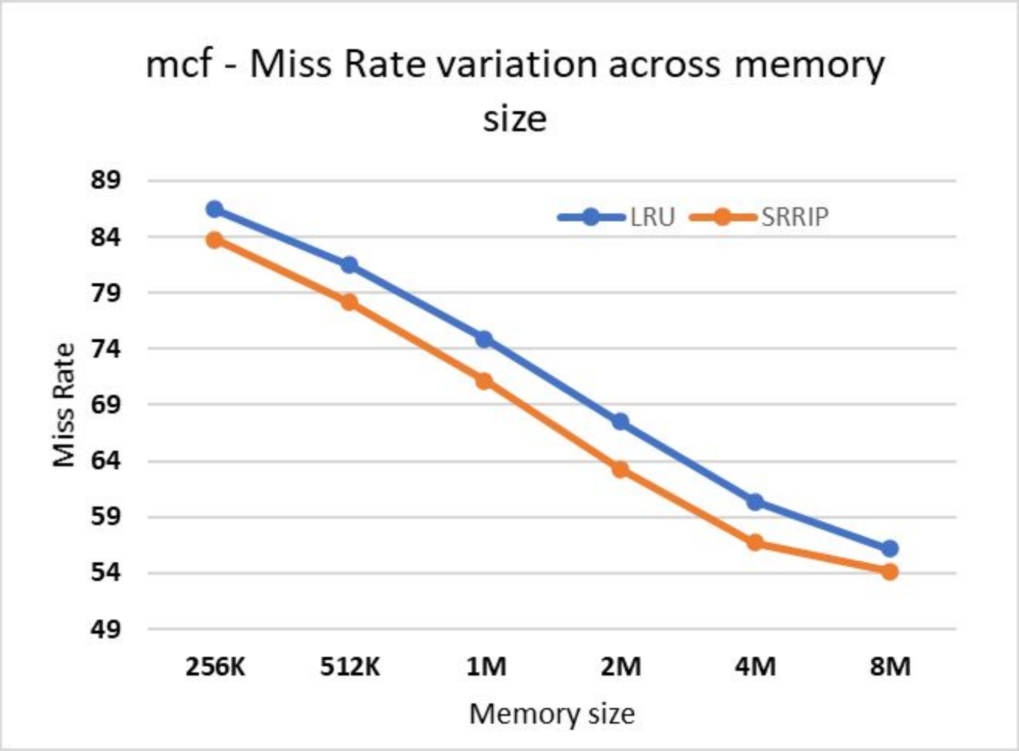
- SRRIP doesn’t show much improvement for this workload.

HMMR- Miss rate variation across memory size



- We don’t see a significant difference in the miss rate.

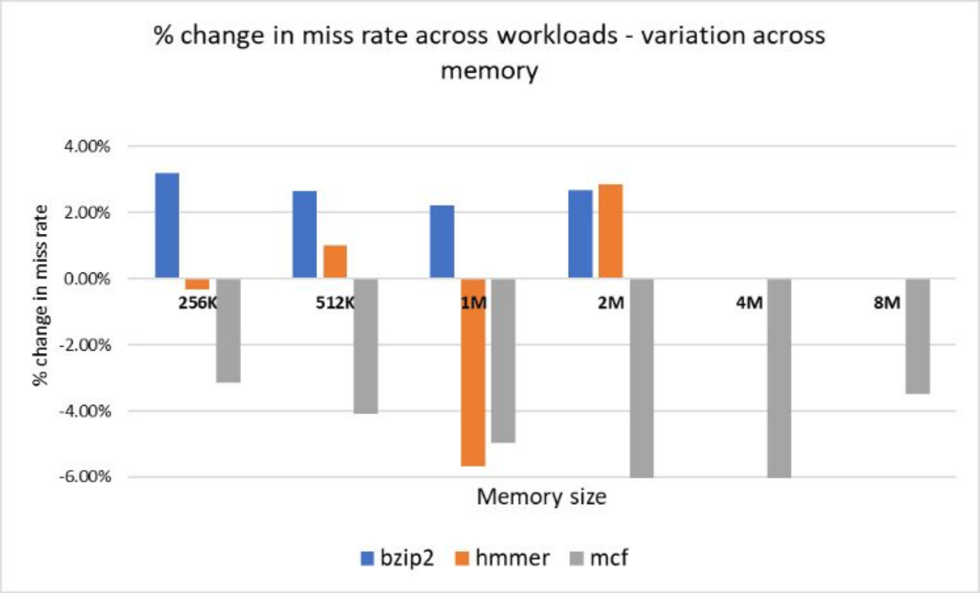
MCF- Miss rate variation across memory size



- There is a steady decrease in miss rate for mcf when we vary the memory size.

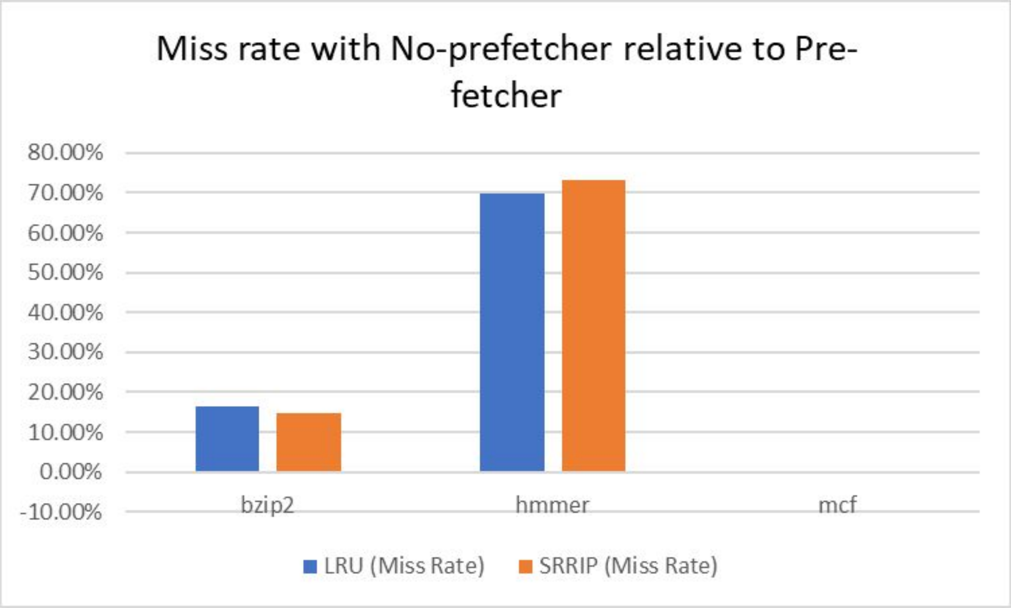
Variation of % change in miss rate with cache size

- BZIP doesn't show any improvement across sizes.
- HMMR shows fluctuation with cache size. So the behaviour is erratic.
- MCF shows a reduction in miss rate till 4MB and then the benefits begin to reduce for greater cache sizes.



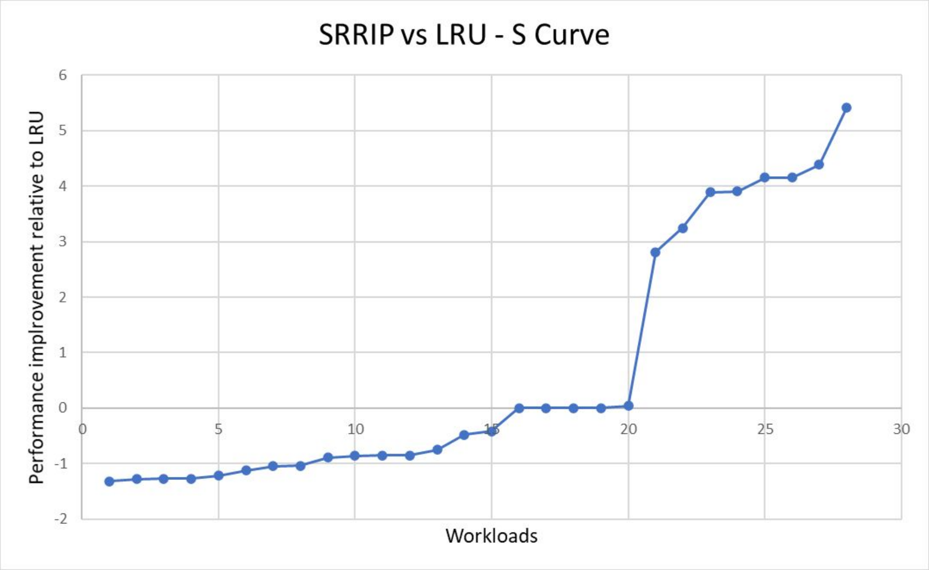
Effect of prefetcher

- As expected the prefetcher prevents compulsory misses.
- So for BZIP and HMMR, the prefetcher decreases the miss rate for both LRU and SRRIP.
- MCF doesn’t have any change, which might mean that compulsory misses could not predicted by the prefetcher.



S-curve

- For the 30 datapoints we analysed, most of the datapoints that show negative performance benefit are from BZIP.
- HMMER shows zero to little benefit, while MCF shows the highest benefit with SRRIP policy as compared to RRIP.



Learnings

- Cache replacement policy is affected by -
  - Workload- Relative size of workload to cache size
  - Memory access pattern in a workload
  - Associativity

References

- https://www.spec.org/cpu2006/.
- Nair AA, John LK, "Simulation points for SPECCPU 2006", Computer Design 2008, ICCD 2008, Pgs 397-403.
- A. Jaleel, K. Theobald, S. C. Steely, and J. Emer, “High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP),” in ISCA-32, 2010.