CS231 Lab 4(Part 2) Report
ChampSim

Arohan Hazarika

1 Replacement Policies

1.1 FIFO AND LRU

To make FIFO work, adjust the LRU code by changing the conditions in the update replacement state. Specifically, remove the condition that skips hits for writebacks, as in FIFO, you're only interested in timestamps when introducing something new.

Now, regarding the rationale for skipping writebacks in the LRU policy for the L2C cache: There are two main approaches—write back and write through. In the case of write through, any change to a cache line. On the other hand, write back updates the next memory level only when evicting a cache line.

1.2 LFU AND BIP

For LFU, I have basically used the concept of frequency, instaed of storing the current cycle in the map, I am storing the frequency of each cache line and increment it on every hit whereas at every miss, I make the frequency zero.

For BIP, I have made the current cycle value either 0 (lru) or 1 (mru), this ensures the existence of two states, implemented the policy accordingly, if we have a hit in lru then change it to mru and if there is a miss, use epsilon to choose either mru or lru cache line for the incoming address. For implementing it, I have used Bernoulli Distribution. One more thing is that when all cycle values reach 1 (mru), then I make every value 0 (lru) which is kind of a reset.

1.3 OBSERVATIONS

n the 602 trace, FIFO shows the highest miss rate, while LRU and all BIPs exhibit the lowest values. This discrepancy may arise from the fact that FIFO tends to discard data elements prematurely, as it doesn't consider the time intervals between accesses. Conversely, LRU and BIPs prioritize recency of use, preventing premature eviction.

Examining the 603 trace, both LFU and BIP (e=0) demonstrate the highest miss rates. This outcome can be attributed to the possibility that certain information is heavily utilized initially in large amounts but becomes obsolete over time. LFU tends to retain such data, causing space wastage. BIP (e=0), acting akin to LRU, evicts without considering usage patterns, leading to potential indiscriminate evictions.

In the 619 trace, LRU outperforms LFU, which exhibits the worst miss rate. This discrepancy may be attributed to a scenario where a small set of data is frequently accessed initially but is never utilized in the future. LRU effectively identifies and evicts these rarely used entries, whereas LFU retains them based on their initial high access frequency.

Speed Ups are almost same for every policy.

2 Stream Prefetcher

Graphs have been plotted and have been shown below.

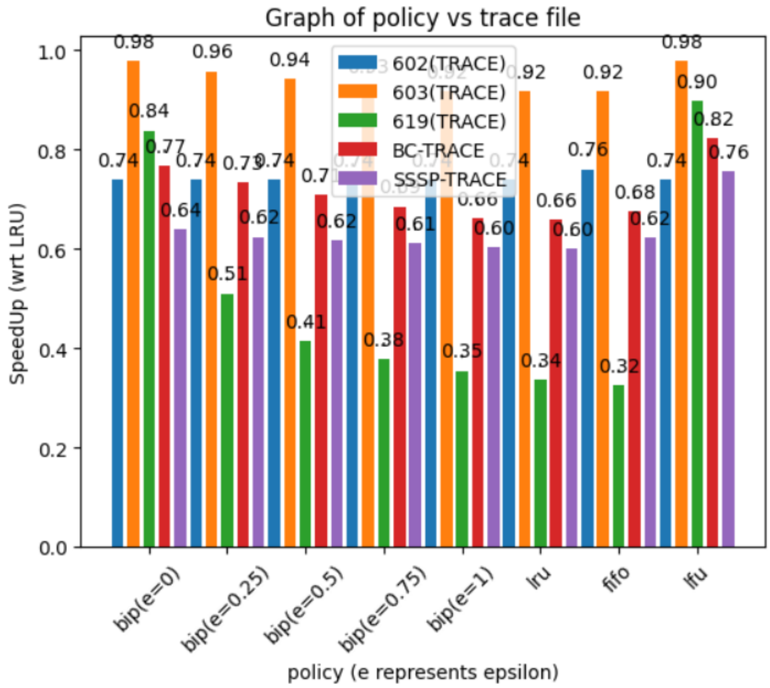

Figure 1: Miss rate graph

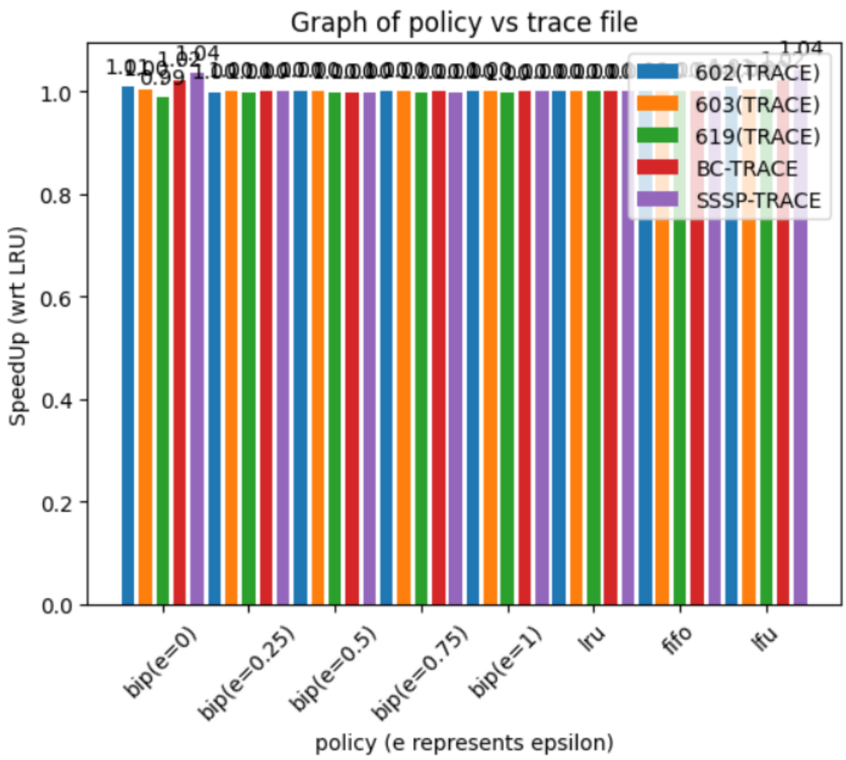

Figure 2: Speed up graph

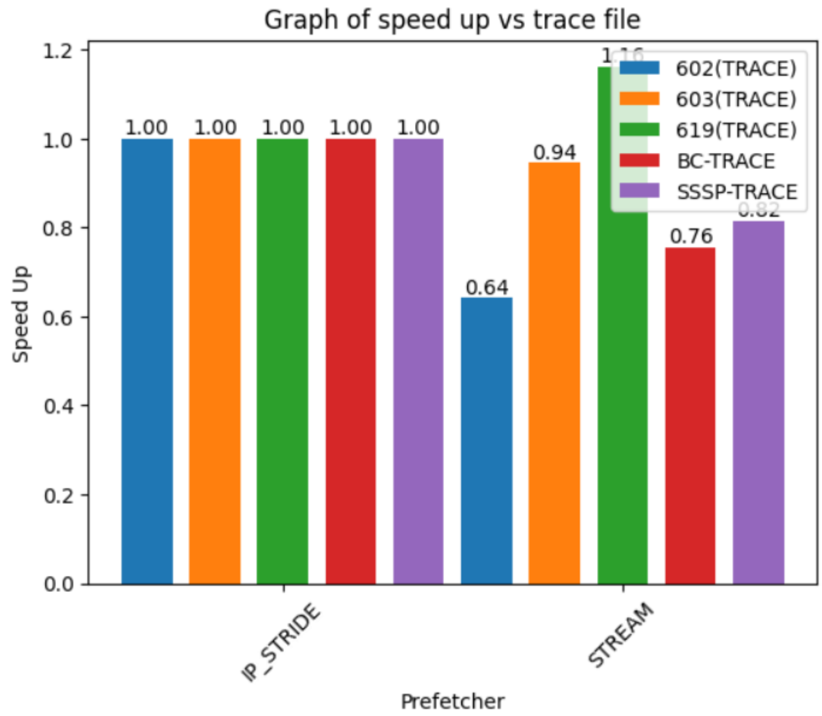

Figure 3: Speed up graph for Prefetcher

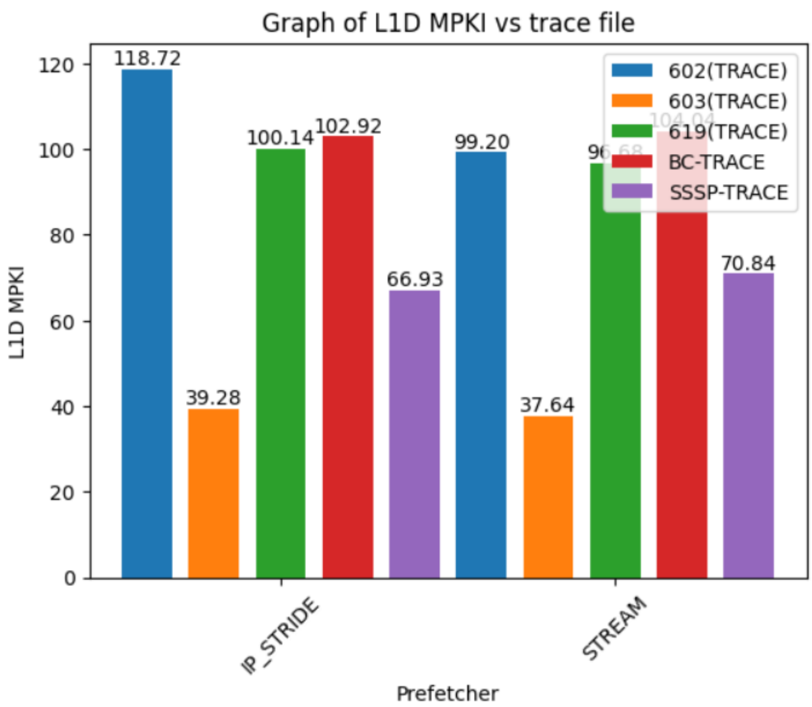

Figure 4: L1D MPKI Graph

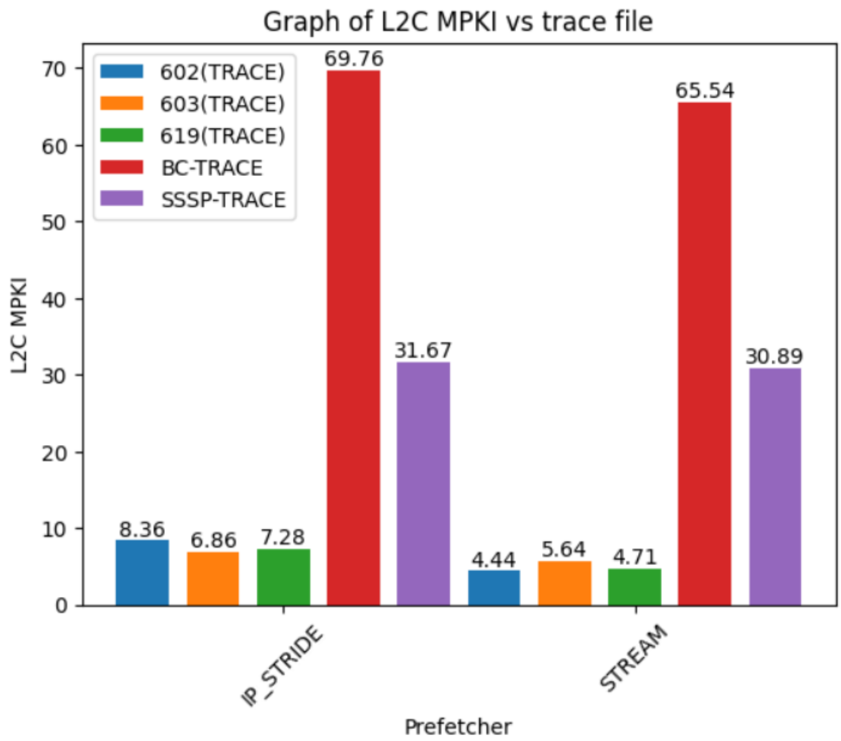

Figure 5: L2C LOAD MPKI

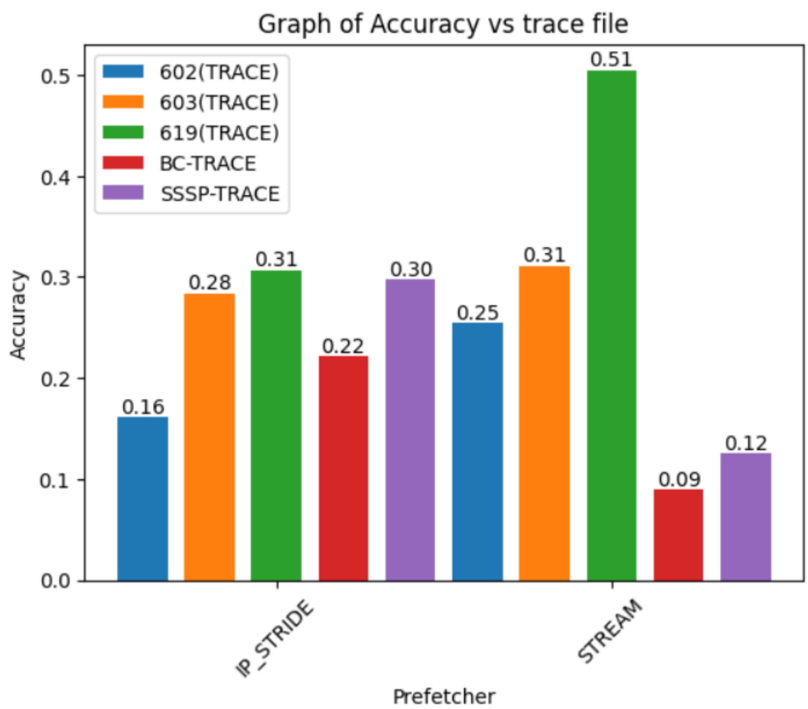

Figure 6: Accuracy graph for Prefetcher