Project 1 Report

Shruti Shivakumar

February 19, 2018

1 Introduction

We need to find the best configuration of $(C, B, S, 100, t)$ such that the average access time (AAT) is the lowest for each of the four traces within the given storage budget. Now,

$$\text{AAT} = \text{HT} + \text{MR} \times \text{MP}$$

It is given that miss penalty is a constant, and that AAT is directly proportional to S, since HT increases linearly with associativity. This makes sense because larger the associativity, larger are the number of tag comparisons per set in the cache to decide if the given access is a hit or a miss. However, decreasing associativity also results in increasing number of conflict misses, which in turn increases miss rate. We also know that miss rate decreases with increasing data storage in the cache i.e. MR decreases with increasing C.
Given a storage budget of 2KB, we have

$$2^{C-B-3}(66 - C + S) + 2^C = 2^{11}$$

Now, the best miss rate can be obtained if the cache size is the largest possible within the given budget. Thus, $C = 10$. Block size B is another parameter that affects AAT. If the block size is too small, we are not taking advantage of spatial locality. On the other hand, large block sizes reduce the number of blocks in the cache, resulting in frequent capacity (since there are too few blocks in each set) and conflict misses (since there are fewer sets). Also, $0 \leq S \leq C - B$.
Now, since we have fixed C, we need to iterate over the space of (B, S) for each of the three kinds of prefetchers to get the best configuration, as done by `best_config.sh` script. The results of all iterations was tabulated in the `aat.xlsx` workbook in the `Reports` directory and histograms were plotted to compare the AAT across configurations. Note that the case of no prefetcher has not been included in the analysis since lack of prediction of next likely address will definitely result in more misses, thereby resulting in higher AAT. Also, $B \leq C - 3$ and $S \geq 1$. Also, the validation outputs for checkpoint 2 is in `output_CP2` directory.

1.1 mcf

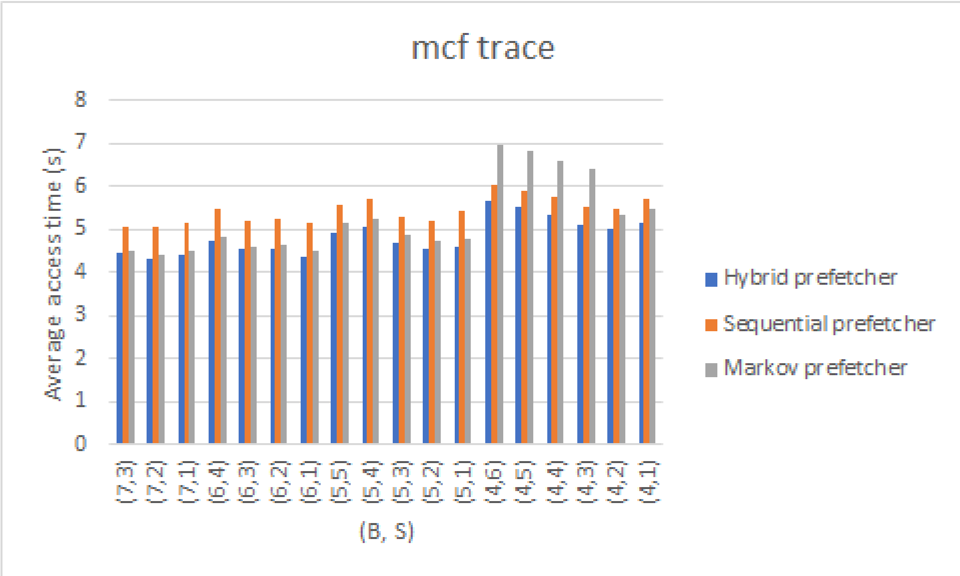

Figure 1: Plot of AAT against different (B, S) configs for the three types of prefetchers.

From the graph, it is clear that the hybrid prefetcher performs the best, followed by Markov prefetcher (when block sizes aren't too small). The best AAT of 4.302 seconds was obtained for the configuration $(C, B, S, 100, t) = (10, 7, 2, 100, 3)$.

1.2 perlbench

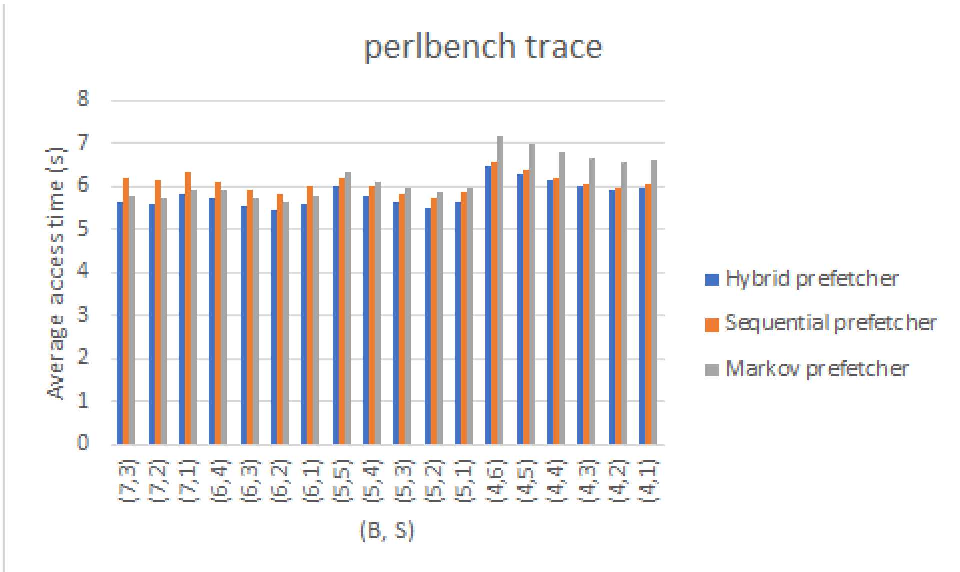

Figure 2: Plot of AAT against different (B, S) configs for the three types of prefetchers.

From the graph, it is clear that the hybrid prefetcher performs the best, followed by Markov prefetcher (when block sizes aren't too small). The best AAT of 5.437 seconds was obtained for the configuration $(C, B, S, 100, t) = (10, 6, 2, 100, 3)$.

1.3 astar

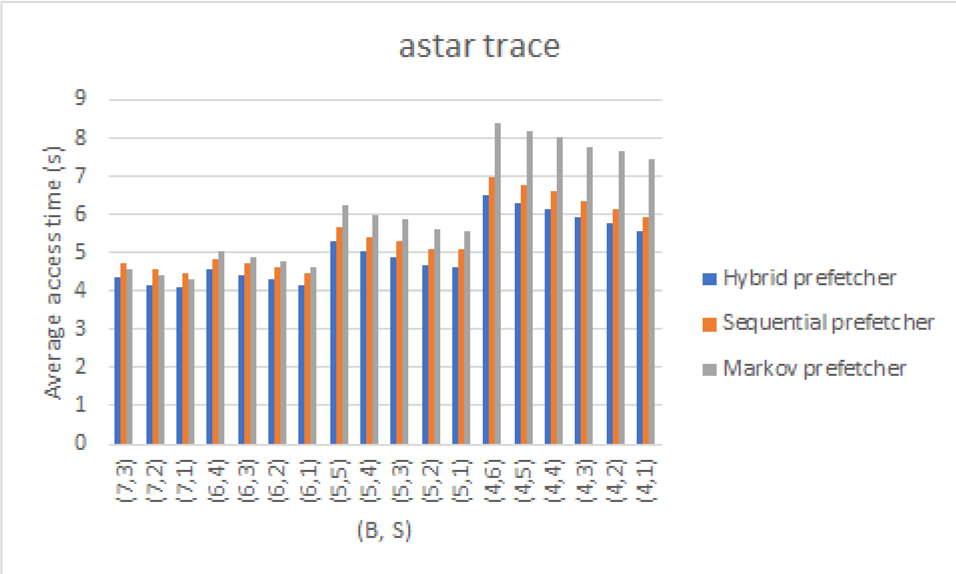

Figure 3: Plot of AAT against different (B, S) configs for the three types of prefetchers.

From the graph, it is clear that the hybrid prefetcher performs the best, followed by sequential prefetcher (unless when block sizes are too large). The best AAT 4.067 seconds was obtained for the configuration $(C, B, S, 100, t) = (10, 7, 1, 100, 3)$.

1.4 bzip2

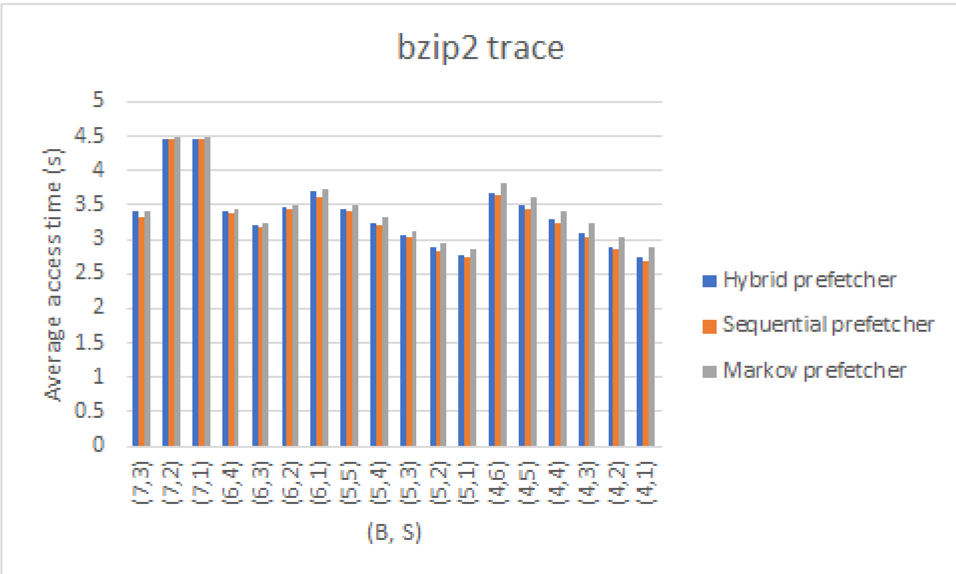

Figure 4: Plot of AAT against different (B, S) configs for the three types of prefetchers.

From the graph, it is clear that the sequential prefetcher performs the best, followed by hybrid prefetcher (though the difference between hybrid and markov prefetcher is very small). The best AAT of 2.734 seconds was obtained for the configuration $(C, B, S, 100, t) = (10, 5, 1, 100, 2)$.