# Cache Simulation Programming Report

Name: Sai Vittal Battula ID: 18XJ1A0238

## GitHub repo link - <https://github.com/saivittalb/cache-simulation-programming>

The above-listed link will be active and visible only 15 days after the deadline.

It is to be noted that our prefetch model is designed given the Cache Design Competition. Our main model for the assignment is the non-prefetch model.

## Instructions to run the code:

Included in the README.md file.

## Results:

Our simulator runs through 100% of the trace file and generates results in seconds. There are no performance issues.

Graphs are attached here as well as included in the *plots* directory. Result data can be found in an excel file with a separate worksheet for each trace in the *exported-data* directory.

For the non-prefetch models, for adpcm.trace and crc32.trace, two different configurations were used to plot the graphs. But from there on, to reduce the workload, only one configuration is used to plot.

Since there are many permutations possible to the practical configurations, and there isn’t much time, plots were carefully chosen for the best possible configurations. Each plot (for both, with and without prefetch) showcases a different configuration. No two same configurations were used to plot the graphs.

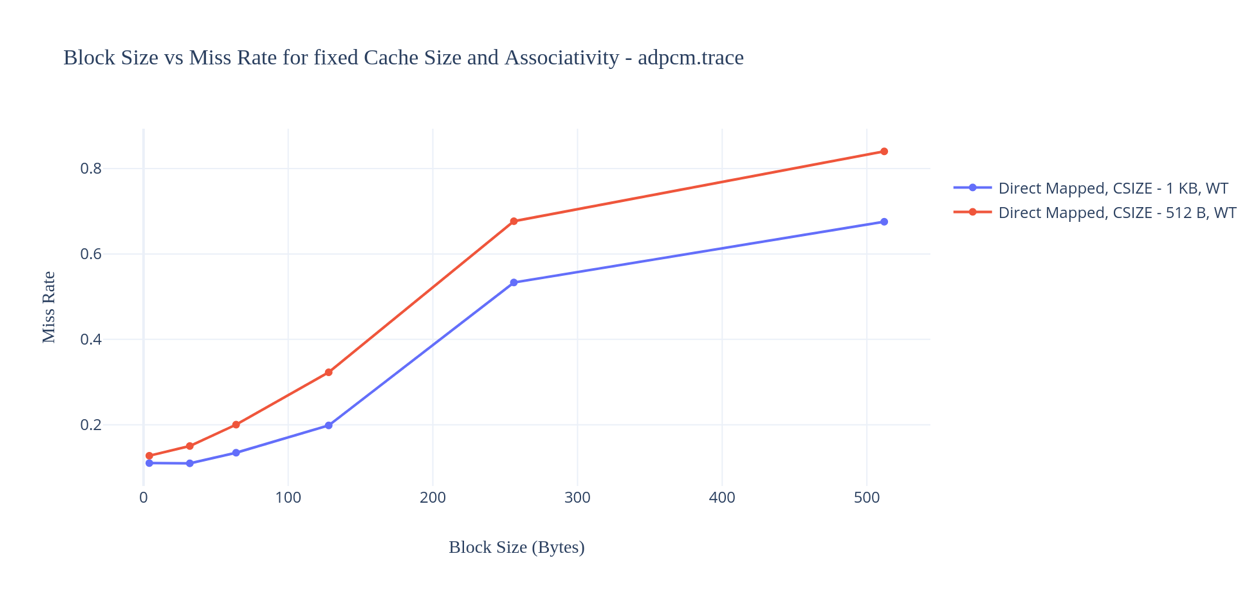
For listing out the best possible cache configurations, memory cost vs size was thoroughly analyzed before concluding. Higher cache size will have less miss rate but it is not efficient. Considering the efficiency from all corners, only plausible design configurations were listed.

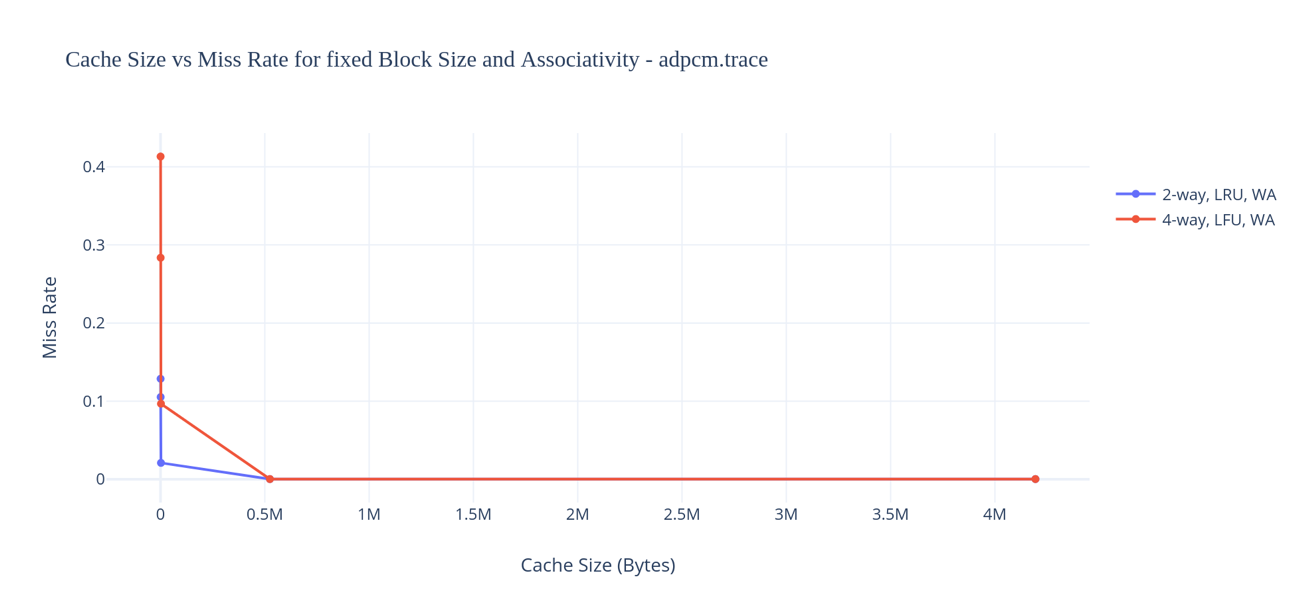
Consider that it is possible to generate a configuration at which the miss rate could reach near zero. Though it was tried best to achieve these results, all the possible configurations weren’t studied. Therefore, it is largely possible that better and efficient designs can be achieved with the same simulator.

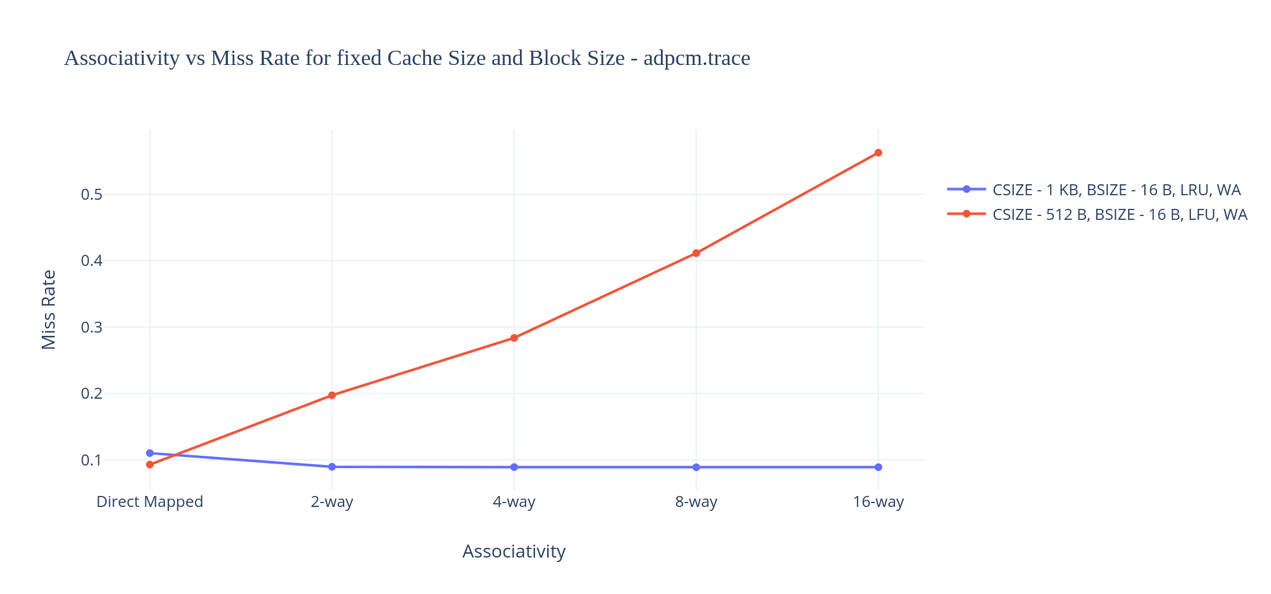
The best cache architecture is presented at the bottom for each trace after showcasing the Block Size vs Miss Rate, Cache Size vs Miss Rate, and Associativity vs Miss Rate for both, without prefetch and with prefetch from the next page.

## adpcm.trace:

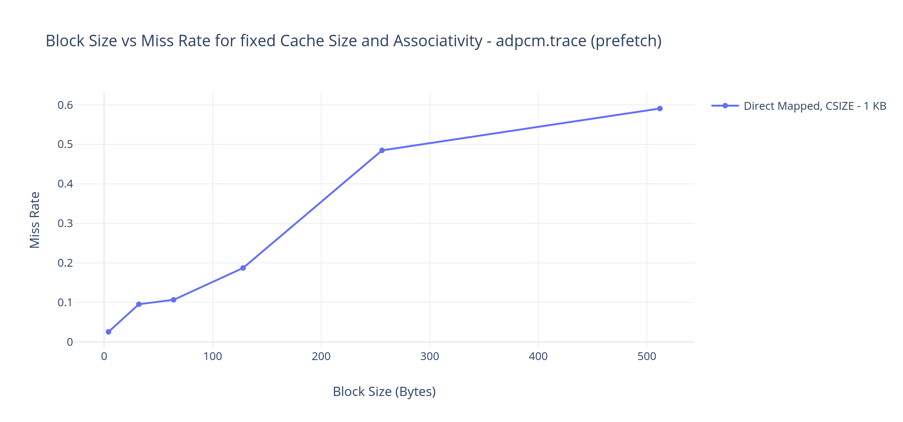
### Without prefetch

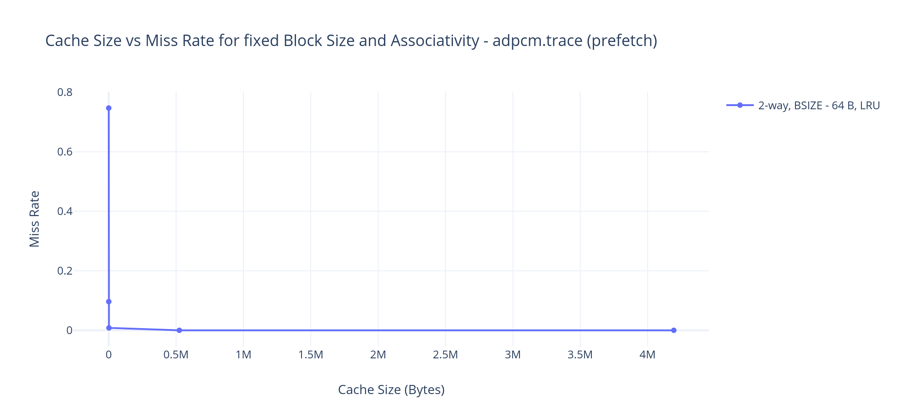


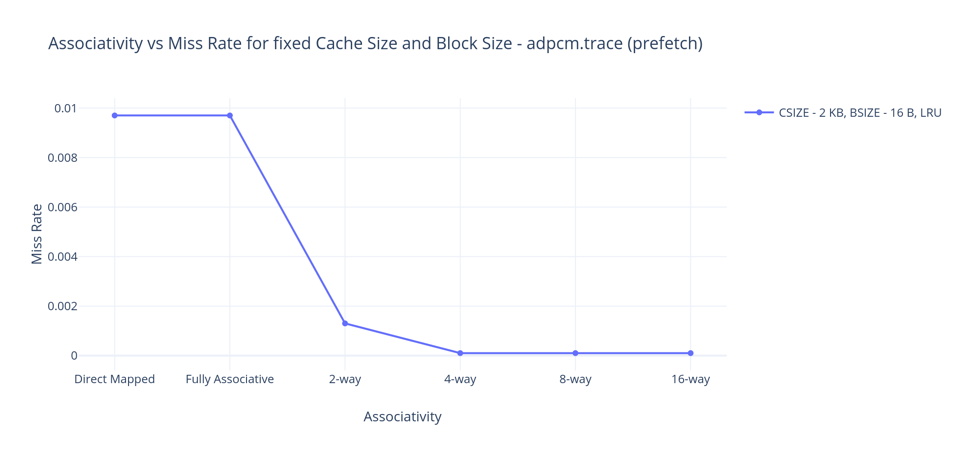




### With prefetch





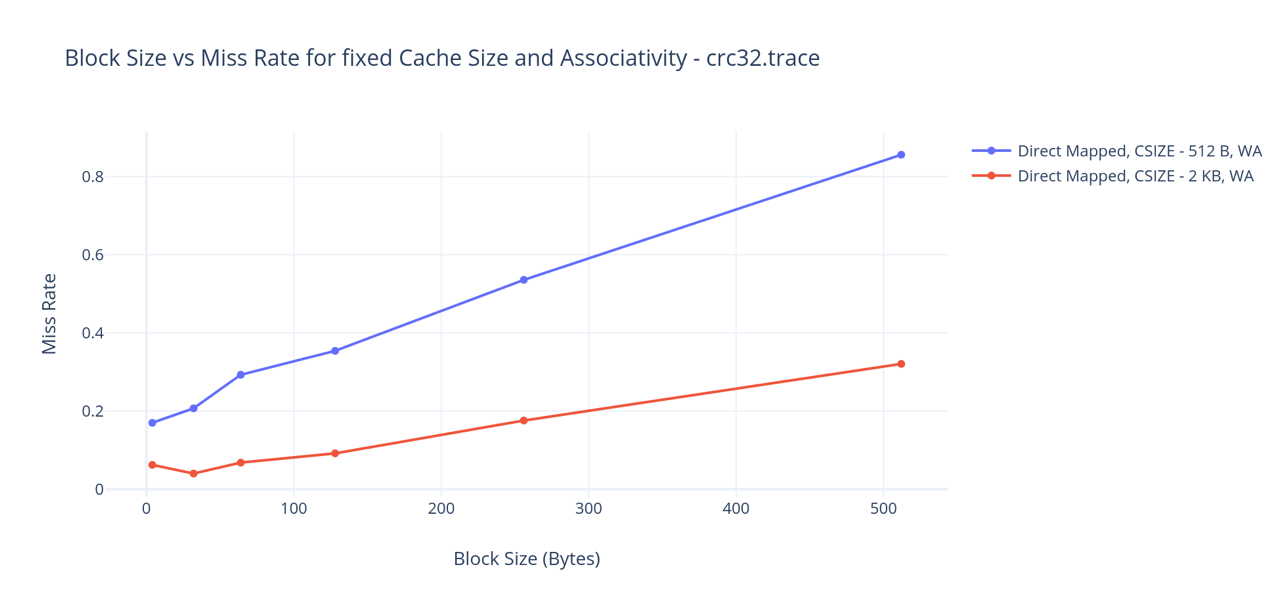


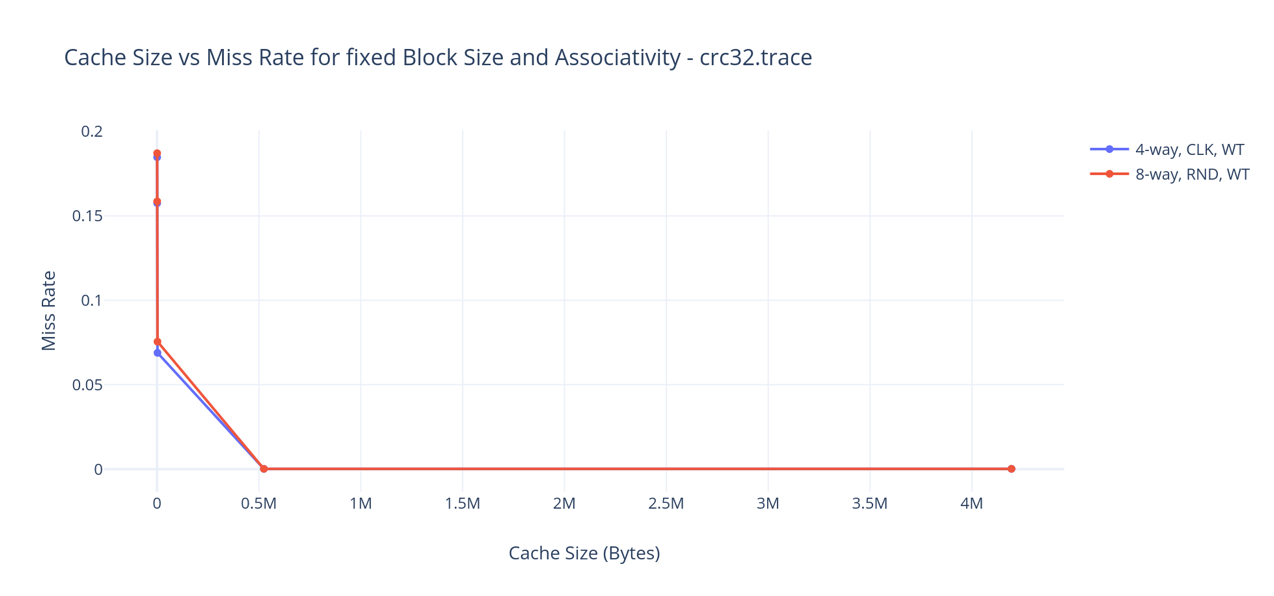
### The best possible cache configuration for the adpcm.trace without prefetch is an 8-way associative cache with a cache size of 1 KB, block size of 16 bytes with LRU and WA. This has a miss rate of 8.88 %.

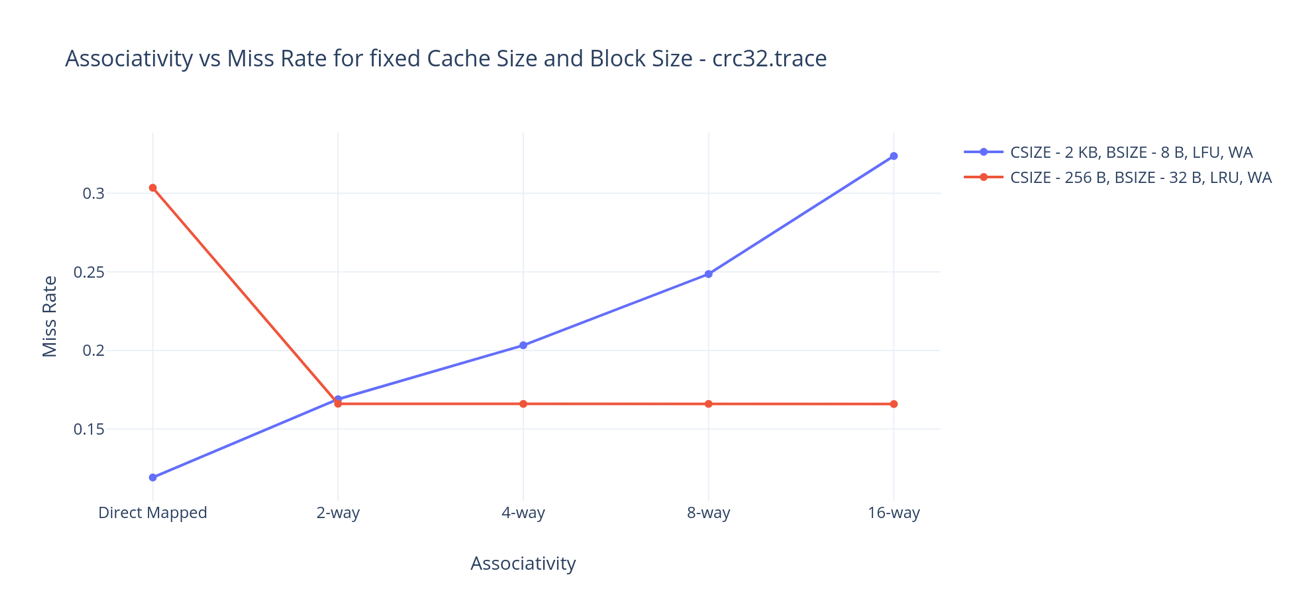
### But when prefetching is done, the best possible cache configuration for the adpcm.trace is a 4-way associative cache with a cache size of 2 KB, block size of 16 bytes with LRU. This has a miss rate of 0.01 %.

## crc32.trace:

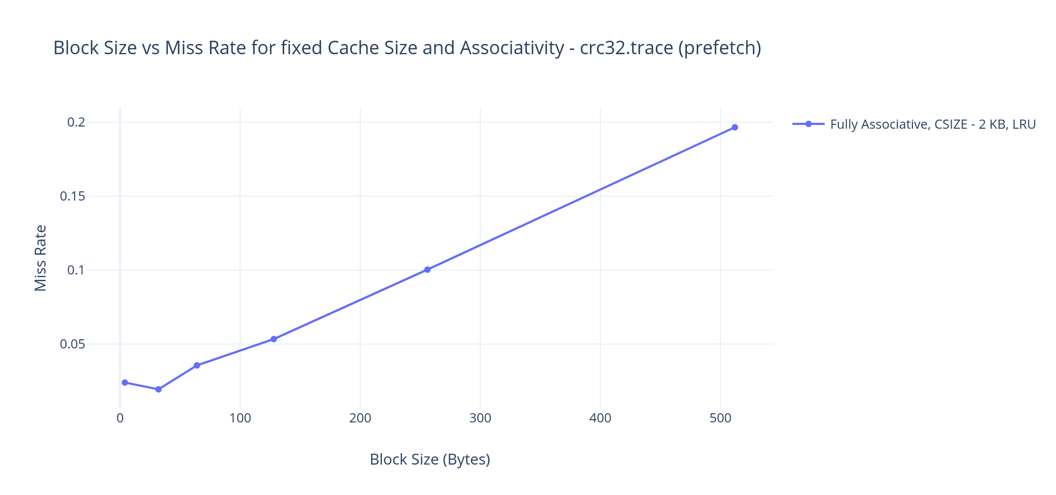
### Without prefetch

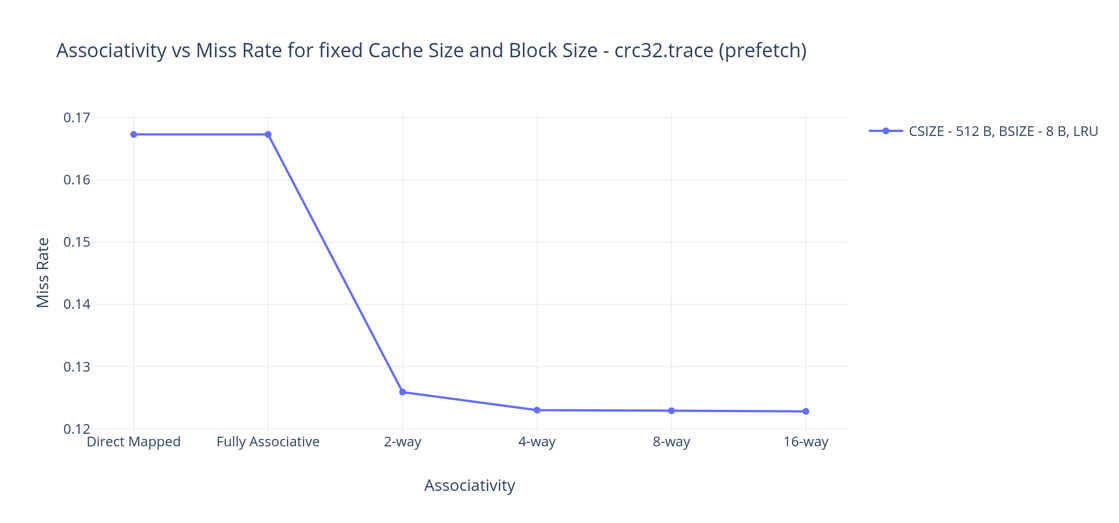
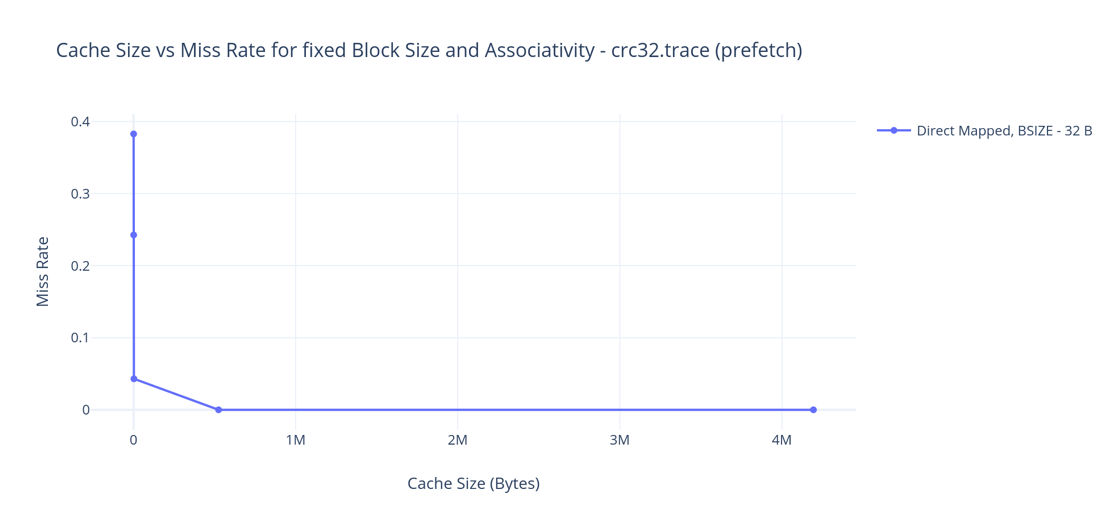






### With prefetch



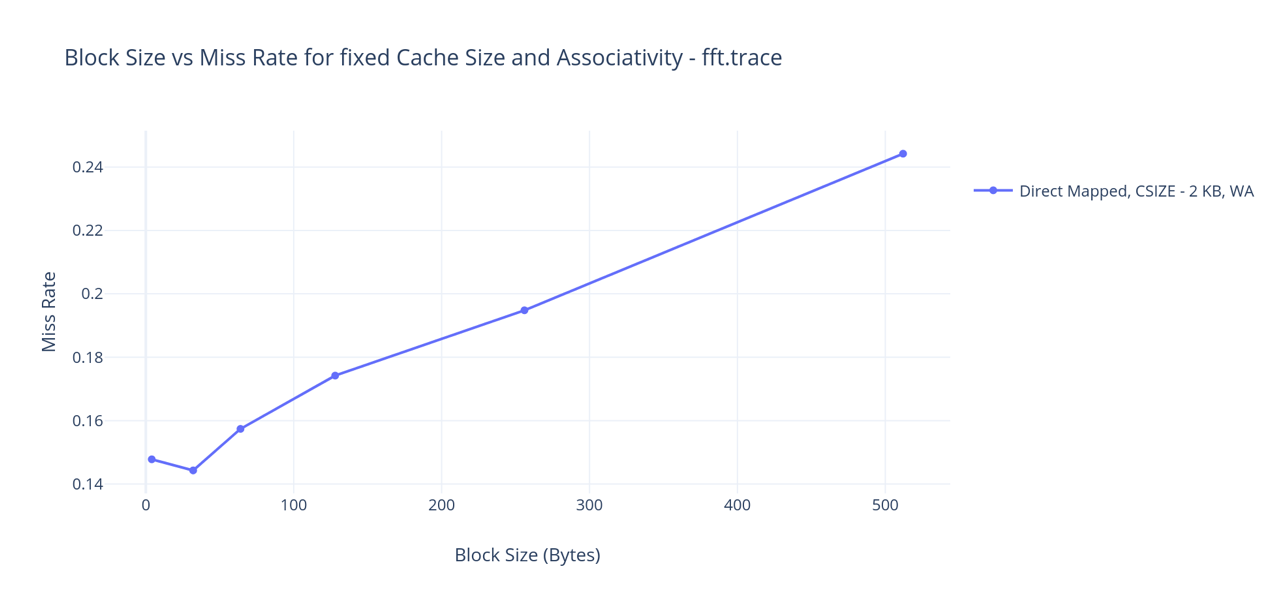


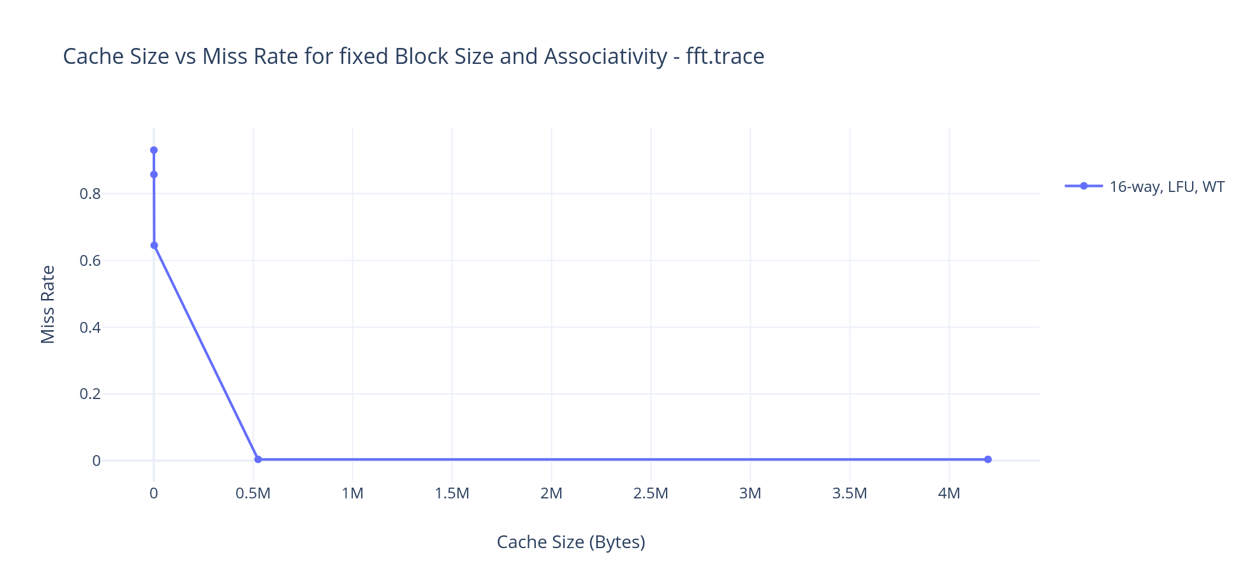
### The best possible cache configuration for the crc32.trace without prefetch is a direct mapped cache with cache size 2 KB, block size 4 bytes with WA. This has a miss rate of 6.2 %.

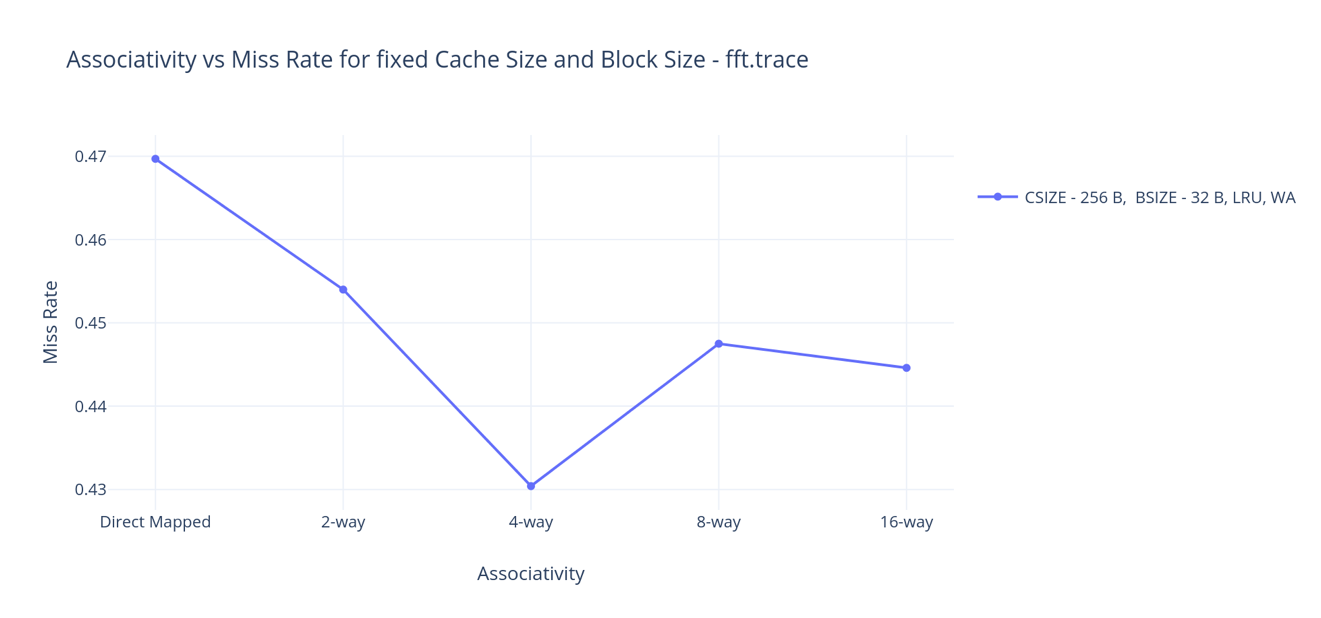
### But when prefetching is done, the best possible cache configuration for the crc32.trace is a fully associative cache with a cache size of 2 KB, block size of 32 bytes with LRU. This has a miss rate of 1.93 %.

## fft.trace:

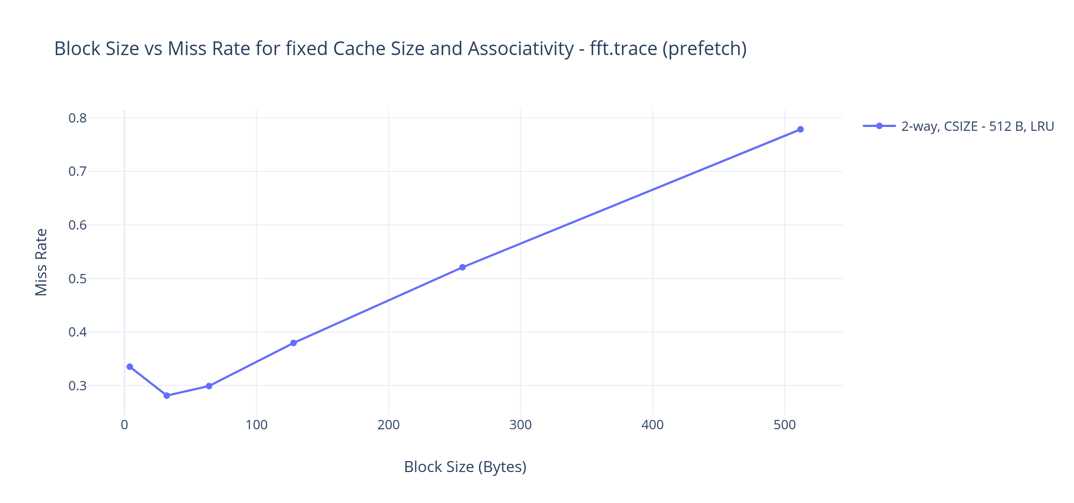
### Without prefetch

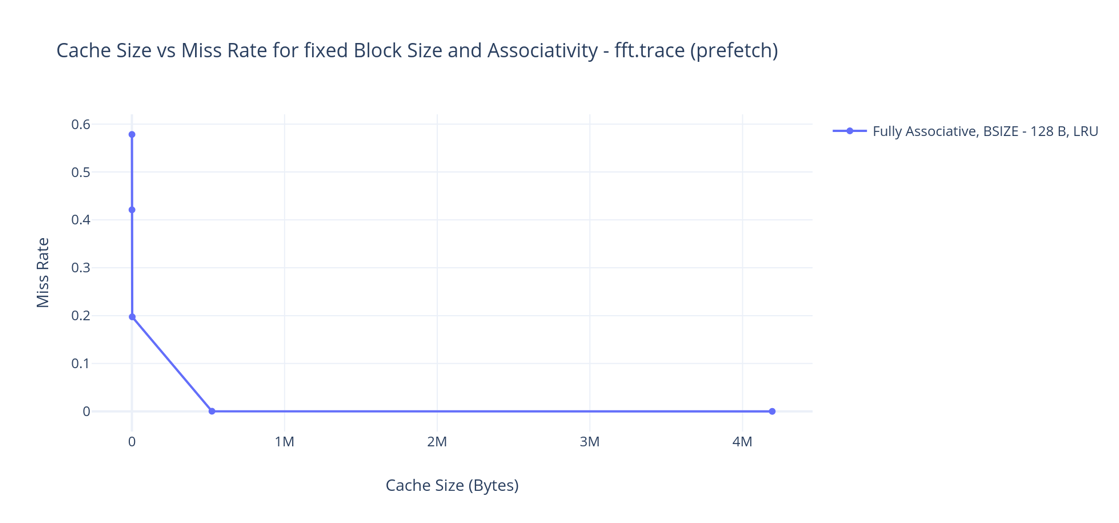


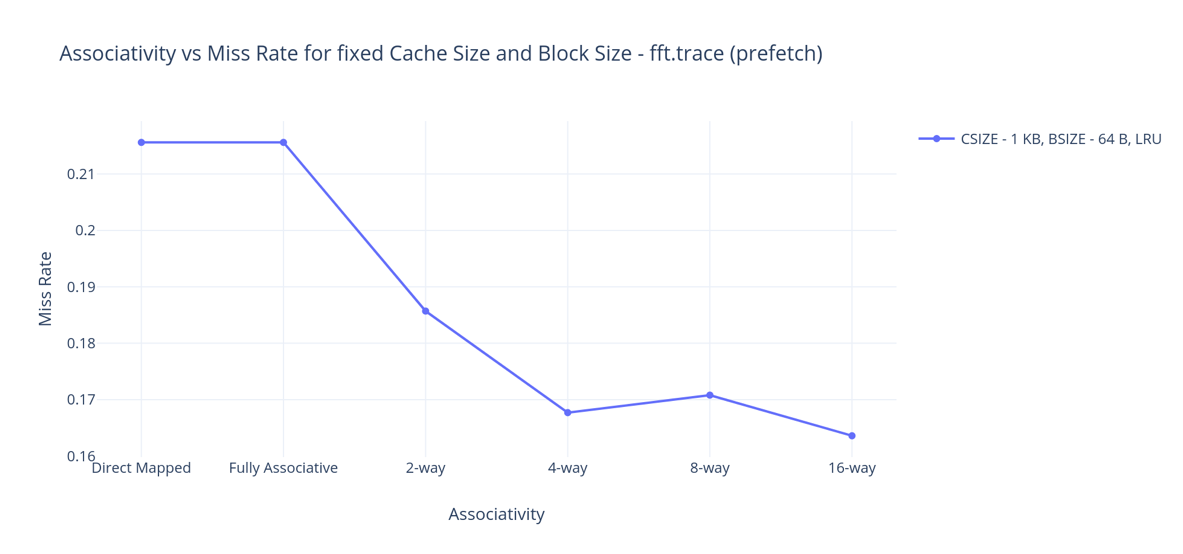




### With prefetch





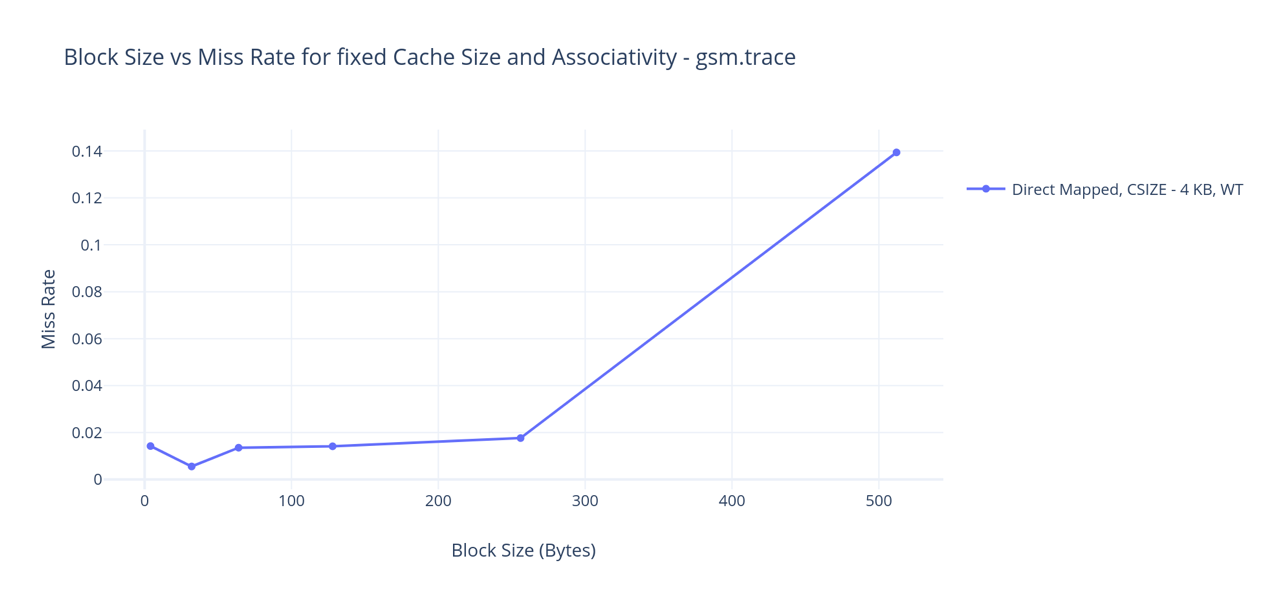


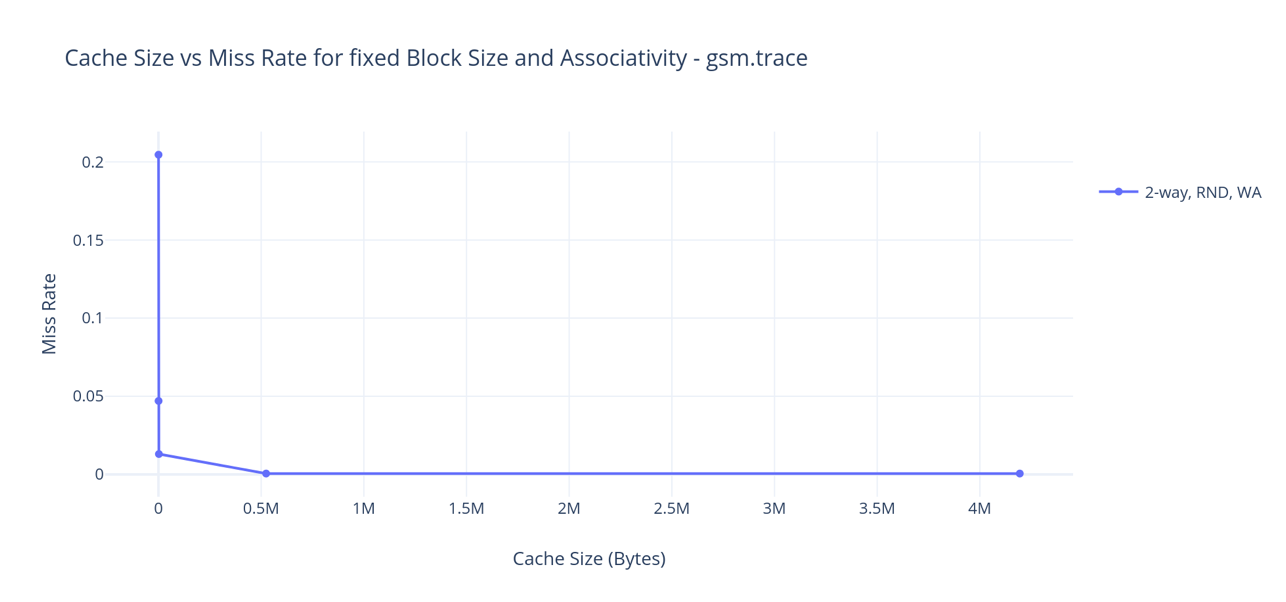
### The best possible cache configuration for the fft.trace without prefetch is a direct mapped cache with cache size 2 KB, block size 32 bytes with WA. This has a miss rate of 14.43 %.

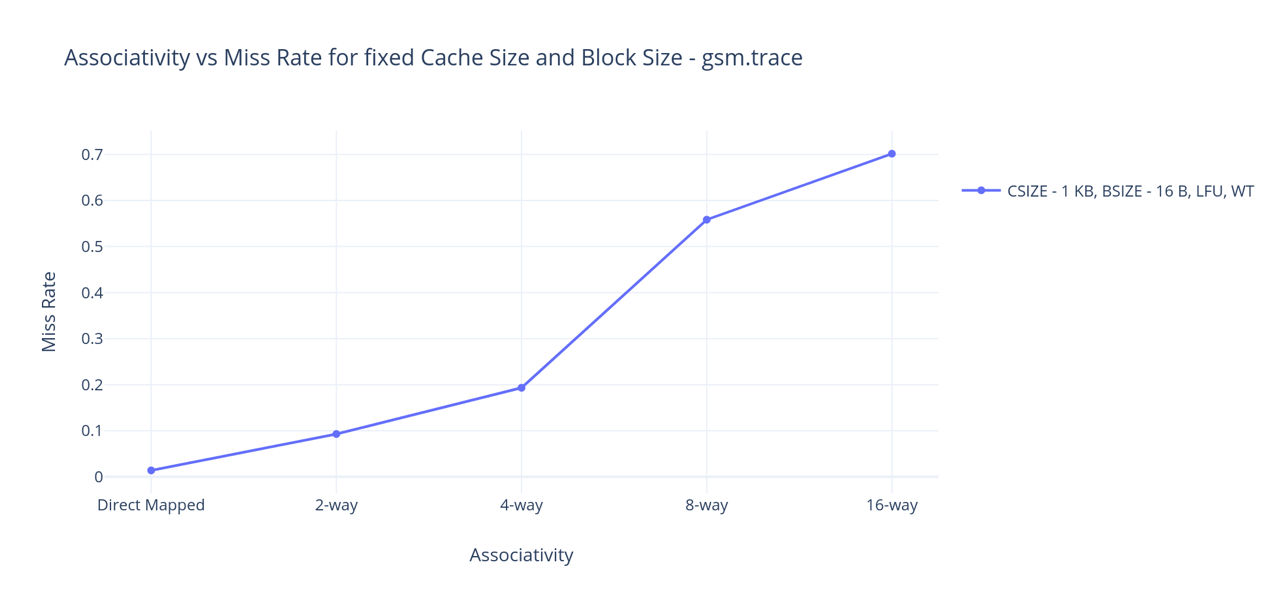
### But when prefetching is done, the best possible cache configuration for the fft.trace is a 16-way associative cache with a cache size of 4 KB, block size of 32 bytes with LRU. This has a miss rate of 9.12 %.

## gsm.trace:

### Without prefetch

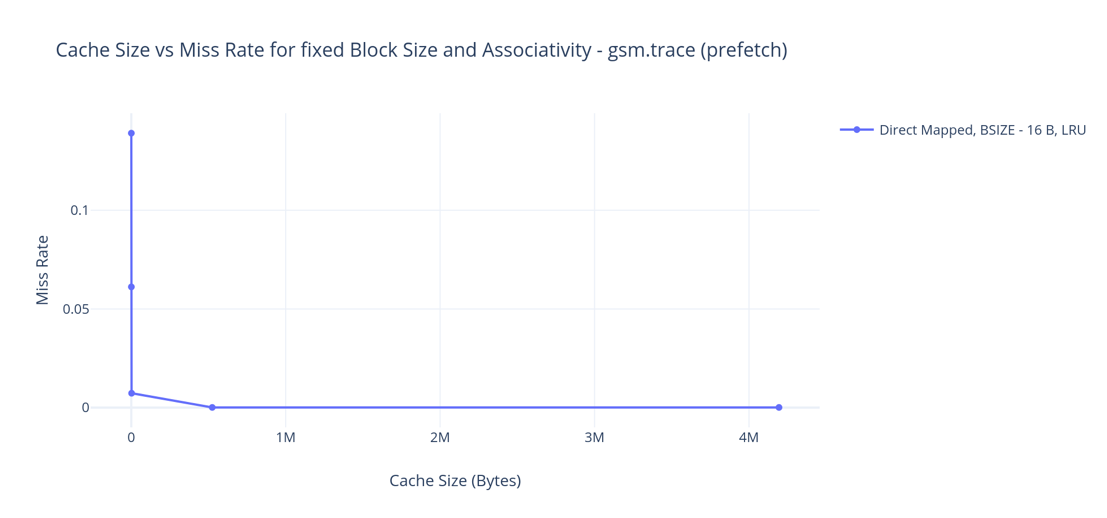


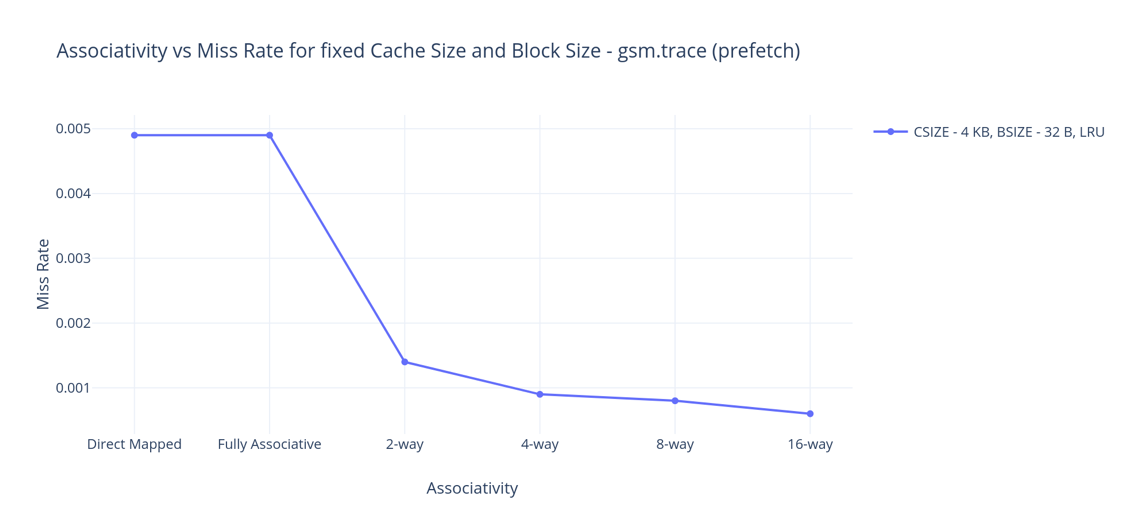




### With prefetch

### 





### The best possible cache configuration for the gsm.trace without prefetch is a direct mapped cache with cache size 1 KB, block size 16 bytes with WT. This has a miss rate of 1.4 %.

### But when prefetching is done, the best possible cache configuration for the gsm.trace is a direct mapped cache with a cache size of 2 KB, block size of 32 bytes. This has a miss rate of 0.65 %.