

Best Cache

Introduction

Our objective is to determine the best possible cache.

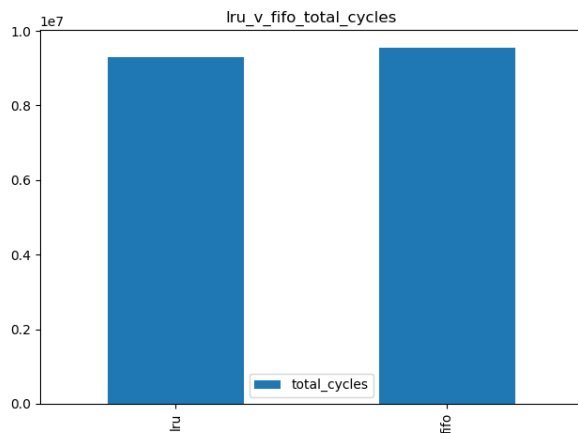
In order to incorporate hit-rate and miss-penalties, I have decided to take total-cycles as my benchmark.

A cache that can out perform other caches on several traces in total cycles (lesser is better), is considered to be better than the others.

Comparisons Ran

Comparison Across LRU and FIFO

Minute differences were observed in medium to large test cases, I couldn't find one absolute winner, although LRU was slightly better than FIFO, when everything else was the same.

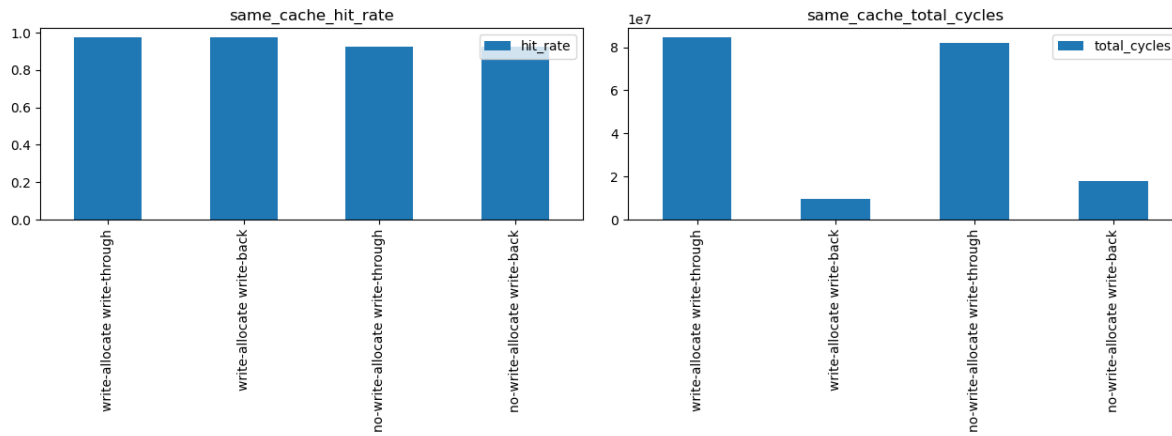


Comparison Across Cache Parameters

We can make a total of 8 combinations playing around with write-allocate-policy, write-type-policy and eviction-type-policy.

We already decided to go with LRU over FIFO.

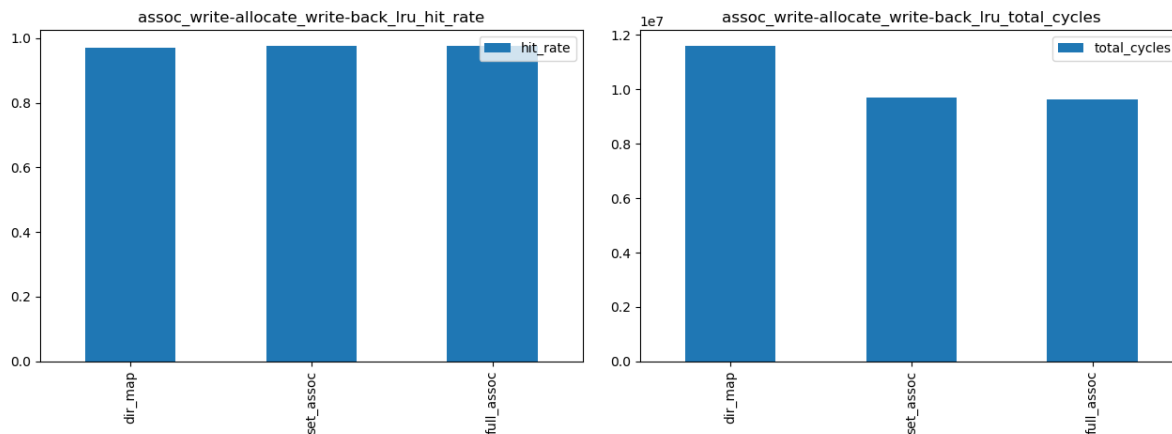
From testing over different tracefiles, it is easily determined that write-through, write-allocate is the best policy.



Comparison Across Associativity

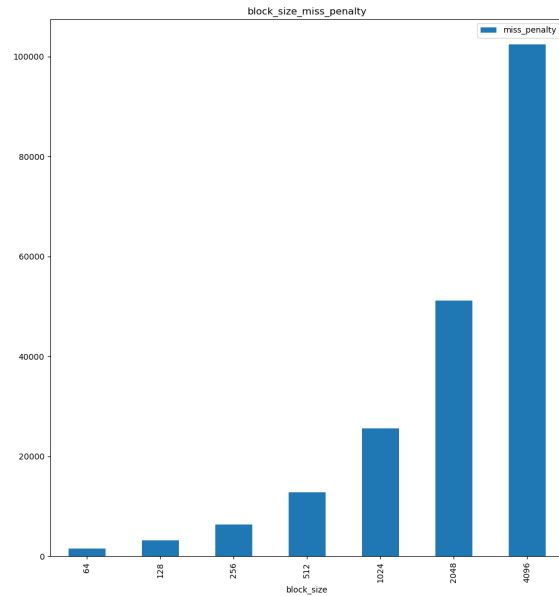
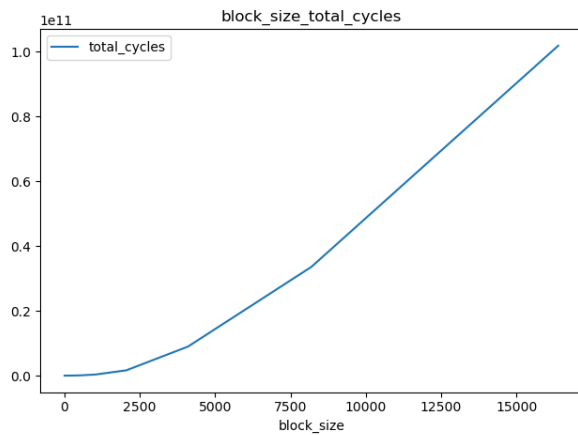
I tried 3 different combinations, amounting to the same cache size. These combinations differ in their configuration, resulting in direct mapped, set associative and fully associative.

As was expected fully associative was objectively the best cache. In our perfect world of simulation, I will choose this cache. However, in reality the hardware cost in implementing a fully associative cache is very high, hence I would decide to go with a set associative approach there.



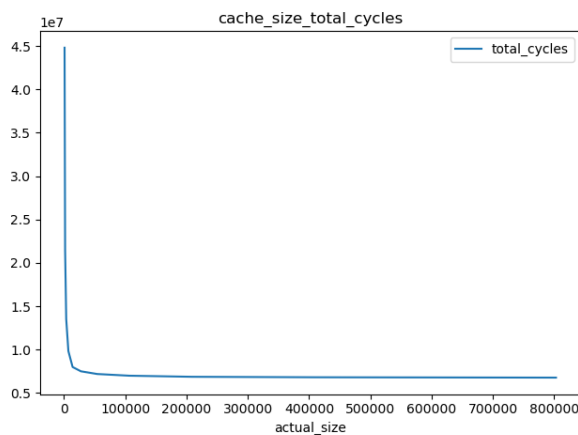
Comparison Across Block Size

For the several trace files used for benchmarking, A block size of 1 word i.e. 4 bytes came out to be the best consistently in minimising total cycles. Hence, I have decided to take block size as 1 word.



Comparison Across Cache Size

As I increased the cache size (basically associativity in this case as everything else has been determined already), the benefits of a bigger cache size flattened out even for the biggest of test cases at around 64 blocks, 4 byte block and 1 set i.e (4032 bits total including tags)



Conclusion

In my testing, I have found a fully-associative, 1 word per block, 64 block cache to be the best for realistic cache traces. Please note all graphs are for `gcc.trace`, as I considered it to be the most realistic trace-file out of the ones provided. There is also a CSV file `data.csv`, that contains the full analysis spreadsheet.