COL216 Assignment 3: Report

Aneeket Yadav (2022CS11116)

April 2024

1 Introduction

The miss ratio was calculated by varying the different cache parameters like:

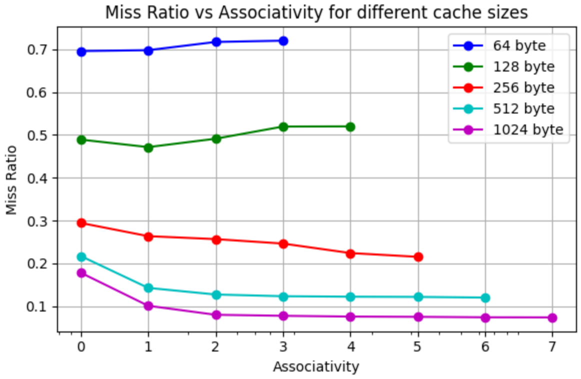
- Number of sets in cache
- Number of blocks in set
- Number of bytes in block
- Write-back or Write-through
- Write allocation
- Eviction policy (LRU or FIFO)

In real processors, number of cycles taken are not directly proportional to average mean access time(and thus hit/miss ratio) since certain instruction may have greater potential to stall the pipeline. Such complexity has not been modelled for the purpose of this assignment.It should be noted that rather minimal cache sizes were considered in the experiment for ease of computation(lower limit: 64-bytes, upper limit: 8192 bytes). Real cache sizes however are greatly larger.

Four experiments were conducted to measure the effects of various parameters.

2 Experiment 1

To gain more meaningful insights rather than cluttering the report with data, a more realistic cache having write-back, write-allocate and LRU policies was considered. The following graph was obtained:

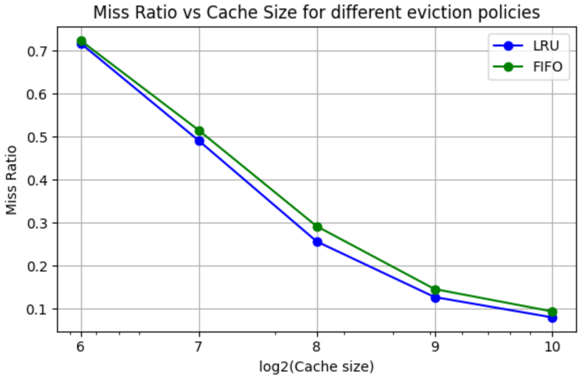


2.1 Observations

- Depending on cache size, results varied. For the 8-byte cache, miss ratio increased slightly on increasing associativity.
- Cache size and miss ratio are negatively correlated.
- Even for small cache sizes, like 512 and 1024 bytes, miss ratio decreases quickly with associativity and quickly becomes almost constant. Such behaviour is therefore expected for larger caches as well.

3 Experiment 2

FIFO and LRU eviction policies were considered for 4-way associative caches of 8-bytes block size and different cache sizes. The below graph was obtained:



3.1 Observations

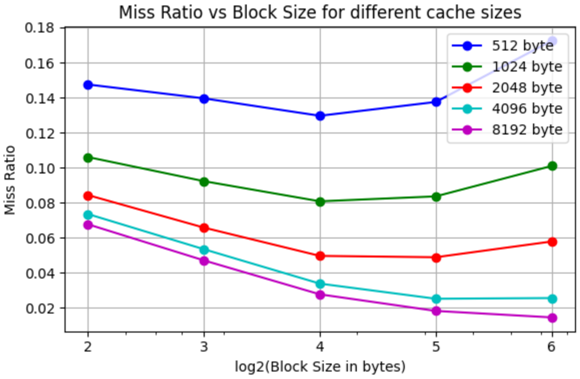
- LRU consistently performed better than FIFO for all cache sizes.
- However, the difference was not great in terms of values though a small difference in miss ratio translates greatly to cache performance.
- This may be due to the implementation of exact LRU instead of approximations and better versions such as second chance LRU.

4 Experiment 3

For a given block size, and given cache size, all possible variations associativities were considered and their mean was taken. For convenience, a write-back, write-allocate LRU cache was considered. However as the previous experiment indicates, similar results would be obtained for the FIFO cache as well. The graph was as follows:

4.1 Observations

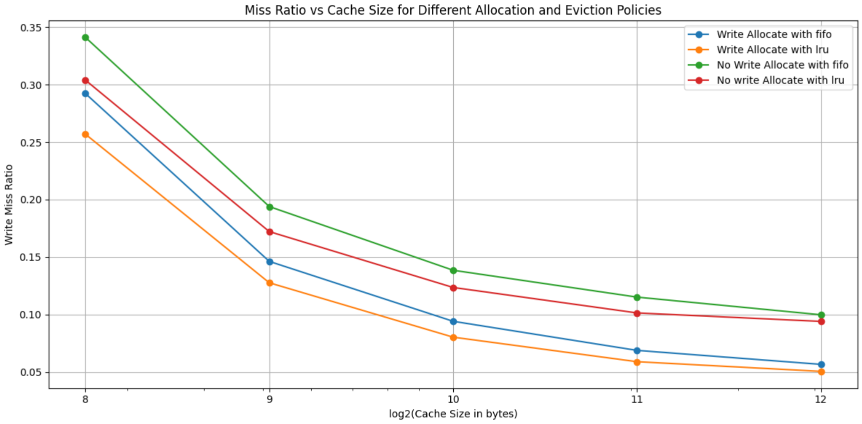
- On increasing block size, the miss ratio first decreases due to improved spacial locality and thus better accessibility of data
- On further increasing block size, a slight increase in miss ratio was observed for smaller caches the reason for which is not clear although such behaviour is expected if miss ratio was replaced by number of cycles since the delay caused by loading a block of larger size

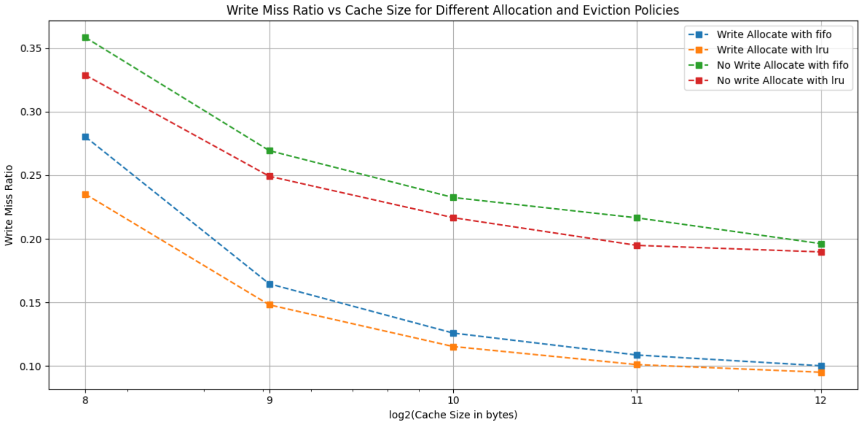


- The miss ratio decreased continuously for larger cache sizes.
- Mean may not be a suitable measure of central tendency for this experiment.

5 Experiment 4

Write allocation and eviction policies were varied for 4-way 8-byte associative caches for different cache sizes. Both the miss ratio and write miss ratio ($Number\_of\_store\_misses/Numberof\_store\_instructions$) were calculated and compared and the following graphs obtained:





5.1 Observations

- Write allocate method results in miss ratio as well as the write miss ratio decreasing significantly for all cache sizes.
- Write miss ratio was slightly higher than miss ratio in all cases.

6 Scope for improvement

In future implementations, approximations such as second chance LRU and dividing cache into L1 and L2 may yield more realistic results. Additionally, caches in multi processor systems could also be modelled. The results relied solely on the trace file `gcc.trace`, further trace files could have been included.