ECE 463 Alan Zheng
Dr. Michela Becchi 4/28/2021

Objective

The goal of this project is to implement a level-1 cache simulated using C++ and understand the behavior of a cache while adjusting some cache variables. Such variable are cache sizes, cache associativity, memory block sizes, and cache policies. Also, to analyze the miss rate of GCC trace while adjusting the variables.

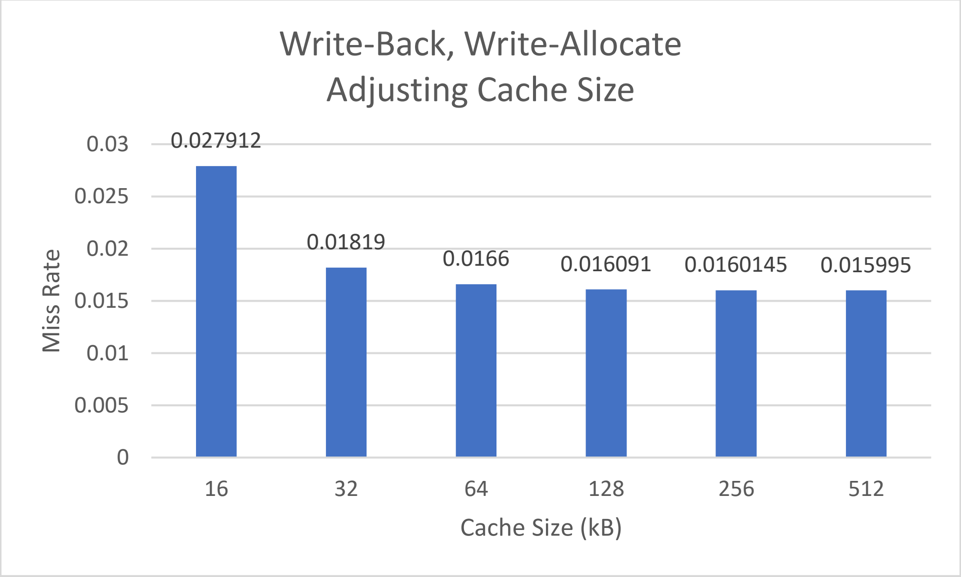
Cache Simulation

The cache is simulated in C++. By using a class to create a cache data structure, we can implement some functions necessary to simulate the cache. Along with the cache data structure, a cache entry is also structured as a two-dimensional vector array. This cache entry data structure contains members that represent the metadata and the tag array. At the end of the simulation, the cache will print of useful statistics that can be used to analyze the cache. The Makefile has been alter with the addition of std=c++11.

Cache Behavior Analysis

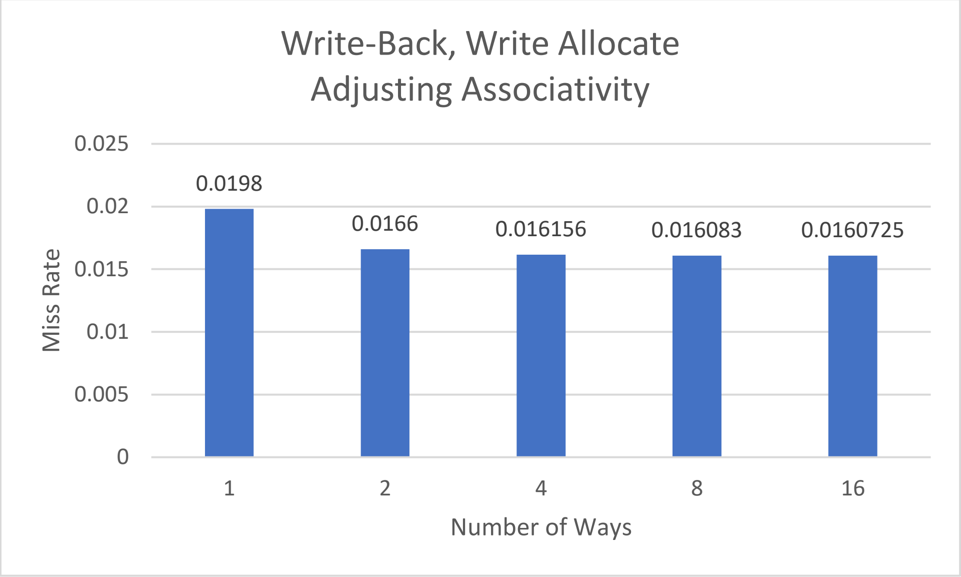
While analyzing the cache variables, it is good to normalize the variables while only altering one. Therefore, cache size will be normally set to 64 kilobytes, associativity to 2-way and the block size of 64 bytes. The influence will be separated by the cache's hit/miss policies. Cache settings will be adjusted for (a) cache size: 16, 32, 64, 128, 256, 512 kB; (b) cache associativity: 1, 2, 4, 8, 16-way; (c) block size: 32, 64, 128, 256 B.

1. *Write-Back/Write-Allocate*
   a. *Influence of Cache Size*



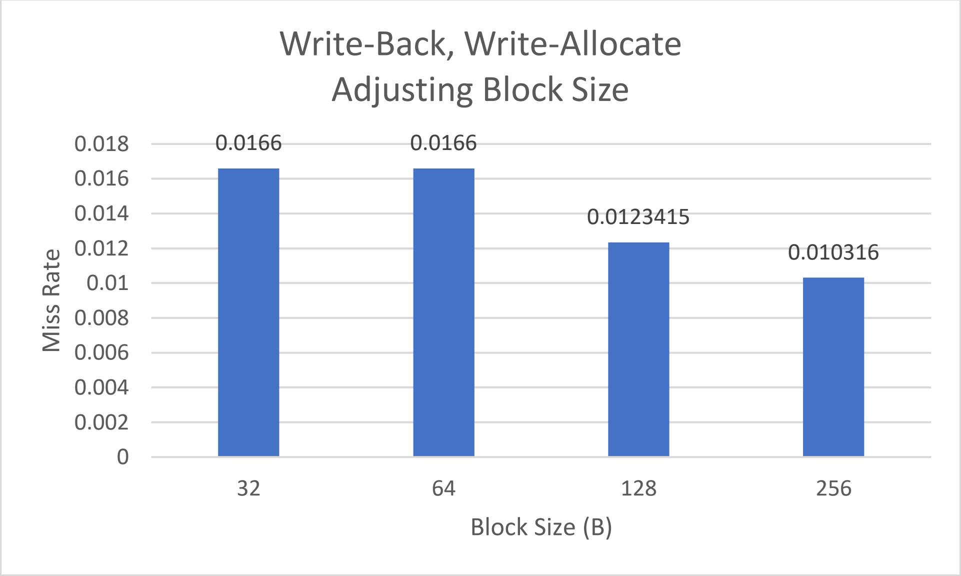
It is observed that as Cache size increase, the Miss Rate decreases. The major change is between 16 kB to 32 kB as the miss rate decreased by 0.9722%. After 64 kB, the change in cache size does not have much of an impact in regard to the Miss Rate.

b. Influence of Associativity



It is observed that as Associativity increases, the miss rate decreases. There are no major changes, as Ways increases, the Miss Rate approaches 1.6%.

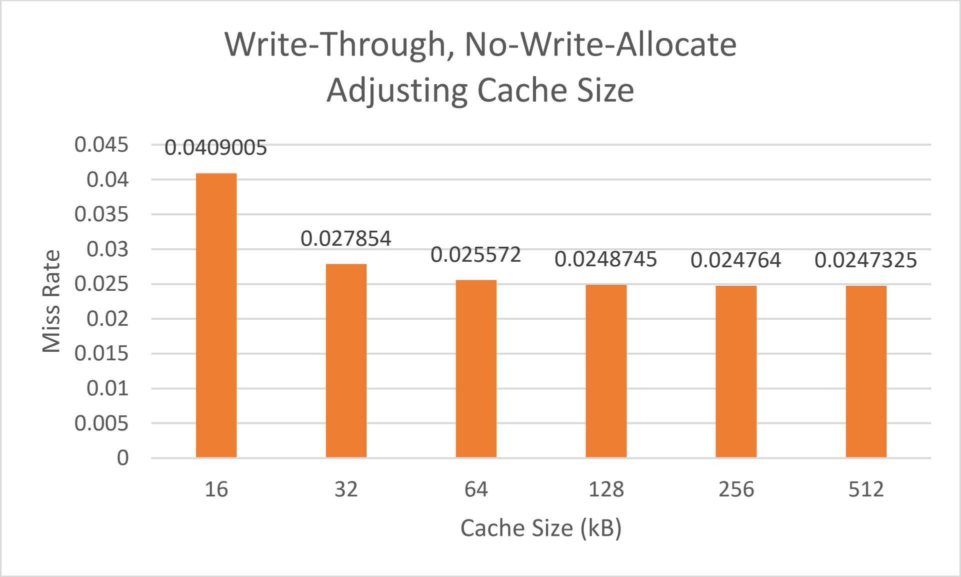
c. Influence of Block Size



It is observed that as Block Size increases, the miss rate changes. However, the Miss Rate is the same for both Block Size of 32 bytes and 64 bytes.

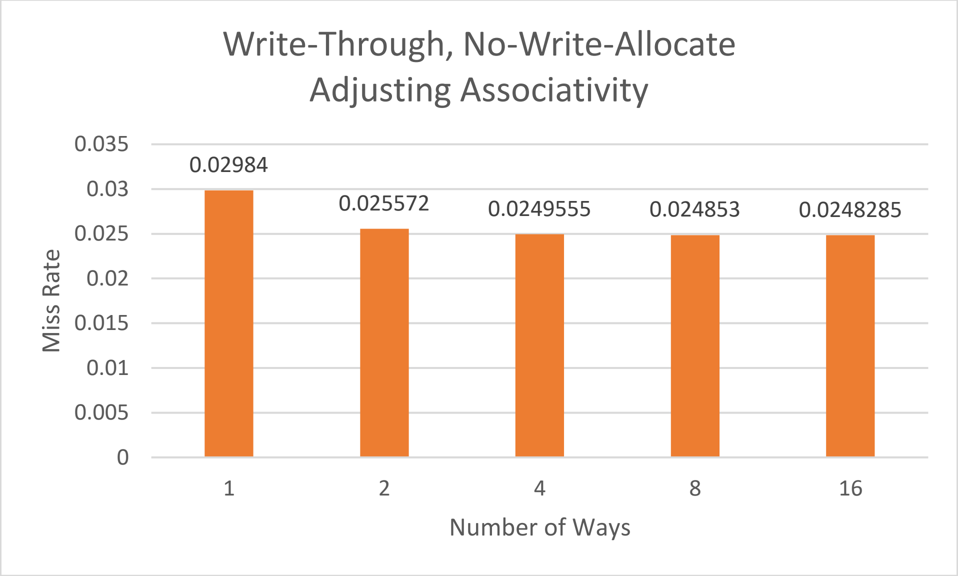
2. *Write-Through/No-Write-Allocate*

   a. *Influence of Cache Size*



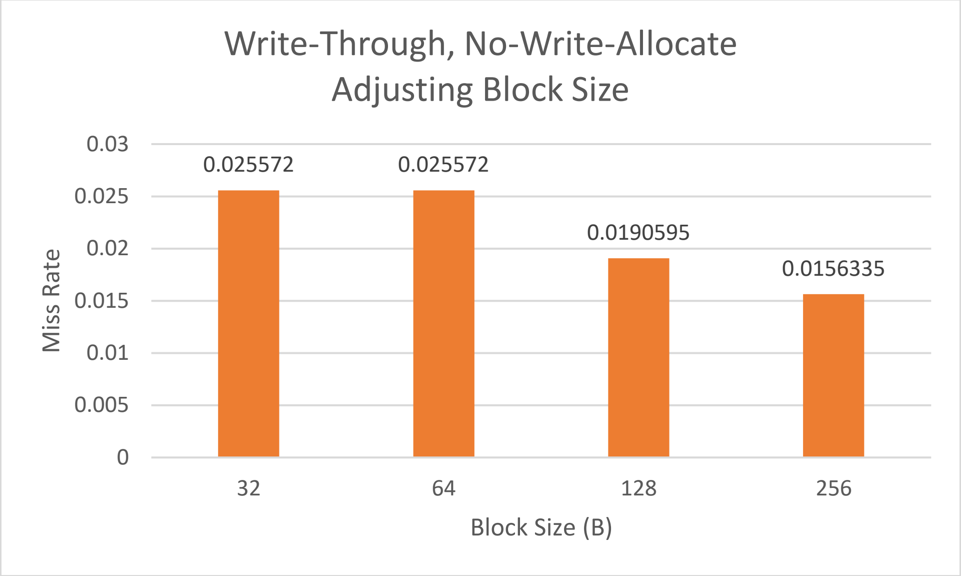
It is observed that as Cache size increase, the Miss Rate decreases. The major change is between 16 kB to 32 kB as the miss rate decreased by 1.3%. After 64 kB, the change in cache size does not have much of an impact in regard to the Miss Rate.

   b. *Influence of Associativity*



It is observed that as Associativity increases, the miss rate decreases. There are no major changes, as Ways increases, the Miss Rate approaches 2.48%.

   c. *Influence of Block Size*



It is observed that as Block Size increases, the miss rate changes. However, the Miss Rate is the same for both Block Size of 32 bytes and 64 bytes.

Conclusion

The effects of changing the Cache size, Associativity and Block Size have similar effects in both Write-Back and Write-Allocate/Write-Through and No-Write Allocate. The major differences are observed in both increasing the Cache Size and Block Size. The Associativity does not have much of an effect after an Associativity of more than 2. The policy does have a major different when compared to each other. The Write-Back and Write Allocate has a overall lower miss rate due to the block in memory always adds an entry to the cache on a write miss. However, there is more eviction due to the Write-Allocate policy. If the trends suggest that the lowest miss rate would be cache size of 512 kB, 16-way, and a block size of 256B in Write-Back and Write-Allocate policy. Although this setting would most likely be the most expensive and is not as cost efficient.