**North Carolina State University**

**Department of Electrical and Computer Engineering**

**ECE 463/563: Fall 2019**

**Project 1: Cache and Memory Hierarchy Design**

**by**

**RHUSHIKESH ANAND PRABHUNE**

**(NCSU ID: 200321627)**

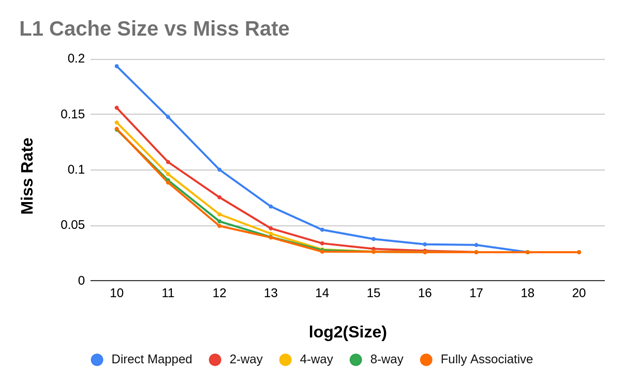
**L1 Cache Size vs. Miss Rate (without L2):**

Cache Configuration:

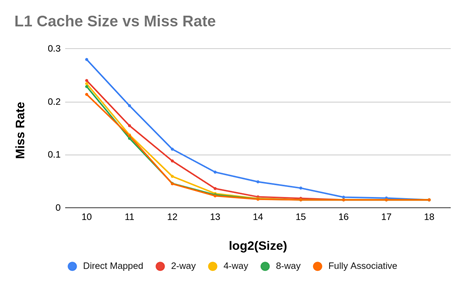
Block size: 32kB

L1 Associativity: Direct Mapped, 2-way, 4-way, 8-way, and Fully Associative

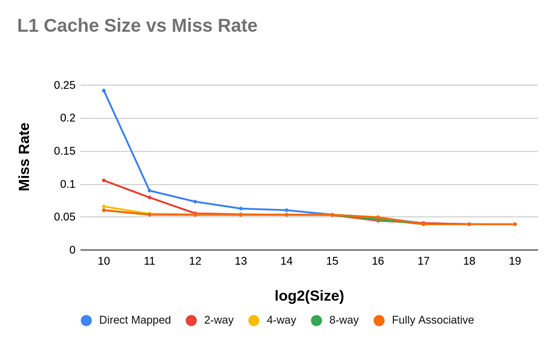
1. **gcc\_trace.txt**

****

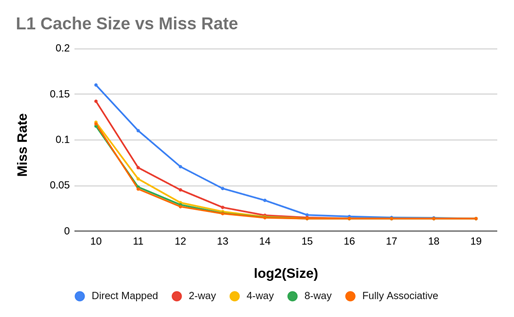
1. **perl\_tace.txt**



1. **go\_trace.txt**



1. **vortex\_trace.txt**



The miss rate goes on decreasing with increasing L1 size as the number of capacity misses go on decreasing with increasing cache size. Also, as the size increases, for high L1 sizes, the miss rate does not decrease to a large extent because the conflict misses and cold misses would not decrease with increasing the size of L1.

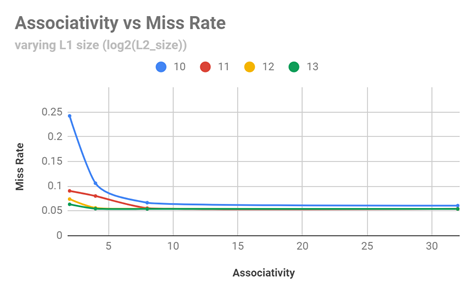
**Associativity vs. Miss Rate**

Cache Configuration:

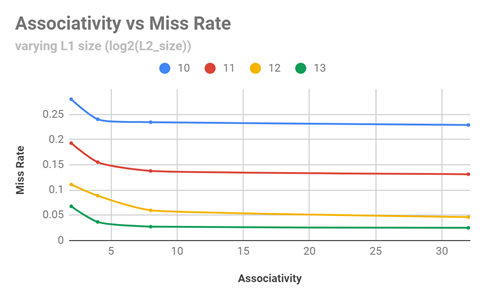
Block size: 32kB

L1 Size:1024kB,2048kB,4096kB,8192kB

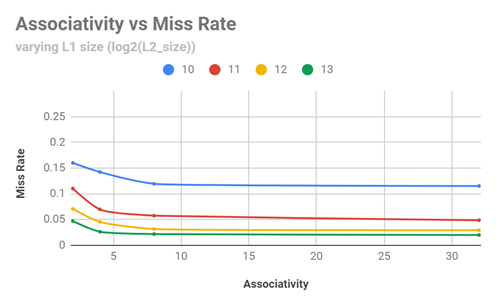
1. **gcc\_trace**



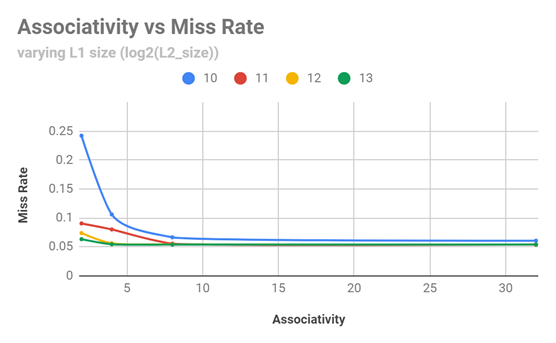
1. **perl\_trace**



1. **vortex\_trace**



1. **go\_trace**



As the associativity is increased, the miss rate goes on decreasing as the number of conflict misses decreases. After certain associativity, the miss rate does not decrease much capacity and cold misses do not decrease with increasing associativity. Therefore, we reach a point where we are unable to reduce the miss rate on increasing associativity.

**L2 cache size vs. Miss Rate(L1 size constant):**

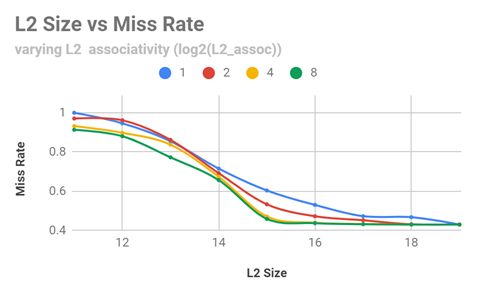
L1 Size: 4096 kB

L1 Associativity: 4

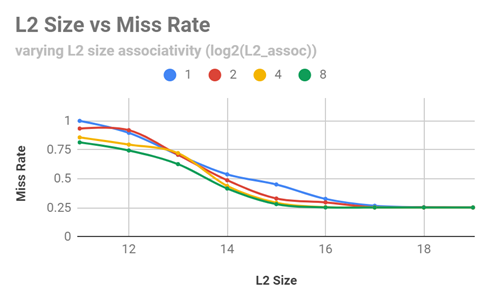
L2 Associativity: 2, 4,16, 256

N=1, P=1

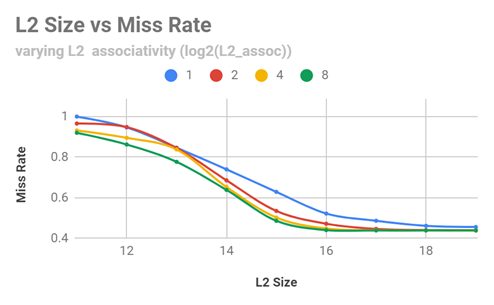
1. **gcc\_trace**



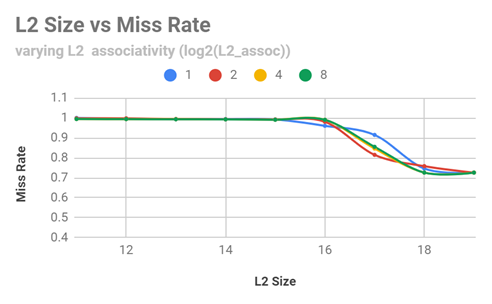
1. **perl\_trace**



1. **vortex\_trace**



1. **go\_trace**



The miss rate goes on decreasing with increasing L2 size as the number of capacity misses go on decreasing with increasing cache size. Also, as the size increases, for high L2 sizes, the miss rate does not decrease to a large extent because the conflict misses and cold misses would not decrease with increasing the size of L2.

**Number of address tags (N) vs. Miss Rate( L1 constant, P constant):**

Blocksize = 32 kB

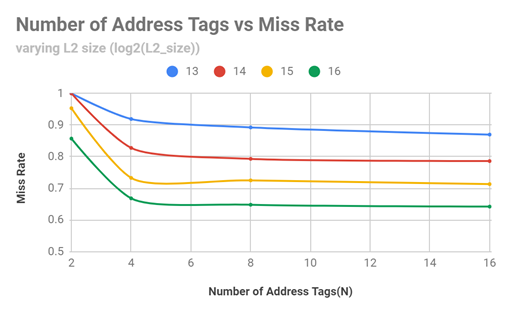
L1 Size = 8192 kB

L1 Associativity = 4

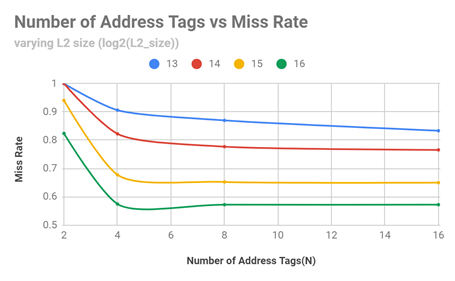
L2 Associativity = 1

P=8

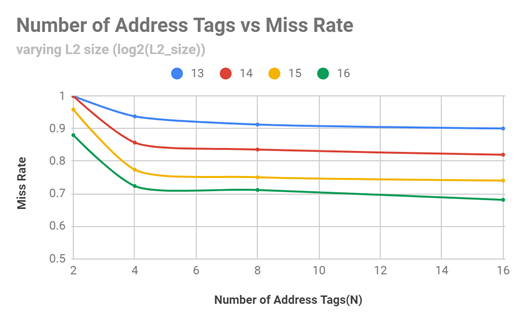
1. **gcc\_trace**



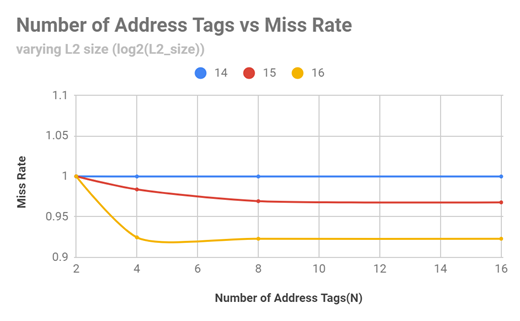
1. **perl\_trace**



1. **vortex\_trace**



1. **go\_trace**



As the number of tag address blocks go on increasing, the miss rate decreases because, the probability of matching a tag with an index increase due to multiple tags corresponding to a particular index. Initially, the tag address blocks get filled resulting in cold misses. But after the tag blocks are filled, the number if misses will decrease for an address with a particular index.

**Number of data blocks (P) vs. miss rate( L1 constant, N constant):**

Blocksize = 32 kB

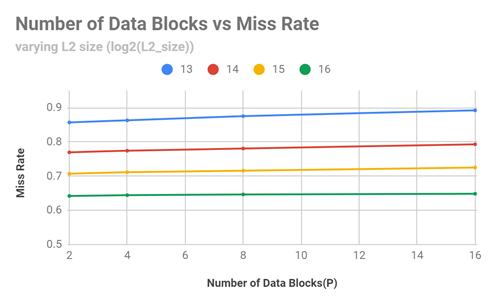
L1 Size = 8192kB

L1 Associativity = 4

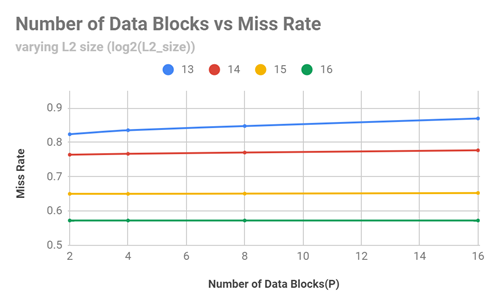
L2 Associativity = 1

N=4

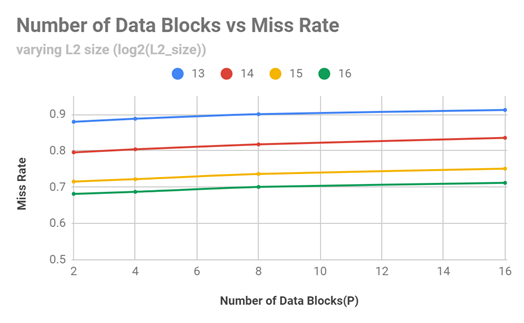
1. **gcc\_trace**



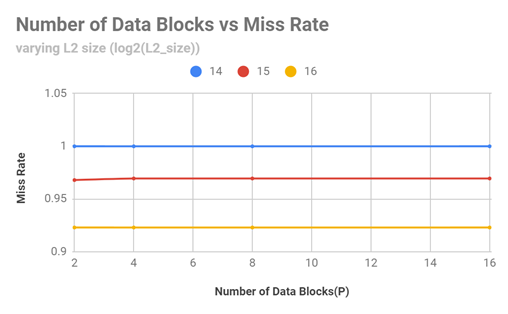
1. **perl\_trace**



1. **vortex\_trace**



1. **go\_trace**



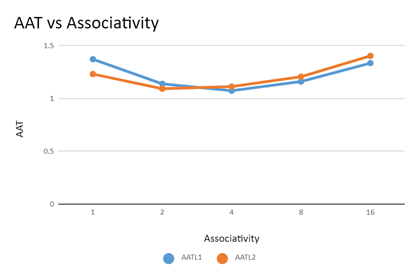
Increasing the number of data blocks hampers the miss rate because for lesser data blocks, a larger number of data blocks will map to a particular tag block which will increase the chances of a hit.

L1 Associativity = 1, 2, 4, 8, 16

L1 Size = 16kB

L2 Size = 64kB

N=4, P=8



As the L2 Associativity is increased the AAT drops until a point(4-way associative) after which it increases. This is because AAT depends on Miss Rate and Miss Penalty. The miss rate will decrease on increasing associativity whereas the miss penalty will increase because of latency.

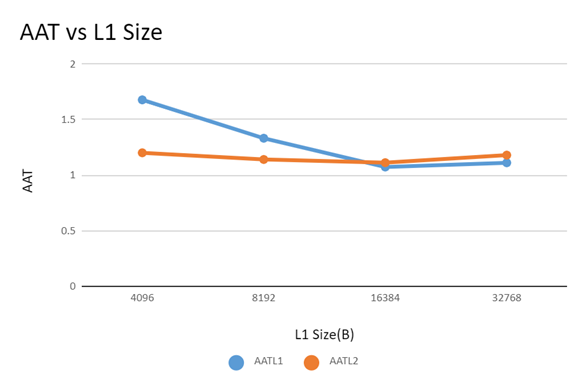
L1 Size = 4kB, 8kB,16kB,32kB

L2 Size = 64kB

L1 Associativity = 4

L2 Associativity = 1

N = 4, P = 8



Increasing the size of the cache will also improve the AAT until a point where the miss penalty dominates. This is because a larger cache will need more time to be accessed.

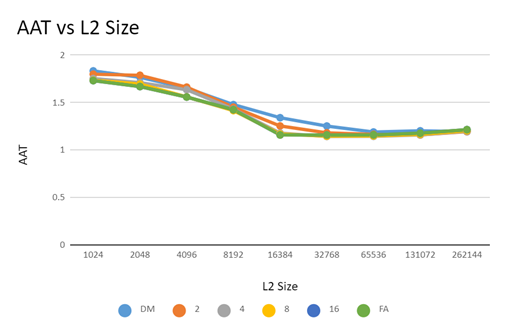
L1 Size = 4kB

L2 Size = 1kB,2kB,4kB,8kB,16kB,32kB,64kB,128kB,256kB

L1 Associativity = 4

L2 Associativity = 1,2,4,8,16, Fully Associative

N=1,P=1



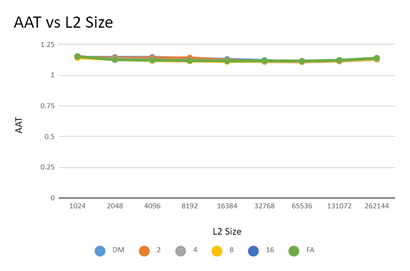
L1 Size = 16kB

L2 Size = 1kB,2kB,4kB,8kB,16kB,32kB,64kB,128kB,256kB

L1 Associativity = 4

L2 Associativity = 1,2,4,8,16, Fully Associative

N=1,P=1



Because the L1 is larger, the AAT is not affected much by changing the L2 size.

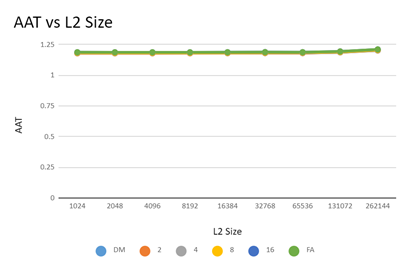
L1 Size = 32kB

L2 Size = 1kB,2kB,4kB,8kB,16kB,32kB,64kB,128kB,256kB

L1 Associativity = 4

L2 Associativity = 1,2,4,8,16, Fully Associative

N=1,P=1



Higher associativity will increase the capacity misses but reduce the conflict misses. The compulsory misses depends on the block size.

**What are the advantages of Decoupled Sector Cache? Does implementing decoupled sector cache give you any benefits over the same size basic cache module? In this project, why we implemented decoupled sector cache over L2 cache and not L1 cache? Give reasons behind your answers.**

The bandwidth of data that needs to be sent to a lower cache decreases significantly because of the use of sectors. This reduces the miss penalty and thus improves the performance of the cache. The hit rate also improves because more than one tag corresponds to an index. The tag store size decreases in a decoupled sector cache which reduces the hardware cost of the cache.

L1 cache needs to be fast. Basically more complex the cache, the slower it will get. Adding more tags and data blocks can slow down the cache. On the other hand, it is okay if L2 is a bit slower. The L2 cache helps reduce the miss rates and latency. A decoupled sector cache improves the spatial locality which is helpful for L2 cache more than L1 cache because L1 cache is accessed first so a better temporal locality suits an L1 cache. L2 is not accessed as frequently as L1 but a spatial locality will definitely help L2 reduce miss rate.