University of Central Florida

Department of Computer Science

CDA 5106: Fall 2024

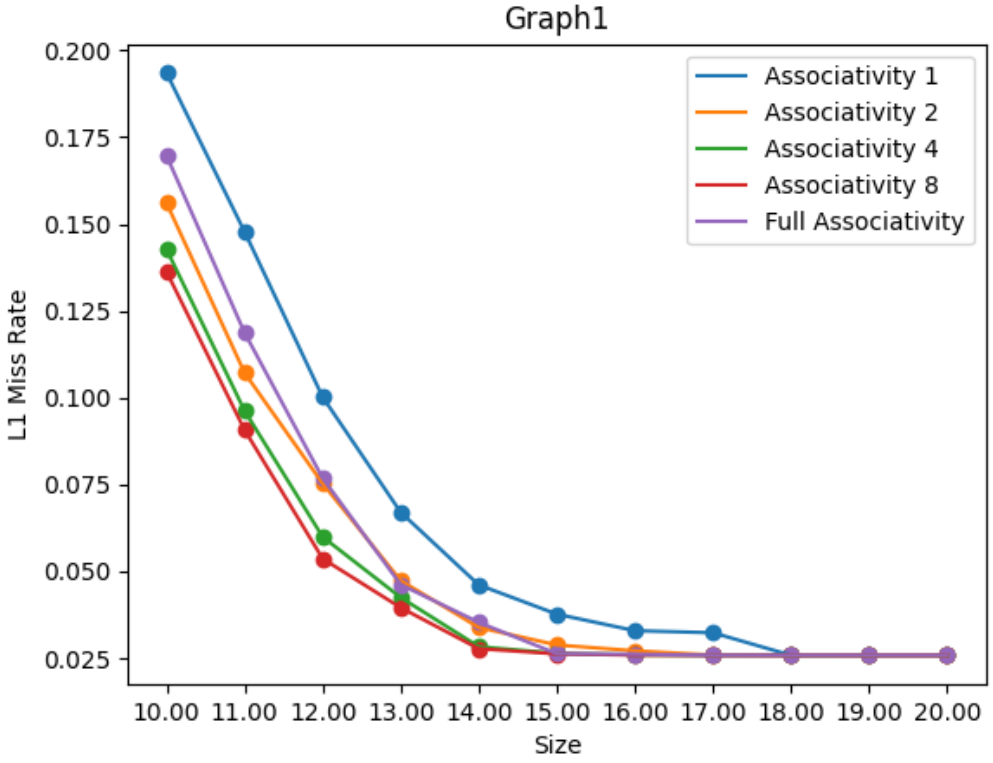
Machine Problem 1: Cache Design, Memory Hierarchy Design

by

<< ASHISH THATIPELLI >>

Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's electronic signature: __ASHISH THATIPELLI___



1. From the graph it can be noticed that higher the cache size lower the miss rate since larger cache size supports holding more data as well larger cache size increases the chance of a block being found in the cache. Higher associativity reduces the miss rate for a given cache capacity by minimizing the conflict misses by enabling a block to map to multiple sets. Greater associativity reduces the likelihood of conflict.

2. For caches with large sizes, these graphs can be used to evaluate the miss rate as they approach zero. During this time majority of misses can be unnoticed because of unfamiliar new which are important for analysis. It appears that the required miss rate is between 0.1 and 0.2.

   So estimated compulsory miss rate is 0.02582.

3. For a given size, the difference between the fully associative curve and the direct-mapped curve can be used to estimate the conflict miss rate. This gap is approximately 0.3–0.5 for smaller cache sizes, such as 1KB, 2KB, and 4KB, indicating high conflict misses because of constrained sets in direct-mapped caches. With an increase in cache size, the conflict misses decrease dramatically.

   We know the Compulsory miss rate. And we also know the L1 Miss Rate for the respective Associativity's. So, Conflict Miss Rate can be calculated as,
   Conflict Miss Rate = Miss Rate – Compulsory Miss Rate

   Direct Mapped (1-Way Set Associative),

   Conflict Miss rate = 0.193460 – 0.025820 = **0. 16764**

2-Way Set Associative,

Conflict Miss rate = 0.156030 – 0.025820 = **0. 13021**

4-Way Set Associative,

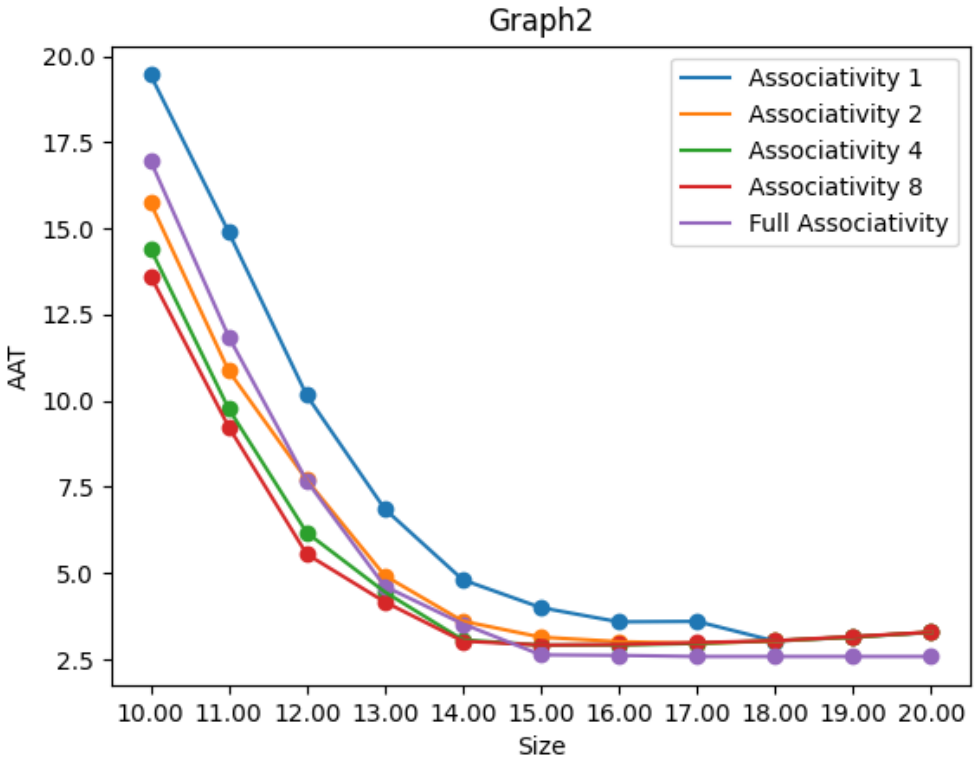
Conflict Miss rate = 0.142700 – 0.025820 = **0. 11688**

8-Way Set Associative,

Conflict Miss rate = 0.136270 – 0.025820 = **0. 11045**

Fully Associative,

Conflict Miss rate = 0.136960 – 0.025820 = **0. 1111**



1. The lowest AAT is produced by a fully associative configuration with a large cache size (like 1MB) for a memory hierarchy with only an L1 cache and BLOCKSIZE = 32. Even with small cache sizes, there are no conflict misses when there is full associativity. By enabling more data blocks to be cached and thereby lowering miss rates, increasing the cache size further lowers AAT. The optimal performance is achieved by fully associative because there is only an L1 cache and no lower level cache, hence minimizing misses.

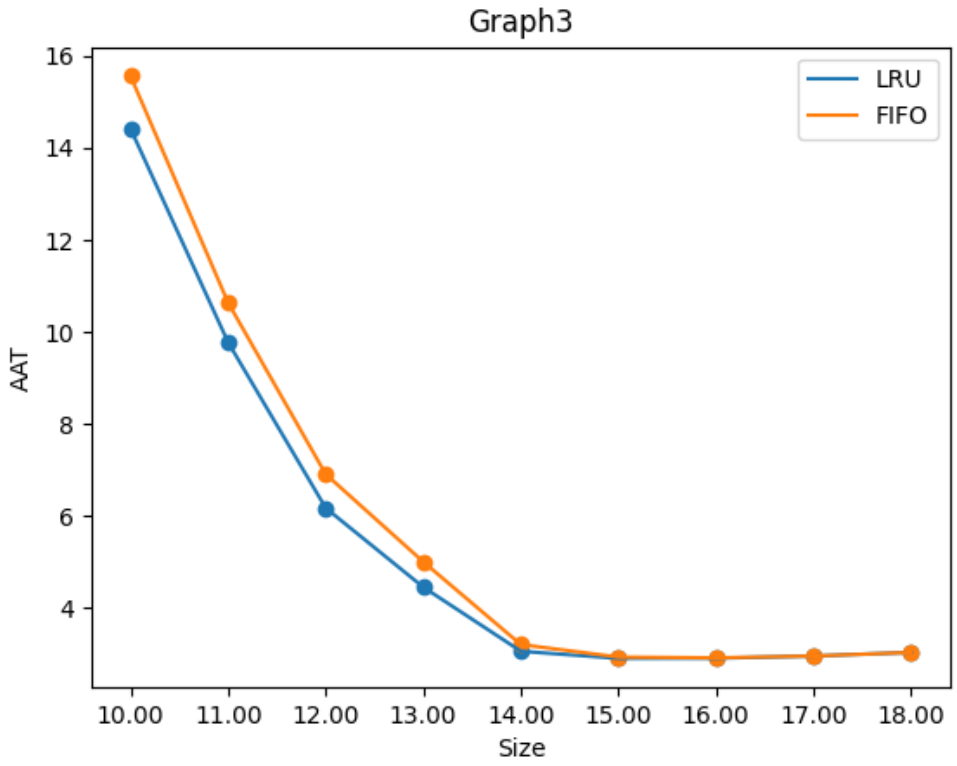
The configuration for which lowest AAT is seen –

$L1\ Cache\ Size = 16KB$

$(\log(Cache\ Size) = 14)$

$Lowest\ AAT = 2.8396080629425$

$Assoc = Fully\ Associative$



1. For all cache sizes the lru replacement policy curve yields lower aat values than fifo because it is consistently lower than the fifo curve as cache size increases the gap between the lru and fifo curves narrows but lru still performs better than fifo this shows that the lru replacement policy yields the best lowest aat values when compared to fifo lru approximates optimal replacement by evicting the block that hasn't been used in a while which reduces the chance that such blocks will be missed in the future conversely fifo is less optimal because it can eliminate blocks that could be accessed once more leading to a rise in misses and aat

Discussions for Graph #3:

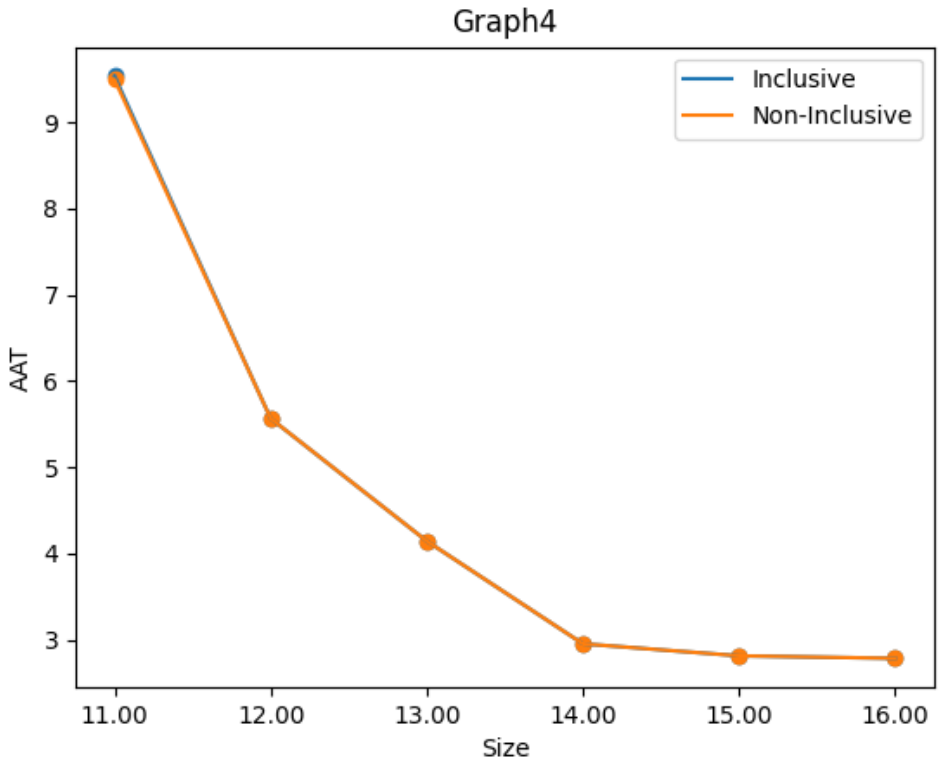
The configuration for which lowest AAT is seen –

$Lowest\ AAT$ = $\mathbf{2.86925004959106}$

$L1\ Cache\ Size$ = $32KB$

$L1\ Assoc$ = $constant(4)$

$Replacement\ Policy$ = $Optimal\ Replacement\ Policy$



1. for various 8-way l2 sizes with lru aat vs log2l2 size for both features and for 1kb 4-way l1 because hits in both caches are guaranteed a higher l2 size reduces aat for all sizes the inclusive curve is lowered non-inclusive does not provide such a guarantee which increases the risk of l1 misses at small l2 sizes at larger sizes the disparity increases marginally as inclusive keeps benefiting from more blocks by enforcing coherence to lower overall misses total yields are less than aat.

Discussions for Graph #4:

The better AAT (lowest) was observed in Graph #4 was for 4KB, L2 Cache and the value was,

AAT = 9.281644

Configurations for the lowest AAT:

$L2\ Cache\ Size$ = $4KB$

$Inclusive\ Property = Non - inclusive\ property$ Configuration for the