University of Central Florida

Department of Computer Science

CDA 5106: Fall 2023

Machine Problem 1: Cache Design, Memory Hierarchy Design

by

Mukund Dhar

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

Student's electronic signature: Mukund Dhar

Note: Using the GCC benchmark for all experiments, that is, use 'traces/gcc_trace.txt' file.

1. L1 cache exploration: SIZE and ASSOC

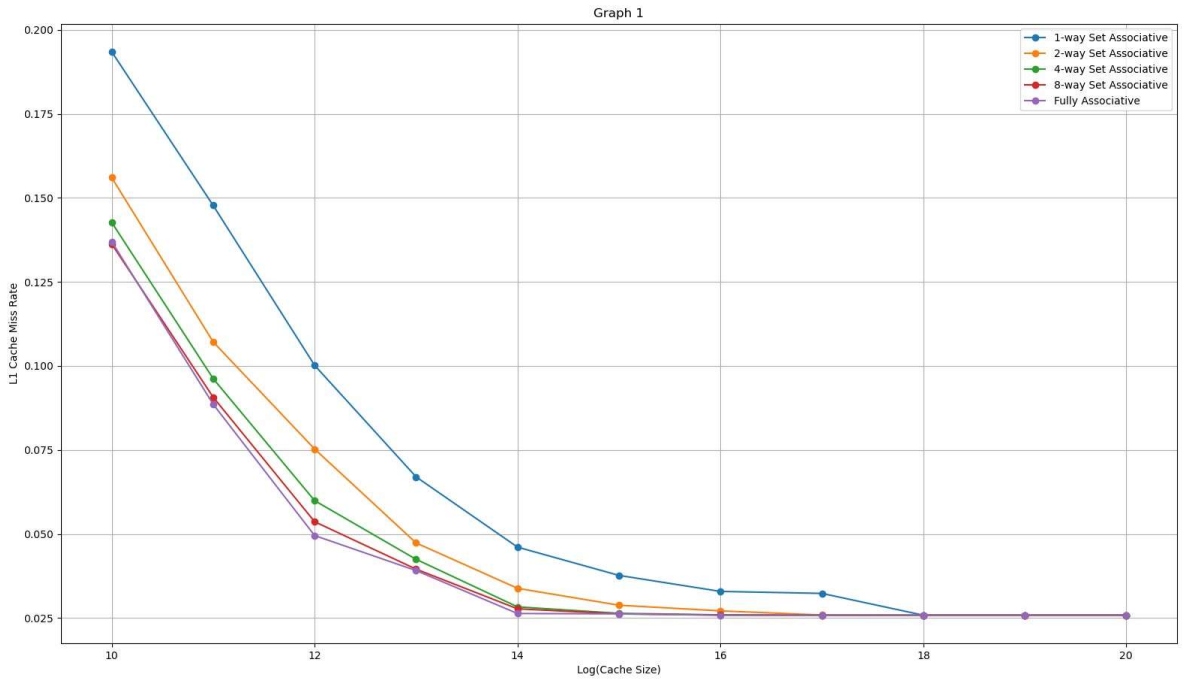
   GRAPH 1: L1 Cache Miss Rate vs Log (Cache Size)

   For this experiment:
   - L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
   - L2 cache: None.
   - Replacement policy: LRU
   - Inclusion property: non-inclusive

Graph 1: L1 Cache Miss Rate vs Log2(Cache Size)						
cache size	**log (cache size)**	**1-way Associative**	**2-way Associative**	**4-way Associative**	**8-way Associative**	**Fully Associative**
1024	**10**	0.19346	0.15603	0.1427	0.13627	0.13696
2048	**11**	0.14774	0.10714	0.09622	0.09069	0.0886
4096	**12**	0.10017	0.07528	0.05992	0.05365	0.04954
8192	**13**	0.067	0.04734	0.04247	0.03954	0.03912
16384	**14**	0.04609	0.03384	0.02832	0.02774	0.02634
32768	**15**	0.03768	0.02881	0.0264	0.02625	0.02624
65536	**16**	0.03292	0.02713	0.02595	0.02589	0.02583
131072	17	0.03233	0.0259	0.02582	0.02582	0.02582
262144	18	0.02584	0.02584	0.02582	0.02582	0.02582
524288	19	0.02584	0.02582	0.02582	0.02582	0.02582
1048576	20	0.02584	0.02582	0.02582	0.02582	0.02582

Discussion:

1. We can see that the miss rates are more for lower associativity. For a given associativity, we can the miss rate decreases initially as the cache size increases, but then plateaus after a point. For a given cache size, as we increase the associativity, the miss rate decreases. This also is valid up to a point, here, after log (cache size) = 18, we see the miss rate becomes constant.
2. The compulsory miss rate represents the percentage of cache misses caused by accessing data or instructions for the first time. From the graph, we can see that compulsory miss rate can be estimated when the cache size is significantly more, and it comes around to be **0.02582**.
3. Depending on the cache size and associativity, conflict miss rates vary.
   Prior to calculating conflict miss rates, capacity misses can be determined by setting the conflict miss rate for fully associative to be 0.
   Conflict miss rates then can be computed for various associativity as the capacity miss rate and mandatory misses remain constant for associativity of the same size.



GRAPH 2: AAT vs Log (Cache Size)

For this experiment:

- L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
- L2 cache: None.
- Replacement policy: LRU
- Inclusion property: non-inclusive

Miss Penalty = 100 ns

Here,

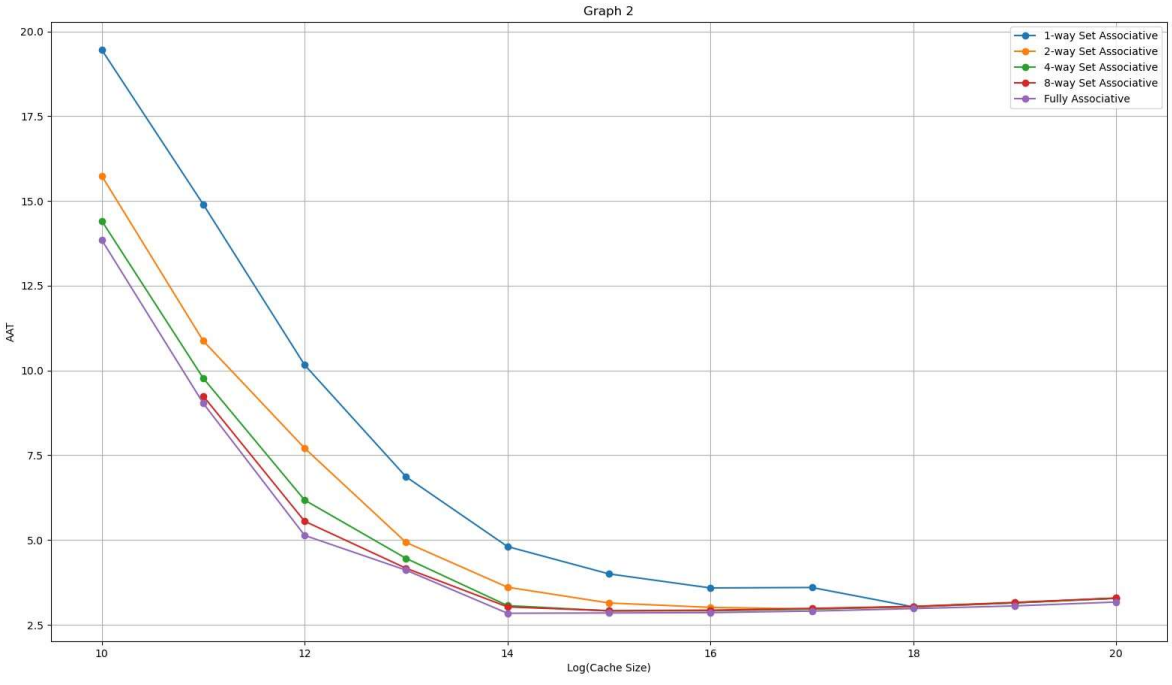
AAT = (Miss Rate * Miss Penalty) + Hit Time

Discussion:

1. For a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, we can see that the configuration having full associativity and cache size = 16,384 bytes (16 KB) yields the best (i.e., lowest) AAT, that is 2.839608.

   The trend is that fully associative configuration is usually better in terms of having lower AAT values, but we can see that after a certain increase in cache size, AAT values for all becomes similar.

AAT						
cache size	log (cache size)	1-way Associative	2-way Associative	4-way Associative	8-way Associative	Fully Associative
1024	**10**	19.460797	15.743329	14.41682		13.851484
2048	**11**	14.90309	10.875691	9.776496	9.249686	9.036515
4096	**12**	10.164005	7.709131	6.177685	5.554065	5.136948
8192	**13**	6.86383	4.928195	4.458173	4.166911	4.110581
16384	**14**	4.807417	3.607917	3.065936	3.028354	2.839608
32768	**15**	4.001353	3.143446	2.91125	2.913511	2.84874
65536	**16**	3.586627	3.013727	2.914481	2.930213	2.859281
131072	**17**	3.5998	2.964603	2.96228	2.983236	2.904486
262144	**18**	3.027812	3.029929	3.039685	3.040925	2.978009
524288	**19**	3.147451	3.149744	3.146418	3.160177	3.057728
1048576	**20**	3.28338	3.288046	3.281607	3.287819	3.170474



2. Replacement policy study

GRAPH 3: AAT vs Log (Cache Size)

For this experiment:

- L1 cache: SIZE is varied, ASSOC = 4, BLOCKSIZE = 32.
- L2 cache: None.
- Replacement policy: varied
- Inclusion property: non-inclusive

Miss Penalty = 100 ns

Here,

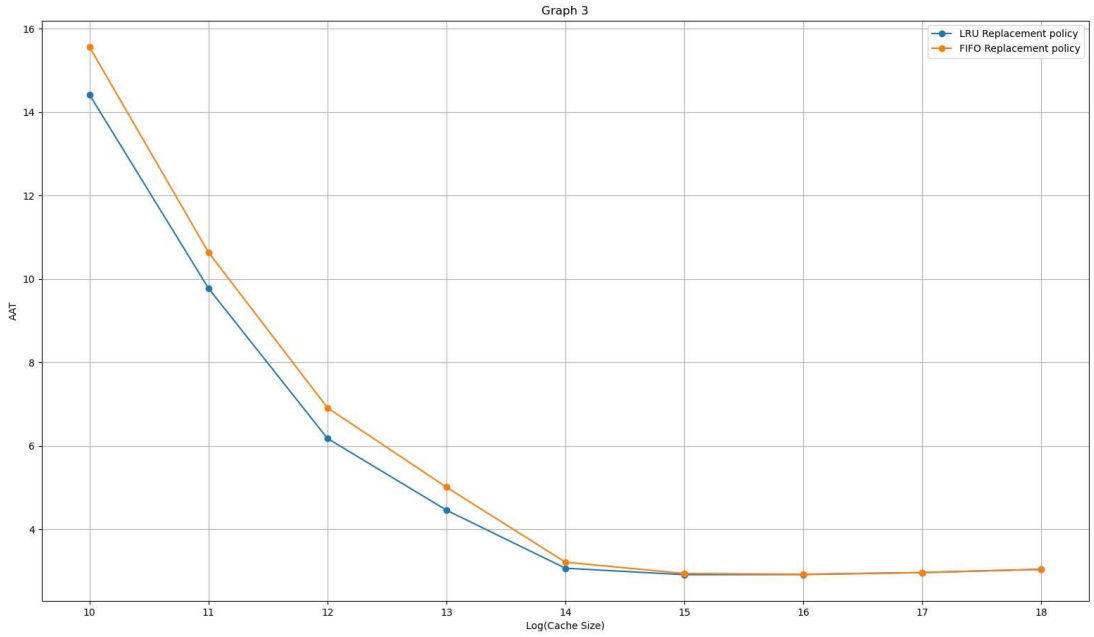
AAT = (Miss Rate * Miss Penalty) + Hit Time

Discussion:

For both LRU and FIFO, we can see that AAT decreases initially as we increase the cache size, but then plateaus. It also becomes similar for both LRU and FIFO as the cache size is more.

The replacement policy yields the best (i.e., lowest) AAT is **LRU, better than FIFO**.

AAT			
cache size	**log (cache size)**	**LRU**	**FIFO**
1024	**10**	14.41682	15.56182
2048	**11**	9.776496	10.637496
4096	**12**	6.177685	6.912685
8192	**13**	4.458173	5.009173
16384	**14**	3.065936	3.210936
32768	**15**	2.91125	2.93925
65536	**16**	2.914481	2.922481
131072	**17**	2.96228	2.96228
262144	**18**	3.039685	3.039685



3. Inclusion property study

GRAPH 4: AAT vs Log (Cache Size)

For this experiment:

- L1 cache: SIZE = 1KB, ASSOC = 4, BLOCKSIZE = 32.
- L2 cache: SIZE = 2KB – 64KB, ASSOC = 8, BLOCKSIZE = 32.
- Replacement policy: LRU
- Inclusion property: varied

Miss Penalty = 100 ns

AAT = HTL1 + MRL1*(HTL2 + (MRL2* Miss Penalty))

where HTL1 is Hit Time of the L1 cache and HTL2 is Hit Time of the L2 cache while MRL1 and MRL2 are miss rates of the respective L1 and L2 caches.

Discussion:

For both inclusion properties, we can see that as the cache size increases, the AAT values decrease initially and then plateaus around a value. For both inclusion properties, the AAT values become similar after a certain cache size value.

Initially, for smaller cache sizes, non-inclusive is giving better AAT values. However, for most of the cache sizes later on, both inclusion properties are giving similar AAT values.

AAT			
cache size	**log (cache size)**	**non-inclusive**	**inclusive**
2048	**11**	9.158608402	9.600056512
4096	**12**	5.541802556	5.581874007
8192	**13**	4.14019545	4.146223303
16384	**14**	2.953123096	2.953123096
32768	**15**	2.81298556	2.81298556
65536	**16**	2.784517195	2.784517195