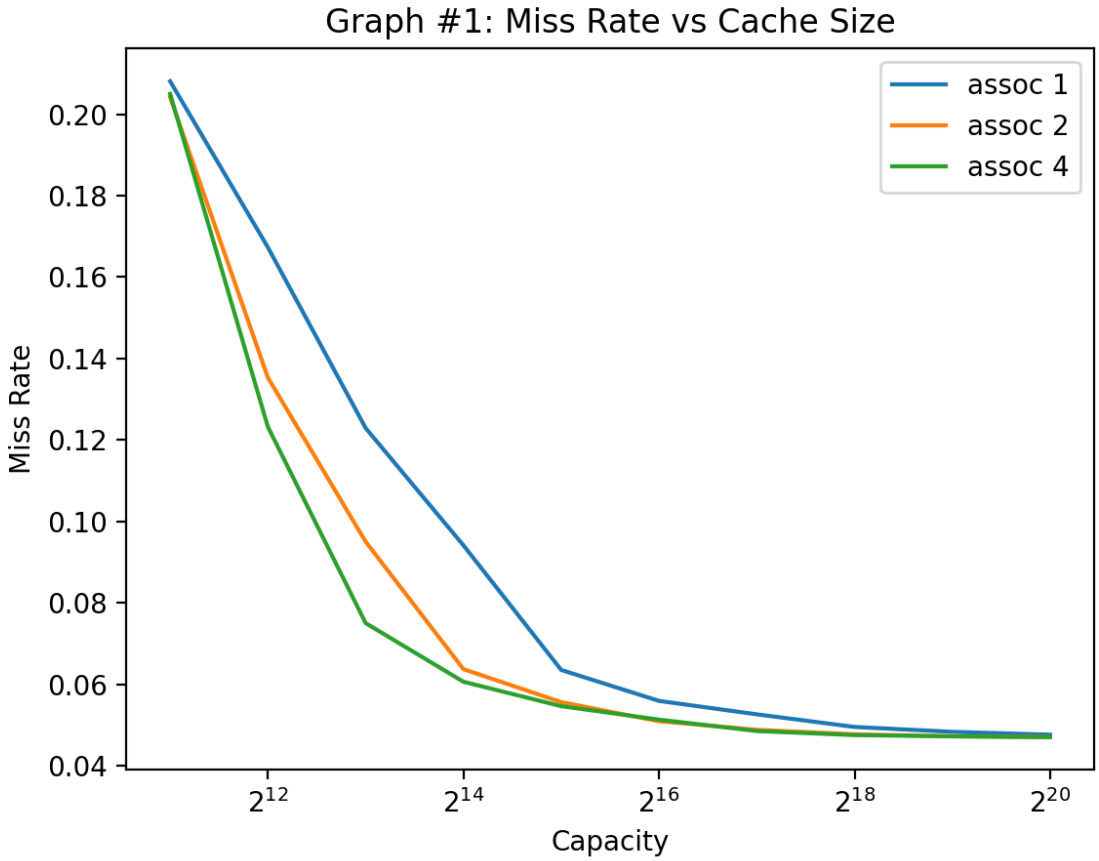


1. What is the smallest capacity that brings the miss rate to less than 10%?

The smallest capacity that brings the miss rate to less than 10% is 16KB.

2. What is the total memory requirement (i.e. maximum memory needed) for this address space? For this RAM size and your answer to question 1, what is the ratio of cache to memory size?

The total memory requirement is 4GB since we have 32 bit addresses and $2^{32} = 4GB$. The ratio of cache to memory size is $1: 2^{18}$

3. Today's processors generally have 32KB to 128KB first-level (L1) data caches. By what ratio does increasing the cache size from 16KB to 32KB reduce the miss rate? (2.0 would correspond to halving the miss rate; 1.0

would correspond to no change in miss rate; less than 1.0 would correspond to an increase in misses).
Miss rate(16KB) = .0941
Miss rate(32KB) = .0636
Miss rate(16KB)/Miss rate(32KB) = 1.483

Increasing the cache size from 16KB to 32KB reduces the miss rate by 1.483 times.

4. By what ratio does increasing the cache size from 32KB to 64KB reduce the miss rate?

Miss rate(32KB) = .0635
Miss rate(64KB) = .0559
Miss rate(32KB)/Miss rate(64KB) =1.138

Increasing the cache size from 16KB to 32KB reduces the miss rate by 1.138 times.

5. When deciding on the ideal cache size, engineers typically look for the "knee" of the curve. When considering various cache sizes, we want the point at which increasing *to* that size yields a great benefit, but increasing *beyond* that size yields far less benefit. What would you say is the ideal cache size for a direct-mapped cache?

32KB would be the ideal cache size for a direct-mapped cache because increasing from 16KB to 32KB decreases the miss rate significantly; however, increasing the cache size from 32KB to 64KB yields diminishing returns.

6. What is the smallest capacity that brings the miss rate of the 2-way set associative cache to less than 10%?

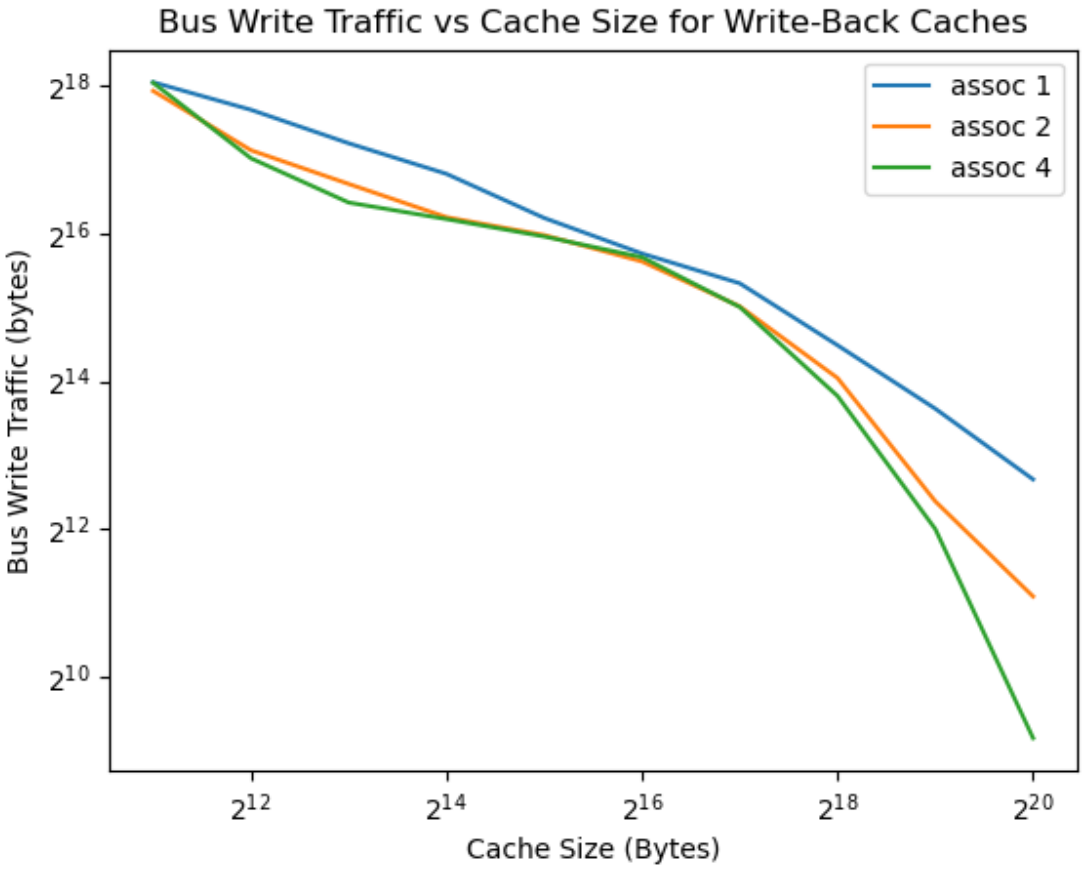
The smallest capacity that brings the miss rate of the 2-way set associative cache to less than 10% is 8KB.

7. What is the smallest capacity that brings the miss rate of the 4-way set associative cache to less than 10%?

The smallest capacity that brings the miss rate of the 4-way set associative cache to less than 10% is 8KB.

8. How large must the direct-mapped cache be before it equals or exceeds the performance of the 8 KB 4-way associative?

The direct-mapped cache has to have 32KB capacity before it exceeds the performance of the 8 KB 4-way associative.

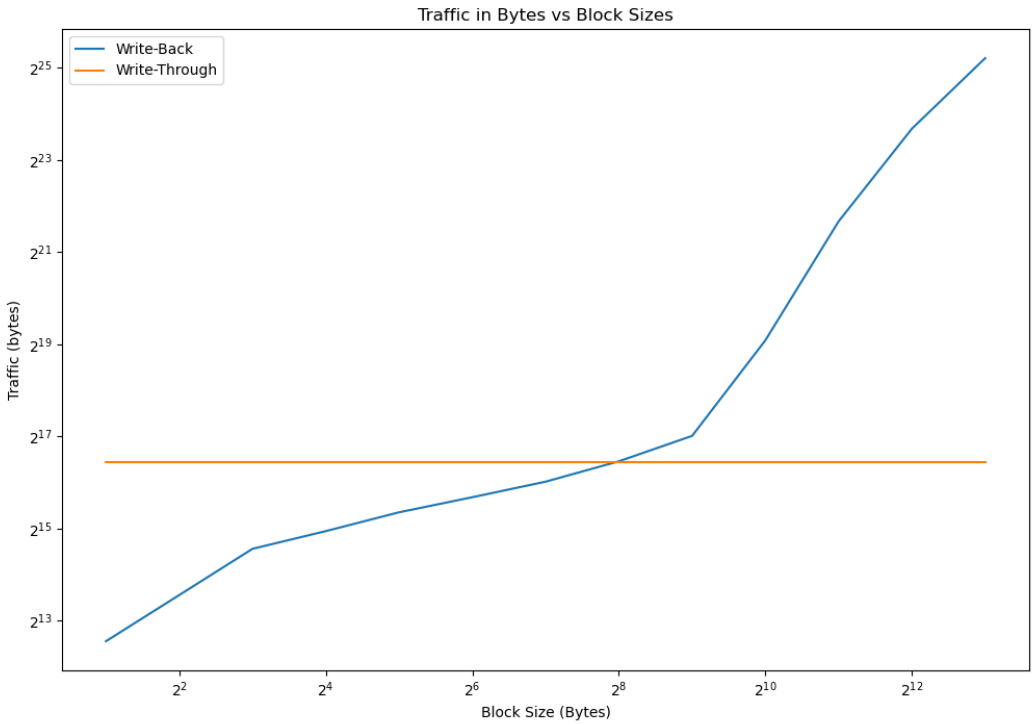


9. What happens to traffic as the capacity increases? Is there a pattern? If so, explain what you see.

As capacity increases, the bus write traffic decreases. The cache can store more lines before reaching capacity, reducing the frequency for write-backs and the need to fetch data from slower levels of the memory hierarchy. Also as capacity increases on the log scale, the traffic for all 3 associativity lines decreases roughly linearly and then the slope of the lines change at a similar point and all get steeper exponentially.

10. What role does associativity play in traffic generation? Is there a pattern? If so, explain what you see.

Associativity compared to direct mapping leads to lower traffic generation as there are multiple ways within a set where a block can be placed, which also reduces conflict misses. Also, higher associativity allows more flexibility for placing blocks within a cache, reducing the chance of evicting a block that may be needed shortly. For the 3 lines, the direct mapped is roughly linear (on the log scale graphs), and the other 2 are a little wavy with the assoc = 4 line being the most wavy.



11. At what block size do the two write policies generate approximately the same number of writes to the bus?

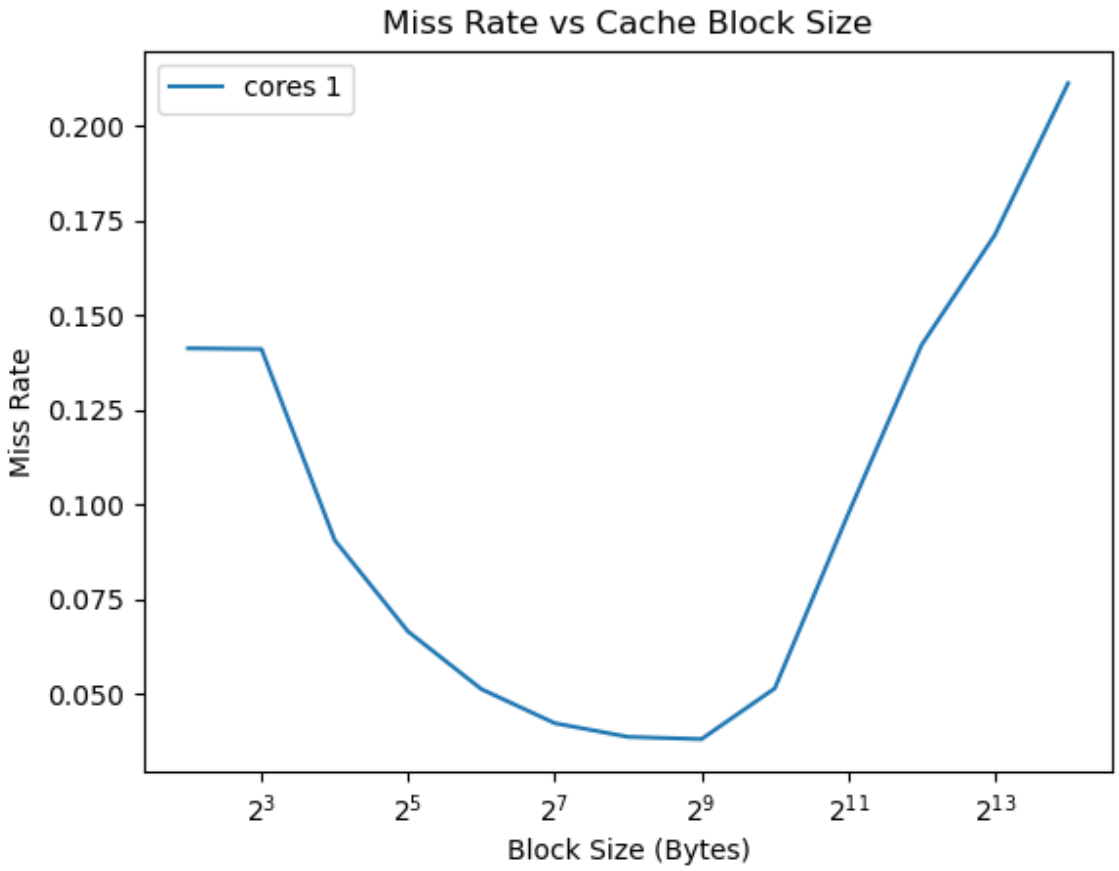
At 256 bytes, the two write policies generate approximately the same number of writes to the bus.

12. Explain the difference in traffic observed for smaller block sizes.

For write through data is simultaneously updated to the cache and memory, leading to constant traffic w.r.t block size. For write-back caches with low block size updates to memory are made only during eviction, and since block size is small, less memory is being written to during eviction.

13. Explain the difference in traffic observed for larger block sizes.

For write through data is simultaneously updated to the cache and memory, leading to constant traffic w.r.t block size. For write-back caches with low block size updates to memory are made only during eviction, and since block size is large, more memory is being written to during eviction.



14. Explain the observed miss rate associated with a small block size.

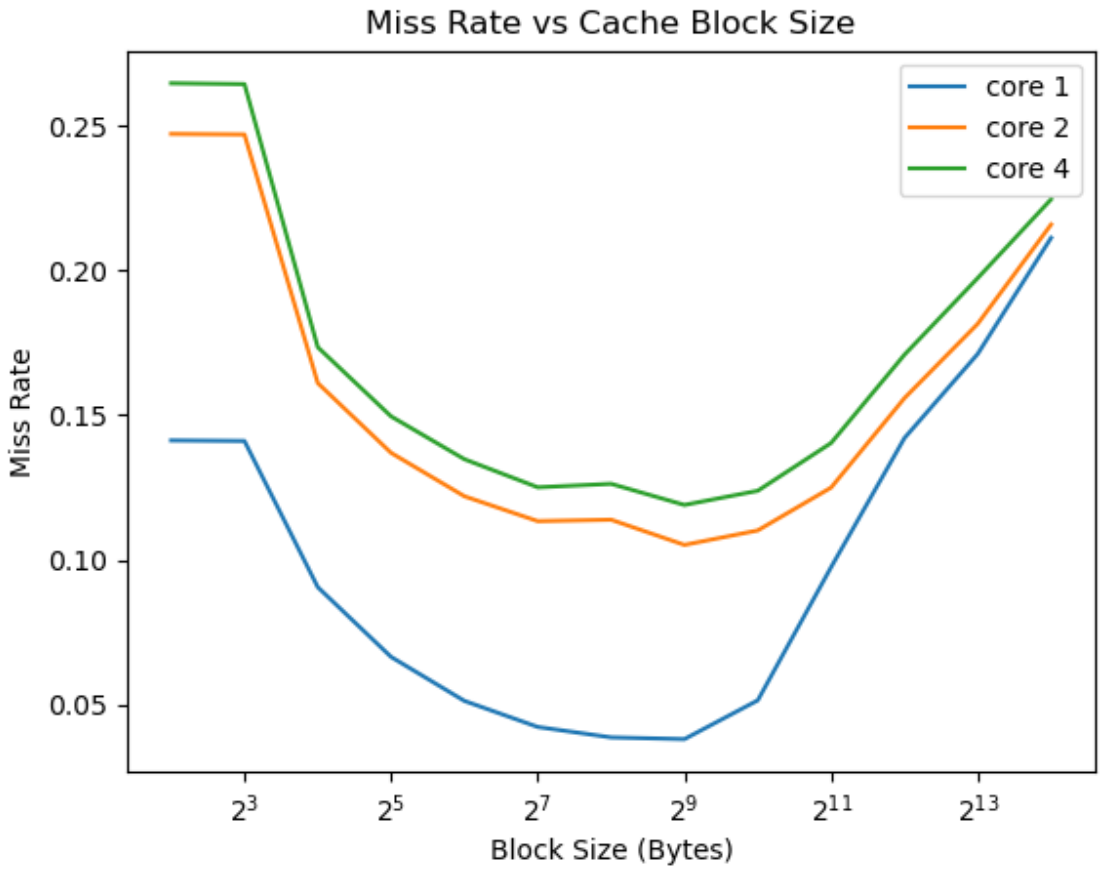
The miss rate is relatively high for smaller cache sizes according to the graph. With smaller block sizes, each line stores fewer blocks of data. Thus these blocks do not take as much advantage of spatial locality, and there may be more cold misses with less prefetching occurring.

15. Explain the observed miss rate associated with a large block size.

For the larger block sizes on this graph, the miss rate is also relatively high. There will be more conflict misses because fewer blocks can be stored in the cache, so there is a higher chance that a block will be displaced when too many addresses map to the same set.

16. What is the block size with the lowest miss rate?

Approximately 2^9th Bytes for our example.



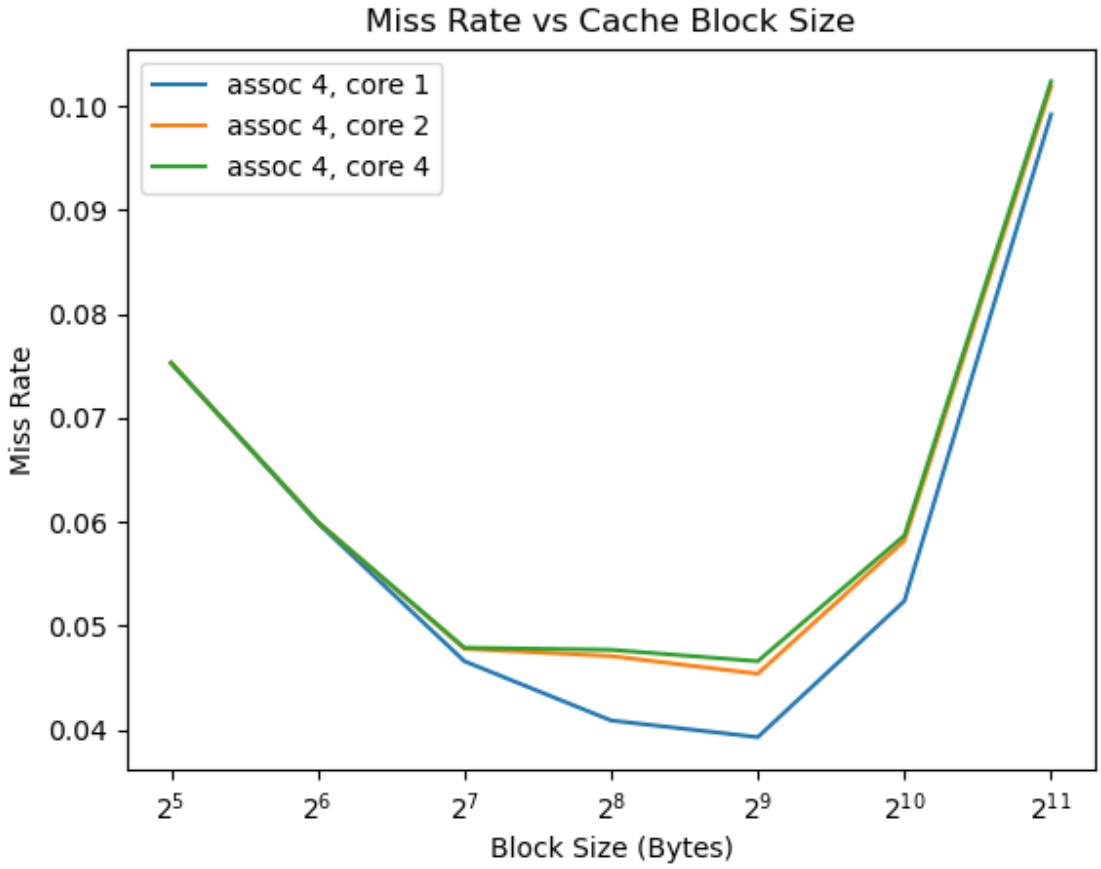
17. Why does the miss rate get worse with more cores?

In VI protocol, only one cached copy is allowed anywhere and shared read-only copies do not exist like in MSI protocol. Only one core will hold the valid data, so the more cores there are, the more contention there is over the valid data which leads to more misses.

18. If the miss rate is so bad, why would one use the VI protocol over no protocol?

Without any coherence protocol, multiple cores could potentially have different versions of the same data, leading to inconsistencies and errors in the program execution. By using a coherence protocol such as the VI protocol, the system ensures that all cores have a consistent view of the data, allowing caches to

"communicate" about the validity of the shared data. This ensures the correctness and reliability of the program execution.



19. For the 1 core trace, which has a lower miss rate: the VI protocol or the MSI protocol? Explain why.

The VI Protocol has a slightly lower miss rate than MSI (3.81% vs 3.93%). With the VI protocol you only miss when the data is invalid but MSI protocol you can have a miss when the data is invalid and also if you are trying to write to data in a shared state.

Thus, in a single core trace, if the workload is such that there is very little sharing of data between the core(s) and there exists a large number of writes to the cache, the VI protocol will have a lower miss rate as the invalidations sent by the MSI protocol could cause more overhead.

20. For the 4 core trace, which has a lower miss rate: the VI protocol or the MSI protocol? Explain why.

The MSI protocol has a lower miss rate than the VI protocol. With the VI protocol, the miss rate increases as the number of cores go up since the protocol only allows one cached copy of the valid data and having multiple cores increases the chance that data is invalidated by another core. However, the MSI protocol allows multiple "shared" copies of cache data to exist, therefore the miss rate does not increase as drastically as the VI protocol's when the number of cores increases.

21. For a 2 core trace, with a block size of 64B, what fraction of bus snoops by Core 0 are "hits" (i.e., the LD_MISS or ST_MISS on the bus is for a cache line that is currently valid in Core 0's cache.)

0.n_bus_snoops 5746

0.n_snoop_hits 180

180/5746 of bus snoops by Core 0 are "hits".

22. For a 4 core trace, with a block size of 64B, what fraction of bus snoops by Core 0 are "hits"?

0.n_bus_snoops 17536

0.n_snoop_hits 518

518/17536 of bus snoops by Core 0 are "hits".