P5: Build-a-Cache Writeup

Computer Systems Organization and Programming

CS3410 2022 Spring

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 1: Cache Tag/Index/Offset Calculations

Two-way set associative 32KB cache with 64B blocks (that is: A=2, B=64B, C=32KB):

Number of offset bits in the address: 6B

Number of sets in the cache: 256

Total number of cache lines in the cache: 512

Number of index bits in the address: 8B

Number of tag bits in the address: 18B

Task 2: Generalizing the Calculations

- A: Associativity (number of ways per set)
- B: Block size (size of a single block, a block is also referred to as a cache line)
- C: Capacity or cache size (total amount of data in the cache)

Number of offset bits in the address: $\log(B)$

Number of sets in the cache: $\frac{C}{A \cdot B}$

Total number of cache lines in the cache: $\frac{C}{B}$

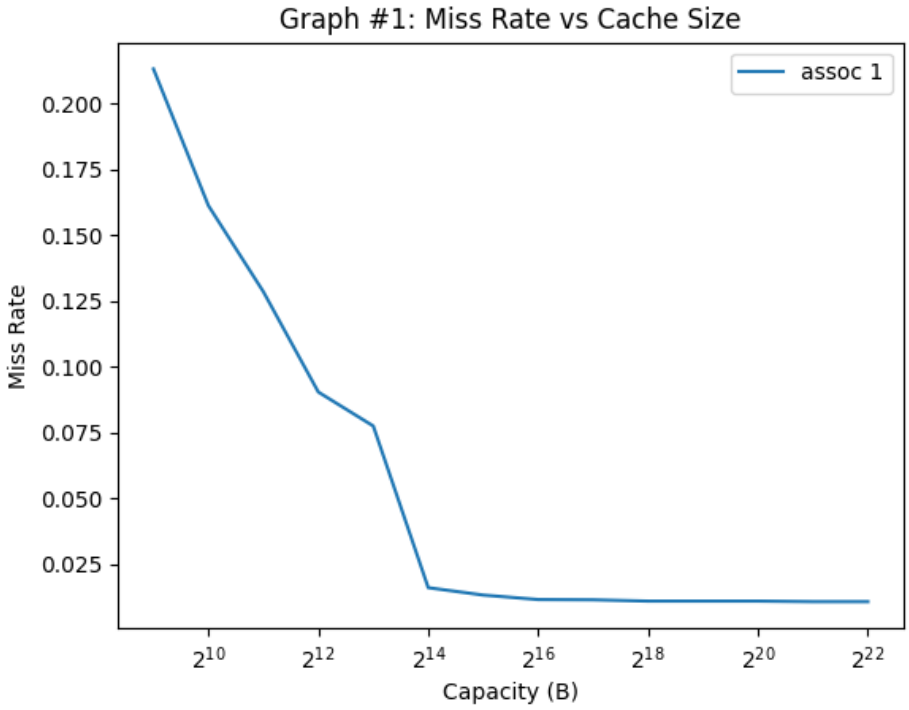
Number of index bits in the address: $\log(C) - \log(A) - \log(B)$

Number of tag bits in the address: $32 + \log(A) - \log(C)$

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 5: Implementing a DM Cache to Explore Hit Rate vs. Capacity

Preliminary Experiment 1 graph:



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% appears to be 2^{12} bytes, or in other words, 4KiB.

2. What is the smallest capacity that brings the miss rate to less than 5%?

   The smallest capacity that brings the miss rate to less than 5% appears to be 2^{14} bytes, or in other words, 16KiB.

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

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 for 16KB: 0.01620

   Miss rate for 32KB: 0.01340

   Ratio: 1.208955

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

   Miss rate for 32KB: 0.01340

   Miss rate for 64KB: 0.01170

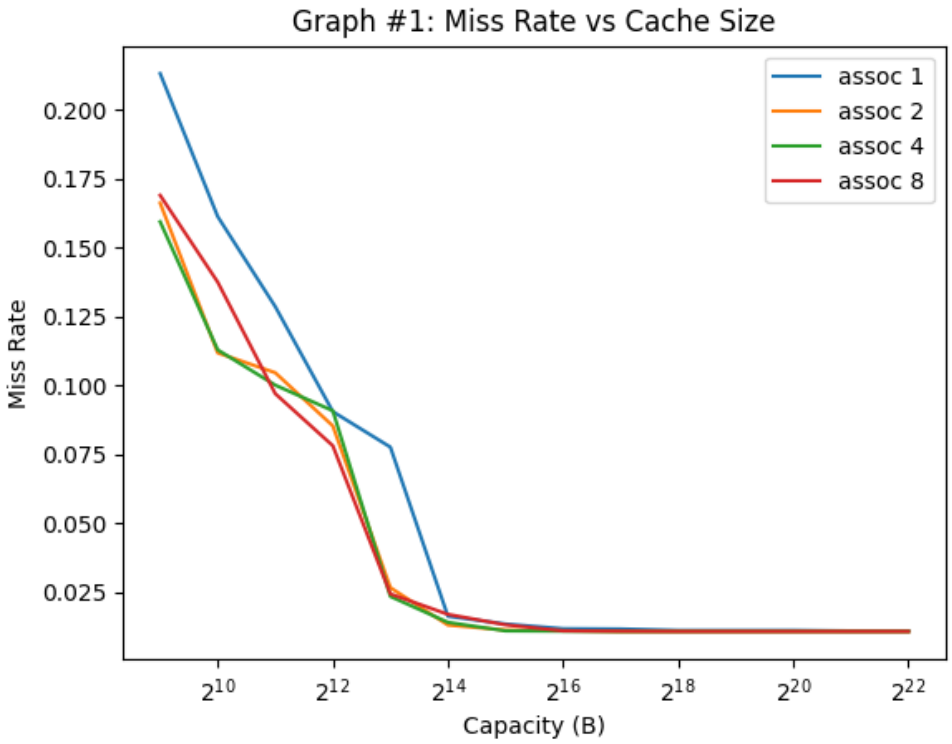
   Ratio: 1.145299

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?

   It would appear that the optimal cache size for a DM cache is 16 KiB.

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 6: Explore Hit Rate vs. Associativity



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 2-way to less than 10% appears to be 2^{12} bytes, or in other words, 4KiB.

7. What is the smallest capacity that brings the miss rate of the 8-way set associative cache to less than 10%?

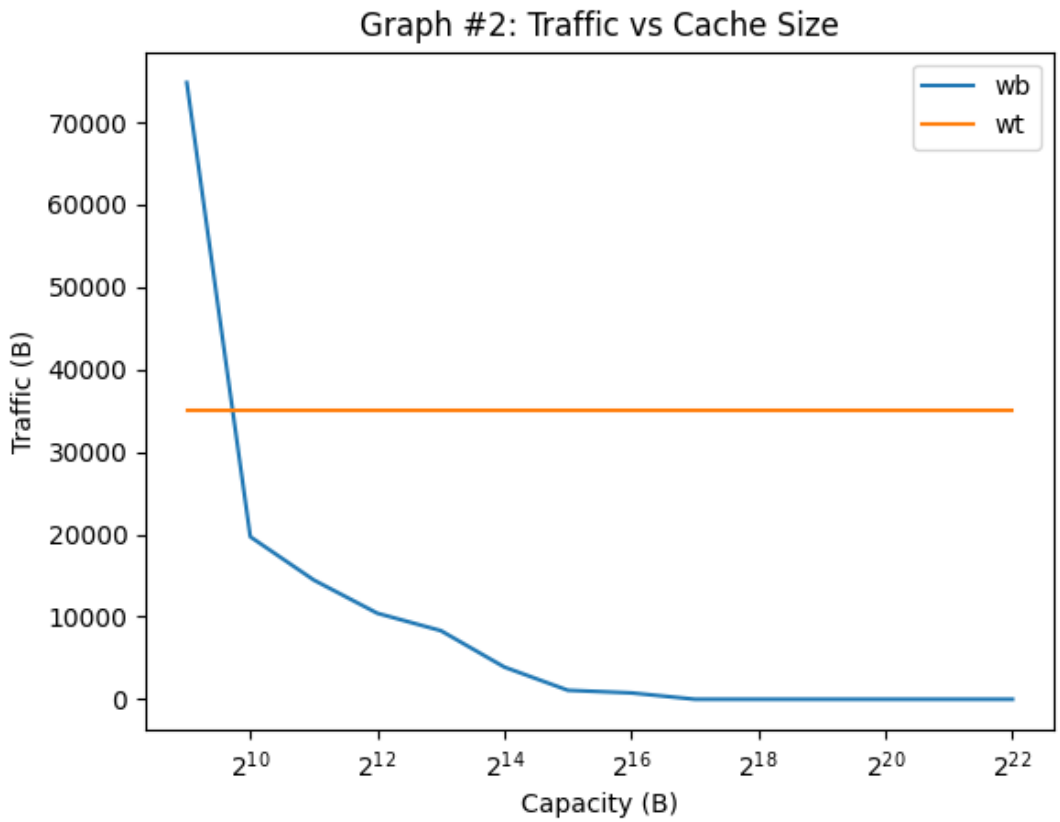
   The smallest capacity that brings the miss rate of 8-way to less than 10% appears to be 2^{11} bytes, or in other words, 2KiB.

8. How large must the direct-mapped cache be before it equals or exceeds the performance of the 1 KB 4-way assoc?

   The DM cache must be at least 2^{12} bytes in size, or 4KiB, to equal or exceed hit/miss performance of a 1KiB 4-way associative cache.

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 7: Tracking Traffic to Explore Write-Back vs. Write-Thru Caches



9. At what cache size do the two write policies generate approximately the same amount of writes to the bus?

   Two two write policies appear to generate approximately the same amount of writes somewhere in between 2^9 and 2^{10} bytes, or in other words somewhere between 512B and 1KiB.

10. Why does the difference between the two schemes diverge at small cache sizes?

    At small cache sizes, the probability of cache misses increases. Write-through always ensures that just the single address that needs to be updated is written back from cache to RAM, whereas write-back needs to write the entire line back on eviction. Therefore, if misses and evictions increase, the traffic of write-back can increase above the constant traffic of write-through.

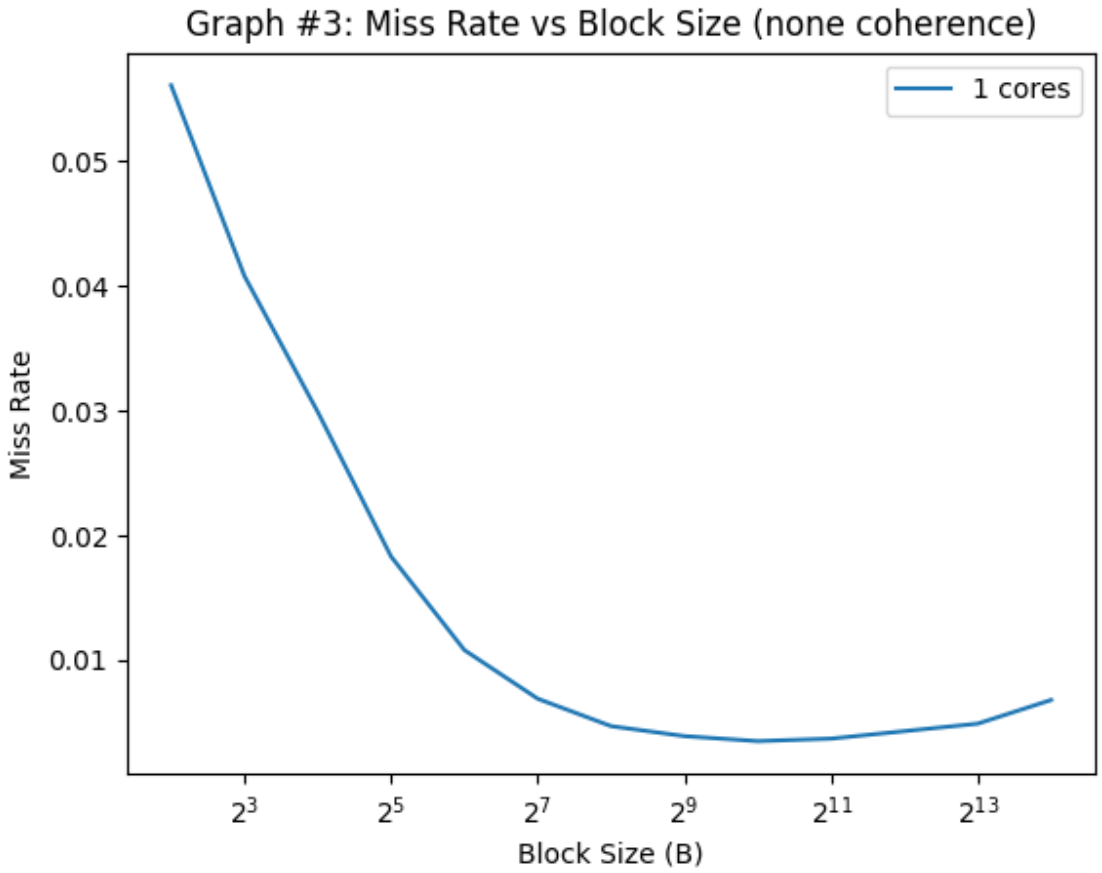
Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

11. Why does the difference between the two schemes diverge at large cache sizes?

    At large cache sizes, it's more probable that the entirety of the desired content is already loaded in cache, so write-backs never actually occur, as opposed to with write-throughs, which occur regularly.

Impact of different cache block sizes

Preliminary Experiment 3 graph:



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

    At small block size, the number of data that gets fetched to the cache is reduced significantly, making the miss rate go up. In other words, the tag bit will be longer, making the tag to be the same harder, because the spatial locality of data has been reduced.

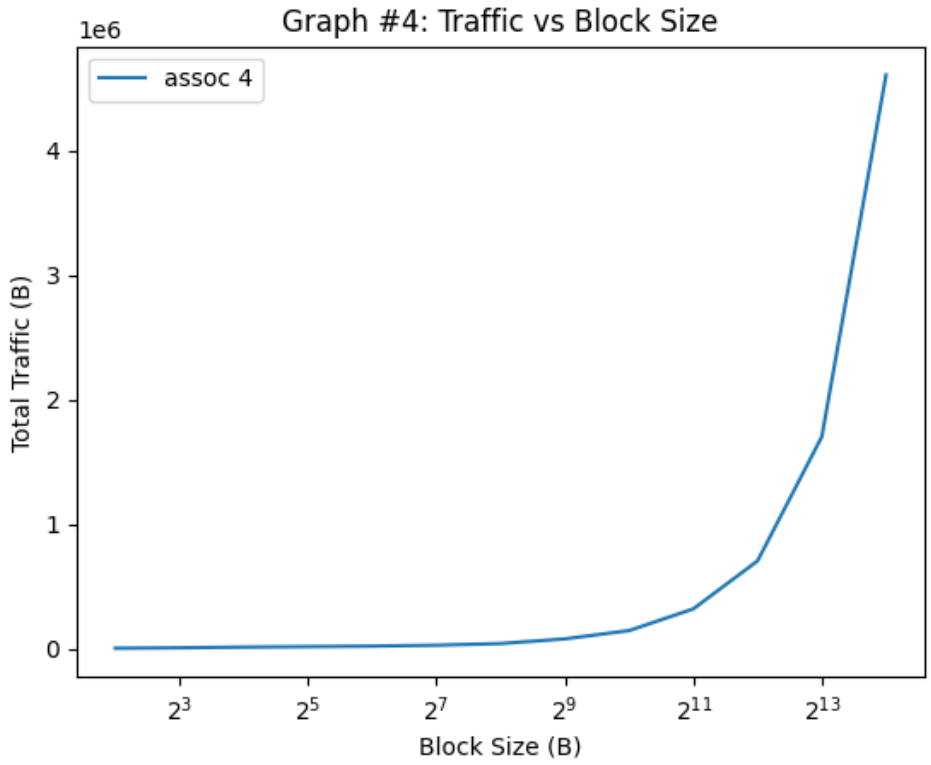
Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

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

    With large block sizes, the number of data that gets fetched to the cache increases, reducing the miss rate. In other words, the tag bit will be shorter, making the tag to be the same easier, because the spatial locality of data has been increased. However, there is a tick up because larger block sizes dominate the entire cache size so there are fewer lines in the cache, decreasing temporal locality.

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

    It is the lowest when the block size is 1KB.



15. What is the block size with the lowest total write-back traffic (transferred in + write-back transferred out)

    The block size with the lowest total write-back traffic is 4B.

16. What are the two sources of additional traffic as the block size grows? Explain why each grows.

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

The two sources of additional traffic as the block size grows come from misses and number of writebacks. The misses increase traffic as a larger block size corresponds with more coherence misses, so if the block size exponentially increases, then the additional traffic that corresponds with these misses will also increase. For number of writebacks, since the block size increases, a larger block size means more stores will be needed to evict the entire block and store it into memory (because the configuration is writeback and not write-through).

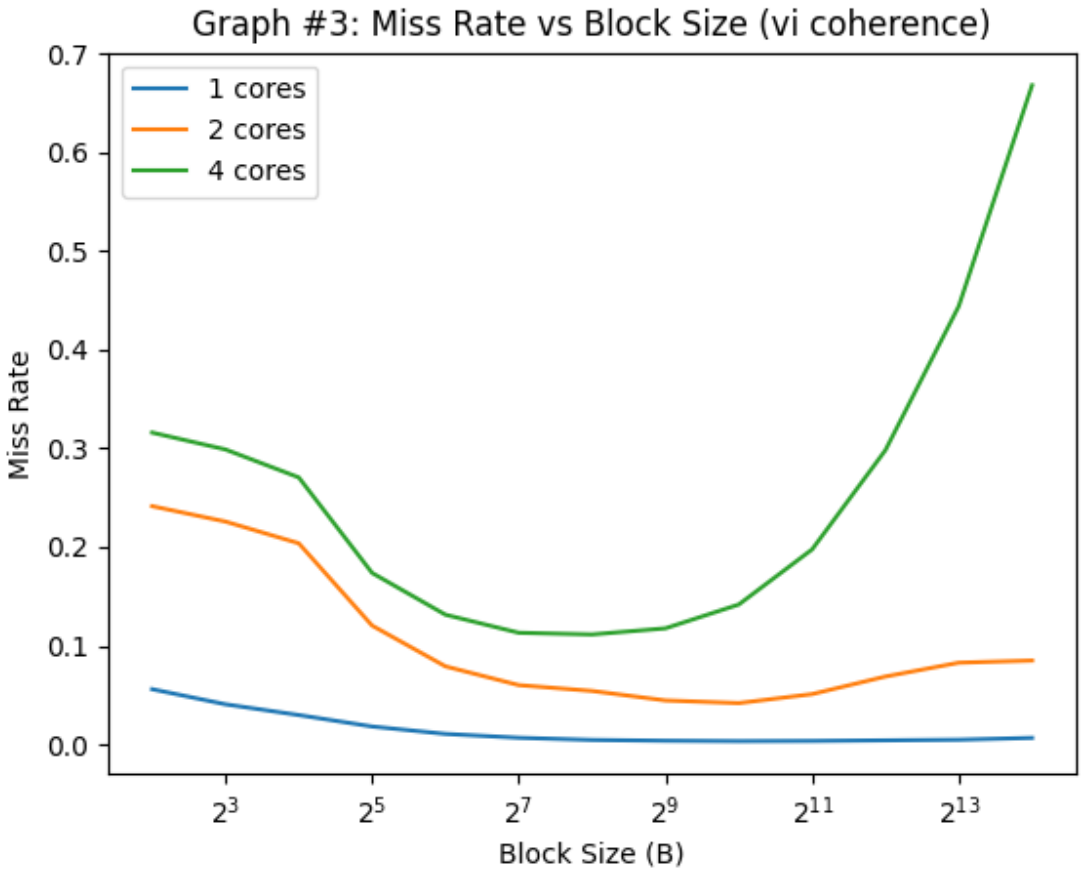
17. Given that current processors typically use, say, 64B blocks, which metric (miss rate or traffic) are today's caches designed to minimize?

    Traffic: Since they have a small block size, it is reasonable to assume that traffic is the priority in minimization.

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 9: VI protocol

Experiment 4 (Graph 3 final version)



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

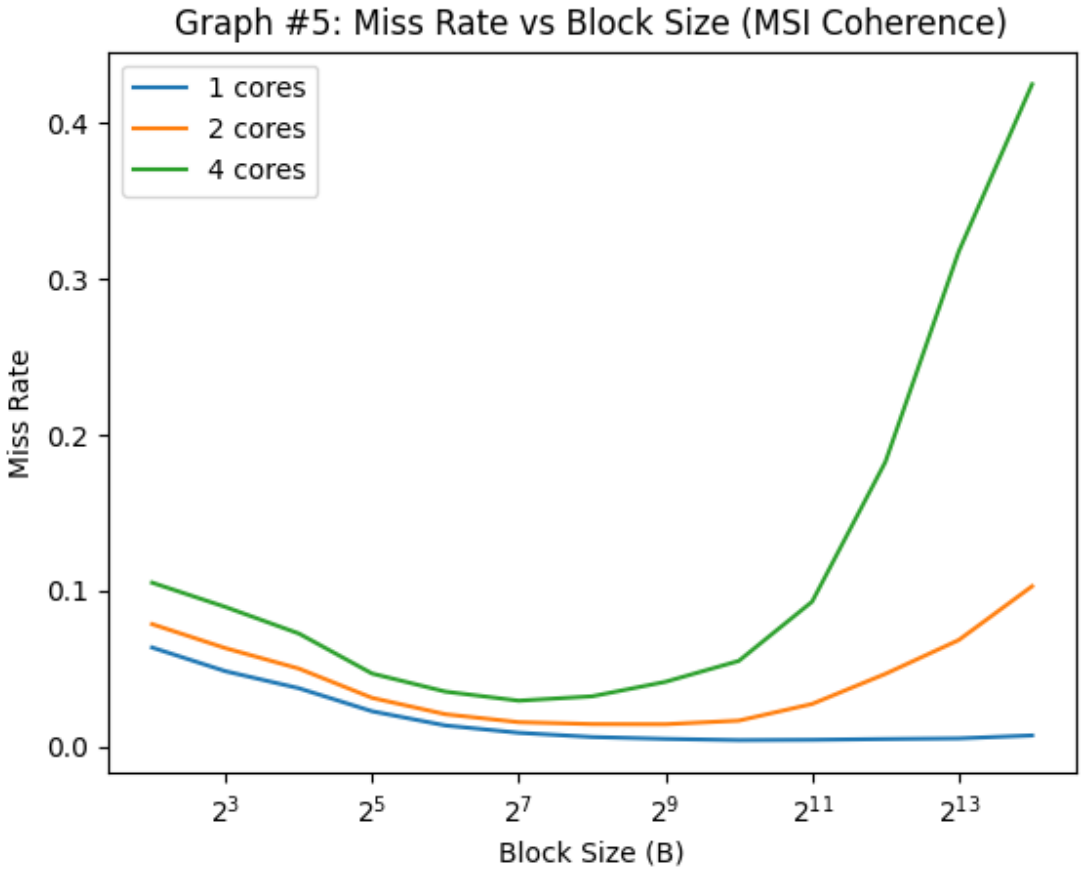
    The miss rate gets worse with more cores because accesses from other cores render data invalid to a given core, resulting in more misses as a result of other accesses. Cores are competing with each other for accesses.

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

    The VI protocol ensures that the information in the caches are coherent across cores (meaning that we trade off speed for the information in the caches actually being correct, which is very important).

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

Task 10: MSI protocol



20. Is there ever a scenario in which the VI protocol would be preferable to the MSI protocol? If so, provide that scenario. If not, explain why not.
    Can you think of a trace or type of workload for which the VI protocol would yield lower miss rates than the MSI protocol? If so, describe that workload/trace/scenario/use case. Otherwise, explain why this would not happen.
    Another possible interpretation: Can you think of a real world situation in which the VI protocol would be preferable to the MSI protocol?

    Note: "working on project 5" is not an acceptable scenario.

    If you are using a simpler multi-core processor, MSI would take more circuitry to implement and could potentially be more of a performance bottleneck than using VI. Dual-core processors also don't appear to differ in performance significantly with MSI

Dave Jung (sj597)
Jerry Xu (jjx6)
Kevin Zhen (kz72)

versus VI, so this would make sense for a very simple dual-core processor where you are trying to save money on your traces/chip fab costs.

Another scenario could be reducing upgrade misses - VI cannot possibly experience an upgrade miss, whereas MSI might on two independent workloads. Therefore, VI shares some of the advantages of MESI over MSI.

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.

    51 snoop hits out of 360 bus snoops, for a result of 14.17% hits.

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

    327 snoop hits out of 1041 bus snoops, for a result of 31.41% hits.

23. What is one characteristic of the workload (visible in statistics in the output) that reconciles the high snoop hit rate and the low miss rates in Graph #5?

    Temporal and spatial locality is very high in the workload’s access. High snoop hit rate and low miss rate means that the cache does not miss and even if a core misses, the data is stored in another core. Therefore, we can assume that the locality was high, allowing the cache to perform well.