Lecture 14: Hardware Approaches for Cache Optimizations

Cache performance metrics, reduce miss rates, improve hit time, reduce miss penalty

Cache Performance Metrics

- Cache miss rate: number of cache misses divided by number of accesses
- Cache hit time: the time between sending address and data returning from cache
- Cache miss latency: the time between sending address and data returning from next-level cache/memory
- Cache miss penalty: the extra processor stall caused by next-level cache/memory access

Cache Performance Metrics

- Calculate cache impact on processor performance

$$\text{CPU time} = \text{IC} \times (\text{CPI}_{\text{execution}} + \text{CPI}_{\text{mem\_stall}}) \times \text{Cycle Time}$$

$$\text{CPI}_{\text{mem\_stal}} = \text{Memory Inst Frequency} \times \text{Miss Rate} \times \text{Miss Penalty}$$

- Calculate average memory access time (AMAT)

$$\text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

Note: Load and store are different!

Cache Performance for OOO Processors

- Very difficult to define miss penalty to fit in this simple model, in the context of OOO processors
  - Consider overlapping between computation and memory accesses
  - Consider overlapping among memory accesses for more than one misses
- We may assume a certain percentage of overlapping
  - In practice, the degree of overlapping varies significantly between
  - There are techniques to increase the overlapping, making the cache performance even unpredictable

Cache Optimizations

Total cache size: Determines chip area and number of transistors

Performance factors:
Miss rate, miss penalty, and hit time

Organization:
- Set Associativity and block size
- Multi-level organizations
- Auxiliary structures, e.g., to predict future accesses
- Main memory and memory interface design
- Many more …

Software Approaches
- Optimize memory access patterns
- Software prefetching
- Many more …

Improving Cache Performance

1. Reducing miss rates
   - Larger block size
   - larger cache size
   - higher associativity
   - way prediction
   - Pseudoassociativity
   - compiler optimization
2. Reducing miss penalty
   - Multilevel caches
   - critical word first
   - read miss first
   - merging write buffers
   - victim caches
3. Reducing miss penalty or miss rates via parallelism
   - Reduce miss penalty or miss rate by parallelism
   - Non-blocking caches
   - Hardware prefetching
   - Compiler prefetching
4. Reducing cache hit time
   - Small and simple caches
   - Avoiding address translation
   - Pipelined cache access
   - Trace caches

Classifying cache misses

- Classifying misses by causes (3Cs)
  - Compulsory—To bring blocks into cache for the first time. Also called cold start misses or first reference misses. *(Misses in even an Infinite Cache)*
  - Capacity—Cache is not large enough such that some blocks are discarded and later retrieved. *(Misses in Fully Associative Size X Cache)*
  - Conflict—For set associative or direct mapped caches, blcoks can be discarded and later retrieved if too many blocks map to its set. Also called collision misses or interference misses. *(Misses in N-way Associative, Size X Cache)*
- More recent, 4th "C":
  - Coherence - Misses caused by cache coherence. To be discussed in multiprocessor

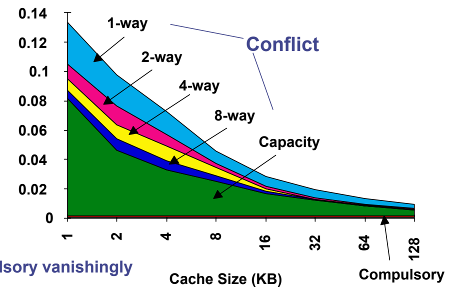
Cache Organization

Cache size, block size, and set associativity

Other terms: cache set number, cache blocks per set, and cache block size

- How do they affect miss rate?
  - Recall 3Cs: Compulsory, Capacity, Conflict cache misses?
- How about miss penalty?
- How about cache hit time?

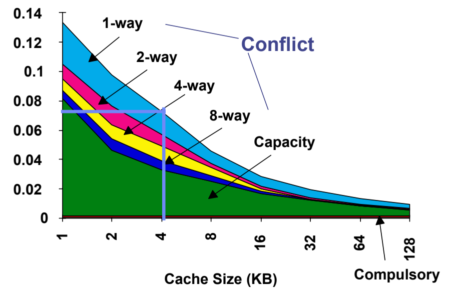
3Cs Absolute Miss Rate (SPEC92)



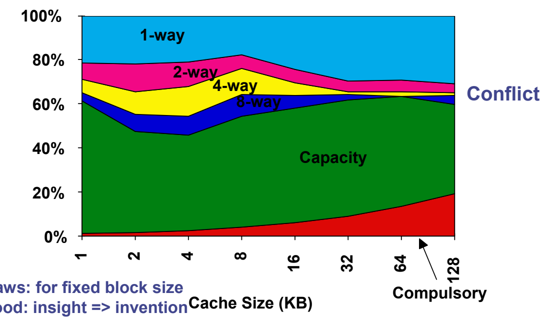
Compulsory vanishingly small

2:1 Cache Rule

miss rate 1-way associative cache size X
= miss rate 2-way associative cache size X/2

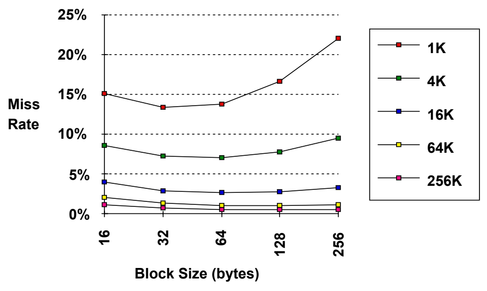


3Cs Relative Miss Rate



Flaws: for fixed block size
Good: insight => invention

Larger Block Size?



Higher Associativity?

- 2:1 Cache Rule:
  - Miss Rate DM cache size N - Miss Rate 2-way cache size N/2
- Beware: Execution time is only final measure!
  - Will Clock Cycle time increase?
  - Hill [1988] suggested hit time for 2-way vs. 1-way external cache +10%, internal + 2%
- Jouppi's Cacti model: estimate cache access time by block number, block size, associativity, and technology
  - Note cache access time also increases with cache size!

Example: Avg. Memory Access Time vs. Miss Rate

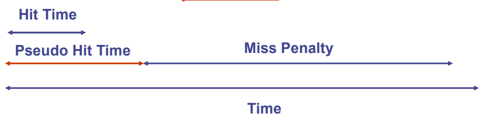
- Example: assume CCT = 1.10 for 2-way, 1.12 for 4-way, 1.14 for 8-way vs. CCT direct mapped

Cache Size	Associativity			
(KB)	1-way	2-way	4-way	8-way
1	2.33	2.15	2.07	2.01
2	1.98	1.86	1.76	1.68
4	1.72	1.67	1.61	1.53
8	1.46	1.48	1.47	1.43
16	1.29	1.32	1.32	1.32
32	1.20	1.24	1.25	1.27
64	1.14	1.20	1.21	1.23
128	1.10	1.17	1.18	1.20

(Red means A.M.A.T. not improved by more associativity)

Pseudo-Associativity

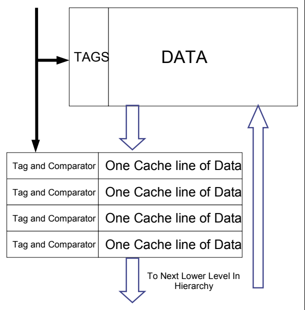
- How to combine fast hit time of Direct Mapped and have the lower conflict misses of 2-way SA cache?
- Divide cache: on a miss, check other half of cache to see if there, if so have a pseudo-hit (slow hit)



- Drawback: CPU pipeline is hard if hit takes 1 or 2 cycles
  - Better for caches not tied directly to processor (L2)
  - Used in MIPS R1000 L2 cache, similar in UltraSPARC

Victim Cache

- How to combine fast hit time of direct mapped yet still avoid conflict misses?
- Add buffer to place data discarded from cache
- Jouppi [1990]: 4-entry victim cache removed 20% to 95% of conflicts for a 4 KB direct mapped data cache
- Used in Alpha, HP machines



Multi-level Cache

- Add a second-level cache
- L2 Equations

AMAT = Hit Time$_{L1}$ + Miss Rate$_{L1}$ x Miss Penalty$_{L1}$
Miss Penalty$_{L1}$ = Hit Time$_{L2}$ + Miss Rate$_{L2}$ x Miss Penalty$_{L2}$
AMAT = Hit Time$_{L1}$ + Miss Rate$_{L1}$ x (Hit Time$_{L2}$ + Miss Rate$_{L2}$ × Miss Penalty$_{L2}$)

- Definitions:
  - Local miss rate— misses in this cache divided by the total number of memory accesses to this cache (Miss rate$_{L2}$)
  - Global miss rate—misses in this cache divided by the total number of memory accesses generated by the CPU (Miss Rate$_{L1}$ x Miss Rate$_{L2}$)
  - Global miss rate is what matters to overall performance
  - Local miss rate is factor in evaluating the effectiveness of L2 cache

Local vs. Global Miss Rates

Example:

- For 1000 inst., 40 misses in L1, 20 misses in L2
- L1 hit 1 cycle, L2 hit 10 cycles, miss 100
- 1.5 memory references per instruction

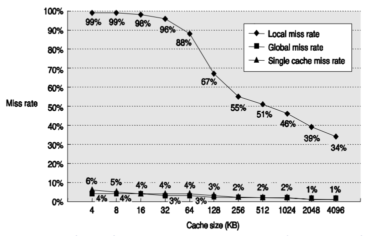
Ask: Local miss rate, AMAT, stall cycles per instruction, and those without L2 cache

With L2 cache
- Local miss rate = 50%
- AMAT=1+4%X(10+50%X 100)=3.4
- Average Memory Stalls per Instruction=(3.4-1.0)x1.5=3.6

Without L2 cache
- AMAT=1+4%X100=5
- Average Memory Stalls per Inst=(5-1.0)x1.5=6

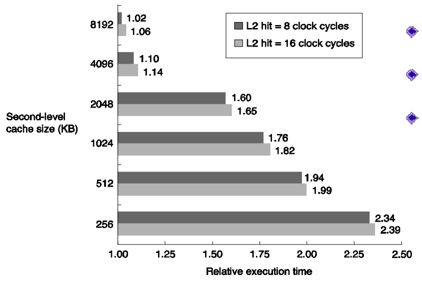
Assume ideal CPI=1.0, performance improvement = (6+1)/(3.6+1)=52%

Comparing Local and Global Miss Rates



- First-level cache: split 64K+64K 2-way
- Second-level cache: 4K to 4M
- In practice: caches are inclusive
- Global miss rate approaches single cache miss rate provided that the second-level cache is much larger than the first-level cache
- Global miss rate is what matters

Compare Execution Times



- L1 configuration as in the last slide
- L2 cache 256K-8M, 2-way
- Normalized to 8M cache with 1-cycle latency

- Performance is not sensitive to L2 latency
- Larger cache size makes a big difference

Early Restart and Critical Word First

- Don't wait for full block to be loaded before restarting CPU
  - Early restart—As soon as the requested word of the block arrives, send it to the CPU and let the CPU continue execution
  - Critical Word First—Request the missed word first from memory and send it to the CPU as soon as it arrives; let the CPU continue execution while filling the rest of the words in the block. Also called wrapped fetch and requested word first
- Generally useful only in large blocks (relative to bandwidth)
- Good spatial locality may reduce the benefits of early restart, as the next sequential word may be needed anyway