Computer Systems Architecture

Lecture 16

Mahadevan Gomathisankaran

March 30, 2010

Term Project

- Project abstract due on 03/30/2010 before class.
- Project reports due on 04/29/2010 before class.
- Project interviews for graduate students from 04/30.

Final Exam

- Take home (just like assignment).
- Will be given in class on 05/04/2010 and will be due before class on 05/06/2010.
- Policies will be strictly enforced.

Memory Hierarchy: Performance Metrics

- Miss rate: NO
- Average Memory Access Time: Hit Time + Miss rate × Miss Penalty
- CPU Execution time

$$\begin{aligned}
\text{CPU Execution Time} &= (\text{CPU Clock Cycles} + \text{Memory Stall Cycles}) \times \\
&\quad \text{Clock Cycle Time} \\
\text{Memory Stall Cycles} &= \text{Number of Misses} \times \text{Miss Penalty} \\
&= \text{IC} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss Penalty} \\
&= \text{IC} \times \frac{\text{Mem Access}}{\text{Instruction}} \times \text{Miss Rate} \times \text{Miss Penalty}
\end{aligned}$$

Improving Cache Performance

- Reduce AMAT
- Three factors in the equation:
  - Miss Penalty
  - Miss Rate
  - Hit Time

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$
\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}
$$

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}$$

- Issues:

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}$$

- Issues:
  - How to compare second level caches ?

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$
\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}
$$

- Issues:
  - How to compare second level caches ? global miss rate

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}$$

- Issues:
  - How to compare second level caches ? global miss rate
  - What is more important for second level cache? speed or miss rate

Miss Penalty Reduction Techniques

Multi-level caches

- Smaller cache is faster (reducing hit time)
- Larger cache has lower miss rate (reducing memory stalls)
- Solution: Use both, multiple level of caches
- Measuring performance of multi-level cache

$$\begin{aligned}
\text{AMAT} &= \text{Hit time}_{L1} + (\text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}) \\
\text{Miss penalty}_{L1} &= \text{Hit time}_{L2} + (\text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}) \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \text{Misses per Ins.}_{L1} \times \text{Hit Time}_{L2} + \\
&\quad \text{Misses per Ins.}_{L2} \times \text{Miss Penalty}_{L2}
\end{aligned}$$

- Issues:
  - How to compare second level caches ? global miss rate
  - What is more important for second level cache? speed or miss rate
  - How data should be placed in multiple levels? inclusion or exclusion

Miss Penalty Reduction Techniques

Example

(Page 414, 3rd Ed)

For every 1000 memory references $\text{Misses}_{L1} = 40$ and $\text{Misses}_{L2} = 20$.
$\text{Hit Time}_{L1} = 1$ clock cycle, $\text{Hit Time}_{L2} = 10$ clock cycles, and
$\text{Miss penalty}_{L2} = 100$ clock cycles. There are 1.5 memory references per instruction. Find various miss rates, AMAT and memory stalls per instruction.

Miss Penalty Reduction Techniques

Example

$$\begin{aligned}
\text{Miss rate}_{L1} &= \frac{40}{1000} = 4\% \\
\text{Local miss rate}_{L2} &= \frac{20}{40} = 50\% \\
\text{Global miss rate}_{L2} &= \frac{20}{1000} = 2\% \\
\text{AMAT} &= 1 + \frac{4}{100} \times \left(10 + \frac{20}{40} \times 100\right) \\
&= 3.4 \text{ clock cycles} \\
\frac{\text{Misses}}{\text{Instruction}} &= \frac{\text{Misses}}{\text{Mem. Reference}} \times \frac{\text{Mem. Reference}}{\text{Instruction}} \\
L1\frac{\text{Misses}}{\text{Instruction}} &= \frac{40}{1000} \times 1.5 = 6\% \\
L2\frac{\text{Misses}}{\text{Instruction}} &= \frac{20}{1000} \times 1.5 = 3\% \\
\frac{\text{Memory Stalls}}{\text{Instruction}} &= \frac{6}{100} \times 10 + \frac{3}{100} \times 100 \\
&= 3.6 \text{ clock cycles}
\end{aligned}$$

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency
- Small the block size higher the miss rate

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency
- Small the block size higher the miss rate
- Why not have a cache of larger blocks but simulate the effect of having smaller blocks ?

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency
- Small the block size higher the miss rate
- Why not have a cache of larger blocks but simulate the effect of having smaller blocks ?
- Two approaches

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency
- Small the block size higher the miss rate
- Why not have a cache of larger blocks but simulate the effect of having smaller blocks ?
- Two approaches
  - Critical word first

Miss Penalty Reduction Techniques

Reduce block latency

- Miss penalty includes the latency to get the whole *cache block* from next level
- Larger the block size larger the latency
- Small the block size higher the miss rate
- Why not have a cache of larger blocks but simulate the effect of having smaller blocks ?
- Two approaches
  - Critical word first
  - Early restart

Miss Penalty Reduction Techniques

Improve read performance over write

- Use Amdahl’s law, read is more common than write

Miss Penalty Reduction Techniques

Improve read performance over write

- Use Amdahl's law, read is more common than write
- *wrie-through* cache: check for conflicts in the write buffer

Miss Penalty Reduction Techniques

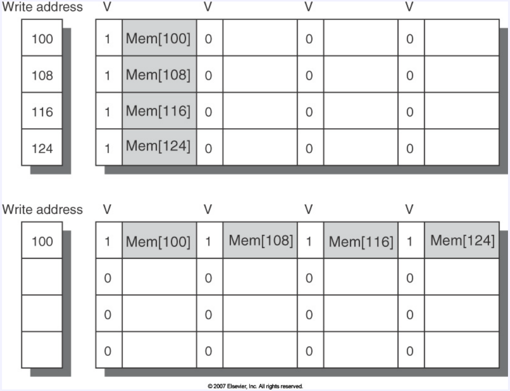
Improve read performance over write

- Use Amdahl's law, read is more common than write
- *wrie-through* cache: check for conflicts in the write buffer
- *write-back* cache: write a dirty block to a buffer instead of the next level

Miss Penalty Reduction Techniques

Reduce the bus traffic

- Merge multiple writes



Write address	V		V		V		V	
100	1	Mem[100]	0		0		0	
108	1	Mem[108]	0		0		0	
116	1	Mem[116]	0		0		0	
124	1	Mem[124]	0		0		0	

Write address	V		V		V		V	
100	1	Mem[100]	1	Mem[108]	1	Mem[116]	1	Mem[124]
	0		0		0		0	
	0		0		0		0	
	0		0		0		0	

Miss Penalty Reduction Techniques

Recycle the victims

- There may be some conflict misses
- Instead of increasing the associativity to reduce the miss add a smaller fully associative cache to store the conflicts or victims