Using Cache Memory to Reduce Processor-Memory Traffic

James R. Goodman[1]

[1]Department of Computer Sciences
University of Wisconsin-Madison

ShanghaiTech University, 2013

Outline

CPU and Memory speed mismatch

- Example

  Motorola MC68000 10 MHz CPU clock; 5 MB/s Memory access rate, half its pins tasked with memory connection.

CPU and Memory speed mismatch

- Example

  Motorola MC68000 10 MHz CPU clock; 5 MB/s Memory access rate, half its pins tasked with memory connection.

  - 10x transistors = 30x memory bandwidth. Not feasible to increase pin number 30 fold.

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.
- The chip should contain as much memory as the CPU needs.

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.
- The chip should contain as much memory as the CPU needs.
- Microprocessors in 1983 need 0.25 MiB of memory, more than possible amount.

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.
- The chip should contain as much memory as the CPU needs.
- Microprocessors in 1983 need 0.25 MiB of memory, more than possible amount.
- Higher performance CPUs apparently require more memory.

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.
- The chip should contain as much memory as the CPU needs.
- Microprocessors in 1983 need 0.25 MiB of memory, more than possible amount.
- Higher performance CPUs apparently require more memory.
- Which leads to:

Debates about On-chip Memory

- Dedicated on-chip memory with a relatively slower CPU may outperform a more powerful CPU with conventional memory.
- The chip should contain as much memory as the CPU needs.
- Microprocessors in 1983 need 0.25 MiB of memory, more than possible amount.
- Higher performance CPUs apparently require more memory.
- Which leads to: On-chip memory is clearly not feasible in 1983, nor is it today.

Issues With Using Cache Memory

- Use of cache has aggravated bandwidth problem.

Issues With Using Cache Memory

- Use of cache has aggravated bandwidth problem.
- Cache optimization aspects:
  - Maximizing Hit Ratio
  - Minimizing Data Accessing Time
  - Minimizing Miss Penalty
  - Minimizing Overhead of Updating Memory, Maintaining Multi-cache Consistency

Issues with Using Cache Memory Cont.

- Optimization Usually Results in Larger Burst Bandwidth Requirement.

- Example

  IBM System/370 model 155
  Cache-Memory transfer rate: 100 MB/s
  Cache-CPU transfer rate is less than 1/3 of that.

Issues with Using Cache Memory Cont.

- Optimization Usually Results in Larger Burst Bandwidth Requirement.

- Example

  IBM System/370 model 155
  Cache-Memory transfer rate: 100 MB/s
  Cache-CPU transfer rate is less than 1/3 of that.

  - Reason: To exploit spatial locality, thus data fetched in large blocks, resulting in high memory bandwidth bursts.

Issues with Using Cache Memory Cont. 2

- To lower the bandwidth from backing store to cache:
  - Transfer small blocks from backing store to cache,
  - Experience long delays while a block is brought from backing store to cache.

Issues with Using Cache Memory Cont. 2

- To lower the bandwidth from backing store to cache:
  - Transfer small blocks from backing store to cache,
  - Experience long delays while a block is brought from backing store to cache.
- Explore the effectiveness of exploiting temporal locality, i.e. blocks fetched from backing store are only the size needed by CPU.
- Effective environment: single-board computer running Multibus or Versabus.

Single Board Computer Application

Usage of Buses

- Buses, if needed, are designed for generality and simplicity not for high performance.

Single Board Computer Application

Usage of Buses

- Buses, if needed, are designed for generality and simplicity not for high performance.
- Example

  Multibus by Intel Corporation

Single Board Computer Application

Usage of Buses

- Buses, if needed, are designed for generality and simplicity not for high performance.
- Example

  Multibus by Intel Corporation

  - Applications are severely limited by bandwidth of Multibus.
  - Try to determine whether a cache memory system can be implemented with Multibus
  - Allocation of memory should be handled by the system instead by the programmer

Caches in Single Board Computers

Proposal

Single-Board Computer

- CPU w/o local memory except for cache
- Backing store provided by Multibus

Caches in Single Board Computers

Proposal

Single-Board Computer

- CPU w/o local memory except for cache
- Backing store provided by Multibus

Question

- Can we build a cache that works with Multibus and supports multiple processors?
- How many processors can we support?

Caches in Single Board Computers Cont.

- Important criterion is maximize use of the bus.
- Optimize system performance by optimizing bus utilization, achieving higher performance by minimizing individual processors' bus requirements.
- Prefer individual processors to remain idle periodically over saturating bus traffic with data that will never be used.

Switching Contexts

Context Switch

Task switch results in cache being reloaded and CPU speeds reduced to bus speed.

- Effects may be minimized if task switching frequency is reduced.
- Utilize multiple processors for multiple tasks.
- Interrupt handling optimized so program only uses small portion of cache.

Write-Through or Write-Back

Write-Through vs. Write-Back

Both Write-Through Write-Back have Pros and Cons.

- When hit rate is 100%, write-back results in no bus traffic.
- When hit rate is 0, write-through to relocated the data from mem to cache, larger bus traffic than write-through.
- For typical R2W and hit ratios and task switching(similar to getting a miss) is infrequent, write-back generate less traffic than write-through.
- However, Write-back has more coherency problems than write-through because the main-memory is not always in time.

New Strategy: Write-Once

Trade-off between write-back and write-through

Here we use two additional bits to label 4 stage.

- Invalid: There is No data in the block.
- Valid: There's data in the block which has been read from backing-store and has not been modified.
- Reserved: The data in the block has been locally modified exactly once since it was brought into the cache and the change has been transmitted to backing store.
- Dirty: The data in the vlock has been locally modified more than once since it was brought into the cache and the latest change has not been transmitted to backing store.

Effect of Write Strategy on Bus Traffic

Take write-back and write-through into consider

A write-back cache use write allocate and a write-throught cache use no-write allocate.

- Normally write-through generates less bus traffic than write-back.
- write-through don't take time to load data to cache from back-storing when get a wirte miss, but write-back will.
- write-back don't take time to load data to back-storing from cache when write-hit, but write-through will.
- As hit ratio is low and the block size is large, the write-back will do a worse work.

Effect of Write Strategy on Bus Traffic

How about Write-Once

Write-once result in bus traffic roughly equial to better of the two even though the initial goal is to get coherence.

- For example, For a 4-way set associative, 2048-byte cache with a block size of 32 bytes, the average bus traffic for three PDP-11 programs for which traced was 30.768% for write-through, 17.55% for write-back, and 17.78% for write-once.

Cold Start vs. Warm Start

- Cold start takes time to deal with cache miss at the beginning because of empty cache.
- The initial miss is amortized over all accesses.
- The longer the trace analyzed, the lower the miss ratio obtained. also for infrequent task switch the cold start will work more efficiently.
- Actually, since cold start is easier to generate, we generally use cold start.

Cache Size

How does Cache size affects miss ratio and traffic bus?

- Generally, larger cache size larger, less capacity miss and conflict miss, so smaller miss ratio, and decrease the traffic bus.
- The hit time is longer because of larger cache size. As hit time is small, it doesn't count.

Block Size

Block size with locality

Hit in the cache on colde start depend heavily on spatial locality, while temporal locality provides many hits when it is warm.

- increasing block size will heavilly affects the spatial locality, and miss ratio decline up to a point, traffic bus increase for either warm or cold starts.

Lowering The Overhead of Small Blocks

Severe Overhead of small blocks

Small blocks are costly in that they greatly increase the overhead of the cache.

Split cache into two parts:

- Transfer block: data transferred.
- Address block: info about the data.(tag,valid bit,dirty bit)

Effect of Large Address Blocks

Waste of traffic

There are case: the transfer blocks is empty, the other transfer blocks in the same address block must be purged so that the new address must be allocated.

- As address block enlarge, the traffic bus decreased and the miss rate declined(less conflict miss).

Conclude after simulation

- Minimum bus traffic is generated with minumum transfer block size.
- Miss ratio is substantially improved by using slightly larger transfer blocks.
- Large address block reduce the cost of the tag memory considerably(less conflict miss).

Summary

- The first main message of your talk in one or two lines.
- The second main message of your talk in one or two lines.
- Perhaps a third message, but not more than that.

- Outlook
  - Something you haven't solved.
  - Something else you haven't solved.

For Further Reading I

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal of This and That, 2(1):50–100, 2000.