

A Framework for Memory Hierarchies

Associativity schemes

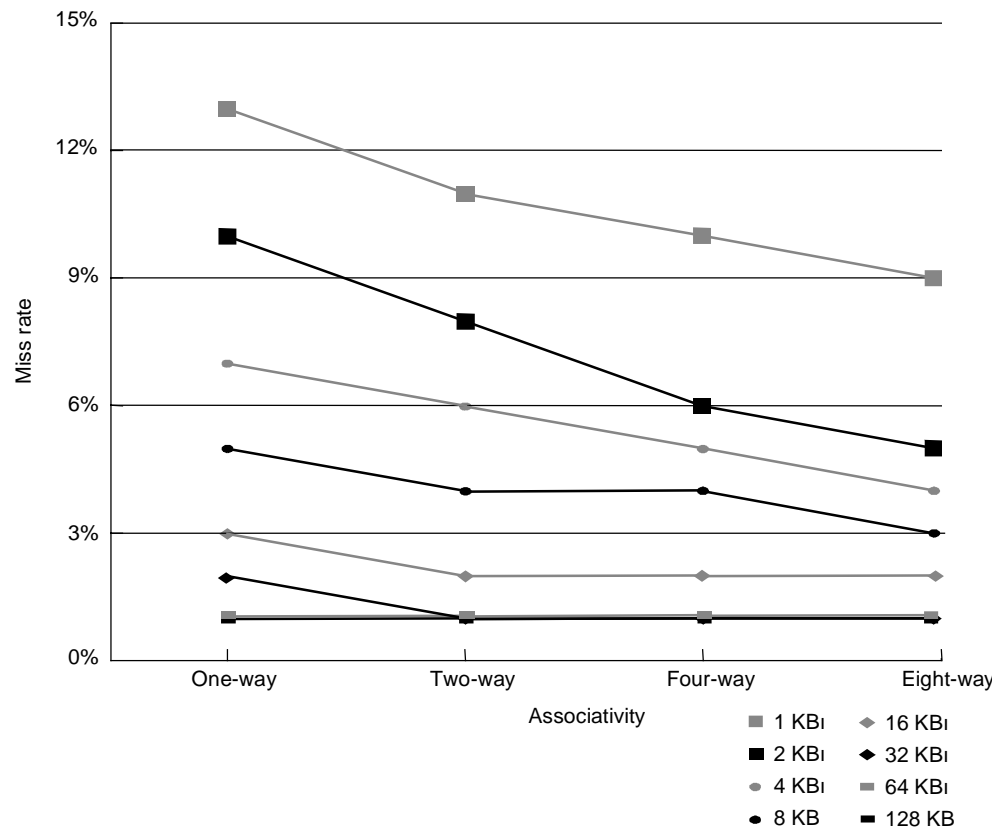
Scheme	Number of sets	Blocks per set
Direct mapped	Number of blocks in cache	1
Set associative	Blocks in cache / Associativity	Associativity (2-8)
Fully associative	1	Number Blocks in cache

Placing blocks

Feature	Typical values for caches	Typical values for VM	Typical values for TLB
Size in blocks	1000 – 100,000	2000 – 250,000	32 – 4,000
Size in KB	8 – 8,000	8000 – 8,000,000	0.25 – 32
Block size in bytes	16 – 256	4000 – 64,000	4 – 32
Miss penalty (clocks)	10 - 100	1,000,000 – 10,000,000	10 – 100
Miss rate	0.1% - 10%	0.00001% - 0.0001%	0.01% - 2%

A Framework for Memory Hierarchies

Miss rates



A Framework for Memory Hierarchies

Finding blocks

Associativity	Location method	Comparisons required
Direct mapped	Index	1
Set associative	Index the set, search among elements	Degree of associativity
Full	Search all cache entries	Size of the cache
	Separate lookup table	0

Why do we use full associativity and a separate lookup table (page table) in VM:

- Misses are very expensive
- Full associativity allows software to use sophisticated replacement schemes to reduce miss rate
- Full mapping table (all pages have entries) allows indexing with no extra hardware and no searching.
- The large page size means that the page table overhead is relatively small.

A Framework for Memory Hierarchies

Choosing a block to replace

- Random choice: The blocks are randomly selected, possibly using some hardware assistance. Used mainly larger caches
- Least recently used (LRU): The block replaced is the one that has been unused for the longest time. Used in VM (reference bit).

Writing blocks

- Write-through:
 - Read misses are simpler and cheaper because they do not require writing blocks to the lower level.
 - Easier to implement, although normally a write-buffer is required
- Write-back:
 - Individual words are written at the rate of the cache
 - Multiple writes within a block require only one write to the lower level memory
 - When blocks are written back a high bandwidth transfer can be used since the entire block is written

A Framework for Memory Hierarchies

The sources of misses

- *Compulsory misses* (cold-start misses): First time access to a block that has never been in the cache
- *Capacity misses*: The cache cannot contain all the blocks needed during the execution of a program. These misses occur because of blocks being replaced are later retrieved again.
- *Conflict misses* (collision misses): Misses that occur in direct or set associative schemes. Multiple blocks compete for the same set (or entry) in the cache.

The challenge

Design change	Effect on miss rate	Possible negative effect on performance
Increase size	Decreases capacity misses	May increase access time
Increase associativity	Decreases conflict misses	May increase access time
Increase block size	Decreases miss rate	May increase miss penalty

A Framework for Memory Hierarchies

Pentium Pro and PowerPC 604

Address translation and TBL

Characteristics	Pentium Pro	PowerPC 604
Virtual address	32 bits	52 bits
Physical address	32 bits	32 bits
Page size	4KB, 4MB	4KB, selectable, 256MB
TBL	Instruction TLB and data TLB Both 4-way associative Pseudo-LRU replacement Instruction TLB: 32 entries Data TLB: 64 entries TLB misses are handled in hardware	Instruction TLB and data TLB Both 2-way associative LRU replacement Instruction TLB: 128 entries Data TLB: 128 entries TLB misses are handled in hardware

First-level caches

Characteristics	Pentium Pro	PowerPC 604
Cache organization	Split instruction and data caches	Split instruction and data caches
Cache size	8KB each (instruction and data)	8KB each (instruction and data)
Associativity	4-way set associative	4-way set associative
Replacement	Approximated LRU replacement	LRU replacement
Block size	32 bytes	32 bytes
Write policy	Write-back	Write-back or write-through