


Administrative Matters

- **Homework #7**
 - Posted
 - Due Friday 12/7
- **Final Exam**
 - Comprehensive
 - Materials from
 - Midterms, quizzes
 - Homeworks
 - Lecture
 - Saturday, 12/15, 11:30AM-2:30PM



©2004 Morgan Kaufmann Publishers 1

Example

- **Example:**
 - CPI of 1.0 on a 5 Ghz machine with a 2% miss rate, 100ns DRAM access
 - Adding 2nd level cache with 5ns access time, miss rate of .5%
- **Miss penalty to the primary memory is:**
 - $100\text{ns} / (1/5\text{Ghz}) = 100\text{ns} / (0.2\text{ns/clock cycle}) = 500$ clock cycle
- **CPI for having a single level of cache is:**
 - $1 + \text{memory stall cycles per instruction} = 1 + 2\% \cdot 500 = 11$
- **Miss penalty to the second level cache is:**
 - $5\text{ns} / (1/5\text{Ghz}) = 5\text{ns} / (0.2\text{ns/clock cycle}) = 25$ clock cycles
- **CPI for having 2 levels of cache is**
 - $1 + 2\% \cdot 25 + 0.5\% \cdot 500 = 4$
- **Reduce CPI from 11 to 4!!!**
- **Alternatively, a hit in second level cache stalls: $(2\% - 0.5\%) \cdot 25 = 0.4$, a miss in second level cache stalls: $0.5\% \cdot (25 + 500) = 2.6$, total CPI $1 + 0.4 + 2.6 = 4$**

©2004 Morgan Kaufmann Publishers 2

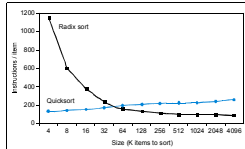
Additional Complication

- **On-chip cache tends to have higher associativity than off-chip**
 - Overhead circuit tends to be able to run faster
- **On-chip L1 cache tends to have**
 - lower associativity and smaller block size
- **On-chip L2 cache tends to have**
 - Higher associativity and larger block size
- **What happen to out of order execution!**
 - No more stalls, miss latency may be overlapped with execution
 - Can't calculate, must simulate

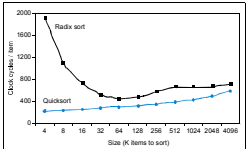
©2004 Morgan Kaufmann Publishers 3

Cache Complexities

- **Not always easy to understand implications of caches:**



Theoretical behavior of Radix sort vs. Quicksort

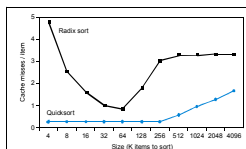


Observed behavior of Radix sort vs. Quicksort

©2004 Morgan Kaufmann Publishers 4

Cache Complexities

- **Here is why:**
(Quicksort is in-place sort, radix-sort is not)



- **Memory system performance is often critical factor**
 - multilevel caches, pipelined processors, make it harder to predict outcomes
 - Compiler optimizations to increase locality sometimes hurt ILP
 - Increase dependency by putting "related" item together
- **Difficult to predict best algorithm: need experimental data**

©2004 Morgan Kaufmann Publishers 5

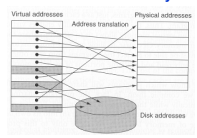
Quick Question

- **What is generally true**
 - First-Level caches are more concerned with hit time, and second-level caches are more concerned with miss rate
 - First-level caches are more concerned with miss rate, and second-level caches are more concerned with hit time

©2004 Morgan Kaufmann Publishers 6

Virtual Memory

- Main memory can act as a cache for the secondary storage (disk)



- Advantages:
 - Illusion of having more physical memory
 - Program relocation
 - Program can be put anywhere in the memory
 - Protection
 - Program from other program, OS from other program...
 - Sharing PHYSICAL memory while having separate VIRTUAL memory space

©2004 Morgan Kaufmann Publishers 7

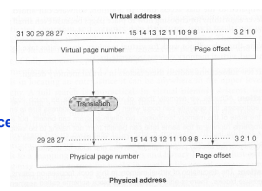
Virtual/physical vs. main memory/cache

- Page – block
- Page fault – cache miss
- Virtual address – physical address
- Physical address – cache index
- Address translation – address to cache index translation
- Page fault (millions of cycles) – cache miss (100's of cycle)
- Usually fully associated – usually no more than 4 ways
- Software fault handling – hardware miss handling
- Clever replacement policy – fast replacement policy
- Always write back – write through or back
- Page table – Tag comparison

©2004 Morgan Kaufmann Publishers 8

Virtual addressing example

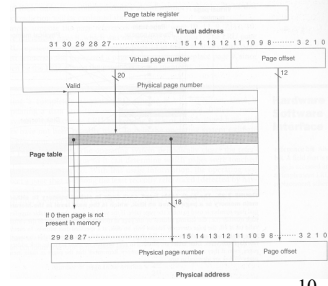
- 32 bit virtual address
 - 4GByte
- 30 bit physical address
 - 1GByte actual memory
- Page size for 12 bit offset
 - 4K Page
- A program may span multiple pages
- In fact, a program address 32 bit space
 - OS control what it actually use



©2004 Morgan Kaufmann Publishers 9

Page Table

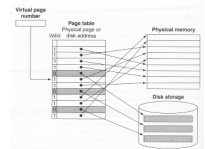
- Each program has a page table
- Page table register point to current one
- 4K byte page
- 4G virtual address space
- 1 G physical address space
- 1 million entries page table
- Valid bit
 - On: good
 - Off: not in memory (exception)
- May contains also protection



©2004 Morgan Kaufmann Publishers 10

Page Fault

- Exception!!!
- Go to disk for the page
 - Need to also look up where on the disk is the page
- Swap out the least recent used page from physical memory
 - E.g. given access pattern: 10, 12, 9, 7, 11, 10
 - 12 is the LRU
- Swap space
 - Full virtual memory space that a memory needs
 - Exist on disk
 - Going to swap space means that the page was swap out
 - By another process
 - Context switch
- LRU is very expensive to implement
- Reference bit
 - 1 single bit to approximate LRU
 - Periodically zap the bit



©2004 Morgan Kaufmann Publishers 11

Pages (continue)

- 32 bit virtual address
- 4KByte pages
- 4 bytes per page table
- $2^{32}/2^{12}=2^{20}$, $2^{20} \times 4=4\text{MB}$ page table size
- Since each program are visible to 32bit address space
 - Which can locate either on disk or in main memory
- Each executing program may need to carry 4MB of page table!
 - Lots of scheme to reduced the page table size:
 - Hashing function
 - Hierarchical page table
 - ...

©2004 Morgan Kaufmann Publishers 12

A few words about page writes

- **Write through is impossible**
 - Millions of cycle disk access time
- **Write back**
 - A.k.a. copy back
 - Keep a “dirty” bit and write page back only when swapped out
 - In theory, it is not likely the page will be access
 - So, take a long time to write is o.k.
 - Also, writing an entire page (vs. write-through) is efficient
 - Disk write time is small compare to access time

Address Translation is slow

- Page table are in main memory
- Each memory access takes twice as long
 - Take one access to get translation (i.e. physical address)
 - Take another access to get the data
- Principle of locality also applies here!
- Translation Lookaside Buffer (TLB)
 - A.k.a Translation Cache

Translation Lookaside Buffer

- Tag holds the virtual page number (or a portion)
- On each access to memory, a TLB look up is first performed
- If hit, Physical address is used to access physical memory
 - Set ref bit, set dirty bit if write
- If miss, translation is swap in
- **Typical TLB parameter**
 - 16-512 entries
 - Block size 4-8 byte
 - Hit time: 1 cycle
 - Miss penalty: 100 cycles
 - Miss rate : 0.5%
 - Write-back
 - Associativity varies

