


MK
Computer Architecture
A Quantitative Approach, Sixth Edition



Chapter 2
Virtual memory

MK
Copyright © 2019, Elsevier Inc. All rights Reserved 1

Introduction

- The virtual address space is much larger than the physical memory (especially with many programs running simultaneously).
- Programmers deal with virtual memory (Have you ever tried to manage transfer from the disk to RAM?).
- Also, need protection (separate code and data, can not access other processes memory).
- Code and data relocation?
- Virtual memory gives the illusion that the programmer has access to memory without worrying about the above points.
- Managed by the OS

MK
Copyright © 2019, Elsevier Inc. All rights Reserved 2

Introduction

- Physical memory acts as a cache for data stored on the disk.
- Bring the data (pages) from disk to cache and access it from there.
- Needs system software to translate
- Also HW to make it faster (TLB).

MK
Copyright © 2019, Elsevier Inc. All rights Reserved 3

Why Virtual Memory?

- Any systems without virtual memory
 - Tiny microcontrollers
 - Really old computers
- Any advantages/disadvantages?
 - Load/store directly access memory
 - Placement? Replacement?

MK Copyright © 2019, Elsevier Inc. All rights Reserved 4

Why Virtual Memory?

- CPU issues virtual addresses (generated by the compiler)

MK Copyright © 2019, Elsevier Inc. All rights Reserved 5

Translation

- Need translation from virtual to physical

MK Copyright © 2019, Elsevier Inc. All rights Reserved 6

Why Virtual Memory?

- CPU issues virtual addresses (generated by the compiler)

Virtual memory

MK Copyright © 2019, Elsevier Inc. All rights Reserved 7

Address Translation

- CPU issues virtual addresses (generated by the compiler)

Virtual memory

MK Copyright © 2019, Elsevier Inc. All rights Reserved 8

Virtual memory

- Page table is used for translation
- What other information is there
- Size of table/page
- Internal Segmentation
- TLB: Translation Lookaside Buffer

Page Table

MK Copyright © 2019, Elsevier Inc. All rights Reserved 9

Example

Example

MK Copyright © 2019, Elsevier Inc. All rights Reserved 10

Steps in memory Access

Memory Access

- CPU generates VA memory reference
- PT (or TLB) translates it to physical memory address
- CPU goes to the cache for that address
- If miss, go to memory
- 2 memory access references for one access.
- Can we use virtual address in cache
- Aliasing

MK Copyright © 2019, Elsevier Inc. All rights Reserved 11

Virtual indexing Physical Tagging

Memory Access

- If we can map the same PA to the same location in the cache, no aliasing problem.
- PA and VA share the low order n bits where $n = \log_2 \text{Page_Size}$
- Example: show limit on cache size

MK Copyright © 2019, Elsevier Inc. All rights Reserved 12

Memory Optimizations

Standard	I/O clock rate	M transfers/s	DRAM name	MiB/s/DIMM	DIMM name
DDR1	133	266	DDR266	2128	PC2100
DDR1	150	300	DDR300	2400	PC2400
DDR1	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1333	2666	DDR4-2666	21,300	PC21300

MK Copyright © 2019, Elsevier Inc. All rights Reserved 16

DDR

- **DDR2**
 - Lower power (2.5 V -> 1.8 V)
 - Higher clock rates (266 MHz, 333 MHz, 400 MHz)
- **DDR3**
 - 1.5 V
 - 800 MHz
- **DDR4**
 - 1-1.2 V
 - 1333 MHz
- **GDDR5 is graphics memory based on DDR3**
 - Wider interface
 - Higher max. clock rate. Normally connected directly to the GPU (as opposed to DIMMs).

MK Copyright © 2019, Elsevier Inc. All rights Reserved 17

Memory Power Consumption

Usage Mode	Background power (mW)	Activate power (mW)	Read, write, terminate power (mW)	Total Power (mW)
Low power mode	~100	0	0	~100
Typical usage	~100	~100	~100	~300
Fully active	~100	~150	~250	~500

MK Copyright © 2019, Elsevier Inc. All rights Reserved 18

Stacked/Embedded DRAMs

- Stacked DRAMs in same package as processor
 - High Bandwidth Memory (HBM)

The diagram shows two methods of DRAM stacking. On the left, 'Vertical stacking (3D)' shows multiple layers of DRAM stacked on top of each other, with an xPU chip positioned below the stack. On the right, 'Interposer stacking (2.5D)' shows a stack of DRAM layers connected to an xPU chip via a central interposer layer.

Memory Technology and Optimizations

MK Copyright © 2019, Elsevier Inc. All rights Reserved 19

Flash Memory

- Type of EEPROM
- Types: NAND (denser) and NOR (faster)
- NAND Flash:
 - Reads are sequential, reads entire page (.5 to 4 KiB)
 - 25 us for first byte, 40 MiB/s for subsequent bytes
 - SDRAM: 40 ns for first byte, 4.8 GB/s for subsequent bytes
 - 2 KiB transfer: 75 uS vs 500 ns for SDRAM, 150X slower
 - 300 to 500X faster than magnetic disk

Memory Technology and Optimizations

MK Copyright © 2019, Elsevier Inc. All rights Reserved 20

NAND Flash Memory

- Must be erased (in blocks) before being overwritten
- Nonvolatile, can use as little as zero power
- Limited number of write cycles (~100,000)
- \$2/GiB, compared to \$20-40/GiB for SDRAM and \$0.09 GiB for magnetic disk
- Phase-Change/Memristor Memory
 - Possibly 10X improvement in write performance and 2X improvement in read performance

Memory Technology and Optimizations

MK Copyright © 2019, Elsevier Inc. All rights Reserved 21

Memory Dependability

- Memory is susceptible to cosmic rays
- *Soft errors*: dynamic errors
 - Detected and fixed by error correcting codes (ECC)
- *Hard errors*: permanent errors
 - Use spare rows to replace defective rows
- Chipkill: a RAID-like error recovery technique

Memory Technology and Optimizations

MK Copyright © 2019, Elsevier Inc. All rights Reserved 22

Fallacies and Pitfalls

- Predicting cache performance of one program from another
- Simulating enough instructions to get accurate performance measures of the memory hierarchy
- Not delivering high memory bandwidth in a cache-based system

MK Copyright © 2019, Elsevier Inc. All rights Reserved 23
