CS 423 – Operating Systems Design

Lecture 20 – Input and Output Software

Klara Nahrstedt

Fall 2011

Based on slides by YY Zhou and Andrew S. Tanenbaum

Overview

- Administrative announcements
  - Homework 1 - deadline **October 10** in class or submission via compass for online students
  - Midterm – Wednesday, **October 12** in class!!
- I/O Software
  - We talked about programming of IO via
    - Programmed IO (polling)
    - Interrupt-driven IO
    - IO using DMA
  - We also talked about Interrupt Handlers
- Today:
  - Talk about Device Drivers and Disk Devices
- Summary

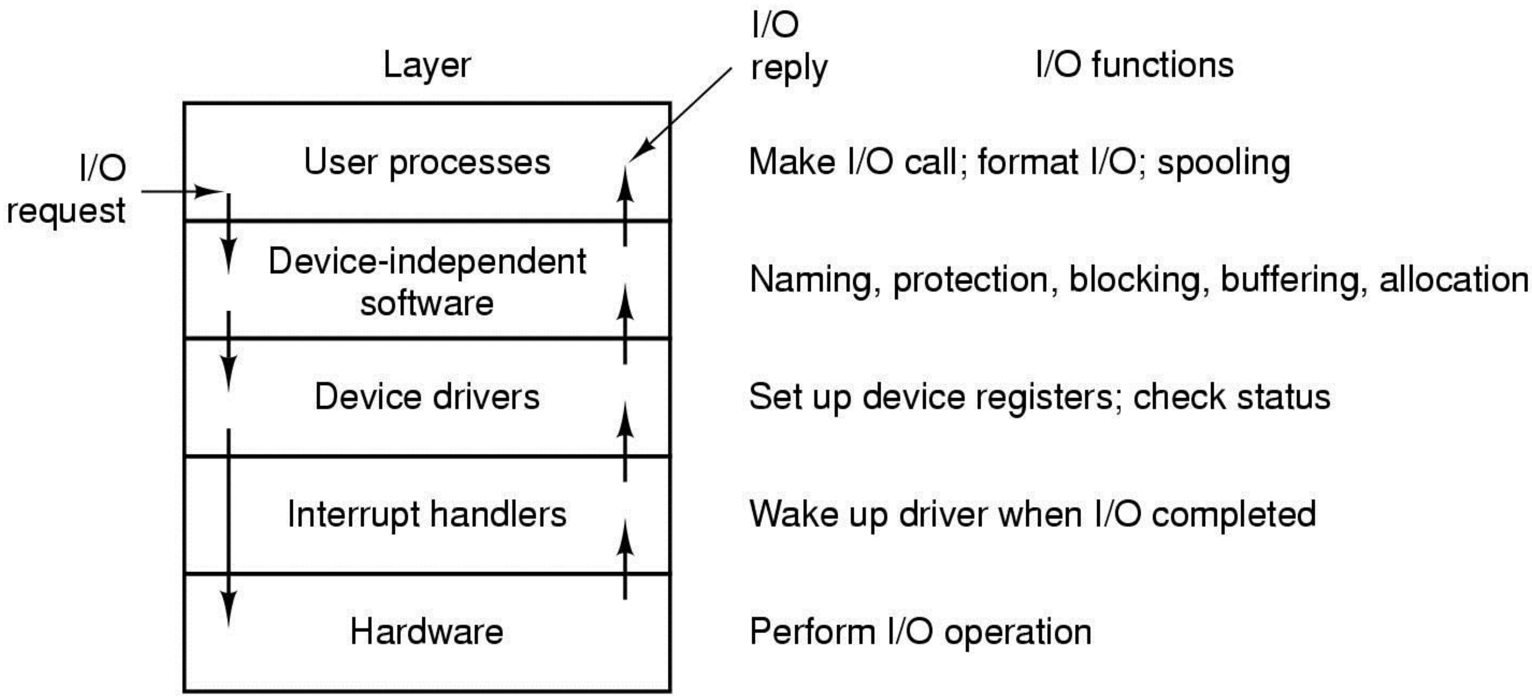
I/O Software: Principles

- device independence
  - possible to write programs that can access any IO device without having to specify the device in advance.
  - read file as input on a floppy, hard disk, or CD-ROM, without modifying the program for each device
- uniform naming
  - the name of a file or a device should simply be a string or an integer and not depend on the device in any way
  - Example: In Unix all files and devices are addressed the same way by a path name.
- Synchronous (blocking) vs asynchronous (interrupt-driven) transfers
  - Example: most devices are asynchronous. User programs are easier to write if the IO operations are synchronous. Hence, it is up to OS to make asynchronous operations look like synchronous to the user programs.

Principles (cont.)

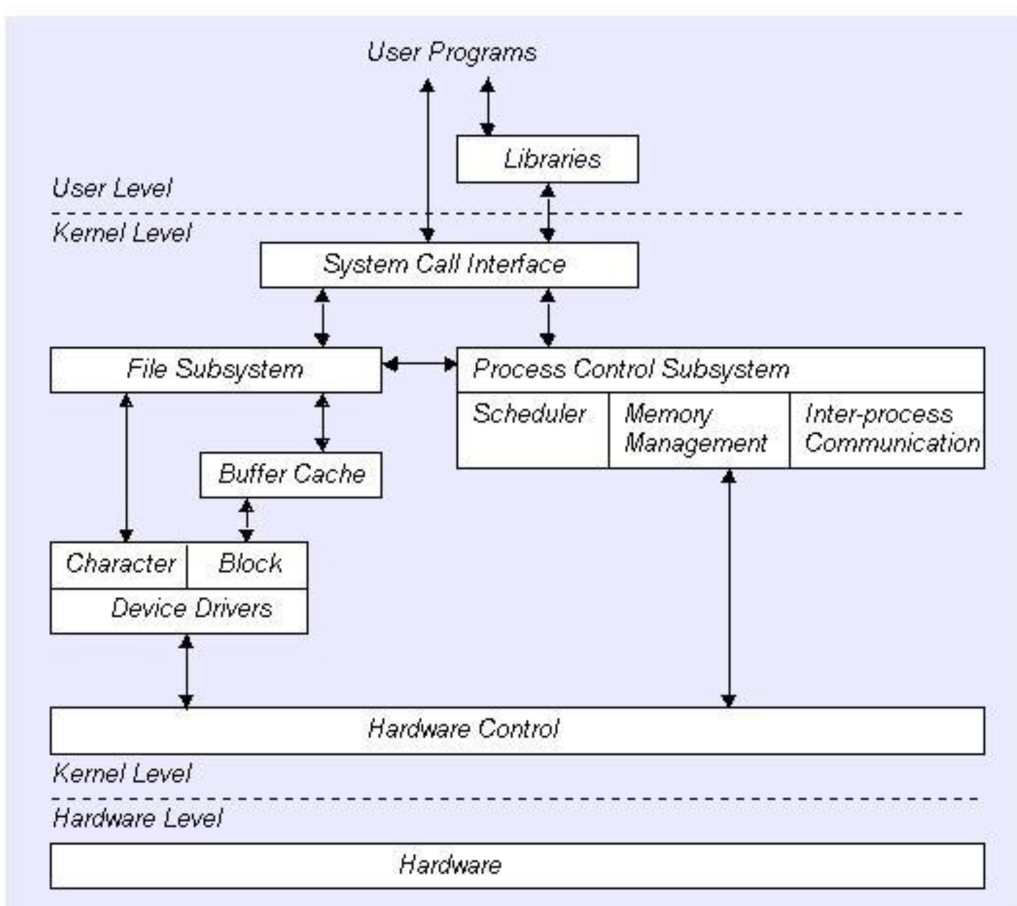
- Buffering
  - data might need to be buffered because they might not be stored immediately
  - Example: Network packets come in off the network, and need to be buffered for protocol processing, audio buffering between audio speakers and network.
- shared vs dedicated devices
  - allowing for sharing, considering two users having open files on the same device, or allowing dedicated devices and deal with deadlock problems.
  - Example: Disk, audio devices.

I/O Software



Layers of the I/O system and the main functions of each layer

I/O Software in Linux



Device Drivers

- Device-specific code to control an IO device, is usually written by device's manufacturer
- A device driver is usually part of the OS kernel
  - Compiled with the OS
  - Dynamically loaded into the OS during execution
- Each device driver handles
  - one device type (e.g., mouse)
  - one class of closely related devices (e.g., SCSI disk driver to handle multiple disks of different sizes and different speeds.).
- Categories:
  - Block devices
  - Character devices

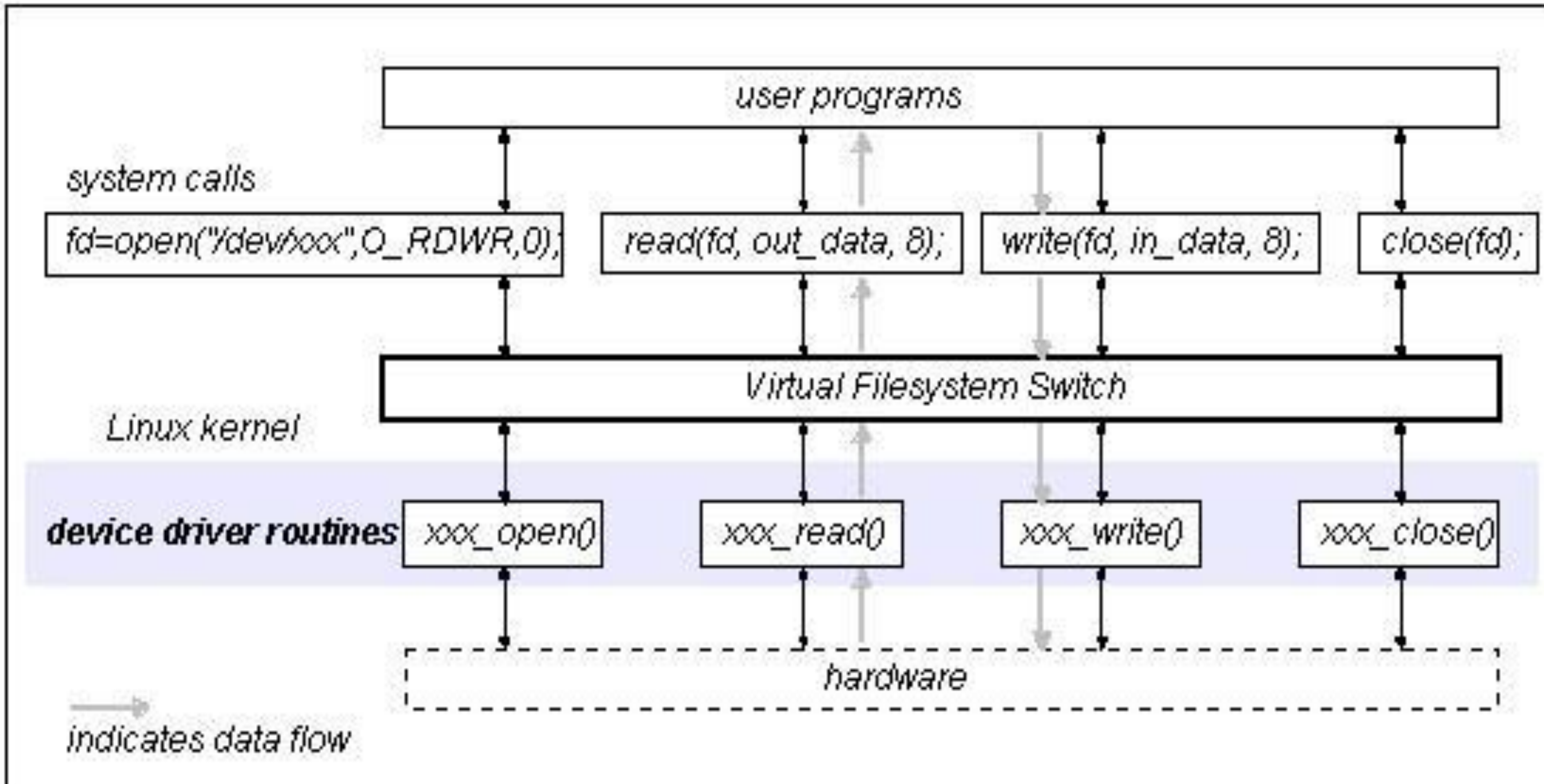
Functions in Device Drivers

- Accept abstract read and write requests from the device-independent layer above;
- Initialize the device;
- Manage power requirements and log events
- Check input parameters if they are valid
- Translate valid input from abstract to concrete terms
  - e.g., convert linear block number into the head, track, sector and cylinder number for disk access
- Check the device if it is in use (i.e., check the status bit)
- Control the device by issuing a sequence of commands. The driver determines what commands will be issued.

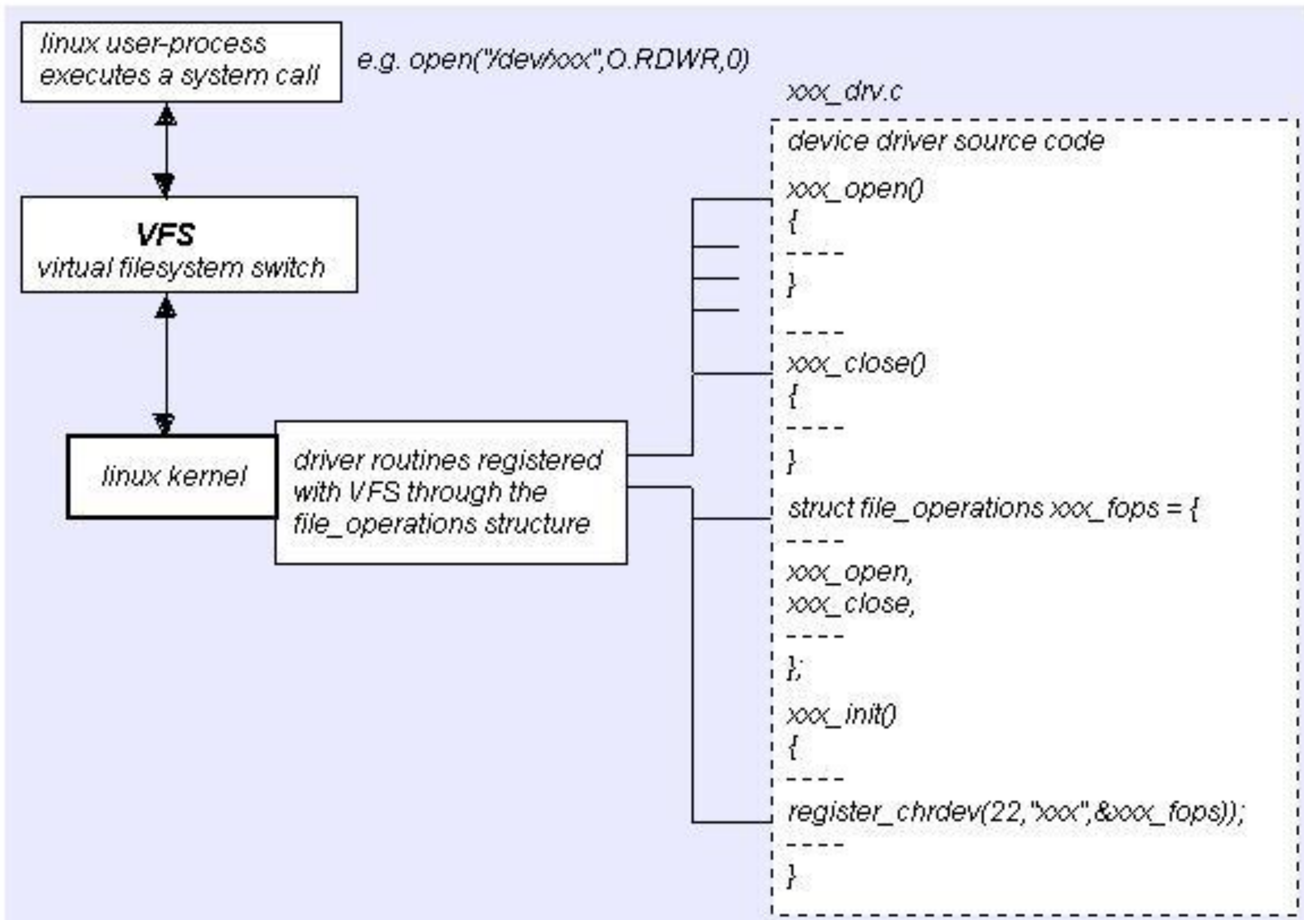
Device Driver Protocol

- After driver knows which commands to issue, it starts to write them into controller's device registers
- After writing each command, it checks to see if the controller accepted the command and is prepared to accept the next one.
- After commands have been issued, either (a) the device waits until the controller does some work and it blocks itself until interrupt comes to unblock it; or (b) the device doesn't wait because the command finished without any delay.

I/O Subsystem in Linux



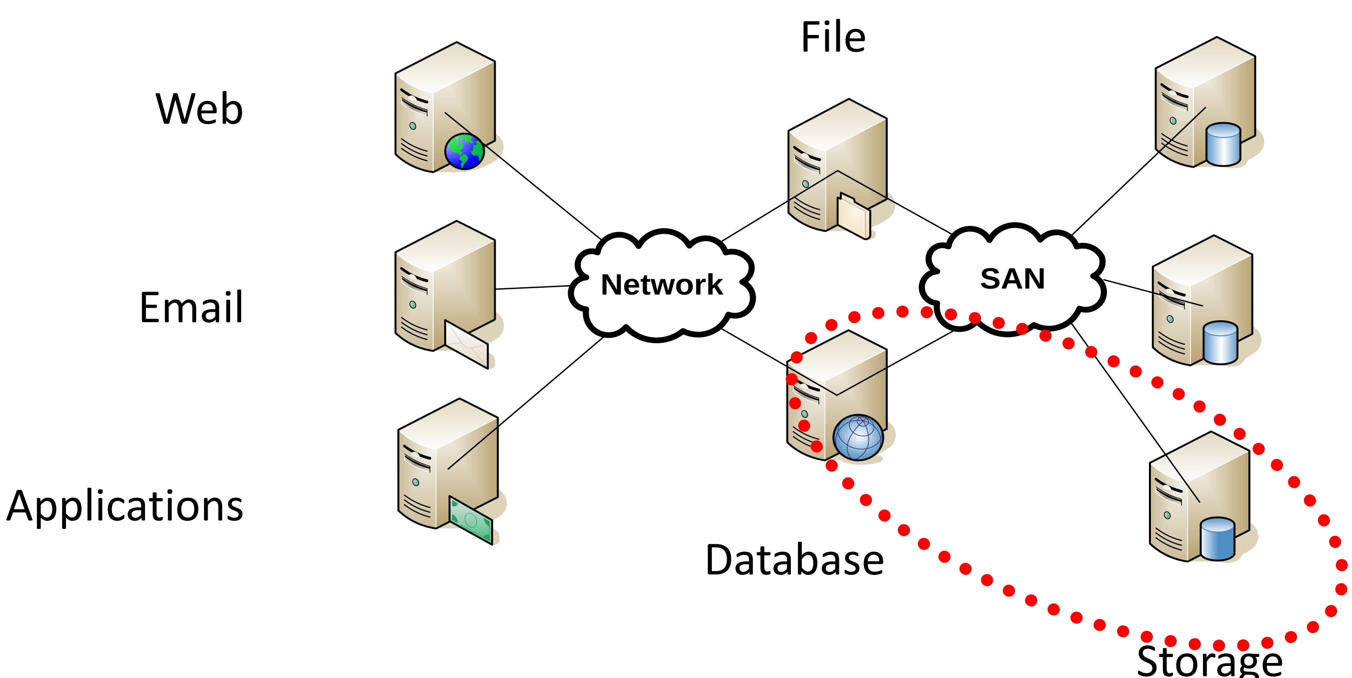
File System Switch



DISK DEVICE

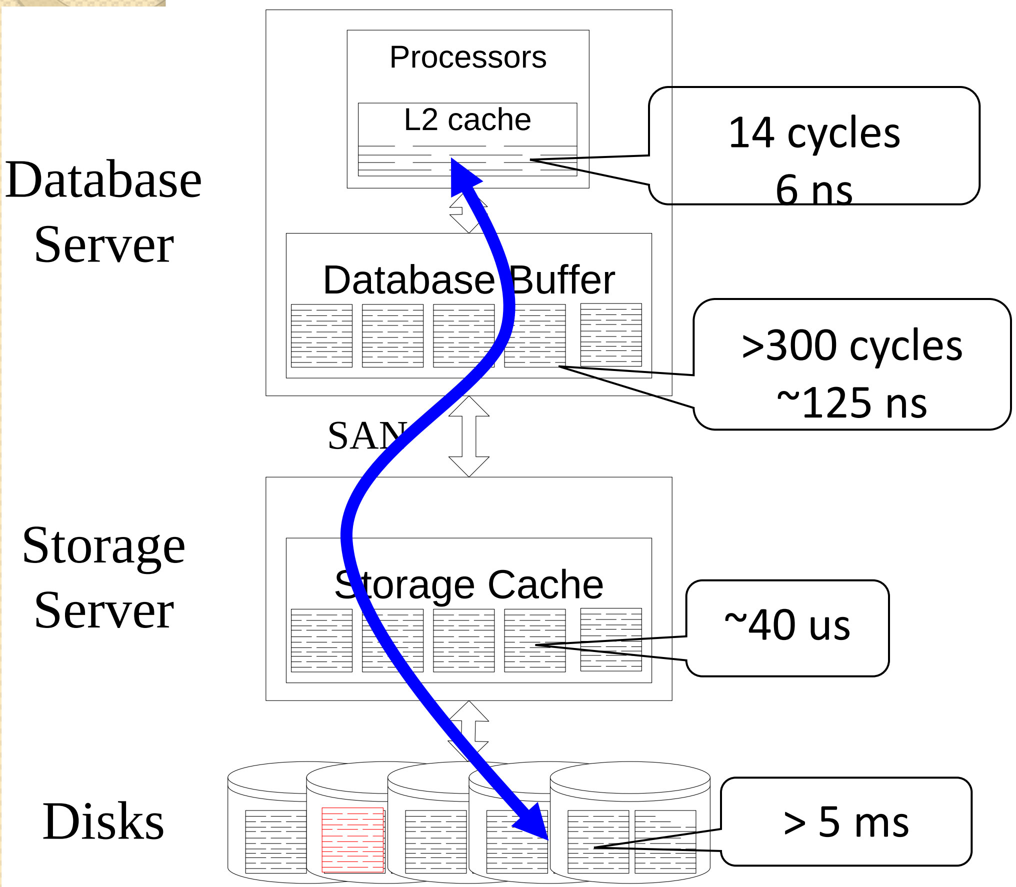
Data Centers

- Core of enterprise computing
  - A cluster of specialized servers
  - Multiple tiers

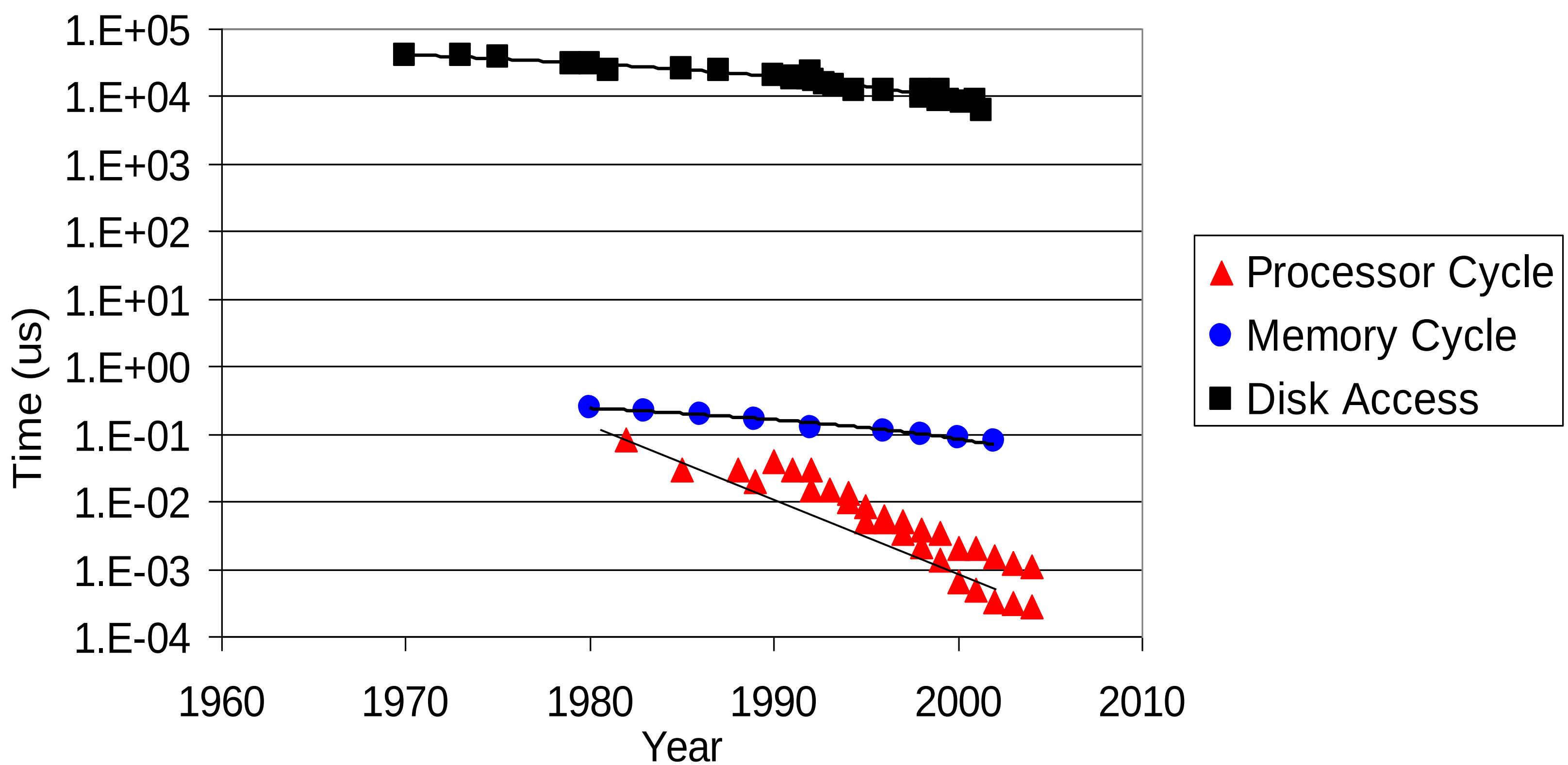


Data Path

- Disks, storage server, and database server
- Widely used caching memory
  - Large access speed gaps
  - Different sizes
  - Various granularity



Two Performance Trends



- **The gaps are increasing large**

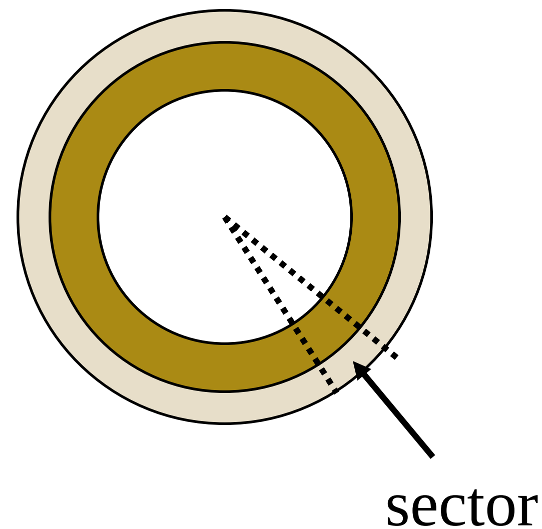
Source: Zhifeng Chen, "Optimization of Data Access for Database Applications", PhD Thesis, 2005, UIUC

Disk Technology Trends

- Disks are getting smaller for similar capacity
  - Spin faster, less rotational delay, higher bandwidth
  - Less distance for head to travel (faster seeks)
  - Lighter weight (for portables)
- Disk data is getting denser
  - More bits/square inch
  - Tracks are closer together
  - Doubles density every 18 months
- Disks are getting cheaper ($/MB)
  - Factor of ~2 per year since 1991
  - Head close to surface

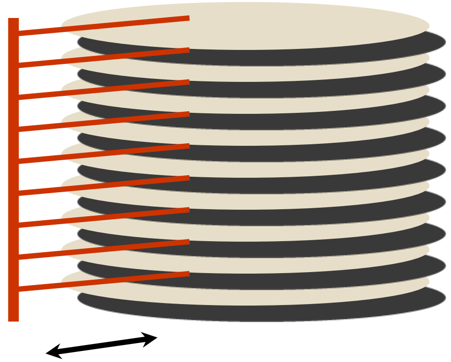
Disk Organization

- Disk surface
  - Circular disk coated with magnetic material
- Tracks
  - Concentric rings around disk surface, bits laid out serially along each track
- Sectors
  - Each track is split into arc of track (min unit of transfer)



More on Disks

- CD's and floppies come individually, but magnetic disks come organized in a disk pack
- Cylinder
  - Certain track of the platter
- Disk arm
  - Seek the right cylinder



seek a cylinder

Disk Example (Seagate Barracuda)

Specifications	1TB[1]	500GB[1]	320GB[1]	250GB[1]
Model Number	ST31000524AS	ST500DM002[2]	ST320DM000[2]	ST250DM000[2]
Interface Options	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ
Performance				
Spindle Speed (RPM)	7200	7200	7200	7200
Cache, Multisegmented (MB)	32	16	16	16
SATA Transfer Rates Supported (Gb/s)	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5
Seek Average, Read (ms)	<8.5	<11	<11	<11
Seek Average, Write (ms)	<9.5	<12	<12	<12
Sustained Data Rate, Sequential-Write (MB/s)	125	125	125	125
Configuration/Organization				
Heads/Disks	4/2	2/1	2/1	1/1
Bytes per Sector	512	4096 or 512[2]	4096 or 512[2]	4096 or 512[2]

Disk Performance

- Seek
  - Position heads over cylinder, typically 5.3 – 8 ms
- Rotational delay
  - Wait for a sector to rotate underneath the heads
  - Typically 8.3 – 6.0 ms (7,200 – 10,000RPM) or ½ rotation takes 4.15-3ms
- Transfer bytes
  - Average transfer bandwidth (15-37 Mbytes/sec)
- Performance of transfer 1 Kbytes
  - Seek (5.3 ms) + half rotational delay (3ms) + transfer (0.04 ms)
  - Total time is 8.34ms or 120 Kbytes/sec!
- What block size can get 90% of the disk transfer bandwidth?

Disk Behaviors

- There are more sectors on outer tracks than inner tracks
  - Read outer tracks: 37.4MB/sec
  - Read inner tracks: 22MB/sec
- Seek time and rotational latency dominate the cost of small reads
  - A lot of disk transfer bandwidth is wasted
  - Need algorithms to reduce seek time

Observations

- Getting first byte from disk read is slow
  - high latency
- Peak bandwidth high, but rarely achieved
- Need to mitigate disk performance impact
  - Parallelize disks access
  - Do extra calculations to speed up disk access
    - Schedule requests to shorten seeks
  - Move some disk data into main memory – file system caching

MIDTERM TOPICS AND READING

MIDTERM: WEDNESDAY, OCTOBER 12, 10AM, 1109 SC (IN CLASS)

CLOSED BOOK/CLOSED NOTES

(YOU CAN USE CALCULATOR, BUT IT WILL NOT BE NEEDED)

Topics Coverage

- Midterm will cover material in lectures
  - Lecture 1-18
  - Topics:
    - Introduction into OS
    - Process Management
    - Virtual Memory Management
    - File Systems
  - No IO topic
- Questions from Lectures
- Use textbook to support your reading of slides if you don't understand some concepts

Supporting Readings from Textbook

- Chapter 1.1.-1.7
  - Review some basic hardware concepts
  - System Call basics
  - OS Structure
- Chapter 2. 1 – 2.4 and 2.5.2
  - Processes versus threads
  - Thread management – states, race conditions,
  - Thread synchronization
    - Critical sections, locks, mutexes, spin lock, TSL, reader-writer locks, monitors with conditional variables
    - Producer-consumer problems (bounded queues)
  - Thread scheduling
    - FIFO, RR, RMS, EDF

Readings from Textbook

- Chapter 3.1-3.6
  - Memory address space – base and limit registers
  - Swapping
  - Managing free memory
  - Virtual memory – paging
    - Page tables
    - TLB to speed up paging
    - Page tables for large memories
    - Page replacement algorithms and their various performance behaviors
    - Design issues for paging systems
    - Page fault handling

Readings from Textbook

- Chapter 4.1-4.3.6 and 4.4
  - File concept
  - File access methods
  - File metadata – role of inode, role of open file table
  - Directory concept
  - Path and link concepts
  - Understanding what happens upon opening file or reading/writing a file
  - Implementing files, directories
    - File allocation methods
  - Log-based vs Journaling File Systems
  - File System Management and Optimization