Disk-based Storage
Oct. 23, 2008

Topics
- Storage technologies and trends
- Locality of reference
- Caching in the memory hierarchy

Disk-based storage in computers

- Memory/storage hierarchy
  - Combining many technologies to balance costs/benefits
  - Recall the memory hierarchy and virtual memory lectures

Memory/storage hierarchies

- Balancing performance with cost
  - Small memories are fast but expensive
  - Large memories are slow but cheap

- Exploit locality to get the best of both worlds
  - Locality = re-use/nearness of accesses
  - Allows most accesses to use small, fast memory

Exam next Thursday
- Style like exam #1: in class, open book/notes, no electronics
An Example Memory Hierarchy

- **Registers**
  - On-chip L1 cache (SRAM)
  - CPU registers hold words retrieved from L1 cache.
- **Main Memory**
  - On-chip L2 cache (SRAM)
  - L1 cache holds cache lines retrieved from the L2 cache memory.
  - L2 cache holds cache lines retrieved from main memory.
- **Local Secondary Storage**
  - Main memory holds disk blocks retrieved from local disks.
  - Main memory holds disk blocks retrieved from local disks.
- **Remote Secondary Storage**
  - Main memory holds disk blocks retrieved from remote disk servers.
  - Local disks hold files retrieved from disks on remote network servers.

Larger, slower, and cheaper (per byte) storage devices

Smaller, faster, and costlier (per byte) storage devices

Disk-based storage in computers

- **Memory/storage hierarchy**
  - Combining many technologies to balance costs/benefits
  - Recall the memory hierarchy and virtual memory lectures

- **Persistence**
  - Storing data for lengthy periods of time
    - DRAM/SRAM is “volatile”: contents lost if power lost
    - Disks are “non-volatile”: contents survive power outages
  - To be useful, it must also be possible to find it again later
    - this brings in many interesting data organization, consistency, and management issues
      - take 18-746/15-746 Storage Systems
    - we’ll talk a bit about file systems next

Page Faults

- A page fault is caused by a reference to a VM word that is not in physical (main) memory
  - Example: An instruction references a word contained in VP 3, a miss that triggers a page fault exception

What’s Inside A Disk Drive?

- Spindle
- Arm
- Platters
- Actuator
- Electronics
- SCSI connector

Image courtesy of Seagate Technology
Disk Electronics

Just like a small computer – processor, memory, network interface.

- Connect to disk
- Control processor
- Cache memory
- Control ASIC
- Connect to motor

Disk “Geometry”

Disks contain platters, each with two surfaces.

Each surface organized in concentric rings called tracks.

Each track consists of sectors separated by gaps.

Disk Geometry (Multiple-Platter View)

Aligned tracks form a cylinder.
Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate. The read/write head is attached to the end of the arm and flies over the disk surface on a thin cushion of air. By moving radially, the arm can position the read/write head over any track.

Disk Operation (Multi-Platter View)

The read/write heads move in unison from cylinder to cylinder.

Disk Structure - top view of single platter

Surface organized into tracks
Tracks divided into sectors

Disk Access

Head in position above a track
Disk Access
Rotation is counter-clockwise

Disk Access – Read
About to read blue sector

Disk Access – Read
After BLUE read
After reading blue sector

Disk Access – Read
After BLUE read
Red request scheduled next
**Disk Access – Seek**

Seek to red's track

**Disk Access – Rotational Latency**

Wait for red sector to rotate around

**Disk Access – Read**

Complete read of red

**Disk Access – Service Time Components**

Seek, Rotational Latency, Data Transfer
Disk Access Time

Average time to access a specific sector approximated by:
- \( T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}} \)

Seek time \((T_{\text{avg seek}})\)
- Time to position heads over cylinder containing target sector
- Typical \( T_{\text{avg seek}} = 3-5 \) ms

Rotational latency \((T_{\text{avg rotation}})\)
- Time waiting for first bit of target sector to pass under r/w head
- \( T_{\text{avg rotation}} = \frac{1}{2} \times \frac{60 \text{ sec}}{\text{RPMs}} \times \frac{1}{\text{track}} \times 1000 \text{ ms/sec} \)
  - e.g., 3ms for 10,000 RPM disk

Transfer time \((T_{\text{avg transfer}})\)
- Time to read the bits in the target sector
- \( T_{\text{avg transfer}} = \frac{1}{\text{RPM}} \times \frac{1}{\text{avg # sectors/track}} \times 60 \text{ secs/1 min} \)
  - e.g., 0.006ms for 10,000 RPM disk with 1,000 sectors/track
  - given 512-byte sectors, \( \sim 85 \text{ MB/s data transfer rate} \)

**Disk Access Time Example**

Given:
- Rotational rate = 7,200 RPM
- Average seek time = 5 ms
- Avg # sectors/track = 1000

Derived average time to access random sector:
- \( T_{\text{avg rotation}} = \frac{1}{2} \times (60 \text{ secs}/7200 \text{ RPM}) \times 1000 \text{ ms/sec} = 4 \) ms
- \( T_{\text{avg transfer}} = 60/7200 \text{ RPM} \times 1/400 \text{ secs/track} \times 1000 \text{ ms/sec} = 0.008 \) ms
- \( T_{\text{access}} = 5 \text{ ms} + 4 \text{ ms} + 0.008 \text{ ms} = 9.008 \) ms

Important points:
- Access time dominated by seek time and rotational latency
- First hit in a sector is the most expensive, the rest are free
- SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
  - \( \sim 100,000 \) times longer to access a word on disk than in DRAM

Disk storage as array of blocks

- Common “logical block” size: 512 bytes
- Number of blocks: device capacity / block size
- Common OS-to-storage requests defined by few fields
  - R/W, block #, # of blocks, memory source/dest

Page Faults

A page fault is caused by a reference to a VM word that is not in physical (main) memory.

Example: An instruction reference that triggers a page fault

```
Virtual address         Physical num
PTE 1: Valid           Phys disk addr
PP 0: null             null
PP 1: null             null
PP 2: null             null
PP 3: null             null
PP 4: null             null
PP 5: null             null
PP 6: null             null
PP 7: null             null
```

"logical block" number can be remembered in page table to identify disk location for pages not resident in main memory
In device, “blocks” mapped to physical store

Disk Sector
(usually same size as block)

Physical sectors of a single-surface disk

LBN-to-physical for a single-surface disk

Disk Capacity

Capacity: maximum number of bits that can be stored
- Vendors express capacity in units of gigabytes (GB), where 1 GB = 10^9 Bytes (Lawsuit pending! Claims deceptive advertising)

Capacity is determined by these technology factors:
- Recording density (bits/in): number of bits that can be squeezed into a 1 inch linear segment of a track
- Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment
- Areal density (bits/in²): product of recording and track density
Computing Disk Capacity

Capacity = (# bytes/sector) x (avg. # sectors/track) x (# tracks/surface) x (# surfaces/platter) x (# platters/disk)

Example:
- 512 bytes/sector
- 1000 sectors/track (on average)
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk

Capacity = 512 x 1000 x 80000 x 2 x 5
= 409,600,000,000
= 409.6 GB

Looking back at the hardware

Connecting I/O devices: the I/O Bus

Reading from disk (1)
Reading from disk (2)
Disk controller reads the sectors and performs a direct memory access (DMA) transfer into main memory.

Reading from disk (3)
When the DMA transfer completes, the disk controller notifies the CPU with an interrupt (i.e., asserts a special "interrupt" pin on the CPU).