CIT 470: Advanced Network and System Administration

Disks

Topics

1. Disk interfaces
2. Disk components
3. Performance
4. Reliability
5. Partitions
6. RAID
7. Failures and backups

Volumes

A volume is a chunk of storage as seen by the server.

- A disk.
- A partition.
- A RAID set.

Storage is organized into layers:

![Diagram showing the organization of storage and volumes](image-url)
Disk Interfaces

**SAS**
Serial attached SCSI (Small Computer Systems Interface)

**SATA**
Serial ATA (Advanced Technology Attachment)

**Fibre Channel**
- High bandwidth (4 Gbps and up)
- Can run SCSI or IP over fiber optic network cabling

**iSCSI**
- SCSI over fast (e.g., 10 Gbps) IP network equipment.

**USB and FireWire**
- SATA drives with converter to use via slow USB/FireWire
- USB 3.0 at 5 Gbps can use full drive throughput

SCSI

Small Computer Systems Interface
- Fast, reliable, expensive.

**Older versions**
- Original: SCSI-1 (1979) 5MB/s
- Recent: SCSI-3 (2001) 320MB/s

**Serial Attached SCSI (SAS)**
- Up to 64k devices
- Up to 6 Gbps throughput, 12 Gbps by 2012
- Can use SATA drives via SATA Tunneling Protocol

IDE/ATA

Integrated Drive Electronics / AT attachment
- Slower, less reliable, cheap.
- Only allows 2 devices per interface.

**Older versions**
- Original: IDE / ATA (1986) 26.4 Mbps
- Recent: Ultra-ATA/133 1.064 Gbps

**Serial ATA**
- Up to 128 devices.
  - 1.5 Gbps (SATA-1), 3 Gbps (SATA-2), 6 Gbps (SATA-3)
### SAS vs. SATA

<table>
<thead>
<tr>
<th>SAS</th>
<th>SATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Higher RPM drives</td>
<td>• Lower RPM drives</td>
</tr>
<tr>
<td>– Up to 15 krpm</td>
<td>– Most 7200 rpm, some 10krpm</td>
</tr>
<tr>
<td>• Smaller drive capacity</td>
<td>– Higher drive capacity</td>
</tr>
<tr>
<td>– Up to 600GB for 15krpm</td>
<td>– Up to 1TB for 10krpm</td>
</tr>
<tr>
<td>• Same as SATA for slower</td>
<td>– Up to 3TB for 7200rpm</td>
</tr>
<tr>
<td>• Higher drive reliability</td>
<td>• Lower drive reliability</td>
</tr>
<tr>
<td>• Higher drive cost</td>
<td>• Lower drive cost</td>
</tr>
</tbody>
</table>

### Hard Drive Components

- **Actuator**: Moves arm across disk to read/write data. Arm has multiple read/write heads (often 2/platter.)
- **Platters**: Rigid substrate material. Thin coating of magnetic material stores data. Coating type determines areal density: Gbits/in²
- **Spindle Motor**: Spins platters from 3600-15,000 rpm. Speed determines disk latency.
- **Cache**: 8-64MB of cache memory
  - Write sequencing and read prefetching
Disk Information: hdparm

```bash
# hdparm /dev/hde
```

```
Model=WDC WD1200JB-00CRA1, FwRev=17.07W17, SerialNo=WDF-WMA8C455367
Config={ HardSect NotMFM HdSw>15uSec SpinMotCtl Fixed DTR>5Mbs FmtGapReq }:
RawCHS=16383/16/63, ThrSize=5760, SerSize=0, ECCbytes=0
BufType=8mbParCache, MaxMultSect=16, MultSect=off
CurCHS=16383/16/63, CurSects=61440, LBA=yes, LBAsect=256414648
IORDY=on/off, PIO={min:120,w/IORDY:120}, DMA={min:120,rec:120}
PIO modes: pio0 pio1 pio2 pio3 pio4
DMA modes: mدام0 mدام1 mدام2
UDMA modes: udma0 udma1 udma2 udma3 udma4
AdvancedPM=no WriteCache=enabled
```

* signifies the current active mode

---

Disk Performance

Seek Time
Time to move head to desired track (3-8 ms)

Rotational Delay
Time until head over desired block (8ms for 7200)

Latency
Seek Time + Rotational Delay

Throughput
Data transfer rate (30-120 MB/s)

---

Latency vs. Throughput

Which is more important?

- Depends on the type of load.
  - Sequential access – Throughput
  - Multimedia on a single user PC
  - Random access – Latency
  - Most servers
Improving Disk Performance

1. Faster Disks
   - Upgrade to a 10krpm or 15krpm disk
2. More Caching
   - Larger caches at OS, controller, or disk level
3. More Spindles (Disks)
   - Spread disk access across multiple disks

Faster Disks: Solid State Drives

- Much lower latency ~ 0.1ms
- Faster throughput ~ 250MB/s
- Much higher cost ~ $1.5-2/GB vs. $0.10/GB
- Flash is slow to erase, so first block write is fast, second block write is slow. Fast SSDs erase asynchronously.

Much Faster Disks: PCIe SSDs

- SSDs saturate 3 Gb/s (300MB/s) SATA bus
- Solution: Use PCIe
- Fusion-IO ioDrive
  - 1424 MB/s read
  - 632 MB/s write
- Drawbacks
  - Only for local disks
  - Unreliable in RAID sets

http://www.brentozar.com/archive/2010/03/fusion-io-iodrive-review-fusionio/
Disk Caching

**Operating System Cache**
- Typically gigabytes in size (uses spare RAM.)
- Power cutoff causes cached data to be lost.
- Use `sync` command to force write to disk.

**Disk Controller Cache**
- Motherboard controllers typically have no cache.
- Server RAID controllers often have 128MB-1GB write cache.
- Some are battery-backed to preserve data if power fails.

**Disk Cache**
- All SATA/SAS disks have 8-64MB cache.
- **Read prefetching** grabs data blocks before requested in anticipation of sequential reads.
- **Write ordering** re-orders writes to minimize head movement (only disk firmware knows drive geometry so higher level caches can’t.)

Cache Types

**Write-Back**
- Write operations finish when data in cache.
- Disk writes happen at a later time (asynchronous.)
- Faster, but if power fails written data disappears.

**Write-Through**
- Write operations finish when data on disk.
- Disk writes happen at same time (synchronous.)
- Slower, but no data will be lost after write complete.

Measuring Disk Performance

- Select an appropriate measuring tool.
- Choose a time when disk is quiescent
  - No background processes making disk accesses

```bash
# hdparm -tT /dev/sda
/dev/sda:
  Timing cached reads: 1954 MB in 2.00 seconds = 977.02 MB/sec
  Timing buffered disk reads: 268 MB in 3.02 seconds = 88.66 MB/sec
```
Whole Disk Reliability

MTTF/MTBF
Average time to failure.
Range from 1 to 1.5 million hours.

Annual Failure Rate (AFR)
Hours per year = 24 × 365 = 8760
AFR = (Hours/year) / MTTF = 0.88% to 0.58%

How many disks fail per year?
int(Disks × AFR)
ex: int(2000 × 0.58%) = 18 disks/year

Bathtub Curve Model

Early phase: high failure rate from defects.
Constant failure rate phase: MTBF valid.
Wearout phase: high failure rate from wear.

Whole Disk Reliability

Failures more likely on traumatic events.
Power on/off.
Annual Return Rates (ARR) > maker AFR
2-4% common
Up to 13%
Why is ARR > AFR?
Sysadmin tests harder than manufacturer tests.
Some manufacturers find 43% of returned drives
good by their standards.
SMART

Self-Monitoring, Analysis, and Reporting Tech
– Monitoring system for disk drives.
– Data collected by `smartd` under Linux.
Provide warnings before drive failure based on
– Increased heat or noise.
– Increased number of transient r/w failures.
SMART categories correlated with disk failure
– First scan error, reallocation errors, probational.
– Enable SMART in BIOS to use then
– Use `smartctl -a /dev/sda` to see SMART data.

Unrecoverable Read Errors

Typical URE rate = 1 in $10^{16}$ bits
– Older drives had much higher rates.
Sector error rate = 1 in $2.4 \times 10^{11}$ sectors
– $10^{16}$ bits / (512 bytes/sector × 8 bytes/bit)
Chance to read a sector correctly
– $1 - 1 / 2.4 \times 10^{11} = 99.9999999995833\%$
Chance to read 1TB correctly
– $(1 - 1 / 2.4 \times 10^{11})^{(10^{12} \text{ bytes} / 512 \text{ bytes/sector})}$
– 99.2%

Partitions and the MBR

4 primary partitions.
One can be used as an extended partition, which is a link to an Extended boot record on the 1st sector of that partition.
Each logical partition is described by its own EBR, which links to the next EBR.
Extended Partitions and EBRs

There is only one extended partition.
- It is one of the primary partitions.
- It contains one or more logical partitions.
- EBRs describe the logical partitions.
- It should contain all disk space not used by the other primary partitions.

EBRs contain two entries.
- The first entry describes a logical partition.
- The second entry points to the next EBR if there are more logical partitions after the current one.

GUID Partition Table

- Replacement for the MBR partition table
  - Allows partitions larger than 2TB, unlike MBR
  - All addresses in LBA, not CHS with various LBA hacks.
  - Protective MBR allows backwards compatibility.
- Used by Intel Macs.
- Can be used by Linux, BSD, 64-bit Windows.

Why Partition?

1. Separate OS from user files, to allow user backups + OS upgrades w/o problems.
2. Have a faster swap area for virtual memory.
3. Improve performance by keeping filesystem tables small and keeping frequently used together files close together on the disk.
4. Limit the effect of disk full issues, often caused by log or cache files.
5. Multi-boot systems with multiple OSes.
RAID

Redundant Array of Independent Disks

RAID capabilities:
- Improve capacity by using multiple disks
- Improve performance with striping
- Improve reliability with redundancy
  - Mirroring
  - Parity

RAID Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Min</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBOD</td>
<td>2</td>
<td>Merge disks for capacity, no striping.</td>
</tr>
<tr>
<td>RAID 0</td>
<td>2</td>
<td>Striped for performance + capacity.</td>
</tr>
<tr>
<td>RAID 1</td>
<td>2</td>
<td>Mirrored for fault tolerance.</td>
</tr>
<tr>
<td>RAID 3</td>
<td>3</td>
<td>Striped set with dedicated parity disk.</td>
</tr>
<tr>
<td>RAID 4</td>
<td>3</td>
<td>Block instead of byte level striping.</td>
</tr>
<tr>
<td>RAID 5</td>
<td>3</td>
<td>Striped set with distributed parity.</td>
</tr>
<tr>
<td>RAID 6</td>
<td>4</td>
<td>Striped set with dual distributed parity.</td>
</tr>
</tbody>
</table>

Striping

- Distribute data across multiple disks.
- Improve I/O by accessing disks in parallel.
  - Independent requests can be serviced in parallel by separate disks.
  - Single multi-block requests can be serviced by multiple disks.
- Performance vs. reliability
  - Performance increases with # disks.
  - Reliability decreases with # disks.
Parity

Store extra bit with each chunk of data.

<table>
<thead>
<tr>
<th>7-bit data</th>
<th>even parity</th>
<th>odd parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000</td>
<td>0000000</td>
<td>1000000</td>
</tr>
<tr>
<td>1011011</td>
<td>1101101</td>
<td>0101101</td>
</tr>
<tr>
<td>1100110</td>
<td>0110011</td>
<td>1110011</td>
</tr>
<tr>
<td>1111111</td>
<td>1111111</td>
<td>0111111</td>
</tr>
</tbody>
</table>

Odd parity
- add 0 if # of 1s is odd
- add 1 if # of 1s is even

Even parity
- add 0 if # of 1s is even
- add 1 if # of 1s is odd

Error Detection with Parity

**Even**: every byte must have even # of 1s.

What if you read a byte with an odd # of 1s?
- It’s an error.
- An odd # of bits were flipped.

What if read a byte with an even # of 1s?
- It may be correct.
- It may be an error where an even # of bits are bad.

Error Correction

XOR each block to get parity information.
\[
A_1 \oplus A_2 \oplus A_3 = (00000111 \oplus 00000101) \oplus 00000000
\]
\[
= 00000110 \oplus 00000000
\]
\[
= 00000110
\]

XOR with parity block to retrieve missing block on bad drive.
\[
A_2 = A_1 \oplus A_3 \oplus \hat{A}_p
\]
\[
= (A_1 \oplus A_3) \oplus \hat{A}_p
\]
\[
= (00000111 \oplus 00000000) \oplus 00000010
\]
\[
= 00000111 \oplus 00000010
\]
\[
= 00000101
\]
RAID 0: Striping, no Parity

Performance
Throughput = \(n \times \text{disk speed}\)

Reliability
- Lower reliability.
- If one disk lost, entire set is lost.
- \(\text{MTBF} = \frac{\text{avg MTBF}}{\# \text{disks}}\)

Capacity
\(n \times \text{disk size}\)

RAID 1: Disk Mirroring

Performance
- Reads are faster since read operations will return after first read is complete.
- Writes are slower because write operations return after second write is complete.

Reliability
- System continues to work after one disk dies.
- Doesn't protect against disk or controller failure that corrupts data instead of killing disk.
- Doesn't protect against human or software error.

Capacity
\(\frac{n}{2} \times \text{disk size}\)

RAID 3: Striping + Dedicated Parity

Reliability
Survive failure of any 1 disk.

Performance
- Striping increases performance, but
- Parity disk must be accessed on every write.
- Parity calculation decreases write performance.
- Good for sequential reads (large graphics + video files.)

Capacity
\((n-1) \times \text{disk size}\)
RAID 4: Stripe + Block Parity Disk

- Identical to RAID 3 except uses block striping instead of byte striping.

RAID 5: Stripe + Distributed Parity

<table>
<thead>
<tr>
<th>Reliability</th>
<th>RAID 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survive failure of any 1 disk.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>RAID 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fast reads (RAID 0), but slow writes.</td>
<td></td>
</tr>
<tr>
<td>- Like RAID 4 but without bottleneck of a single parity disk.</td>
<td></td>
</tr>
<tr>
<td>- Still have to read blocks + write parity block if alter any data blocks.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity</th>
<th>RAID 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n-1) x disk size</td>
<td></td>
</tr>
</tbody>
</table>

RAID 6: Striped with Dual Parity

- Like RAID 5 but with two parity blocks.
- Can survive failure of two drives at once.
Nested RAID Levels

Many RAID systems can use both
– Physical drives.
– RAID sets.

as RAID volumes:
– Allows admins to combine advantages of levels.
– Nested levels named by combination of levels,
e.g. RAID 01 or RAID 0+1

RAID 01 (0+1)

Mirror of stripes.
If disk fails in RAID 0 array, can be tolerated by using disk from other RAID 0.
Cannot tolerate 2 disk failures unless both from same stripe.

RAID 10 (1+0)

Stripe of mirrors.
Can tolerate all but one drive can failing from each RAID 1 set.
Uses more disk space than RAID 5 but provides higher performance.
Highest capacity, performance, and cost.
RAID 51 (5+1)

Mirror of RAID 5s
Capacity = (n/2 - 1) × disk size
Min disks: 6

RAID Failures

RAID sets still work after single disk failure
- Except RAID 0 and 6
- Operate in degraded mode
RAID set rebuilds after bad disk replaced
- Can take hours to rebuild parity/mirror data.
- Some hardware allows hot swapping, so server doesn’t have to be rebooted to replace disk.
- Some hardware supports a hot spare disk that will be used immediately on disk failure for rebuild.

RAID 5 Write Hole

RAID 5 does incremental parity updates
- Read of old parity block
- Read of old block to be replaced
- Write of new data to old block
- Write of new parity block
Blocks can become out of sync with parity since
- You cannot write to two drives atomically
If a block goes bad, RAID 5 will not notice
- Until a drive fails and you rebuild the array
- Receiving garbage in place of that block.
**Hardware vs. Software RAID**

**Hardware RAID**
- Dedicated storage processor, possibly cache.
- OS sees RAID volume as single hard disk.
- Easy to boot or rebuild after drive failure.
- Hardware RAID sets will not work on other controllers.
- SNIA Raid Disk Data Format (DDF) std may fix this in future.

**Software RAID**
- RAID controlled by operating system; will use 1-10% of CPU.
- Must configure OS to boot off either drive of mirror so system can boot if a drive fails.
- Software RAID sets will work on any controller.

**Fake RAID**
- Hardware RAID without a dedicated storage processor; uses 1-10% CPU.
- Cheap, supports limited RAID types, may come with motherboard.
- Fake RAID sets will not work on other controllers.

**UREs and RAID**

Ex: RAID 5 of 8 1TB disks (7TB storage)
- What happens if a disk fails and is replaced?
- RAID5 rebuild reads 7TB to reconstruct 1TB

Probability to read a sector correctly:
- \( \frac{1}{\frac{1}{2.4 \times 10^{11}}} = 99.9999999995833\% \)

Probability to rebuild RAID 5 array:
- \((1 - \frac{1}{2.4 \times 10^{11}})^7 \times \frac{7 \times 10^{12}}{512} \text{ bytes/sector} \)
- 94.5%

We need RAID 6 today.

**You still need backups**

Human and software errors
- RAID won’t protect you from \( z_m - z_f \) / or copying over the wrong file.

System crash
- Crashes can interrupt write operations, leading to situation where data is updated but parity not.

Correlated disk failures
- Accidents (power failures, dropping the machine) can impact all disks at once.
- Disks bought at same time often fail at same time.

Hardware data corruption
- If a disk controller writes bad data, all disks will have the bad data.
References