Backups

Portions courtesy Ellen Liu

Quick Digression: Scripts

- You probably need to write simple scripts for backups (and lab 3)
- A script is just a list of shell commands in a file
  - With permissions set executable, and the shell name at the front:

```
#!/bin/sh
ls | grep pdf | wc -l > pdf-count.txt
```


Outline

- Storage hardware and interface
- RAID
- Storage management layers
- Linux filesystem types and commands
- Backups

Local Storage Hardware

- Basic storage: hard disks, flash memory, magnetic tapes, optical media
  - Last two lack instance access and rewritability. Are mainly for backups
  - Solid state disks (SSD): flash-memory based devices
  - Hard disks (HD): continuous exponential increases in capacity

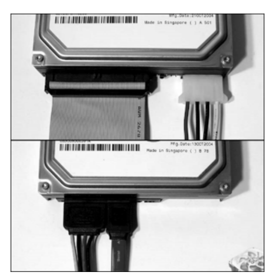
Characteristic	HD	SSD
Size	Terabytes	Gigabytes
Random access time	8ms	0.25ms
Sequential read	100MB/s	250MB/s
Random read	2MB/s	250MB/s
Cost	$0.10/GB	$3/GB
Limited writes	No	Yes

Storage Hardware Interfaces

- Metrics: speed, redundancy, mobility, and price
- PATA: parallel ATA. Commonly called IDE. 40- or 80-conductor ribbon cable. Medium to fast in speed, large capacity, very cheap
- SATA: serial ATA, successor of PATA. Higher transfer rate. Longer maximum cable length. Hot-swapping, command queueing (out-of-order command execution)
- SCSI: still popular. Supports multiple disks on a bus
- Fibre channel: a serial interface. High bandwidth. Can have many storage devices attached to it. Enterprise use
- USB and FireWire: serial interface. For external HDs

ATA Interfaces

- PATA on the left. SATA on the right.



PATA on top, SATA on bottom

SCSI, SAS, and SATA

- SCSI: was popular for high-end disks, tape drives, scanners, printers.
  - Most external devices now use USB
  - Distinguish parallel SCSI, and serial attached SCSI (SAS)
  - SAS improved over parallel SCSI. High-end devices now use SAS
- SCSI hold premium prices, used by the fastest and most reliable drives
  - SATA cheaper and good enough for many uses, limited number of devices
  - SAS faster and can handle many storage devices

Stony Brook University CSE/ISE 311: Systems Administration

RAID

- A disk failure on a server can be disastrous
- RAID: "redundant arrays of inexpensive disks" distributes or replicates data across multiple disks
  - Avoid data loss, minimizes downtime due to disk failure
  - Can be implemented by dedicated hardware, or by OS's reading/writing multiple disks with RAID rules
- Two capabilities
  - Stripe data across multiple drives, allow several drives to supply or absorb a single data stream at the same time
  - Replicate data across multiple drives, decreasing the damage when a single disk fails

Stony Brook University CSE/ISE 311: Systems Administration

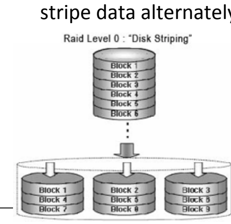
RAID Replication

- Mirroring: data blocks are reproduced bit-for-bit on several difference drives
  - Faster, consumes more disk space
- Parity schemes: one or more drives contain an error-correcting checksum of the blocks on remaining data drives
  - Disk-space efficient, lower performance
- Parity example: Have data 1, 1, 1, 0, 0, 1, 0, 1. With even parity, the parity bit is 1. I.e., the number of 1's in both data and parity is even.
  - If 1st data is changed to 0, what's the new parity bit?
  - If 4th data is changed to 1, what's the new parity bit?

Stony Brook University CSE/ISE 311: Systems Administration

RAID Levels

- Linear mode: concatenate the block addresses of multiple drives to create a single, larger virtual drive
  - No data redundancy or performance benefit
- RAID 0: combine two or more drives of equal size, stripe data alternately among the disks in the pool



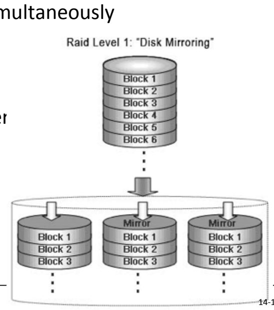
RAID 0: disk striping
- Increased performance
- No data redundancy
- Failure rate of a two-drive array is higher than a single drive

Stony Brook University CSE/ISE 311: Systems Administration

RAID 1

- RAID 1: known as mirroring. Writes are duplicated to two or more drives simultaneously

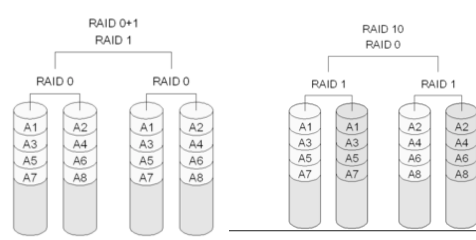
RAID 1: mirroring
- Writes slightly slower
- RAID 0 read speed
- Data redundancy



Stony Brook University CSE/ISE 311: Systems Administration

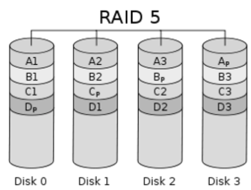
RAID 0+1

- RAID 0+1: Mirrors of stripes
- RAID 1+0: Stripe of mirrors
- For both performance and redundancy



RAID 5

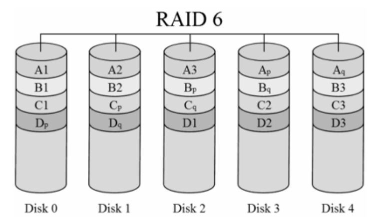
- RAID 5: stripe both data and parity information. In the graph, parity Ap computed for blocks A1, A2, A3. Parity Bp for B1, B2, B3, and so on.
- Parity bits are distributed among the drives



RAID 5: striping with parity
- Added redundancy: the parity bit
- Improved read performance
- More efficient use of disk space than RAID 1
  - N disks, N-1 data

RAID 6

- RAID 6: Two parity blocks (disks). Can withstand the complete failure of two drives without losing data



Drawbacks of RAID 5

- RAID 5 or others do not replace regular off-line backups
  - It does not protect against **power supply failures,** accidental deletion of files, fires, hackers, etc.
- RAID 5 write needs two reads and two writes
  - Reading old data and old parity, compute new parity, write new data and new parity
  - It does not compute parity using all old data, fast but less reliable. Thus an earlier erroneous parity causes error in all subsequent parities. Called "write hole", it backfires if a disk fails
  - Can use "scrubbing" to validate parity blocks while idle

Storage Management Layers

- A hard disk can be conceptually divided into partitions or logical volumes for data management
- To manage files, a filesystem mediates between raw disk blocks and standard filesystem interface
- So roughly three layers
  - Storage device and RAID on the bottom, Logical volumes and partitions in the middle, Filesystem on the top
- There are different types of filesystems
  - UNIX allows co-existence of more than one filesystem types
- Filesystem implementation: inodes, superblock, etc.
  - Typically a chapter in an OS course

Linux filesystems: the ext family

- Ext2: the second extended filesystem. Mainstream Linux filesystem type for a long time
- Ext3: added journaling capability to ext2, increases reliability. Default for Red Hat
- Journaling: ext3 sets aside an area on disk for a journal
  - When a filesystem operation occurs, the required modifications are first written to the journal
  - If it completes, the normal filesystem is modified
  - If a crash occurs during the update, journal is used to reconstruct a consistent filesystem
- Ext4: an update to the above ones. Common default.

Filesystem Commands

- df: report filesystems' disk space usage
- mkfs: create new filesystem on a device, disk partition
- fsck: check and repair filesystems
- mount: attach the filesystem on some device to the big UNIX file tree
- umount: detach a filesystem from the big tree

```
[root@vl120 ~]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      19679908   1917152  16746948  11% /
/dev/sda1               101086     26390     69477  28% /boot
tmpfs                   126192         0    126192   0% /dev/shm
```

Stony Brook University CSE/ISE 311: Systems Administration

Backups

- Backups: the process of making copies of data so that these additional copies may be used to restore the original after a data loss event
  - One of the most important tasks of sysadmins
  - Backups must be done carefully and on a strict schedule
  - Backup system and media must be tested regularly to verify that they are working correctly

14-19

Stony Brook University CSE/ISE 311: Systems Administration

Hints on Backups (1)

- Perform all backups from a central location
  - Run a script from a central location that executes dump on each machine, or use a backup software package
  - Centralization facilitates administration and restoration
- Label your media
  - Write lists of filesystems, backup dates, format of backups, the exact syntax of the commands used to create them
  - Allow quick restoration
- Pick a reasonable backup interval
  - More often backups are done, less data is lost in a crash
  - Backups use system resources and operator's time

14-20

Stony Brook University CSE/ISE 311: Systems Administration

Hints (2)

- Choose filesystems carefully to backup
  - Filesystems that rarely change need less frequent backups
  - If only a few files change, copy them daily to a partition that is backed up regularly
- Make daily dumps fit on one piece of media
  - E.g., a single tape. If a dump spans multiple tapes, operator must be present to change the media. Hard if it is 4am every day
- Keep media off-site
  - Keep an off-line copy of data always
  - Off-site increases reliability

14-21

Stony Brook University CSE/ISE 311: Systems Administration

Hints (3)

- Protect your backups
  - Encrypt the backup media. Do not lose the encryption keys
  - Physical security too. With safes, lock and key
  - Make duplicates
- Limit activity during backups
- Verify your media
- Develop a media life cycle
- Design your data for backups
- Prepare for the worst

14-22

Stony Brook University CSE/ISE 311: Systems Administration

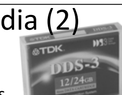
Backup Devices and Media (1)

- Optical media
  - CD-R/RW, DVD+R/RW, DVD-R/RW, DVD-RAM, Blu-ray
  - For small, isolated systems: CD <1GB, DVD 4.7-8.5GB
  - -R or +R are write-once, RW are re-writable
  - DVD-RAM has built-in defect mgmt, reliable, expensive
  - Quality varies. Shelf-life: 1-5 years
  - Blu-ray: 25-100GB
- Portable / removable hard disks
  - Up to few terabytes. SSD lower
- Magnetic tapes
  - Vulnerable to sources of electrical or magnetic fields: audio speakers, power supplies, motors, disk fans, etc.

14-23

Stony Brook University CSE/ISE 311: Systems Administration

Backup Devices and Media (2)



- Small tape drives, DDS/DAT
  - low end tape storage. Up to 10yrs' life
  - up to 80GB, 6.9MB/s speed, 100 backups
- DLT/S-DLT: reliable, affordable, capacious
  - up to 800GB, 60MB/s, 20-30years
- Others
  - AIT, SAIT: advanced intelligent tape
  - VXA: a tape backup format
  - LTO: Linear Tape-Open, a tape tech.
  - Jukeboxes, stackers, tape libraries
  - Hard disks
  - Cloud backup services

14-24

Backup Summary

- Data needs to be in multiple machines
  - Multiple physical locations, and off-line (why?)
    - Protect against hackers, machine failure, natural disaster, etc.
  - And encrypted (why?)
    - Protect privacy of data on the backup
    - But don't lose the keys!
- Backup intervals are a balance: data lost vs. load
- Incremental vs. full backups
  - Incremental only saves changes, but can't lose the full
- Periodically (~yearly) check that you can actually restore from your backups using different hardware

Backup Summary (2)

- Periodically check the *integrity* of your backups
  - Is the media ok?
  - Are the same number of files on the backup as on the system?
  - Spot check file contents (compare md5sum hashes)
- If the local file system doesn't support snapshots, you may have some weirdness with concurrent use + backups
  - Note: Databases usually need special steps to backups

Backup Tools

- Lots available
- Often divided into file system vs. block-level backups
  - Default windows backup is a block-level backup. Main drawback is that you can only restore onto a same-sized device
  - Apple Time machine is a file system-level backup
- I (Don) like rdiff-backup
  - Linux-compatible, does full and incremental backups
  - Weekly cron script containing:

```
rdiff-backup /filer /backup
```


A Note on Destroying Media

- Don't just put media in the recycling
  - Even if you cut up a tape, easy to re-spool; cheap services to read platters taken out of a disk
  - Someone might find and read sensitive data
  - Even encryption tools may be broken later
- Use a secure erase tool
  - shred is a good start – writes zeros over every sector
    - Can miss remapped sectors
  - hdparm/sdparm and other utilities include something that clears remapped sectors