Storage Systems

Sponsored by Green Hills Software

Green Hills make the world's highest performing compilers, most secure real-time operating systems, revolutionary debuggers, and virtualization solutions for embedded systems.

Storage Systems

Outline:

- Background
- RAID
- Hardware RAID Controllers
- Logic Volume Manager (lvm)
- ZFS
- btrfs

Storage Systems

- Where this is useful
  - Network Access Storage servers
  - … or just about any server
- Presentation goal:
  - Enough knowledge to know what you can do
  - (too much material for workshop-style coverage)

Background: Disks

- Hard disks
  - Spinning platters
    - Unpredictable performance
    - High failure rate
  - Cheap, high capacity

- Solid State Disks (SSDs)
  - No moving parts
    - No seek delays
    - More reliable?
  - Newfangled, poorly supported

RAID

- Problems:
  - Drives are unreliable
  - High capacity drives are too expensive

- Solution:
  - **R**edundant **A**rray of **I**nexpensive **D**isks

RAID: Striping

- Idea:
  - Split data across multiple physical drives
  - Spread data out so we can access it in parallel

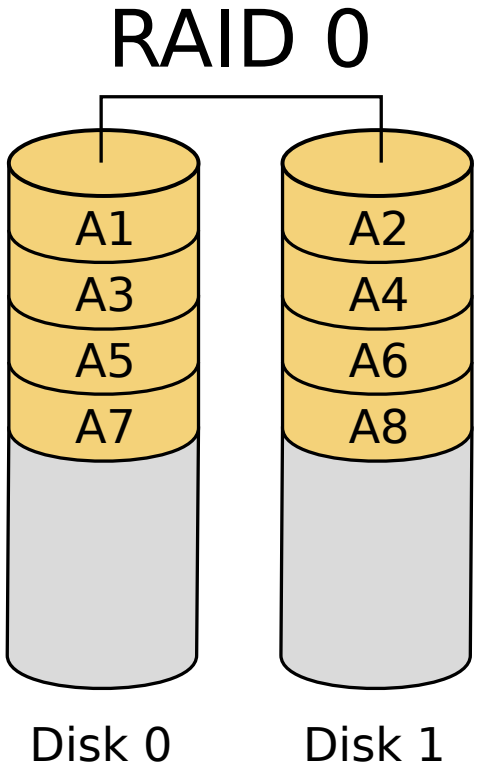

Photo credit: wikipedia
http://en.wikipedia.org/wiki/File:RAID_0.svg

RAID: Mirroring

- Write everything onto two disks
  - If one fails, we have a backup

- Performance
  - Writes: slower
    - Two writes done in parallel
      - … but seek delays may be different
  - Reads: faster
    - Read from whatever one is convenient
      - or a little of both

RAID: Parity

- Idea:
  - Take XOR of all the bits in a striping setup
  - Can now calculate missing bits if one goes down
- Distributed parity
- Extra parity disks

RAID: Levels

- Look them up on wikipedia
  - http://en.wikipedia.org/wiki/RAID#Standard_levels
- Most popular:
  - RAID 0: Striping
  - RAID 1: Mirroring
  - RAID 6: Striping + double distributed parity
- Hot spares

Hardware RAID

- Fully in the hardware
  - Software is oblivious

- Advantages
  - Sometimes faster
  - Easier to configure
- Disadvantages
  - Oblivious to disk contents
    - May “rebuild” uninitialized data
  - Cannot migrate disks across different hardware

LVM: Overview

- Linux **L**ogical **V**olume **M**anager
- Layer of indirection between disks and filesystem
- Features:
  - RAID
  - Easy addition/removal of capacity
  - Copy-on-write snapshotting
  - Thin provisioning
  - etc

LVM: Layout

- Physical Volumes:
  - Drives
  - Or more exotic things: partitions, files in existing file systems, etc
- Volume Group:
  - Set of Physical Volumes
    - (possibly only one)
- Logical Volume:
  - Exported “disk partitions”
  - ie. where you actually put your file system

LVM: Layout

Photo credit: wikipedia http://en.wikipedia.org/wiki/File:RAID_0.svg

LVM: Conclusion

- Advantages
  - Lots of nice features

- Disadvantages
  - Decreased compatibility
  - Increased complexity
    - More things to go wrong
  - Oblivious to file system contents/format
    - Makes some operations more expensive (eg. snapshots)

ZFS: Overview

- Copy-on-write file system
  - ie. never write on top of live data
  - More robust (?)

- Features:
  - RAID
  - Easy addition/removal of capacity
  - Copy-on-write snapshotting
  - Thin provisioning
  - etc

  - … look familiar?

ZFS: Comparison to LVM

- Advantages:
  - Aware of file system contents/structure
    - Faster rebuilds in sparse RAIDs, etc
  - Data integrity checks
    - And fixing if using a RAID
  - Copy-on-write volumes

- Disadvantages
  - Uses a lot of memory
  - Traditionally poorly supported under Linux
    - No longer an issue

btrfs

- Like ZFS, only... not ZFS
- Advantages over ZFS:
  - Not much...

NFS

- **N**etwork **F**ile **S**ystem

- Export folders on local file systems as network-mountable volumes
- Fairly performant and reliable

- Security:
  - IP whitelisting and Unix permission bits
    - Need to trust ever host who can mount it

- Simple and functional

NFSv4

- Newer and better?

- Security improvements
  - … very complicated
-
- Other improvements
  - … more complexity