ECE 574 – Cluster Computing
Lecture 17

Vince Weaver

http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

3 November 2015

Announcements

- HW#6 was posted

- Don't forget project topics

Pi-Cluster Update

- 10 nodes up, 7 GFlops

- Old core2 = 17 GFLOPS, Haswell (hw machine) = 123 GFLOPS

- Lots of problems getting it going with MPI.
  Forgot to compile with mpicc
  Need to tell slurm one-instance-per-node
  Need to set PxQ value in HPL.dat

- Need another day of soldering

Big Data

- Until now compute or network bound systems

- What if we want to do a lot of disk/IO? Big Data?

- Where is Data Used a lot?

  – Google
  – Worldwide LHC Computing Grid (WLCG) as of 2012
     25 petabytes of data/year (petabyte=1000 terrabytes)
     300 GByte/s of data incoming

- Big Data is about big data sets (Terrabytes?) and trying to get useful information out of them. Data Mining.

Types of Storage

- Hard Disks
  - spinning rust – can be slow latency wise
  - SSD – faster, why?
  - Traditional vs Advanced features
    Shingled (SMR) Disks
    Perpendicular (PMR) Disks
    `https://www.youtube.com/watch?v=xb_PyKuI7II`
    Helium
    Caches

- Other flash / SD cards
- Memristors/ Non-volatile RAM
- Tape – robot tape libraries, etc
- Optical – CD/DVD/Blueray

RAID

- Redundant Array of (Independent / Inexpensive) Disks

- Patterson Gibson and Katz 1987: replace expensive mainframe disks with arrays of relatively cheap desktop drives

- RAID0: striping, spreading across multiple disks, can increase performance, increases size of disk, bad things happen if one drive fails

- RAID1: mirroring – same thing written to both drives

can increase performance as either drive can answer request

- RAID2: hamming code, each bit on separate drive. Not really used.

- RAID3: byte-level striping with parity. not common

- RAID4: block-level striping with dedicated parity.

- RAID5: block-level striping with distributed parity. can handle failure of single disk, can rebuild based on parity.

Not recommended, as you have to read entirety of all other disks to rebuild, likely to fail other disks if all of same vintage

- RAID6: block-level striping with double parity. Recommended

- Hybrid: RAID10 = RAID1 + RAID0

- Software vs Hardware

- Some filesystems include RAID like behavior: ZFS, GPFS, brfs, xfs

Non-RAID

- nearline storage – not offline but also not fully online

- MAID – Massive Array of Idle Drives
  Write once, read occasionally.
  Data that you want to save, but really don't access often
  so taking seconds to recall is OK.
  What kind of data? Backups? Old Facebook pictures?
  Old science data?

Cluster Filesystem

- Filesystem shared by multiple systems

- Parallel filesystems – spread data across multiple nodes

Shared-disk Filesystem

- Shared-disk filesystem – shares disk at block level

- SGI CXFS
  IBM GPFS
  Oracle Cluster Filesystem (OCFS)
  RedHat GFS
  Many Others

- RedHat GFS2
  Distributed Lock Manager

SAN – Storage Area Network

- (Don't confuse with NAS – network attached storage)

- A network that provides block-level access to data over a network and it appears to machines the same as local storage

- DAS – direct attached storage – typically how you hookup a hard-drive

- NAS – like a hard-drive you plug into the Ethernet but

serves files (not disk blocks) usually by SMBFS (windows sharing), NFS, or similar

- NAS: appears to machine as a fileserver, SAN appears as a disk

- SAN often uses fibrechannel (fibre optics) but can also be over Ethernet

Concerns

- QoS – quality of service. Limit bandwidth or latency so one user can't overly impact the rest

- Deduplication

Cluster Storage

- Client-server vs Distributed

- Multiple servers?

- Distributed metadata – metadata spread out, not on one server

- Striping data across servers.

- Failover – if network splits, can you keep writing to files

• Disconnected mode.

Network Filesystems

- NFS, SMBFS, Netware

Distributed Filesystem Architectures

From *A Taxonomy and Survey on Distributed File Systems*

Can be any combination of the following

- Client Server – like NFS

- Cluster Based – single master with chunk servers

- Symmetric – peer to peer, all machines host some data with key-based lookup

- Asymmetric – separate metadata servers

- Parallel – data striped across multiple servers

Stateless

Can you reboot server w/o client noticing? Lower overhead if server stateless because the server doesn't have to track every open file in the system

Synchronization/File Locking

- Multiple users writing to same file?

- Always synchronous?

- Same problems with consistency that you have with caches/memory

Consistency and Replication

- Checksums to validate the data

- Caching – if you cache state of filesystem locally, how do you notice if other nodes have updated a file?

Failure Modes

Security

Distributed Filesystems

- Follow a network protocol, do not share block level access to disk

- Transparency is important. User doesn't need to know how it works underneath.

- Ceph, GFS, GlusterFS, Lustre, PVFS

Lustre

- "Linux Cluster"

- Used by many of the top 500 computers

- Can handle tens of thousands of clients, tens of perabytes of data across hundreds of servers, and TB/s of I/O.

- One or more metadata servers: keep track of what files exist, metadata, etc, can load balance.

- One or more object storage servers

- **Distributed Locking**

- **Fast networking. Use RDMA when available.**