Lessons Learned in Deploying the World's Largest Scale Lustre File System

Presented by

David Dillow

Galen Shipman, David Dillow, Sarp Oral, Feiyi Wang, Douglas Fuller, Jason Hill, and Zhe Zhang

Brief overview of Spider

- 10 PB storage to users
- 244 GB/s demonstrated bandwidth
- Currently serves 26,887 clients
- Based on Lustre 1.6.5 plus Cray and Oracle patches

Spider Hardware

- 13,696 1 TB SATA Drives
  - 13,440 used for object storage
  - 256 used for metadata and management
- 48 DDN 9900 Couplets (IB)
- 1 Engenio 7900 Storage Server (FC)
- 192 Dell PowerEdge 1950 Object servers
- 3 Dell R900 Metadata servers
- Other various management servers

Why Spider?

- Data availability
  - Better MTTI/MTTF than locally attached Lustre
  - Available during system maintenance periods
- Data accessibility
  - No need to copy from simulation platform to visualization/analysis clusters
  - Allows use of dedicated transfer nodes for movement off-site

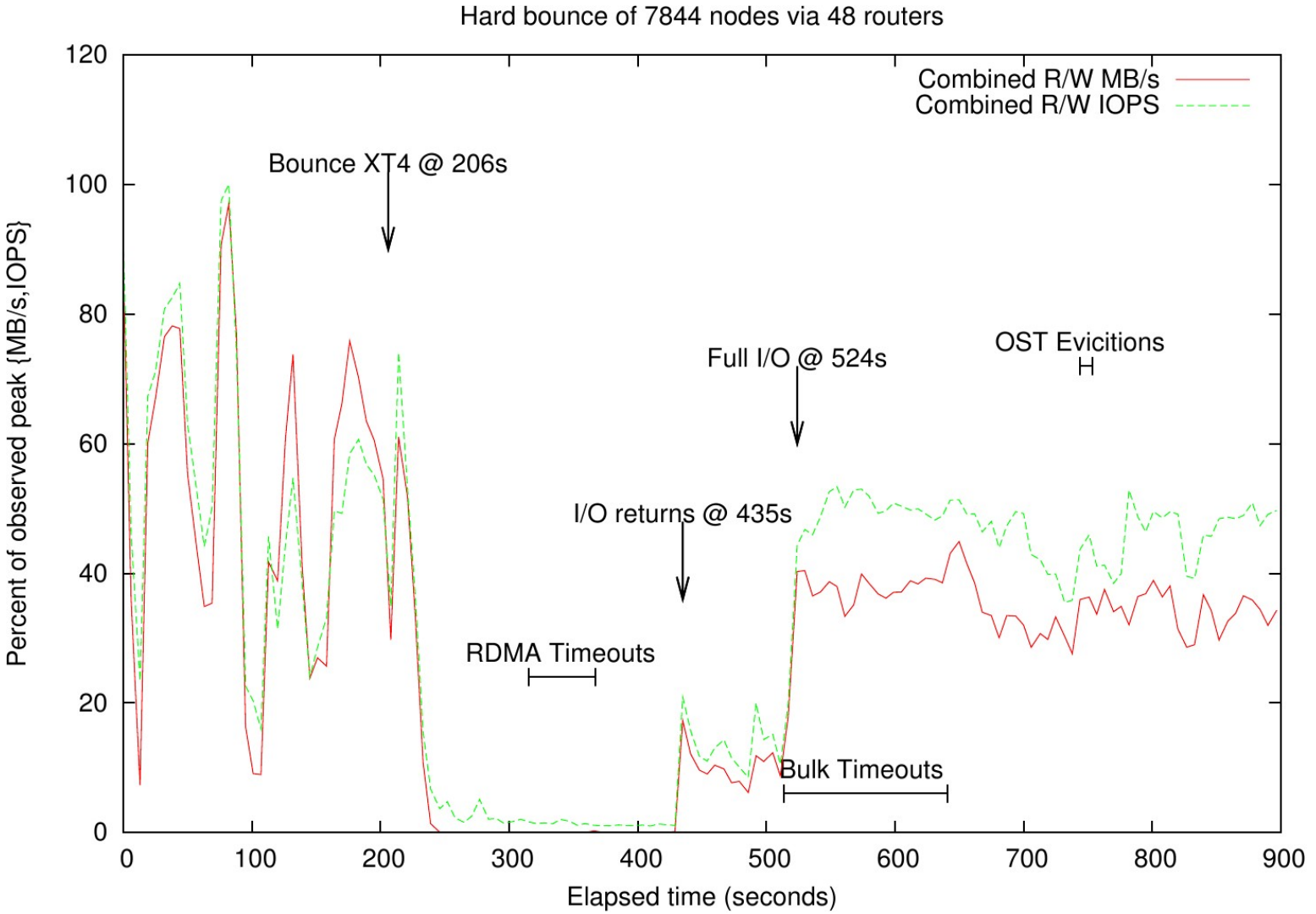
What could go wrong?

- Interference from other systems
- Interactive performance
- Hardware failures
- Management headaches
- Lustre bugs
- Scapegoat syndrome

System Interference

- It's a shared resource
- Big jobs on the XT5 necessarily impact smaller clusters during heavy I/O
- Turns out to be mostly a non-issue
  - Most users I/O needs seem to be more modest
  - Occurs within a single cluster as well if multiple jobs are scheduled
  - “Mostly....”

I/O Shadow


Hard bounce of 7844 nodes via 48 routers
Combined R/W MB/s
Combined R/W IOPS
Bounce XT4 @ 206s
Full I/O @ 524s
OST Evicitions
I/O returns @ 435s
RDMA Timeouts
Bulk Timeouts
Percent of observed peak {MB/s,IOPS}
Elapsed time (seconds)
0
20
40
60
80
100
120
0
100
200
300
400
500
600
700
800
900

Metadata scaling

- Conventional wisdom is that one needs huge disk IOPS for metadata service
- Provisioned the Engenio for MDT
  - RAID10 of 80 1 TB SATA disks
  - Short stroked to an 8 TB volume
  - Write-back caching enabled
    - With mirroring!
- Achieved 18,485 read and 23,150 write IOPS
    - 4 KB requests, random seeks over 8TB

Metadata scaling

- Conventional wisdom may be a bit inaccurate
  - I/O rates very low during steady state operation
    - Bursts of ~1000 8K write requests every 5 seconds
    - Bursts of ~1500 to 3000 8K writes every 30 seconds
    - Occasional reads
  - MDS memory sizing is paramount!
    - Try to fit working set size into memory
    - We have 32 GB which does well so far
  - IOPS more important when cache cold after mount

Metadata scaling

- Lock pingpong hurts large jobs with shared files
  - Opening a file with O_CREAT holds lock over a full round trip to client from MDS
    - 65,536 core job with O_CREAT takes 50 seconds
    - 65,536 core job without takes 5 seconds
  - Lustre 1.8.3 fixes this if the file exists
  - Still takes an exclusive lock, though
  - But why does this hurt other jobs?

Metadata scaling

- Lock pingpong hurts interactivity due to Lustre's request model
- Every request is handled by a thread
- If the request needs to wait on a lock, it sleeps
- If you run out of threads, request handling stalls
- No quick fix for this
  - Can bump up the thread count
  - High thread counts can cause high CPU contention

Hardware failures

- We've dodged many bullets
  - Server hardware has been very reliable
  - Relatively few disk failures (one to two per week)
  - More singlet failures than we'd like
    - Upper bound of about 2 per month
    - Some of those were likely software faults rather than HW
    - Multipath has had good results surviving a singlet failure

Hardware failures

- We've also had some issues
  - SRP has had a few issues releasing the old connection
  - Leaf modules in the core IB switches seem to prefer Byzantine failures
  - OpenSM has not dealt gracefully with flapping links
  - No one seems to make a good power supply
  - MDS soft lockups
  - OSTs transitioning to read-only due to external event

Management issues

- How do you find needles with particular attributes when you have 280 million of them?
  - lfs find -R –obd <OST>
    - Over five days to complete
  - Ne2scan
    - Almost two days
  - Find
    - Three days with no arguments that require a stat call

Management issues

- Who is beating up the file system?
- Who is using the most space?
- How does our data and IO operation rates trend?
  - Can we tie that to an application?

Scapegoat Syndrome

- Lustre has bugs
- Modern Lustre is much more of a “canary in the coal mine” than the walking trouble ticket it has been in the past
- User's first response is that “the file system is slow!”
  - Even if the root cause is that the HSN is melting down

Questions?

- Contact info:

  David Dillow

  865-241-6602

  dillowda@ornl.gov