Outline

• Overview/Diagram of “Spider”
• “must haves” for any Lustre FS
• Some Uptime/Outage information
• User Experience
• Administrative Experience
• Where we’re heading from here
• Wrapup/Questions
“Must Haves”

• Budget is the final driver
• RAID + dm-multipath
• Redundant SAN (As much bandwidth as you can buy)
• Lustre Failover node
• High Availability Solution
• Diskless Boot Environment
  – If not available, use Centralized Configuration Management
  – Probably good to use even with Diskless Env.
• Policy for deleting data
  – Sweep for files that are older than 14 days (daily)
Uptime

• Hard to measure
  – MDS/MGS/OSS uptime?
  – Currently MDS up short time
    • Downtime to add threads
  – Some OSSes have over 130 days of uptime
    • Ones that don’t have had backend storage problems

• MTTI – ex. OST/OSS unavailable

• MTTF – MDS or filesystem unavailable
Hiccups?

• We see some IO “hiccups”
  – Run MDS out of servicing threads

• XT dies, large IO job holding locks
  – Spend odb_timeout trying to talk to the clients
IO Shadow

Hard bounce of 7844 nodes via 48 routers

Bounce XT4 @ 206s

Full I/O @ 524s

OST Evictions

I/O returns @ 435s

RDMA Timeouts

Bulk Timeouts

Combined R/W MB/s

Combined R/W IOPS

Percent of observed peak (MB/s, IOPS)

Elapsed time (seconds)

0 100 200 300 400 500 600 700 800 900
Outages/Saved Outages

• dm-multipath saves the day
  – Since 07/13/2009, 25 controller failures resulted in 6 service
    interrupts
• Other outages for LBUG’s (panic on lbug set)
• Testing time (center wide, rare)
• Issues with ib_srp
• Backend storage problems
Feb 25th 2010

• Backend storage problem caused us some grief
• Complicated by bug in e2fsck
  – Freeing inodes that were not used, but are part of a valid stripe
• What did we learn?
  – Verify connectivity to disks on both controllers before rebooting,
    check status of all disks
  – e2fsck –n
  – We need better tools
Tools

- Need better way to get what files are on an OST
  - Have ne2scan output, but that’s not current as of time of failure
  - Recommendation from Oracle folks was lfs getstripe –obd=
    • Took 5 days on OST with ~1M objects

- Need way to map inode on MDS to a filename/path

- Lustre error message marking OST as read only could use a note about the path it’s using
  - Very hard to determine what OST was marked read only because of IO errors from backend storage
Admin Experience

• At this scale “Lustre” can be difficult to work with
  – Log parsing from 210 servers with differing filesystems
  – Log correlation between 27k clients and 210 servers more difficult
  – Host Monitoring
    • We killed our existing Nagios Infrastructure by putting ~1500 more checks into it just for Lustre (more coming)
  – Storage monitoring (not Lustre’s job)
  – Maintenance windows
    • We’re at the center of everything. When the FS is down we stop processing.
Admin Experience

• Developed some tools to help
  – DDN monitoring tool (Ross Miller)
  – MDS/OSS RPC Tracing application (David Dillow)

• Still need to develop some more tools
  – IB Fabric monitoring
  – Log parsing/Event notification

• Never sure of an applied patch until you try it against 27k Lustre clients
Admin Experience

• Some key admin tools
  – Syslog-ng
  – SEC
  – Nagios
  – DDNTool
  – Tail, grep, etc.
  – LCTL
  – Routerstat
  – MDS/OSS trace
  – Ne2scan/genhit/purge/fsfind

• DDNtool is working its way to being released
Admin Tools

• DDNTool
  – Use DDN API to get information
  – Classes of information
    • Performance Stats (fast polling – every 2 seconds)
    • Failure Data (medium polling – every 60 seconds)
    • Environmental Data (slow polling – every 30 minutes)
  – Can write application to query this data in anything
  – Each update erases the data from the previous run
  – If you want to track it, slurp the data into your own database
    • It gets large **very** quickly
Admin Tools

- Syslog-ng
  - Use rules here to put types of messages into the same logfile
    - Weekly run of all kernel messages from the oss nodes for an example
    - Lots of possibilities here

- SEC
  - Simple Event Correlator
  - Write rules to alert if you see a specific syslog message
  - We trigger on OSS reboot
  - Have some rules for non-contiguous memory for IB page allocation
  - Need to continue to dig deeper into error messages to send more
  - Can have issues after long uptimes with the message suppression algorithm in Lustre (don’t see messages in a timely fashion because lots are suppressed).
Admin Tools

– Nagios
  • Make sure it’s robust for large filesystems
  • We check for server health
    – Ping, SSH, Environmentals, Voltage
    – Multipath health
    – Load\n  • Probably more to check for, and we’ll add that as the Nagios infrastructure allows
  • Also monitor ping for DDN controllers
  • Haven’t investigated snmp traps from DDN yet, the API is working well for us
Admin Tools

- lctl
  - Disable OST’s, remove OST’s completely
  - Manage routes
  - All the normal stuff
- Routerstat
  - Locally modified to put in date and time to the output
  - Use on the MDS to see:
    - Messages in flight (and max messages)
    - Errors
    - MB Sent/Received
    - LNET information
Admin Tools

- MDS/OSS trace
  - Lctl dk >> /dev/null
  - Echo + rpctrace >> /proc/sys/inet/debug
  - Wait
  - Echo – rpctrace >> /proc/sys/inet/debug
  - Lctl dk >> /tmp/myfile

- Use this data to correlate information back to nodes (LNET nids)
- Gather apstat information from XT5 and XT4 and do further correlation
- Generate report and e-mail if the average LDLM_ENQUEUE time is longer than 1 second
- Do this every 10 minutes; We get ~50 emails a day
- Run by hand on OSS currently
Admin Tools

- Ne2scan
  - Get all information off the mdt (modified e2scan by Nick Cardo @ NERSC)
- GenHit
  - Generate list of files that are older than a specified date from the ne2scan output
- Purge
  - Stat the file, make sure mtime/ctime allow for deletion
  - Unlink
- Fsfind
  - Find files for a specific user
  - Find files on a specific OST
  - Find files that are setuid/setgid
- Talk to Nick – should be going through tech transfer here soon
Addressing Scalability Challenges

• Periodic user reports that the filesystem is slow
  – Very helpful!

• Mainly interactive performance is poor
  – Is –l /tmp/work/user

• Our observations are that we’re running out of locks when large jobs spin up or do checkpoint
  – Ex. Single shared file with 48k cores

• Some things we’ve done to help
  – Turn off color ls
    • Doesn’t help if user is doing ls –l
  – Changed mds_num_threads from 128 (default) to 2048
    • Is a local modification/patch, not generally advised
    • Had tried up to 16k
    • At that scale saw 1.6 Million context switches per second during MDT unmount
    • MDT unmount took over 20 minutes
Addressing Scalability Challenges

• Things to think about moving forward
  – Give users information about changing their .vimrc to move the swap file (or turn it off)
  – Potentially recommend alternate workflows that have less impact on the MDS
  – Deploying multiple (order 2-3) shared filesystems
    • Allows better segregation of users and use cases
    • Gives smaller collision domain if one project/code has a pathological IO behavior

• All of these are band-aids to the thread model on the MDS
  – Fixing thread sleeping model is not easy
Moving Forward

• Administratively we have a lot of work to do
  – Deploying HA/Failover
  – Nagios checks
    • Mounted OST’s, memory utilization, IB Health
  – Deploying Lustre 1.8
  – With that a focus on getting better MDS performance for interactive use
    • This is a question, not a solution
    • Is there a way to get part of the namespace assigned to a particular MDS?
    • Could be bridge to clustered metadata
Performance Numbers

Bandwidth GB/s

Timeline (February 2010)
Performance Numbers

Bandwidth GB/s

Timeline (March 2010)

ReadBW GB/s

WriteBW GB/s
Wrapup

- Always plan for failure
- Develop procedures for scenarios
- We’ve had pretty good luck, but Murphy strikes all
- Lots of tools to develop
- More bugs to squash