Monitoring Tools for Large Scale Systems

Presented by

David Dillow

Ross Miller, Jason Hill, David Dillow,
Raghul Gunasekaran, Galen Shipman, Don Maxwell

Brief overview of Spider

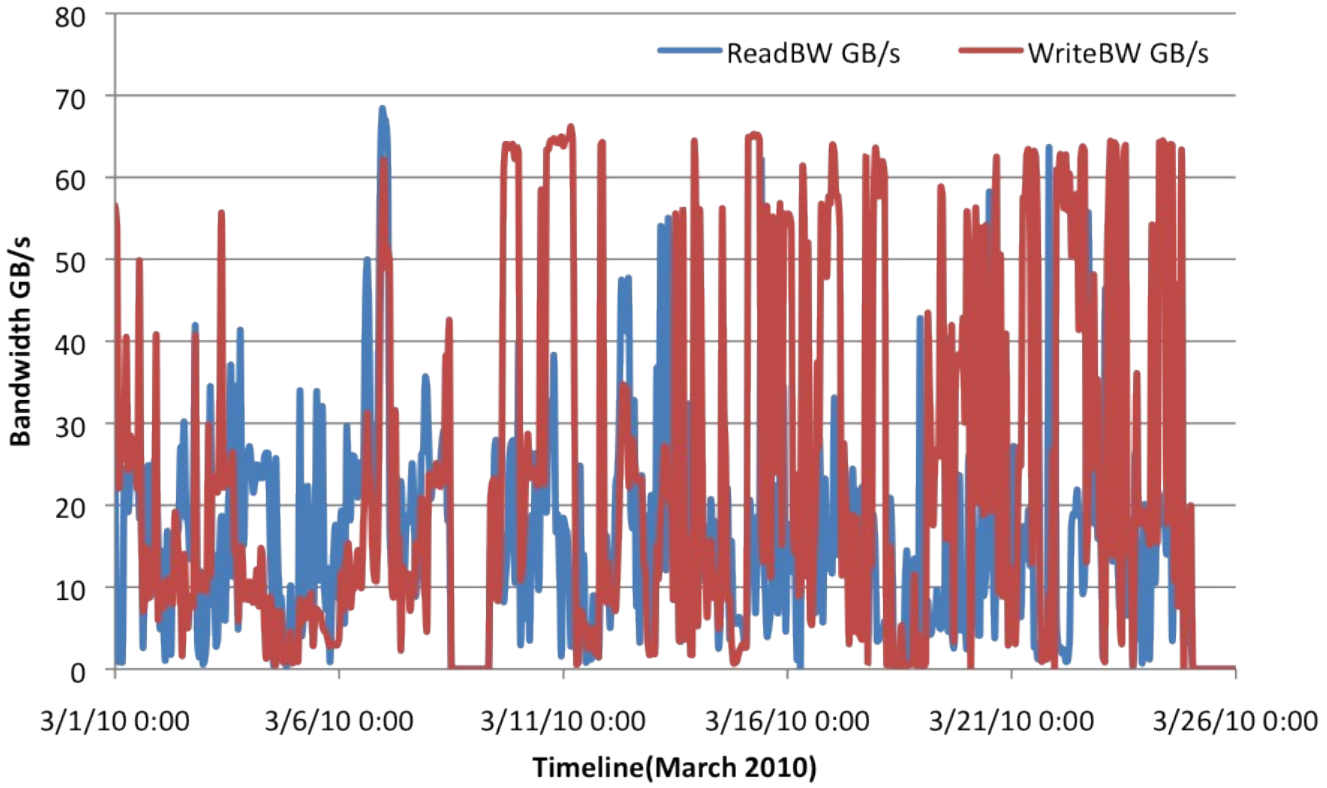
- 10 PB storage to users
- 244 GB/s demonstrated bandwidth
- Currently serves 26,887 clients
- Based on Lustre 1.6.5 plus Cray and Oracle patches

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

Monitoring Aggregate Performance

- What does day to day usage look like?
- What is the duty cycle?

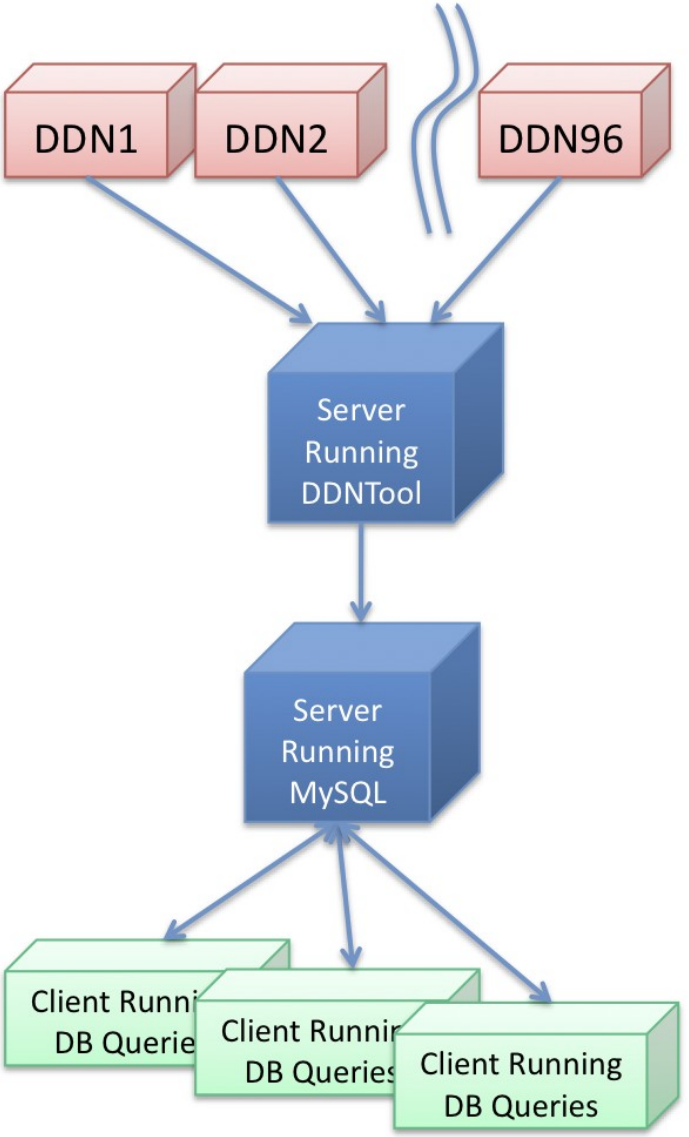


Monitoring Aggregate Performance

- Vendor tools insufficient for gathering this data
  - Serialized polling
  - No data history
  - Limited data selection
- DDNTool
  - Polls all controllers in parallel
  - Allows clients to collect history
  - Rich assortment of data collected

Monitoring Aggregate Performance

- Data is stored in in-memory MySQL tables
  - Transient storage, overwritten with each polling period
  - Clients can pull to more permanment storage
- Multiple clients can retrieve same data without overloading the DDNs
- Clients use standard SQL to get to data
  - Well known API
  - Multiple language bindings
- Clients only need know about the subset of data they care about
  - Performance client retrieves bandwidth and IOPS history every two seconds
  - Environmental client retrieves temperature sensor data every minute
  - Configuration client checks zoning once an hour



Monitoring Metadata Performance

- Is the metadata server under water?
- If so, who is causing this?
- Useful to know, to correct application IO patterns
  - and to avoid tar and feathers!
- Basic tools like “routerstats” can give LNET loading, but do not indicate queue times or correlate to applications

MDSTrace

- Analyzes a nominal 60 second RPC trace
  - can be less if debug logs overflow
  - can analyze longer periods if you can get the data and don't mind waiting
- Aggregates RPC data and associates it with running applications
- We run it every 10 minutes
  - general health check
  - can miss “bursty” performance issues
- Not a turn-key tool – much interpretation needed!

MDSTrace Legend

- Lustre terms
  - LDLM_ENQUEUE is an open() or stat() call
  - LDLM_CANCEL is a lock release
  - MDS_REINT is usually mkdir() or mknod()
  - MDS_READPAGE is readdir()
- Abbreviations for request times
  - pmin/pavg/pmax is Processing Time
  - tmin/tavg/tmax is Total Time

Sample MDSTrace Output

Begin 1274308813.824492 (Wed May 19 18:40:13 EDT 2010)
End 1274308861.475305 (Wed May 19 18:41:01 EDT 2010)
Elapsed time ~ 48 seconds

Minimum ENQUEUE/REINT/STATFS observable latency: 48us
Average ENQUEUE/REINT/STATFS observable latency: 4943us
Maximum ENQUEUE/REINT/STATFS observable latency: 1802519us

Total: 50401 RPCs ~1050 per sec
- 25169 LDLM_ENQUEUE RPCs ~524 per sec
  - pmin 33us pavg 5130us pmax 1802494us
  - tmin 48us tavg 5286us tmax 1802519us
- 16299 MDS_CLOSE RPCs ~339 per sec
  - pmin 38us pavg 17333us pmax 989077us
  - tmin 64us tavg 17507us tmax 990601us
- 7240 MDS_REINT RPCs ~150 per sec
  - pmin 49us pavg 3693us pmax 188734us
  - tmin 77us tavg 3777us tmax 188879us
- 1456 LDLM_CANCEL RPCs ~30 per sec
  - pmin 21us pavg 148us pmax 2591us
  - tmin 43us tavg 302us tmax 27380us

Sample MDSTrace Output

XT5 Application 2011794 ('application A', user A, res 2526)
5133 RPCs from 1015 of 1024 nodes
~106 per sec
4649 MDS_CLOSE RPCs ~96 per sec
pmin 51us pavg 60501us pmax 989077us
314 LDLM_CANCEL RPCs ~6 per sec
pmin 42us pavg 487us pmax 2591us
170 LDLM_ENQUEUE RPCs ~3 per sec
pmin 57us pavg 22227us pmax 1308450us
Overall times
pmin 42us pavg 55562us pmax 1308450us

XT4 Application 3587244 ('application A', user A, res 1584)
668 RPCs from 1 of 257 nodes
~13 per sec
338 MDS_CLOSE RPCs ~7 per sec
pmin 46us pavg 79us pmax 1120us
330 LDLM_ENQUEUE RPCs ~6 per sec
pmin 63us pavg 1323us pmax 135118us
Overall times
pmin 46us pavg 694us pmax 135118us

Sample MDSTrace Output

Service node 'XT4 login12' sent 13849 RPCs ~288 per sec
- 8340 LDLM_ENQUEUE RPCs ~173 per sec
  - pmin 33us pavg 1029us pmax 1648279us
- 5319 MDS_CLOSE RPCs ~110 per sec
  - pmin 38us pavg 70us pmax 6151us
- 69 MDS_REINT RPCs
  - pmin 122us pavg 275us pmax 563us
- 67 LDLM_CANCEL RPCs
  - pmin 24us pavg 43us pmax 84us
- 54 MDS_READPAGE RPCs
  - pmin 192us pavg 1295us pmax 41438us
- Overall times
  - pmin 24us pavg 653us pmax 1648279us

Sweeping the System

- Spider is a scratch filesystem
  - If it is important, copy it elsewhere
- Not using quotas, so need a way to keep space usage under control
  - Periodic sweeps to remove file older than policy allows
- ne2scan/genhit/purge trio helps immensely
  - approx two days to generate file list w/ 280 million files
  - file list reused to generate “hit” list for purges
    - 14 hours to generate 133.4M candidates
    - 14+ days to delete those 133.4M files
    - Normal purge runs are much faster (12 to 24 hrs / 4 to 5 M files)

Monitoring the System Health

- System administrators need to sleep sometimes
- Nagios monitors the system health for us
  - watches for hardware faults
  - especially useful for alarming on sustained high load averages
    - potential signal of a pathalogical I/O pattern
- Simple Event Correlator watchs log messages
  - simple rules: watch for string indicating a reboot
  - complex rules: only alarm if prior event happened
- Working on analysis tools to chew through large logs and help automate isolation of failures

WIP: Who's using my space?

- LustreDU gives a sideband “du”
  - du is terribly slow on a file system of this size
  - a parallel du exists, but can easily swamp the system
- We can reuse the ne2scan file list to quickly generate usage reports
  - Just need a bit more information from the OSSes, which we can get out-of-band
- Can only give usage at time ne2scan was run
  - That's often all that is needed

Questions?

- Contact info:

  David Dillow

  865-241-6602

  dillowda@ornl.gov