Monitoring Lustre
Getting more from performance info
Piles of data on performance, back to when the system was installed.
Tons of emails and metrics on failures

Service: DG_OST_05.OST_05.Virtual Disk
Host: hss4-osts-01-a
State: CRITICAL

Date/Time: Mon Dec 8 08:25:17 EST 2014

Additional Info:
Degraded

Summary:
Filesystem is OK
Total number of services in CRITICAL/WARNING/UNKNOWN state: 6/3/0
Somewhere you have a pretty picture like this of your system.
Somewhere else you've got a database full of interesting configuration and layout.
What to do (Failures)?

- Read it all, and try to figure it out
  - Works for small systems
- Just keep replacing things
  - Ignores underlying problems
- Send equipment back to manufacturer for analysis
  - Takes a long time
  - May not give a root cause
  - Expensive
What to do (Performance)?

- Read it all, try to find correlations
  - Ok for small systems, nearly impossible for large ones
- Hand create displays based on configuration
  - Takes time, lots of information
- Create displays based on items used during jobs
  - Takes a lot of time, needs lots of data
- Probably won't tell you much, because these are data which may not be available anymore
Visualizing Failures

- Temperature anomaly
- Looks good
- Correlation is obvious
- Replaced open floor tile in front of the rack, things cleared up
Visualizing Failures

- Bad disk batch
- Same visualization, not very helpful, everything looks bad
Visualizing Failures

<table>
<thead>
<tr>
<th>Batch</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

- Got the serial numbers
- Use input from MFG to correlate them into batches
- Spotting the errors is straightforward
Visualizing Performance

• If you know your OST pools creating views is simple
• Sometimes this is all you need to see that you need to rethink your approach
Visualizing Performance

- RPC size over time can give insight into how well-behaved I/O is
- Can you stripe some I/O into different OST pools, so that it better fits the pattern? (SSD tiers, etc.)
- Genome workloads are a good example
  - Initial dataset is lots of small files
  - Output is one large file
Merging Data

• How to tell when a system is overloaded?
• Various performance counters help, but you need an Nagios-style interface to report them
• How to know if your metadata performance is MDS/MDT bound, or OSS/OST bound?
  – Compare outstanding RPCs to OSSes, CPU load on MDS, IOPs to disk, etc.
Merging Data

- A failed over OSS pair will have different performance limits from a fully redundant OSS pair
- An OST that is rebuilding will have different performance
- How to convey this information to the admins and on to the users?
Questions?