CSCS Site Update

Stefano Gorini, Matteo Chesi, Carmelo Ponti, CSCS
May 31st, 2017
CSCS IT Architecture

Staff login

Local Area Network
(Ethernet, up to 100 Gbit/s)

Researcher login

Switch
(Ethernet, 100 Gbit/s)

Piz Daint
Sonexion 1600 (scratch)
Sonexion 3000 (scratch)

Mönch
NEC Lustre (scratch)

Escha & Kesch
CRAY CLFS Lustre (scratch)

Data Parallel Transfer

ELA Login Server

Data Centre Network
(Infiniband 56 Gbit/s + Ethernet up to 100Gbit/s)

Project

Store

Tape Library

Traditional Element
- Authentication and Authorization
- Nagios
- Database
- Login server for each SuperComputer
- Slurm
- Firewall
- FlexLM
- ...
Lustre Filesystems @ CSCS
## Lustre for the Flagship System

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>GB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>/scratch/snxA600</td>
<td>2.7 PiB</td>
<td>120</td>
</tr>
<tr>
<td>/scratch/snxA000</td>
<td>6.4 PiB</td>
<td>80</td>
</tr>
</tbody>
</table>

- optimized for very big files
- optimized for writes
- Lustre 2.5
- ~6K client nodes
- Robinhood for cleaning policies
Lustre for TDS and R&D System

- Test and Development Systems: Cray Sonexion 1600 & 2000

- Cray Sonexion 2000 for R&D systems
  - Lustre 2.5
  - Declustered RAID (GridRAID)
  - New Expansion Storage Units
  - 4 OSSs with 2 OSTs each one
  - 41 disks (113 TiB) per OST
  - stripe_count=1

- Management Infrastructure (Nagios, Ganglia, Puppet, Greylog, custom solutions…)

MCH System – CRAY CLFS Lustre

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Escha /scratch</td>
<td>73 TiB</td>
</tr>
<tr>
<td>Kesch /scratch</td>
<td>73 TiB</td>
</tr>
</tbody>
</table>

NetApp 2760 (2TB drives, NL-SAS)
2 CLFS Servers (OSS, MDS, MGS)

Server:
CentOS release 6.4 (Final)
Lustre: 2.5.0

Client:
Red Hat Enterprise Linux Server release 6.7 (Santiago)
Lustre: 2.5.4
Monch – NEC Lustre

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monch /scratch</td>
<td>350 TiB</td>
</tr>
</tbody>
</table>

**Server:**
CentOS release 6.4 (Final)
Lustre: 2.1.6

**Client:**
CentOS release 6.7 (Final)
Lustre: 1.8.9
Data Movers
The Data Mover nodes are managed via SLURM in order to create dependency and a clear workflow with HPC Jobs and data movement via specific tools:

- GRIDFTP
- move
- Cp
- Rsync
- …
Each Lustre at CSCS as a dedicate RobinHood Server to perform the proper cleaning policy

Unfortunately on the main HPC System we are not able to do a real time check of the file system:

- Changelog is too slow in respect of the change rate we have on the FS
- Cleaning Policy is running base on a 24h FS Scan
Application Vs Lustre
Description of the Problem

- Application (pre/post processing Fortran tool) slowdown

<table>
<thead>
<tr>
<th>Condition</th>
<th>zone_reclaim_mode</th>
<th>Number of Runs</th>
<th>Average [s]</th>
<th>Standard dev [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>15</td>
<td>198.533</td>
<td>12.928</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>38</td>
<td>440.921</td>
<td>337.741</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>62</td>
<td>193.677</td>
<td>27.617</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>161</td>
<td>499.379</td>
<td>1133.936</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>173</td>
<td>199.08</td>
<td>11.316</td>
</tr>
</tbody>
</table>
Is it the FS?

- Lets try GPFS…..
  - No Variation the application always perform the same
- So is it Lustre FS storage HW? ....No
Dedicated Test and Analysis Session

- All the problems are not related to an high load on the Lustre file system
- The kernel parameter reclaim \texttt{vm.zone\_reclaim\_mode} has a significant effect on the slowdown (“condition 5”)
- Running the suite on the same node mitigates the slowdown

**Important Remark:**

During the analysis of the application process with \texttt{perf}, in case of slowdown, The application was spending a lot of time with the kernel function \texttt{clear\_page\_c\_e}:

Samples: 1M of event 'cycles', Event count (approx.): 854374192198
13.12% \ [kernel.kallsyms] [k] \texttt{clear\_page\_c\_e}
7.58% \ application\_12.2.0\_gnu4.9.3\_opt\_omp \ [.] \ spumb\_c\
7.35% \ [kernel.kallsyms] [k] \texttt{compaction\_alloc}
Solution

- The customer redesigned the initialization of data arrays (~40 GB on disk) by doing this initialization stepwise.
- With this new version of the library no significant performance fluctuation has been seen.
- Running the test case during more than 12 hours without cache cleaning on all nodes (“condition 5”)
- The new initialization even improves the performance
- BUT the problem is still there:

“Lustre 2.9: is it fixed in this version?”
Next Challenges