Installing & Configuring Lustre on KVMs

Naomi Cheeves, Gabe Maxfield

August 2, 2022
Team Members

- Naomi
  - Senior at University of California, Davis
  - Computer Science
  - Graduating December 2022

- Gabe Maxfield
  - Sophomore at Brigham Young University
  - Computer Science
  - Graduating 2025
Goals

Our mission was to find a scalable filesystem for our HPC environment.

We decided to create a small, proof of concept cluster on some kernel-based virtual machines.

Our goals were as follows:

- Investigate Lustre
- Create a miniature lustre cluster utilizing three kvms
- Investigate and install a backend filesystem for lustre (zfs vs ldiskfs).
- Run some benchmark tests
Lustre

- Why use Lustre in HPC?
  Lustre has...
  - Exascale Capacities
    - Lustre uses distributed, object-based storage managed by servers.
      - Very large files can be distributed amongst different data objects and amongst several servers.
  - Data Integrity (only ZFS)
    - Data is written frequently in the event of a node crashing
  - Massive Scalability
    - Object Storage is low cost and efficient
    - I/O throughput and capacity are easily scaled by dynamically adding servers
  - POSIX-compliant on a Linux-based OS
The Lustre Architecture

- **Components of a Cluster**
  - MGS: Management Server, stores configuration information.
  - MDS: Metadata Server, manages MDTs.
  - MDT: Metadata target that stores file information/location.
  - OSS: Object Storage Server, manages OSTs.
  - OST: Object Storage Target, storage device, hosts files.
  - Client(s): Access and use the data.
Why ZFS over LDISKFS?

- Larger Capacities
- Integrated Data Integrity

<table>
<thead>
<tr>
<th>Feature</th>
<th>LDISKFS</th>
<th>ZFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Volume Size</td>
<td>32PB</td>
<td>512PB</td>
</tr>
<tr>
<td>Maximum Lustre File Size</td>
<td>512PB</td>
<td>8EB</td>
</tr>
<tr>
<td>Native Data Protection</td>
<td>None</td>
<td>Mirror, RAIDZ{1,2,3}, DRAID (Future)</td>
</tr>
<tr>
<td>Detect/ repair silent data corruption</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>File system repair</td>
<td>Offline: FSCK</td>
<td>Online: ZFS Scrub</td>
</tr>
</tbody>
</table>
In the end, our setup consisted of:

- 3 Centos 7 Kernel-based virtual machines
- A single MGS/MDS/MDT and OSS/OST setup with one client
- A small Lustre cluster built from source utilizing a ZFS backend filesystem

What was not accomplished:

- Benchmark tests
  - Lustre utilizing local storage
  - Lustre utilizing JBODs
Challenges

- Outdated resources
  - Tutorials written for an older RHEL version
  - More niche technology with few public discussions
  - Poor documentation (Except Lustre manual)
    - Documentation used old methods and syntax

- Installing Lustre
  - Lustre packages hard to find/ Don’t support ZFS
  - Setting up the development environment
    - LNET
    - Mounting Lustre Devices
Possible Future Goals

- **High Availability**
  - Utilize pacemaker to maintain high availability in the event of server failure

- **Explore Hierarchical Storage Management (HSM)**
  - Integrate Cheap Long-term Storage Solutions
  - Automatically move old data to/from a cheaper/slower storage medium
  - Invisible to end client

- **Explore Clustered Trivial Database (CTDB)**
  - Interface for the Server Message Block protocol (Windows protocol)
  - Extend filesystem support for different clients and operating systems
Sources

- https://wiki.lustre.org/
- https://wiki.whamcloud.com/
- https://en.wikipedia.org/wiki/Lustre_(file_system)
- https://www.netapp.com/data-storage/storagegrid/what-is-object-storage/