What a Lustre Cluster

(Improving and Tracing Lustre Metadata)

Team Saffron

Amanda Bonnie
Zach Fuerst
Thomas Stitt
Overview

- Motivation
- Configuration
- Tracing Metadata
- Improving Metadata Hardware
- Multiple Lustre Clients via Virtualization
- Conclusions & Future Work
Motivation

- **Tracing Metadata Motivation**
  - Can we get enough information without too much overhead?

- **Improving Metadata Hardware Motivation**
  - MDS can be a performance bottleneck
  - Faster MDT ☞ better performance?

- **Lustre Client Virtualization Motivation**
  - Single Lustre Client/Node underutilized IB device
  - Higher throughput ☞ Less transfer agents needed
  - Multi-VM nodes ☞ better throughput?
Lustre Configuration

- TAMIRS
  - MASTER (sa-master)
  - 4 X OSS (sa02-sa05)
    - Single disk RAID0
  - 1 X MGS/MDS (sa01)
    - hdd, nvme, KOVE
  - 5 X CLIENTS (sa06-sa10)

- PROBE
  - MASTER (n01)
  - 5 X OSS (n02-n05,n11)
    - 8 disk RAID0
  - 1 X MGS/MDS (n06)
  - 2 X CLIENTS (n07-n08)
  - 2 X VM CLIENTS (n09-n10)
MDS Tracing
Tracing Metadata

- Test tool: mdtest
- Tracers
  - Lustre Debug
  - debugfs (ftrace)
- Mask
  - ftrace - create, open, link, unlink, readdir, getattr, setattr
  - Lustre Debug - no mask
Tracing Metadata - Results

mdtest results with various tracing routines

- file.create
- file.stat
- file.read
- file.removal
- tree.creation
- tree.removal

Operations/second

- Quite an overhead
- Not too bad
- Ideal

tracer:
- both
- ftrace
- Lustre Debug
- none

Ideal not too bad
quite an overhead
MDS Hardware
Improving Metadata

- **HDD**
  - meh. (96.7 MB/s write & 206 MB/s read)

- **NVMe**
  - Fast! (686MB/s write & 1.3GB/s read)

- **KOVE Express Disk (XPD)**
  - RAM Storage Appliance
  - FAAAST! (2.8GB/s write & 3.5GB/s read)
Improving Metadata Hardware - Testing

- **mdtest**
  - Concerned with node caching (dropped caches!)
  - Performance still “low”

- **MDS-Survey**
  - Runs on MGS/MDS
  - Independent of CLIENT and OSS nodes.
## Improving Metadata Hardware - Results

<table>
<thead>
<tr>
<th></th>
<th>hdd to nvme (%)</th>
<th>hdd to kove (%)</th>
<th>nvme to kove (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>19.57</td>
<td>20.12</td>
<td>0.46</td>
</tr>
<tr>
<td>lookup</td>
<td>-1.67</td>
<td>0.99</td>
<td>2.70</td>
</tr>
<tr>
<td>md_getattr</td>
<td>-0.12</td>
<td>4.72</td>
<td>4.85</td>
</tr>
<tr>
<td>setxattr</td>
<td>287.45</td>
<td>244.46</td>
<td>-11.09</td>
</tr>
<tr>
<td>destroy</td>
<td>43.45</td>
<td>46.83</td>
<td>2.36</td>
</tr>
</tbody>
</table>

PERCENT INCREASE FROM NVME TO HDD, KOVE TO HDD, & KOVE TO NVME
Lustre Client Virtualization
SR-IOV

Standard I/O

VM-1  VM-2

Hypervisor

NIC

Pass-through

VM-1  VM-2

Hypervisor

NIC

SR-IOV

VM-1  VM-2

Hypervisor

VF-1  VF-2

NIC
Multiple Lustre Clients via Virtualization

- Enable SR-IOV
- KVM hypervisor with Centos 6.6 VMs on top
- Attach \( n \) Virtual Functions (VF) to the Physical Function (the device)
  - Virtual Functions just interfaces
  - \( n \in [1-11] \)
Testing Client Performance

- IOR
- Trinity Test from NERSC
  - POSIX Only
- N to N writes/reads
  - 44.7 GiB File per Client
- 10K, 100K, 1MB transfer sizes
IOR Write Results

IOR Write Results for 1-11 VMs on a Single Host

Write Speed (MB/s)

Single Host VM Count

(dashed lines are native installs)
IOR Read Results

IOR Read Results for 1-11 VMs on a Single Host

Read Speed (MB/s)

Single Host VM Count

(dashed lines are native installs)
VM Problems

- Hardware Restrictions
  - More than 2GB Ram Needed
  - Only 12 physical Cores
- IB Subnet Manager Needed on Host
- VMware’s ESXi Hypervisor
  - Mellanox drivers for ESXi didn’t support SR-IOV, only pass-through
  - Not Free
Conclusions

- **MDS Tracing**
  - Large Overhead or Not Extensive
- **MDS Hardware**
  - Improvements << Cost
- **Virtualization of Clients**
  - Scalable!
  - Worth Further Exploration
Future Work

● More Virtualization!
  ○ Put VMs in a VM so we can virtualize our virtualization allowing us to virtualize while we virtualize (and manage SR-IOV better)
    ■ Changing the number of VFs requires a reboot which is slow
  ○ Greater number of VMs (>11)

● Local subnet on each host

● SR-IOV with verbs on ESXi
Future Work

● More Virtualization!
  ○ Put VMs in a VM so we can virtualize our virtualization allowing us to virtualize while we virtualize (and manage SR-IOV better)
    ■ Changing the number of VFs requires a reboot which is slow
  ○ Greater number of VMs (>11)

● Local subnet on each host
● SR-IOV with verbs on ESXi
Acknowledgements

Mentors: Brad Settlemyer, Christopher Mitchell, Michael Mason

Instructors: Matthew Broomfield, Jarrett Crews

Administration: Carolyn Connor, Andree Jacobson, Gary Grider, Josephine Olivas
Questions?

such fast
many vm
many GBs
such bandwidth
quite speedy

lots storage
wow verbs wow

wow

much Lustres
hi lester

so questions

such professional

such verbs