Lustre Scalability Workshop
Requirements Gap Response
October 2009

Introduction

The Lustre Center of Excellence (LCE) at Oak Ridge National Laboratory hosted a workshop on scalability of the Lustre file system on February 10 and 11 of 2009 and a follow up workshop in May. One of the purposes of the workshops was to identify key requirements for supporting the next generation of large scale parallel systems with the Lustre file system. As part of the February workshop the attendees identified ten gaps between their anticipated requirements for high performance storage and IO and the Lustre roadmap presented by the Lustre team. The Lustre team agreed to review and respond to these gaps. This document is that response.

The gaps are listed in order of priority assigned by during the workshop from highest to lowest.

Detailed information on the Lustre Roadmap or on specific features can be found at: http://wiki.lustre.org.

Top Ten Lustre Gaps and Responses

1. Asymmetric impact of failures

Gap Description - Hardware or software failures on a subset of file system resources should only impact those resources residing on the failed/failing equipment. For instance, a failed OSS node should only preclude access to files controlled by that OSS – it shouldn't snowball into MDS hangs because of exhausted threads or router backups. A problem on one file system should never impact another file system in any way.

Gap Response - Today, Lustre failures are detected through communication timeouts between clients and servers. This can result in failures remaining

undetected and uncorrected for long periods. A separate virtual health network and an imperative recovery capability will be designed and implemented to enable the rapid detection and eviction of failed clients and initiate the restart of failed servers. This network will be able to integrate with external RAS systems.

2. MDS Performance

Gap Description - The attendees desired known milestones for order of magnitude incremental MDS performance improvements on servers of sufficient configuration over next two years. CMD performance will become a significant/gating need after two years. The performance of Metadata operations is a current and growing issue. While Clustered Metadata Servers (CMD) are going to address this, there is a need for improved MDS performance before CMD is available. The workshop participants would like to know what interim performance improvements to expect and when.

Gap Response - The long term solution for MDS performance and scalability is CMD which will enable the scaling of metadata performance by adding multiple MDSs to a single file system. In the short term, SMP scalability enhancements are being developed and tested which will substantially improve the performance of a single MDS. These enhancements include locking on the server, replacing highly contended locks in LNET and Lustre with multiple locks, along with creating per-CPU data structures to avoid unnecessary contention and enabling greater parallelism and scalability. In addition, CPU affinity, keeping the processing for particular peers on particular CPU cores has resulted in substantial performance improvements.

3. Lustre ZFS Licensing

Gap Description - There was concern over the license used with ZFS (CDDL). While CDDL is an Open Source license people were concerned that it is not compatible with GPL and that this may impact ZFS in Lustre the following ways:

- Community involvement - The Lustre community may be unwilling to invest in enhancing ZFS if it not licensed with GPL.
- Protect investment - Concern that the CDDL does not sufficiently protect any investments made in ZFS. Sun could drop ZFS and the user investment in the code could be lost or encumbered.
- Contaminating – Concern that programmers working on the ZFS code will be exposed to Sun owned Intellectual Property and then not be free to work on other file systems.
- Link level – compatibility of licenses – Will the kernel implementation of the ZFS DMU be compatible with both letter and spirit of the GPL and the Linux kernel's license checks?

Gap Response - The response to this issue is still being developed.

4. Quality of service

Gap Description - The ability to assign quality of service levels to individual clients

is required to handle:
• Individual jobs competing for bandwidth on a single system (direct attached)
• Example: an aggressive reader to a single OST can slow a competing simulation job during a checkpoint.
• Individual machines competing for bandwidth in a center wide configuration
• Example: A visualization cluster may consume a disproportionate amount of bandwidth in a center-wide file system reducing the I/O performance of other simulation platforms.
• Metadata processing rates to provide a more real-time responsive system.
• Example: Users on login nodes receive poor file system responsiveness when other metadata "hungry" applications are running
Gap Response - Network Request Scheduling (NRS) will provide a basis for Quality of Service. NRS re-orders request execution to present a workload to the backing file system that can be optimized more easily while avoiding request starvation. Extending NRS to implement Quality of Service (QoS), allowing specific clients or clusters to force priority request handling, is a natural extension of the current NRS prototype.

5. Performance Variability

Gap Description - Users should see consistent performance across a continuum of workloads assuming the individual I/O operations are reasonably large. For example, users should achieve a large fraction of the ideal I/O rates for large block I/O regardless of alignment, exact size, and number of clients.
Gap Response – Sun's plan is to have a Lustre IO Performance engineer examine Lustre layer by layer, determine where performance anomalies exist, and work with Lustre engineering to fix such inconsistencies.

6. Policy engine

Gap Description - A Policy Engine is needed to set allocation, migration, tier classes, locality to client OST performance stats: load avg, etc. Such a policy engine would use the following information to help implement such policies:
OST fill information: percent full, file count, etc
OST group/pool info Uid/gid info File size/type/mime-info
Gap Response - There is a policy engine in HSM that monitors filesystem disk usage and file access time and migrates files to an archive based on this information and admin policy. This could be extended to manage functions beyond file archival and retrieval.

7. Manageable at scale:

Gap Description - This is the ability to quickly and precisely identify failures and potential failures. Examples are:
Offering more information in proc Improved syslog information RAS interface that will give 3rd parties something to develop tools against? Other mechanisms of aggregating this information in Lustre Manage normal operations

Gap Response - This response is still in progress.

8. Failover duration

Gap Description - Failover is not widely used today because of difficulty in configuring it and because the time required for the system to detect a failure and complete the failover can be longer than the time required to reboot the system. A mechanism is needed to significantly speed up the process.
Gap Response - The health network and imperative recovery described in item 1 will address this issue.

9. Small file performance

Gap Description - Small file performance and efficiency. Improve the efficiency and performance of small file management by aggregating files on OST's and placing small files on MDS.
Gap Response - The key to optimizing small file performance is aggregating multiple requests into a single RPC. The Write Back Cache (WBC) feature will enable metadata operations to complete on the client before they are flushed to the server and to be sent to the server in bulk. The Size on Metadata (SOM) feature will at least double the stat performance for small files because accessing files will not require and additional RPC's to the OSTs to get the file size. We are also considering keeping small files on the MDS.

10.Wide stripe performance

Gap Description - Users would prefer to not have to worry about setting striping for their files. A system wide default should achieve a large fraction of the best case I/O without requiring the user to manually set the stripe count. One likely consequence of this is a small file (on the order of the stripe size) that is widely striped should perform as well as narrowly striped file.
Gap Response - This area is still under investigation. This can potentially be addressed through extending IO libraries or middleware. ORNL are investigating middleware approaches to this. Another possible approach is to develop a mechanism to identify (either by user action or automated in some way) files that should be widely-striped and make the single-vs-wide-striping decision at first open. Also, Lustre's striping limits are being increased. The maximum number of stripes has been increased from 168 to 512 and the maximum stripe size has been increased to 4GB.