Cray's Lustre Support Model and Roadmap

Cory Spitz, *Cray Inc.*

ABSTRACT: *Cray continues to deploy and support Lustre as the file system of choice for all of our systems. As such, Cray is committed to developing Lustre and ensuring its continued success on our platforms. This paper will discuss Cray's Lustre deployment model, and how it ensures both a stable Lustre version and enables productivity. It will also outline how we work with the Lustre community through OpenSFS. Finally, it will roll out our updated Lustre roadmap, which includes Lustre 2.2 and Linux 3.0.*

KEYWORDS: Lustre, OpenSFS, esFS, Direct-Attached Lustre, Sonexion, CLE, roadmap, file systems

1. Introduction

The Lustre file system is a key component of Cray systems. Cray has historically provided value to its Lustre users by performing many duties include development, integration, performance scaling, stringent testing, and support. To accommodate all of this work, the release of Lustre software as a part of larger, integrated Cray release always lags the general availability of Lustre software.

This paper will discuss Cray's Lustre deployment model, and how it ensures both a stable Lustre version and enables productivity. It will also outline how we work with the Lustre community through OpenSFS. Finally, it will roll out our updated Lustre roadmap, which includes Lustre 2.2 and Linux 3.0.

This paper assumes that the reader is familiar with Cray's existing software release models, specifically for CLE. In short, Cray uses a "train" model with quarterly updates. The updates can contain new features, but the base kernel version (not necessarily the OS service pack level) remains constant. More information on Cray CLE software releases and roadmap can be found in Cray Operating System Roadmap from the CUG proceedings [1]. The paper also assumes that the reader is somewhat familiar with the Lustre community and OpenSFS. More information about OpenSFS is available at http://www.opensfs.org.

2. Cray's Lustre Model

Cray's Lustre model is designed to ensure stability and productivity. It now starts with an investment 'upstream' in OpenSFS with $500K dues (paid annually). This money is intended mainly to fund the Lustre features that OpenSFS members feel are important for HPC. Cray was a founder of OpenSFS because we felt that it was important to nurture Lustre development in an open format. OpenSFS ensures that funded features land to the Whamcloud/OpenSFS canonical tree[1]. Oracle no longer maintains the canonical tree.

Cray then acquires Lustre from the new canonical source. Our intent is to regularly rebase from that source to ensure that we incorporate the features that we and our customers desire. However, we then stabilize that Lustre version on our hardware and greater software offerings before releasing our version of Lustre to the community.

We do not consider our version a fork, but rather a version "plus patches". We ensure that all the changes that we commit to our tree are pushed and landed upstream if they have not been already.

Lustre at Cray has traditionally been a part of CLE, and that tradition continues. Even though we have other Lustre software products coming such as esFS and Sonexion (covered later), we continue to release Lustre in a monolithic style. Each hardware platform is tested with

[1] Whamcloud has been awarded a "tree maintenance" contract by OpenSFS to offset the costs of performing Lustre releases

a specific version of kernel and Lustre and this is released and supported as a set.

These products (save Sonexion) use a common source base, which is beneficial because if we, say, fix a client bug for CLE we can then easily leverage that fix for the esLogin Lustre client.

Support

Cray has signed a Lustre Level III support contract with Xyratex and we are no longer using Oracle support. The contract with Xyratex includes all of our products including legacy versions of Lustre that were included in CLE versions that are now at their end-of-life.

Cray and Xyratex work privately on bugs, so this is a difference from the past. That is, bugs reported to Xyratex are not public. However, once an issue is confirmed, then Xyratex will open a public Jira ticket with Whamcloud (http://jira.whamcloud.com) and push any proposed fix through for review through their open Gerrit (http://review.whamcloud.com). We do not consider bugs 'closed' until all patches for a ticket are accepted and landed to the upstream 'master'[2].

3. Cray Participation in OpenSFS

Cray is an original founder and promoter of OpenSFS. We are given a seat on the board (and a vote) along with our promoter level membership. Dave Wallace holds our seat on the board.

After founding OpenSFS, Cray continues to take a leadership role on Lustre within the organization, primarily with the Technical Working Group (TWG). John Carrier co-chairs the TWG. Cory Spitz is also a participant. This group engages the community to define requirements for functionality and performance and then craft RFPs to commission work to address the most popular or needed enhancements based on the collected requirements. The TWG has a whitepaper discussing their role at http://ww.opensfs.org. Both Cory and John are members of the TWG RFP sub-team that craft the RFPs and work with the board for approval. Johan and Cory are also members of the TWG Project Approval Committee (PAC) that oversees and approves deliverables and progress on the funded projects.

Cory is also a member of the Community Development Working Group (CDWG). That group provides a forum for community development. That mainly works on co-ordinating other development and testing needed that is not covered by the main TWG award contracts[3]. John participates with the Benchmarking Working Group (BWG) where we have promoted our HPCS I/O Scenarios as a basis for an open Lustre benchmark [2].

Please join OpenSFS and make your requirements heard. The TWG is currently collecting input for the next round of funding. We hope to gather responses and make recommendations to the board by 5/24, decide the projects that will be funded and issue an RFP @ ISC. Then, we are targeting creating a SOW before SC '12.

The projects from the last round of funding are landing. Parallel DirOps landed in 2.2 and SMP affinity is landing in 2.3. See the community roadmap on the Whamcloud wiki at to see more[4].

Your participation is also needed because OpenSFS needs the funds to fully fund Lustre community releases. The current tree maintenance contract only covers about half of the costs.

4. Cray development in the last year

Cray has shipped Lustre with four CLE releases in the last year. CLE 4.0 UP00, UP01, and UP02 all included Lustre 1.8.4. We actually did not deliver on our previous roadmap. Last year, we announced that we would include Lustre version 1.8.6 in 4.0 UP01 and UP02. However, we held that version because it did not meet our standards for quality. After addressing a few quality issues, we were prepared to release and Lustre 1.8.6 is included in CLE 4.0 UP03. However, by the time we made it to UP03, we have rebased from Whamcloud, rather than Oracle. So the Lustre in CLE 4.0 UP03 is actually based on 1.8.6-wc1.

We are currently working on bringing Linux 3.0 support to Lustre. We require this because we need SLES 11 SP2 for Intel processor support for our Cascade line and it is based on Linux 3.0. We are working with the community on Linux 3.0 support and we are pushing changes to LU-812.

We have also quickly moved to the external Lustre model with external servers and LNET routers in the mainframe. To reduce contention in the both the high speed network (HSN) and the external IB fabric, we've been working on Fine Grained Routing (FGR) for LNET based upon the work that ORNL did to avoid I/O congestion [3]. We have recently reported some of our progress in identifying scaling problems with LNET routing and FGR at LUG '12 and solutions for Cray systems at CUG '12 [4].

Finally, last year Cray announced that we would productize the external services offerings, esLogin and esFS. Thusly, we have extended lustre_control to support

[2] "master" is the head of the line of development for Lustre

[3] For example, community development is announced and tracked at http://wiki.whamcloud.com/display/PUB/Lustre+Community+Development+in+Progress

[4] http://wiki.whamcloud.com/display/PUB/Community+Lustre+Roadmap

esFS and esLogin nodes. While we were at it, we added numerous new features to lustre_control, which is covered later.

5. Lustre for External Services

Cray is standardizing the Lustre software on esFS and esLogin and forming a product around that software. ESF is the codename for the new esFS SW release. Correspondingly, ESL is the codename for the new esLogin SW release. ESF and ESL is a moniker for a bill of materials that includes the base OS, Lustre, OFED, HCA drivers, etc.

The Lustre is ESL & ESF use the same Lustre stack as CLE so that there is one common source tree for all three SW products. ESL is tied to specific CLE release, which is the same SLES release as CLE. ESF is not tied to CLE at all. In fact, it is based on CentOS. However, it will still be paired and tested with specific CLE releases

The support around Lustre for external services includes esFSmon for esFS failover. However, this feature requires an esMS Management Server. As stated previously lustre_control was extended to external service for command and control. It is the same lustre_control as CLE for familiarity and a common code base. It also requires an esMS.

The ESL and ESF roadmap is as follows. ESL and ESF for Koshi UP01 will GA in December '12 w/CLE Koshi UP01. It will include Lustre 2.2. This is the first GA Cray Lustre release to include Lustre 2.x. Then ESL & ESF for Nile UP02 will GA in March '13 w/CLE Nile UP02. It will also include Lustre 2.2. ESL & ESF releases are supported for 18 months, just as CLE is.

Cray will provide a migration plan for upgrades. This is important for two reasons. First, the on-disk format changes in Lustre 2.x with the new FID format (and then again with large xattr support for wide striping in version 2.2). Second, Lustre 2.x clients are not compatible with 1.8.x servers, so care must be made when planning deployments. That is, ESF w/Lustre 2.2 must be installed before Lustre 2.x CLE clients.

Even though Lustre 2.x clients are not backward compatible with Lustre 1.8.x servers, the opposite is possible. That is, Lustre 2.x servers are backwards compatible with Lustre 1.8.x clients. This means that it will be possible to deploy external Lustre 2.x based file systems and mount them with 1.8.x clients from a Cray mainframe. That is fortunate, because it means that clients and servers do not need to be upgraded at the same time.

6. Planning for Direct Attached Lustre EOL

So-called Direct Attached Lustre where Lustre servers are housed on Cray I/O nodes is the traditional Cray Lustre offering with Lustre servers on mainframe I/O service nodes. As announced last year, Direct Attached Lustre is planned for end-of-life.

Here is the release roadmap. CLE 4.0 UP03 w/Lustre 1.8.6 is available now and Cray will include patch support through mid-2013. CLE Koshi UP01 w/Lustre 1.8.x will GA in December. The Lustre version is TBD, likely 1.8.7-wc1 or 1.8.8-wc1 based. It will include patch support through mid-2014.

Koshi is the last line to support both Lustre 1.8.x and Direct Attached Lustre. This means that Lustre 2.x will not be available for Direct Attached Lustre.

7. lustre-utils enhancements

As stated earlier, there are lustre_control enhancements beyond basic esFS and esLogin support. Those enhancements include:

- Improved file system definition and configuration
- Operation on multiple file systems with a single command
- Automatic updates to the SDB if using Direct Attached Lustre w/failover
- Control of mount/umount of service node clients and compute nodes
- Failover and failback control including interface with esFSmon for esFS
- Lustre server status reporting
- Parallel Lustre target consistency checking with fsck
- Configuration verification (verifies correct target for correct device)
- Lustre tunables emplacement via lctl set_param

To assist routed LNET deployments and support with FGR, we have added a new tool to lustre-utils called clcvt, the Cray LNET configuration and validation tool. It will initially support Sonexion LNET FGR only. Its main purpose is to automatically generate 'ip2nets' and 'routes' for LNET configuration. However, it will also generate a cable map to aid install and performs live validation of the cabling and LNET configuration.

8. Conclusion

Lustre file systems are a very important resource for Cray systems and Cray's customers. In order to provide the best Lustre experience Cray executes substantial testing and software stabilization as a part of distributing CLE, and now ESL & ESF. These efforts will continue for both 1.8.x and 2.x versions going forward.

Cray is very involved with OpenSFS to ensure its success, the success of Lustre 2.x, and future Lustre feature development.

We follow a roadmap for our Lustre offerings that is summarized here:

- **Koshi UP01 GA December '12**
  - Includes new lustre_control and clcvt
  - CentOS 6.2 for ESF
  - SLES 11 SP1 for ESL and CLE
  - Lustre 1.8.x for Direct Attached Lustre in CLE
  - Lustre 2.2 client for ESL & CLE
  - Lustre 2.2 server for ESF
  - Patch support ends 18 months after GA – mid-2014
- **Nile UP02 GA March '13**
  - CentOS 6.2 for ESF
  - SLES 11 SP2 for ESL and CLE
  - Lustre 2.2 CLE client for ESL & CLE
  - Lustre 2.2 server for ESF
  - Patch support ends 18 months after GA – late-2014

Acknowledgements

I would like to thank OpenSFS founders for having the vision to create a truly open development ecosystem for Lustre. The success of OpenSFS will have a direct impact upon Cray's future Lustre offerings and in turn the Cray system experience.

I would also like to thank John Carrier and Dave Wallace for helping to define Cray's Lustre requirement and thus a roadmap.

About the author

Cory Spitz is the team lead for Lustre integration and manager of I/O Filesystems in Cray's OSIO division. He can be reached via email at spitzcor@cray.com and works at 380 Jackson Street, St. Paul, MN, 55101.

References

[1] C. Carroll, "Cray Operating Systems Road map," Proc. Cray User Group, 2012

[2] HPCS I/O Scenarios, John Carrier, Cray, Inc. http://www.opensfs.org/wp-content/uploads/2011/11/Carrier_LUG12_HPCS-Scenarios.pdf

[3] D. Dillow, G. Shipman, S. Oral, Z. Zhang, "I/O congestion avoidance via routing and object placement," Proc. Cray User Group, 2011

[4] C. Spitz, N. Henke, D. Petesch, J. Glenski, "Minimizing Lustre Ping Effects at Scale on Cray Systems," Proc. Cray User Group, 2012