Sun Lustre Centre of Excellence
at Oak Ridge National Laboratory

LCE Summit Meeting Report

Burlington, Massachusetts
February 7-8, 2008

Edited by Dan Ferber and Sheila Barthel, Sun Microsystems

Abstract

This two-day event gathered key customers and partners to discuss strategic Lustre topics in a five-year planning window. Upcoming priorities for the Lustre product and user community were presented and ranked. The highest-ranked topics were discussed in detail. These discussions yielded strategic and tactical input for the Lustre Group at Sun, and the Lustre community in general.

LCE Summit – Objective and Goals

The objective of the Lustre Centre of Excellence (LCE) summit was to create a far-reaching, strategic vision to bring Lustre to the next level for high-end HPC customers[1]. The principal goal was to get feedback from the 10 biggest Lustre sites worldwide, and from strategic partners on Lustre's development and evolution in the next five years. A secondary goal was to promote the Lustre community – to get together, share challenges and ideas, and begin working together as an integrated community. This emphasis on the community nature of Lustre and making it more open, is a main focus since Sun's acquisition of CFS (and Lustre) in October 2007. The event was sponsored by the Lustre Centre of Excellence at Oak Ridge National Laboratory.

Strategic input, user experiences, and community discussion are critically important to help guide Lustre development and to support what we see as emerging hundred petaflop systems with thousands of storage servers managing an exabyte of data. With that motivation, the expected outcomes of the LCE summit were:

- Engage the Lustre HPC community in visionary discussions about the future of technology sites, to communicate where they are going and what they need from Lustre.
- Establish community priorities based on a ranked list of top issues and their underlying context, for Lustre in HPC within the next five years.
- Advise the user community of Sun's current plan and strategy for Lustre in the next five years.
- Subsequent to the meeting, develop a strategic plan to address the community priorities.

Dr. Peter Braam, the founder of CFS, and now a Sun VP and Lustre Chief Architect, opened the LCE summit. Peter said that on October 1, 2007, the Sun acquisition of CFS was completed. Sun and CFS agreed that the overriding strategy for Lustre was its continuity. In keeping with that strategy, Lustre remains open source under GPL[2], with all design documents and Lustre internals course materials available on the lustre.org community wiki. Lustre CVS (source control) is open, and architecture discussions are also open via the lustre-devel email community alias. Sun continues to work with all of the Lustre partners from CFS, including DDN, HP, Bull, Cray, and others. There are no special versions of Lustre for any customer or partner, including Sun. Additionally, Peter said that all Lustre customers are on-board, as well as former CFS employees. In fact, what we saw at the LCE summit was that the greatest change to Lustre this past year has been wider adoption and use in production environments.

[1] See Appendix A for the complete meeting agenda, attendee list, presentation slide decks, and detailed notes.

[2] See Sun's open letter to the community, at http://www.sun.com/software/clusterfs/lustre_community_ltr.pdf.

Peter went on to describe several key management changes in the Lustre Group.

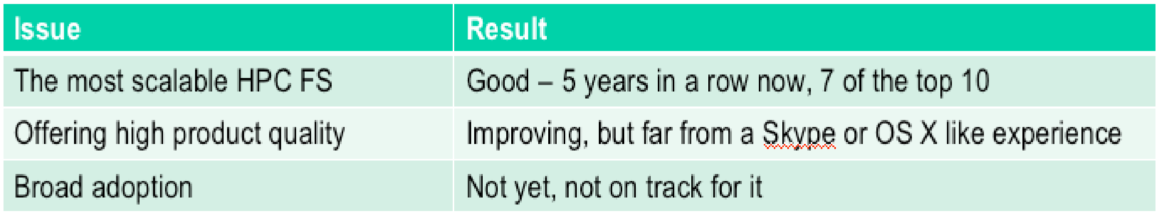
- Peter Braam is now the Lustre Chief Architect, responsible for customer requirements and product vision.
- Peter Bojanic now manages the Lustre Group and is responsible for support and project planning. Peter reports to John Fowler.
- Eric Barton is now the Lustre CTO and implementation lead.
- Lustre Marketing and Sales have moved to other groups within Sun, in order to better leverage Sun's infrastructure, and keep the Lustre Group focused on product architecture engineering.

Peter indicated that 7 of the top 10 HPC sites are running Lustre, as are 50% of the top 30, and 30% of the top 100. At least three DOE sites are running multi-cluster Lustre (LLNL, ORNL, Sandia). Additionally, HPC use of Lustre continues to grow in various industry sectors, such as Oil and Gas, EDA, Manufacturing, Media, ISPs, Business Strategy, and DOD. Peter commented that the short-term vision for Lustre is product quality, with HPC scalability and WAN features to follow.

Sun's Lustre Roadmap and Thoughts on the Future of Lustre

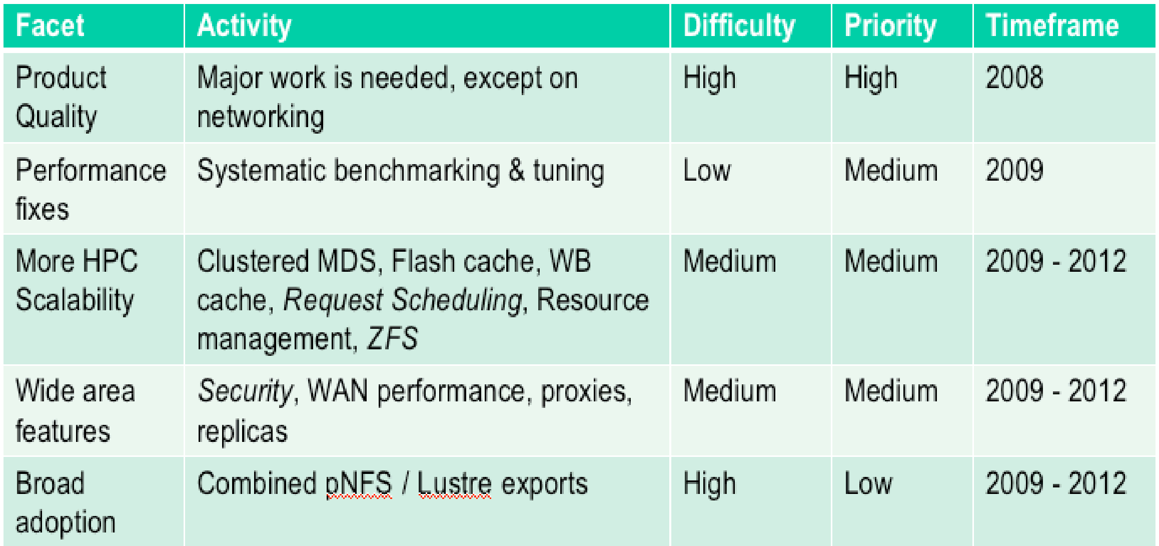
After this introduction, Peter Braam provided his perspective on strategic areas and features that, from a conceptual view, he believes are important for Lustre's future development. These features are not to be confused with the attendees' ranked priorities, discussions, and outcomes described later in this paper.

Here is Peter's view of Lustre's current macro-level status.



Issue	Result
The most scalable HPC FS	Good – 5 years in a row now, 7 of the top 10
Offering high product quality	Improving, but far from a Skype or OS X like experience
Broad adoption	Not yet, not on track for it

While Peter was careful to say that these features, and the specifics that follow, are visionary in nature and not firm commitments, the diagram below illustrates his perspective of the areas in which Lustre requires future development, and a general time frame to implement these features.



Facet	Activity	Difficulty	Priority	Timeframe
Product Quality	Major work is needed, except on networking	High	High	2008
Performance fixes	Systematic benchmarking & tuning	Low	Medium	2009
More HPC Scalability	Clustered MDS, Flash cache, WB cache, *Request Scheduling*, Resource management, *ZFS*	Medium	Medium	2009 - 2012
Wide area features	*Security*, WAN performance, proxies, replicas	Medium	Medium	2009 - 2012
Broad adoption	Combined pNFS / Lustre exports	High	Low	2009 - 2012

The key elements of the Lustre product vision are described below.

ZFS DMU Backend

In the current Lustre server implementation, servers are in a patched Linux kernel and a modified ext4 (ldiskfs) is required. However, Lustre customers require:

- Portability
- No kernel patches
- Platform independent API
- Scalability
- Hardening

To work all of these requirements into ext4 would take an estimated 24-person years. CFS explored user space servers layered directly on top of the ext4 filesystem, but MDS was not possible. CFS also considered hardening and scaling ext4. Although possible, the development effort to do this was too involved, and the ext4 community is moving too slow. Therefore, ZFS DMU was chosen as an alternative[3]. The benefits of this choice are that ZFS (1) can be in user space, (2) it is scalable, hardened, and portable, (3) it is estimated to be 3x less work to implement, and the Sun ZFS team will help.
The downside of choosing ZFS is that the performance work the Lustre Group did for ext4 will have to be re-applied to ZFS (although the team knows how to do this).

ZFS DMU also has data integrity features that Lustre can leverage. Lustre will add checksumming of all file data passing over the network. When this feature was tested, Lustre was able to detect data corrupting network cards.

[3] See http://arch.lustre.org/index.php?title=Architecture_ZFS_for_Lustre.

Network Request Scheduler

Today, Lustre file servers process a request queue as FIFO. A better approach is to have the servers re-order requests in a manner similar to disk elevator to optimize request ordering, and ensure fairness or priority among clients. It will also re-order I/O to make it sequential in the diskfs, and pre-fetch metadata to avoid blocking. Lustre will use this scheduler in conjunction with OST and MDT write caches. A second generation of the network request scheduler would add server coordination.

Flash Cache

In general, there was quite a bit of discussion about flash cache at the LCE summit.. Peter Braam said that the Lustre Group plans to take full advantage of the storage hardware revolution. There is very high bandwidth available from flash, and adding flash cache OSTs to total RAM capacity in the cluster is appealing. The cost is a small fraction of the cost of RAM in the cluster. The flash device will not be Sun-specific.

This allows very fast I/O from compute node memory to flash storage, and then the flash data can be drained to disk storage independently. For example if disk is ~5x slower than flash, and if the cluster finishes I/O in 10 minutes and it is on disk in 50 minutes, then we will need 5x fewer disks. Lustre will manage a coherent view of the file system. There was some discussion of MDS performance. Currently, it is the case that the MDS is CPU-bound and gets little benefit from RAM or flash cache. The benefit is to stage I/O and let allocation be done logically afterwards.

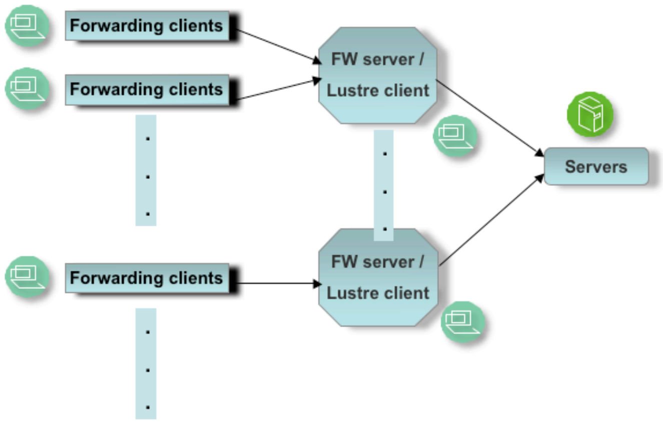
Metadata Write-Back Cache

Disk-based file systems make updates in memory, but network file systems do not as metadata operations require synchronous RPCs. Ideally, Lustre should only require synchronous RPCs for cache misses. Key elements of this design will be:

- Clients can determine file identifiers for new files
- Change log is maintained on the client
- Parallel re-integration takes the place of a log to clustered metadata servers
- Sub-tree locks – enlarge lock granularity

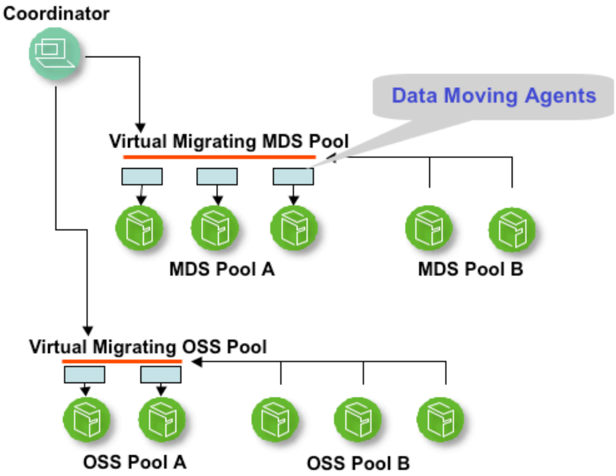
There will be many uses of this feature. In HPC, I/O forwarding makes Lustre clients act as I/O call servers, and these servers can run on write-back cache (WBC) clients. For exa-scale clusters, WBC enables last-minute resource allocation. In WAN-based Lustre, this eliminates latency from wide area use for updates, as it removes the synch delay to the MDS. For HPCS, this will dramatically increase small file performance.

Note that for the client/server model with a forwarding agent on cluster nodes, Lustre's focus is on the file system and not the forwarding agent.



Migration

There are many uses and requirements for migration. Some examples are between ext3/ZFS servers, for space re-balancing, to empty servers for replacing other servers, in conjunction with HSM, and to manage caches. The migration model that Peter has in mind is pool-to-pool, not disk-to-disk. A coordinator and agents would do the work. The coordinator is a server, but it can run anywhere that the client runs.



Caches and Proxies

Peter briefly discussed the possibilities for Lustre caches and proxies. This could be used in HSM environments where Lustre is a proxy cache for third-tier storage, collaborative read cache, or wide area cache where repeated “local” reads come from cache. Some of the technical elements that need to be addressed are migrating data between storage pools, re-validating cached data with versions, and the hierarchical management of consistency.

Broad Adoption Vision

For broader Lustre adoption, Peter talked about pNFS integration. Soon, users will be able to have pNFS exports from Lustre clients on Linux. Longer term, we could enable Lustre servers to offer pNFS and Lustre protocol. This would require an interesting Lustre storage layer, making LNET an RDMA transport for NFS. This functionality would offer proven Lustre features to NFS standards efforts.

HPC Community’s Top Lustre Priorities

Each Lustre customer and partner that attended the LCE summit came prepared to present and discuss their top three strategic priorities for Lustre. (See Appendix A, which contains the slides presented by each summit attendee regarding their strategic priorities.) Each attendee presented their strategic topics, and then answered questions and responded to comments about it.

After the Lustre priorities were introduced by the participating organizations, we consolidated all of the priority topics into a master list, with similar topics grouped together. Then, each Lustre customer and partner cast five votes for topics on the master list, to develop a rank order of priorities. The idea was that each customer/partner would likely vote for the three topics they presented, and select two more. This ranking was used to determine the order in which topics would be discussed, with the most popular topics taken up first. With limited time available, we did not expect to be able to discuss all topics on the master list.

The complete master list, in rank order, is in Appendix A. At the summit, we had enough time to discuss the following priority topics:

1. System and File System Administration (included usability)
2. Improved support for multi-clustered environments (included Quality of Service)
3. Data integrity
4. Evolve Lustre towards a more community-driven development model
5. Support for very/ultra-large clusters and WAN

The main objective of these topic discussions was to ensure that the Lustre Group and the Lustre user community fully understand the requirements and context of these priority items, including any use case examples.

Discussion summaries for each of these five topics are described below.

1. System and File System Administration (includes Usability)

This topic was ranked at the top of the priorities list because so may LCE attendees now depend on Lustre for production-oriented work on very large systems.

We talked about free space management and the activities that are important:

- Need to drain an OST
- Rebalancing of existing files

Within this topic, PNNL indicated it wants to be able to fill one OST at a time. Sandia said that running out of space can kill jobs at very inopportune times. The Lustre Engineering team discussed providing a low-level verb syntax, whereby users would script the policy to each site's policies. Instrumental mentioned that Harriet Coverston is already doing this.

The discussion then turned to the need for better diagnostics, and diagnostic topics in general. NERSC indicated that it would monitor network traffic to determine if an OST has gone silent. LMT was discussed, and we explored using it in a Lustre RPM. Several people commented that LMT is too specific to LLNL and, further, that LLNL may not have resources to support this at the current time. Initially, LLNL used LMT because nothing else was available.

CEA suggested that SNMP support should be extended, and that a single SNMP daemon to collect all information would be useful. Andreas stated that not many people ask for SNMP.

A big problem is the straggler. The Lustre community needs tools to find slow hardware, slow processes, etc. One suggestion was that graduate students or the lustre.org community could build off of hooks currently in Lustre.

Failover was a significant topic. LCE attendees agreed that failover must be bullet-proof. Failures can come from multiple places: controllers, line cards, switches, etc. One attendee's large GPFS file system has gone 180 days without a user-observable failure. They have 30,000 cores, 12,000 nodes, and can fail over within 15 seconds. The group discussed data integrity and clustered metadata consistency needs and cases. Most participants do not use failover, due to operational complexities. Several said that fast integrity scans are needed before backing up / restoring.

Other discussion threads in this topic included:

- NERSC uses CACTI. Other participants mentioned ganglia, collectl
- Multi-cluster management is an important topic for several attendees.
- NERSC does not always like Kerberos – too hard to implement. They would like a lighter-weight solution. They might need multiple plug-in modules and policies for different cases within one site. nfsv4 is using gssapi.
- NERSC does not use firewalls, but instead uses intrusion detection. Systems typically do not trust each other.
- OST pools were discussed, because they let users define groupings of OSTs and restrict clients to a specific pool, and could be used for migration.
- Lustre 1.6 has stripe allocation balancing, but this introduces its own set of problems: emptier OSTs fill up proportionally faster, resulting in congestion.
- A primary issue with tools is advertising. Often, tools are not known or easy to find.
- Configuration templates. The Lustre management server can dump configuration information into a CSV spreadsheet, and then the spreadsheet can be used to export the configuration to other servers. However, this method is neither parallel nor automated.

2. Improved Support for Multi-clustered Environments (including QoS)

Sun asked about rolling upgrade requirements; if two Lustre versions simultaneously are required. The resounding feedback was that rolling upgrades are required and a fact of life. Sites can no longer afford to do monolithic upgrades. For example, any 1.8 server should work with any 1.6 client. Sun noted that Engineering resources spent on compatibility are not working on new features, but attendees agreed that compatibility is a must-have aspect of Lustre.

Sun indicated that Lustre does not support more than two different versions at any one time. As an example, the latest version of Lustre 1.4 interoperates with the latest version of Lustre 1.6, but version 1.6 will not work with all versions of 1.4. Cray mentioned that Sun took on forward and backward compatibility for NFS. It took awhile to make it work, but the benefit now is that compatibility between NFS versions is not even questioned. Sun wanted to bound the cases so compatibility is on a known set of components.

Some attendees asked to have compatibility between major versions, i.e. all 1.8 versions work with all 1.6 versions, and if protocol changes appear incrementally, then Sun is handling its releases incorrectly.

We discussed the ZFS upgrade, as well as LNET IPv6 as an eventual change. For ZFS, maintaining current performance levels is key. IPv6 presents a wire format change. This will require major engineering work if Lustre must interoperate between IPv4 and IPv6. TCP can run over most interconnects, and this may be a fallback strategy. Can Lustre compatibility be specified in the procurements? Yes, probably so. Procurements cannot necessarily limit competition by specifying Lustre, especially now that Lustre is owned by Sun. This situation could get into competitive conflicts.

The participants agreed that compatibility issues are relentless and must be addressed.

3. Data Integrity

There was much discussion about end-to-end data integrity checking, and if corruption occurs, having the ability to know where it occurred. We discussed the need for and challenges in having the POSIX standard modified to support this. Sun indicated that Integrity checking must be done in Lustre, since it supports so many types of hardware. It cannot simply rely on exploiting an integrity feature in one vendor's hardware.

Although ZFS offers additional data integrity features, attendees said that current performance levels cannot be sacrificed for ZFS, even if there are significant ZFS feature benefits. An acceptable delta limit was within 10% of ldiskfs. Sun indicated that based on the Lustre Group's track record with ext3 and ext4, they are confident that they and the ZFS team will get ZFS performance to where it needs to be.

Related to performance and data integrity, we talked about integrity code needing to be scalable across cores, as core density is increasing. A good example of this are checksums.

The group discussion then circled back to failover. Some attendees stated that since we do not know what disk environment looks like, Server Network Striping (SNS) is a nice option. Addressing concerns about failover would help. Sun has started architecting SND. They came up against the issue of locking stripes sequentially to avoid a cascading abort problem, which produces a performance problem. So we have either an abort problem or a performance problem. We may need to put this issue on the lustre-devel list for continued discussion or hold a workshop on it.

4. Evolve Lustre towards a more community-driven development model

Sun would like to see the Lustre Group leverage the larger user community in the evolution and development of Lustre. Regarding development efforts, the CEA HSM project is a good example of community involvement. Summit attendees were asked to review CEA's designs and work with Jacques-Charles on the project, where possible.

In discussing this topic, the subject of Sun's contributor agreement was brought up. Sun requires the agreement to be signed before open source code can be accepted. The code must be able to be re-distributed by Sun, which is why joint copyrights are needed.

The group discussed access to the CVS tree and if it would really be used. Sun indicated that some limited access to checked-in code could possibly be granted to select community individuals.

Major Lustre roadmap strategies were also discussed. Several attendees said that they do not want to be forced into a change like ZFS without some community discussion.

A simple, community-driven charter was suggested (which others have used). See Peter Bojanic's comments at the end of this paper for 'next steps' regarding community involvement.

5. Support for Very/Ultra-Large LargeClusters and WAN

Peter Braam opened this topic by saying we must think about scaling to over 100,000 clients. At this point, 50-100,000 clients are not a problem. Even 500,000 clients might have issues, but are workable. Some attendees commented that we need a collectl project at ORNL. Two challenges in handling large numbers of clients will be that the recovery protocol between MDS and thousands of OSSs will not scale, and the pinger may have scalability issues.

Eric Barton indicated that Lustre is modeling for 1,000,000 clients, but it is expected to support 100,000 clients. We need to avoid "concerted denial of service" problems by getting one millionth of the bandwidth available one second before a new system comes online.

Diagnostics were discussed again. The issue was posed that with large clusters and WAN, it is critical to know where the problems are. As a team, how do we solve this problem at scale? Scaling is more than just the number of clients. It is also things like scaling of OSSs, striping limits on the OSTs, scaling of routers, and so on.

We reviewed the striping limit in ldiskfs. This can be fixed in ZFS, but it needs a new disk format. Andreas indicated that a user can combine files, regardless of striping, and create a new MDS record to combine them. Then, the client gets the combined list of OSTs. This functionality works, but the code is still in beta. fsck may miss some orphaned elements.

Increasing the number of OSTs will be a problem for recovery since the interaction with OSTs and the MDSs will be onerous. Bandwidth of 1 TB/s will not be a problem, but the number of OSTs for the 10 TB/s bandwidth may be an issue.

ORNL indicated that we need information on resource issues. What is the impact of an additional client on the OSTs and the routers? Sun stated that scaling at LLNL benefits from going through routers so that 100,000 clients can be supported without 100,000 TCP connections to the servers. It is a multi-cluster issue, not a WAN issue. We agreed that I/O forwarding could also help scale, and noted that several different I/O forwarding schemes are in use.

Next Steps

This last section summarizes important discussion points from the LCE summit, recommendations that Sun took away from the meeting, and adjustments to the Lustre Group's plans based on feedback at the summit.

Lustre CTO's Summary

Eric Barton, Lustre's CTO, focused on the overriding theme of stability that he heard at the meeting, and stated his intention to drive this objective in the Lustre Group. As a result of the summit, Eric said he will focus the team on these areas:

Code Ownership and Coverage. Eric said he did not want obscure coding or unexplored areas of the Lustre code. He will strive for clear, documented internal APIs from which to continue to build a foundation for stability and scalability. Eric wants to establish Lustre Group experts for each Lustre subsystem.

Engineering Process. Eric talked about improving the engineering process. Sun will focus on stronger branch management, which will enable improved, concurrent feature development.

Realistic Roadmap. Eric also talked about the need to fulfill commitments to Lustre partners and customers, and set appropriate expectations for feature development. He believes this can be accomplished by building on solid ground, having a realistic roadmap, ensuring interoperability, following enumerated use cases, and limiting product complexity.

Quality Engineering Improving Lustre depends, in large part, on quality engineering. Eric underscored the importance of regression test automation, and leveraging customer site testing.

Summit Recap

Peter Bojanic, Director of the Lustre Group, summarized the meeting and what we can take away from two days of roundtable discussion.

Peter opened by saying he was pleased with the turnout at the LCE Summit. Approximately 30 of Lustre's largest customers and partners made time to attend this event. The meeting's focus was on big, strategic and HPC uses of Lustre. The invitations were made largely by ORNL and Sun. For the next LCE summit, Peter requested feedback on how to manage the invitations, yet keep the group at a workable size, and the need for commercial representation.

In reviewing what was accomplished at the summit, he noted the excellent level and quality of engagement by the participants. There was candid, valuable discussion of the participants' vision and priorities for Lustre, and a deep dive into as many of the ranked topics as we could discuss in two days. Peter was glad that so many Lustre engineers could attend the summit.

On the subject of improvements for the next LCE Summit, we noted that Eric Barton and the team came away with the lion’s share of the work. While this is not surprising, and not necessarily a negative result, Peter would like to see the user community contribute more to Lustre's design documents and code submissions. He talked about the need for greater community ownership of projects.

To promote greater community involvement, Peter asked us to establish a permanent LCE Council, with the summit participants as the founding members. He asked who would step forward to help lead this effort. Peter suggested several tasks to establish the LCE Council:

- Identify leaders from the LCE Summit.
- Define a group mandate, with responsibilities, authorities and accountabilities.
- Define the criteria for membership.
- Use the mailing list lce-council@lists.lustre.org, with public subscriptions and approved posters.
- Continue to use the Lustre wiki at http://wiki.lustre.org and the Lustre architecture wiki at http://arch.lustre.org.
- Elaborate on requirements on the website, and use the wiki for discussion, change tracking, notification.

After discussing these tasks, Peter talked about managing Lustre requirements. He said that the Lustre Group would post the community priorities and then tie them to the Lustre roadmap in an explicit and traceable way. We will build on established community standards and continue to ask for community input as Lustre evolves.

As an example, the CEA HSM project is the most ambitious Lustre community project ever undertaken. The high-level design is published on lustre-devel@lists.lustre.org and feedback is highly encouraged. CEA underscored this to the summit participants.

Next Step Actions for Sun and the Community

Peter talked about the themes that Sun took away from the LCE Summit. Although many Lustre topics and priorities were discussed, the themes of stability, interoperability, maintaining performance and scaling, and introducing features to support this overall context were heard very clearly by Sun. Indeed, in a postscript to the LCE Summit, the Lustre Group has taken these themes to heart in its preparation for a week-long internal planning and resource allocation session to be held near the end of March 2008.

Peter closed his remarks by saying that the LCE event was very worthwhile, and he thanked everyone for attending. We discussed when and where to host a follow-on meeting. Most people agreed that the Fall of 2008 time frame made the most sense. The site of the next LCE Summit will most likely be at Oak Ridge National Laboratory.