Author: Nikitas Angelinas <nikitas_angelinas@xyratex.com>
Reviewers: Peter Bojanic <peter_bojanic@xyratex.com>
Nathan Rutman <nathan_rutman@xyratex.com>

Date: 23 October 2011

The following document contains some notes and impressions from the European Open File System Lustre Workshop held at the Pullman Bercy hotel in Paris during the 26th-27th of September 2011, as well as the post-workshop Lustre developer summit that took place on the 28th of September 2011 on the same venue. Slides from the 2-day workshop presentations are available at http://www.eofs.org/index.php?id=5; slides and notes from the developer summit are scheduled to be made available on the EOFS website soon.

EOFS Lustre Workshop:

Workshop attendees were approximately 60 in number, mostly from Europe and USA, and included Lustre developers, administrators, support, sales and management personnel. In general, it was a good opportunity to meet people that carry out different roles based around the filesystem, and get an appreciation of what their role entails and how it relates to each other's work. The presentations given could be broadly segregated into two categories, filesystem administration-oriented, and project/development-oriented, with an approximately even distribution amongst the two groups.

The first day started off with Peter Jones of Whamcloud giving an overview of Whamcloud's maintenance and feature release streams for Lustre (currently based on versions 1.8 and 2.0 respectively), and giving some information on the work performed by Whamcloud and other community members on the impending (at the time of the presentation, now released) 2.1 release. Peter followed this with some information and an overview of features for the 2.2 release, as well as an overview of Whamcloud's development roadmap for the next two years or so.

The next talk was given by Roland Laifer of KIT, Karlsruhe, who presented an overview of Lustre deployments (four presented in total) that have taken place at KIT sites. The talk started by giving a good amount of information on fabric interconnects and filesystem characteristics, and continued with lessons learned from administering the aforementioned filesystem deployments, as well as a list of areas that could be improved to make administrator tasks more efficient.

The next presentation was given by Lu Wang of IHEP, Beijing, and also revolved around aspects of Lustre administration. The talk started by giving a description of current and previous Lustre deployments at IHEP, and then proceeded with presenting a good amount of performance-related information pertaining to the current deployment. This was followed by a description of issues encountered in managing the filesystem along with steps taken to resolve these and a list of nice-to-have Lustre features from the administrator's point of view. The talk concluded by providing some information on what must be a much-anticipated first ever Lustre workshop to be held in China towards the end of November this year.

The last presentation for the morning session was from Stéphane Thiell of CEA, and also revolved around administration-related aspects, but was strictly dedicated to version 2.0 deployments. The talk started by giving some information on the adoption of Lustre 2.0 releases at CEA, and proceeded with an overview of the software ecosystem used for administering Lustre deployments at CEA. Special reference is made to Shine, a library and associated tools to setup and manage Lustre filesystem deployments, followed by a list of important tips for performance tuning, tools used for monitoring, and patches applied that resolved important issues. The overall impression from CEA is that the current version 2.0 deployment of Lustre that includes all the necessary patches, is very stable; it is probably worth noting that CEA mentioned being able to conduct fsck runs in a realistic time frame by enabling the flex_bg ext4 feature in ldiskfs.

The afternoon session started by Alexandre Louvet of BULL presenting issues relating to how BULL provides Lustre support to their customers. The talk started by giving some information on the problem space when having to debug Lustre-related issues, with reference made to the difficulties involved, the types of problems encountered, and tools used to arrive at solutions. The talk then proceeded to present a couple of case studies of real-world issues (a recovery-related issue on the MDS, and an MDS hang issue), along with steps of how these were addressed by making use of some of the aforementioned tools.

The next presentation was from Michael Kluge of TU Dresden; this revolved around Vampir, a novel tool that uses an RPC tracing mechanism to provide with information on RPCs as they travel throughout the filesystem, such as average RPC completion times and RPC queuing times on the servers, in the form of heat-map-like graphs. The talk made reference to the challenges encountered in implementing the tool, along with techniques and data structures employed to overcome these challenges, and finishes with some suggestions on the next steps to be taken on the project.

The next presentation was given from Johann Lombardi of Whamcloud, and revolved around some of the new features introduced in Lustre 2.x. Special emphasis was given to FIDs, the older analogous scheme in versions 1.8.x, and some of the problems FIDs help overcome. The talk then proceeded by making reference to the internal structure of FIDs, FID Sequences, LMA and Link extended attributes and IGIF FIDs. The talk finished by highlighting some important interoperability and upgrade issues between 1.8 and 2.x versions of Lustre, and Ext4-related changes in Lustre 2.1.

The next presentation was given by Torben Kling Petersen of Xyratex, and described Xyratex's ClusterStor family of products for HPC. The talk started by listing the motivating factors for engineering an integrated solution for Lustre appliances, and proceeded with listing advantages offered by ClusterStor product hardware, namely, the highest available in the market storage density, redundant components throughout the storage enclosure level and Xyratex IP in the form of Metis, a battery-enabled mechanism that allows the dump of data stored in volatile memory in the case of AC power failure to the enclosure. The talk then proceeded with giving some information on the storage and OSS configuration and characteristics for ClusterStor 3000. This is followed by a description of the software stack used in the product, with emphasis given on Xyratex-developed software components such as GEM (Generic Enclosure Management, the storage enclosure firmware) and ClusterStor Manager (integrated installation and management front-end).

The next talk was from Stephen Simms of Indiana University. It pertained to a technique employed alongside the Data Capacitor project that helped the IU-lead team win the Bandwidth Challenge competition in SC07, in order to provide homogeneous UIDs on a Lustre deployment across a large scale WAN. The talk started by giving some information on the changes required to the mainline MDS Lustre code, as well as data structures used to implement the solution at this scale, and proceeded with listing some limitations of the current IU UID patch, and how a new iteration of the code is scheduled to overcome these limitations in Lustre 2.x.

The final talk for the first day was from Eric Barton of Whamcloud, and revolved around architectural additions that will enable Lustre to become a feasible choice in future Exascale computing cluster configurations. The talk started by setting the scene encountered in Exascale computing environments, by giving some information on the order of complexity of environment characteristics, depicting non-volatile memory technologies as an integral component in the design of a two-level storage scheme that was proposed later in the presentation, and listing the core requirements for a filesystem capable of operating at this scale and resultant design ideas that aim to satisfy these requirements. The talk then proceeded to propose a model of I/O that aims to overcome the limits imposed by POSIX (which most experts seem to agree should be abandoned for Exascale); this uses a container-based object storage scheme that along with application-specific libraries will allow Lustre to serve the I/O needs of exascale applications, from the same namespace as 'legacy' (i.e. in the future, so 'current generation') applications.

The second day of the workshop started with a talk from Alfonso Pardo Díaz of CETA-CIEMAT, describing the experiences with the use of Lustre at the above site. The talk started by listing some of the advantages of using Lustre compared to a previous NFS deployment, and then proceeded to give some information related to cluster availability, reliability and design concerns. These included failure scenarios for filesystem components along with measures that should be taken to compensate in such occurrences, experiences with different Lustre releases, and the adoption of bind mounts and grid computing middleware to solve problems faced.

The next talk was from Greg Matthews of Diamond Light Source (DLS), UK, and presented information relating to the use of Lustre at the above site. The talk started by giving detailed information on the type of work carried out at DLS, and continued with depicting typical usage scenarios encountered for the movement of data throughout its lifetime at the site, and equipment used that are the source of data generation. The talk then continued to give information on the network topology and current Lustre deployments, and finished with performance benchmarking data and impressions from the use of the said Lustre deployments.

The next talk was from Frank Heckes of Forschungszentrum Juelich, Germany, and revolved around impressions from the use of Lustre deployments (24 filesystems in total) at the above site. The talk started by giving information on the cluster configurations, and continued by highlighting some problematic use cases as well as a large deviation in overall available throughput in one of the filesystem deployments. Possible reasons for this were given from the presenter, and Whamcloud employees (Eric and Oleg in particular) seemed to be making a note of this; it is assumed that this matter may require more investigation in order to determine whether it is a setup-related issue or a performance-related issue with Lustre. The talk then continued with some information on the institute's plans on performing a storage upgrade, and of providing a test cluster for the community with the ability to run automated installation and smoke test runs, of builds generated from Whamcloud's Jenkins server.

The next talk was from Daniel Kobras of Science+Computing AG; by drawing from experiences of deploying and administering various scalable storage solutions for different customers of S+C AG, the presenter was able to give a talk revolving around a comparison between Lustre, and GPFS (as the most important representative of competition in the scalable filesystem arena) from the administrator's viewpoint. After listing the categories used for the assessment of the two filesystems, the talk proceeded to consider each category in turn, by first presenting an idealized list of features and then assessing each filesystem and on most cases giving examples of exhibited behaviour from each filesystem, for each category. Perhaps the most important things that were highlighted as advantages of Lustre are the ability for comprehensive configuration, the community support, openness, and OST pools feature, while most important disadvantages were the lack of a unified method for managing configuration options, the steep learning curve for administrators, the non-admin-friendly and non-enlightening nature of error messages, and also the lack of GSSAPI support, transparent data migration and HSM (although HSM seems to be currently scheduled for landing in Lustre version 2.4 and is worked on by CEA, and transparent data migration seems to - at least at some point - have been planned to be enabled by the implementation of Epochs).

The next talk was from Thomas Leibovici of CEA, and revolved around Robinhood, a tool developed by CEA to manage the content of different types of filesystems mainly on HPC environments, and offers special features when used with Lustre (OST, OST pool and FID-aware, uses changelogs to avoid filesystem scanning). The tool can provide filesystem statistics and fast queries, data management and alerts based on administrator-set rules, scratch filesystem management, as well as data archiving, and HSM-binding. If memory serves right, at least one more administrator that was not a member of CEA mentioned they also use Robinhood at their site, and the subject seemed to gather some interest from attendees in general; more information on the tool can be found on the tool's home page at http://sourceforge.net/apps/trac/robinhood.

The next talk was from John Spray of Whamcloud, and revolved around a new management project for Lustre designed by Whamcloud, codenamed 'Hydra'. The aim of this project is to simplify provisioning, management and monitoring of Lustre deployments by establishing a unified framework that storage controller hardware can offer support for in the near future; Hydra-enabled devices will pass configuration and status information to higher software layers in order to enable the aforementioned functionality offered by Hydra. The project makes use of a number of open source tools to achieve distributed operation across the cluster, uses dependency-aware schemas to characterize the cluster, and exports relevant outputs using a web interface.

The last presentation of the workshop was from Nathan Rutman of Xyratex, and revolved around the possible exploitation of various hardware features with the aim of improving performance and providing enhanced data integrity capabilities for Lustre. Topics presented were hardware CRC acceleration, improvements to MDRAID, an end-to-end data integrity scheme using T10-DIF/DIX, the utilization of flash drives in Lustre servers, the concept of Ext4

Hybrid Volumes, and improvements related to HA and failover. The topic of CRC acceleration covered the usage of the CRC32 instruction on Intel Nehalem and successor processors (presumably also in other x86 compatible CPUs that support the SSE4.2 instruction subset) in order to add a new checksum type (CRC32C) to Lustre (this feature seems to have landed on Whamcloud's software repository for version 2.2 of Lustre, see Whamcloud Jira ticket LU-241), and the possibility of using the PCLMULQDQ instruction in Intel Westmere and successor CPUs (and also in other x86 compatible CPUs that support the CLMUL instruction set) in order to accelerate the existing CRC32 checksum type calculation in Lustre. The T10-DIF/DIX-based part of the talk depicted a scheme that aims to allow for end-to-end data integrity of file data from the client to the T10-DIF/DIX-enabled disk drive at OSS nodes; this consumed a fair amount of the total time for the talk and seemed to require further discussion between all interested parties in order to agree on the best way forward (some consensus was reached on this topic during the developer summit that followed the EOFS workshop, please see below). The presentation of the Ext4 Hybrid Volumes concept received a comment from Oleg Drokin regarding Whamcloud engineers having tried something similar in the past but finding that it did not benefit performance much due to the filesystem journal (if memory serves right, on the MDS) rearranging most writes into large contiguous IO; Nathan feels that more information would be useful in fully determining the relationship between the work performed by Whamcloud engineers and the potential usefulness of having an Ext4 Hybrid Volumes implementation.

Lustre Summit:

Following the EOFS workshop, a one-day Lustre summit took place amongst some of the developers, and engineers responsible for carrying out architectural design work; attendees included representatives from Whamcloud, Xyratex, ORNL, LLNL, CEA and Cray (apologies if I neglected to include someone). The day started with a talk by Alexander Zhuravlev of Whamcloud regarding the OSD restructuring work that is being carried out by Whamcloud and LLNL; this project aims to restructure some of the server code in order to allow the use of different back-end filesystems in the future (MDS and OBDFILTER layers are being removed, and new layers LOD and OSP are being added, along with the ensuing cleanup for currently duplicated code) and both OSS and MGS servers are to transition to the new OSD interface (MDS servers already make use of the current OSD interface, although they have not transitioned completely). The code for this seems to live under the 'orion' branch on the Whamcloud development repository; some information on this project can be found on an announcement by Andreas Dilger on the Lustre-devel mailing list at http://lists.lustre.org/pipermail/lustre-devel/2011-June/003831.html.

The next talk was from Eric Barton of Whamcloud, in which he presented plans for the Distributed Namespace (DNE) feature. DNE aims to provide the ability for more scalable metadata access patterns for directory operations, by partitioning the filesystem namespace across more than one MDS. DNE will be developed against the OSD-restructured codebase, requires the FID-on-OST feature to be implemented in order to allow files to scale to more than 8 MDTs, and is planned to land in two phases. Phase 1, 'remote directories', aims to deliver some of the benefits of DNE and allow for feedback from real-world DNE deployments to become available. By requiring administrator privileges in order to regulate directory entry allocation, Phase 1 intends to allow spreading home and project directories across multiple MDTs, but will constrain subdirectories and files to the same MDT; rename and hardlink operations between files and directories on different MDTs will not be handled at this stage. Phase 2, 'striped directories', will allow a single directory to be distributed over multiple MDTs, while allowing file create operations to take place in an isolated manner in each directory stripe in order to improve performance for create operations in a single shared directory, and will allow for distributed hardlink and rename operations to take place. More information on the DNE feature can be found at http://www.opensfs.org/wp-content/uploads/2011/03/OpenSFS-Contract-Final-2011-07-29-Article11.pdf .

The next talk was lead by Nathan Rutman of Xyratex; Nathan presented a proposal for a bitmap-based wide striping method that would allow a larger maximum number of objects for a single file, plans for IPv6 support in Lustre, and analyzed the T10-DIF/DIX-based end-to-end data integrity scheme he had presented on the second day of the workshop. The wide striping scheme received some concern regarding how the OST objects would be identified, and how the scheme could work in conjunction with the FID-on-OST feature that will be required to be implemented in anticipation of DNE. The topic has since been followed up on in a discussion in the Lustre-devel mailing list at http://lists.lustre.org/pipermail/lustre-devel/2011-October/003895.html . On IPv6 support, community members wanted to start scoping the amount of effort required to make the necessary changes to LNET in order to provide for interoperability between IPv4 and IPv6-based nodes. The topic has since been followed up on in a discussion in the Lustre-devel mailing list at http://lists.lustre.org/pipermail/lustre-devel/2011-October/003919.html; scoping of the work required has been started by Isaac Huang of Xyratex. Regarding the T10-DIF/DIX-based end-to-end data integrity scheme, Whamcloud engineers seemed to be more in favour of making use of Sun's Merkle tree approach, in order to better integrate with the end-to-end checksumming capabilities in ZFS, when the latter is employed as a back-end filesystem for Lustre. Some discussion took place on a method that could be used in order to accommodate both approaches; the consensus was that the end-to-end data integrity method to be supported for a given client connection should be determined via negotiation at client connection time, and that the underlying protocol should be made flexible enough in order to handle either of the approaches.

The next talk was from Nikitas Angelinas of Xyratex, and revolved around the Network Request Scheduler (NRS) feature, which is being developed as a collaborative project between Whamcloud and Xyratex. The talk presented the main motivating factors behind developing the NRS component, different types of NRS policies and their applications, and gave an overview of high-level aspects of the design; during the talk it was mentioned that Xyratex aims to develop a version of the Object-Based Round Robin policy for NRS, in addition to the currently existing ones developed by Whamcloud. Eric Barton explained the role of the binary heap implementation that is added to libcfs and is part of the NRS project, and also gave some performance data for a pair of insert and remove operations for the binary heap. Some of the attendees raised a request for having the different NRS policies be available as separate loadable kernel modules, in an effort to ease development of new policies in the future. From early discussions between involved parties, it seems possible that this feature will be implemented at some point (especially since it will only require some small changes to the existing code base), probably once the current work on NRS is finished.

Towards the end of the summit, the possibility of porting the Lustre sources on a currently unsupported CPU architecture was mentioned, with aim of scoping the amount of work this would require. Xyratex engineers informed the attendees that they had successfully attempted ports of different 1.8.x versions of Lustre on a MIPS64-based storage controller in the past with little difficulty. Eric Barton also mentioned that there may be additional complications that could arise due to differences between endianness-related characteristics of various CPUs, although the fact that the MIPS64 CPUs used by Xyratex (that were of configurable endianness (and page size)) for running the ported Lustre codebases were used in big-endian mode (x86 and x86-64 are liitle-endian CPU architectures), should be encouraging for similar undertakings in the future.