LUSTRE® in HPC, AI and Big Data: Widening Scope, New Features and Roadmap

Tuesday, May 31st / 10:30 – 11:30 AM / Hall E at CCH Hamburg

Frank Baetke (EOFS & for HPE)
Hugo Falter (EOFS & ParTec)
Kevin Harms (OpenSFS & ANL)
Carsten Beyer / Thomas Ludwig (DKRZ)
Peter Jones (EOFS & Whamcloud-DDN)
Andreas Dilger (Whamcloud-DDN)
Jacques-Charles Lafoucriere (EOFS & CEA)
LUSTRE BoF @ ISC 2022 - Agenda

10:30 – 10:35 Welcome and Introduction
Frank Baetke (EOFS - for HPE)
Hugo Falter (ParTec)
Kevin Harms (ANL)

10:35 – 10:40 Next Generation Lustre System at DKRZ
Carsten Beyer (DKRZ)

10:40 – 10:50 Lustre News & Community Update
Peter Jones (DDN/Whamcloud)

10:50 – 11:00 Lustre Features and Community Requests
Andreas Dilger (DDN/Whamcloud)

11:00 – 11:20 General Discussion

11:20 – 11:25 How to Get Involved With Lustre
Jacques-Charles Lafoucriere (CEA)

11:30 Adjourn

Short Q&A
EOFS President / Chairman of the Board
• Frank Baetke (acting for HPE)

EOFS Vice-President:
• Jacques-Charles Lafoucriere (CEA)

EOFS Directors:
• Hugo R. Falter (ParTec AG)
• Peter Jones (DDN/Whamcloud)

Members of the Administrative Council:
• Eric Monchalin (Atos)
• Jacques-Charles Lafoucriere (CEA)
• Thomas Stibor (GSI)
• Frank Baetke (acting for HPE)
• Johann Lombardi (Intel)
• Arndt Bode (LRZ)
What is OpenSFS?

- OpenSFS facilitates a community around Lustre
  - Organization for both Vendors (Participants) and Users (Members) to discuss features and directions
- Promote Lustre and the Lustre community
- Ensure Lustre remains vendor-neutral and open
- Organize the LUG conference

- Co-owner of the LUSTRE trademark, logo and assets

http://opensfs.org
Deutsches Klimarechenzentrum
(German Climate Computing Centre)
DKRZ

Next Generation Lustre System

Carsten Beyer
(beyer@dkrz)
German Climate Computing Centre

Lustre at DKRZ

Old system Mistral
ClusterStor CS9000 with 21 PiB (Infiniband)
ClusterStor L300 with 33 PiB (Infiniband)
Will be switched off in June.

Plan: reconfigure L300 as Cloud storage without Lustre
HPC system Levante

3 Filesystems based on NVMe / HDD and Infiniband

- HOME
  - 116 TiB NVMe (4 MDS/MDT / 4 OST) – DDN ES400NVX
  - Home directories for users
  - Software tree
  - User quota
  - Directory stripping over all MDT (User toplevel dir)
PROJECT / WORK

- 118 PiB HDD (8 MDS/MDT + 80 OSS/160 OST) – DDN ES7990X
  - WORK directories for customer projects
  - SCRATCH directories for user
  - Pool for common data
- Usage of Progressive File Layout (PFL)
- Directory stripping over all 8 MDT (toplevel dir)
- Project Quota for each WORK/SCRATCH directory
- Planned: Stratagem/LIPE policy engine for reporting
FASTDATA

- 3.2 PiB NVMe / HDD (4x ES400NVX / 1x ES7990)
  - Approx. 200 TiB NVMe / 3 PiB HDD
  - 16 MDT / 4 OST HDD / 16 OST NVMe
  - 2 Pools (one each for NVMe / HDD)

- Currently a collaboration project with DDN
  - Hybrid usage NVMe/HDD
  - Hot data/pools
  - Stripping to pools (e.g. with PFL)

- Maybe later integration in PROJECT/WORK
German Climate Computing Centre

Challenge

How to copy approx. 45 PiB of data from old to new system

- Not enough or no LNET Router available
- Configure second set of IP addresses on the DDN Storage as additional @tcp device
- Mounting the new Lustre via @tcp on old system
- Usage of SLURM jobs with pftool/rsync for each project
- Duration approx. 2 months
Thank you

Questions?

Carsten Beyer
beyer@dkrz.de
Short Q&A
Lustre News & Community Update

Peter Jones
Whamcloud/DDN
Lustre LTS Releases

Lustre 2.12.8 went GA in December
- http://wiki.lustre.org/Lustre_2.12.8_Changelog

Lustre 2.12.9 coming soon
- RHEL 8.6 client support
- https://wiki.lustre.org/Lustre_2.12.9_Changelog

Upcoming Lustre 2.15 release will be next LTS
- Will be transition period between 2.12.x and 2.15.x
- Likely Lustre 2.12.10 will be last 2.12.x release

Which Lustre versions do you use in production? (select all that apply)

https://wiki.opensfs.org/Lustre_Community_Survey
Lustre Major Releases

Lustre 2.15.0 coming soon
• RC4 likely to be GA version
• http://wiki.lustre.org/Release_2.15.0
• https://wiki.lustre.org/Lustre_2.15.0_Changelog
• OS support
  • RHEL 8.5 servers/clients
  • RHEL 8.5/SLES15 SP3/Ubuntu 20.04 clients
  • More current distro support will appear in future
2.15.x maintenance releases

Lustre 2.16.0 landings underway
• Roadmap refresh will finalize after 2.15 GA
Community Events

Community events were all virtual throughout the pandemic but some in-person events during 2022

Rice Energy 22 Mar 3rd Hybrid event
- https://2022energyhpc.blogs.rice.edu

LUG22 May 9th – 11th Virtual event
- https://www.opensfs.org/events/lug-2022/

ISC22 Lustre BOF May 31st In-person event (now 😊)
LAD22 Sep 26th – 28th Hybrid event
- https://www.eofs.eu/events/lad22

Dates and format TBA for Australia, China and Japan 2022 LUG events and possible SC22 Lustre BOF

Data analysis of top500.org lists
Lustre 2.16 and Beyond

Andreas Dilger
Lustre Principal Architect
Planned Feature Release Highlights

► **2.16** opening to land new feature patches
  - **LNet IPv6 addressing** – allow 160-bit NIDs, more flexible server configuration (SuSE)
  - **Optimized Directory Traversal** (WBC1) – cross-directory statahead (WC)

► **2.17** has several major features already lined up
  - **Client-side data compression** – use client CPU to reduce network and storage usage (WC)
  - **Metadata Writeback Cache** (WBC2) – low latency file operations in client RAM (WC)
  - **File Level Redundancy - Erasure Coding (EC)** – efficiently store file redundancy

► **2.18** feature proposals in early discussion stages
  - **Lustre Metadata Redundancy (LMR1)** – MDT0000 service redundancy
LNet Improvements (2.15/2.16)

- **Multiple TCP sockets for 100GigE+ performance** ([LU-12815](#), WC)
  - Add `conns_per_peer=N` for `sock1nd` (4.1GB/s→9.5GB/s on 100GbE)
  - Auto-configure based on interface speed (e.g. 10Gbps=>2, 100Gbps=>4, ...)

- **LNet Network Selection Policy (UDSP)** ([LU-9121](#), WC)
  - Allow policies for local/remote interface prioritization by NID
    - e.g. primary IB with TCP backup, select "best" router NID for client/server

- **IPv6 NID support** ([LU-10391](#), SuSE)
  - Variable-sized NIDs (8-bit type, 8-bit size, 16-bit network, 128-bit+ address)
  - Interoperable with existing current LNDs whenever possible

- **Simplified/dynamic server node addressing** ([LU-14668](#), WC)
  - Detect added/changed server interfaces automatically ([LU-10360](#))
  - Reduce (and eventually eliminate) static NIDs in Lustre config logs
Client Improvements

(ORNL, SuSE, WC)

► **GPU Direct RDMA** - directly into GPU, bypass CPU ([LU-14798](#), WC, NVIDIA, HPE)
  - A100 2x200Gb IB **36GB/s** write, **39GB/s** read, **174GB/s** with 8x200Gb IB

► **Parallel large DIO** optimization ([LU-13798](#), [LU-13799](#), HPE, WC)
  - Improve single-thread read()/write() (1.5GB/s->**15.8GB/s**)!
  - Particular benefits for AIO/DIO and io_uring in client kernels 5.1 and later

► Improved "lfs find -printf" option for scanning files ([LU-10378](#), ORNL)

► **o2iblnd** cleanups for in-kernel OFED ([LU-8874](#), ORNL)

► Buffered/DIO/mmap performance/efficiency improvements ([LU-13805](#), WC)

► Ongoing code style/structure cleanup for upstream submission (ORNL)

► Ongoing updates for newer kernels (ORNL, SuSE)
Backend OSD Improvements (2.16+)

► Parallel e2fsck for pass2/3 (directory entries, name linkage) (LU-14679, WC)
  • Now slowest part of e2fsck (was 7% of total time, now 70% after pass1/pass5 speedups)

► ZFS 2.1 dRAID VDEVs - declustered parity and hot space (LLNL, HPE, Intel)

► fallocate() and FALLOCATE_FL_PUNCH_HOLE for ZFS (LU-14157, AEON)

► Improved ldiskfs mballoc efficiency for large/full filesystems (LU-14438, Google, WC)
  • O(1) lookup of power-of-two free space, O(logN) lookup of other sizes

► Improved ldiskfs "-o discard" efficiency (LU-14712, Kuaishou, WC)
  • Allow real-time TRIM of flash storage to maintain peak performance

► OST object directory scalability for large OSTs (LU-11912, WC)
  • Large OSTs (500-1000TB) have billions of objects, only 32 dirs per MDT!
  • Wider dir fanout not better, object create/remove access all dirs randomly
  • New OST FID Sequences more often (e.g. 32M vs. 4B objs), retire old SEQ
  • Groups objects by age to limit directory size and improve efficiency
Batched Cross-Directory Statahead (WC 2.16)

- **Batched RPCs** for multi-update operations ([LU-13045](#))
  - Allow multiple getattrs/updates packed into a single MDS RPC
  - More efficient network and server-side request handling

- **Batched statahead** for `ls -l`, `find`, etc. ([LU-14139](#))
  - Aggregate getattr RPCs for existing statahead mechanism

- **Cross-Directory statahead** pattern matching ([LU-14380](#))
  - Existing statahead only detects `readdir()`-ordered `stat()`
  - Detect pattern for alphanumeric ordered traversal + `stat()`
  - Detect breadth-first (BFS) depth-first (DFS) directory traversal
  - Direct statahead to next file/subdirectory based on pattern
Create new dirs/files in client RAM without RPCs
- Lock new directory exclusively at mkdir time
- Cache new files/dirs/data in RAM until cache flush or remote access

No RPC round-trips for file modifications in new directory

Batch RPC for efficient directory fetch and cache flush

Files globally visible on remote client access
- Flush top-level entries, exclusively lock new subdirs, unlock parent
- Repeat as needed for subdirectories being accessed remotely
- Flush rest of tree in background to MDS/OSS by age or size limits

Productization of WBC code well underway
- Some complexity handling partially-cached directories
- Need to integrate space usage with quota/grant
MDT DNE Improvements (WC 2.15+)

► **DNE MDT Space Balance** - load balancing with normal `mkdir` ([LU-13439, LU-13440])
  - Round-robin/balanced subdirs, prefer to stay on parent, limited layout inheritance depth
  - Keep MDTs within free inodes/space (`mdt.*.mdt_qos_threshold_rr=5%`)

► **Single-dir migration** - "`lfs migrate -m -d <dir>`" ([LU-14975])
  - Move only one directory level, instead recusing down full subdirectory tree

► **Balanced migration** - "`lfs migrate -m -1 <dir>`" ([LU-13076])
  - Auto-select less-full MDTs for each directory, keep inodes local to parent

2.15

2.16  

► **OST object directory scalability for large OSTs** ([LU-11912])
  - Request new OST FID Sequences more frequently

► **DNE locking, migration, remote RPC optimization** ([LU-15528])
  - Improve distributed transaction performance, reduce lock contention
Lustre Metadata Redundancy (LU-12310) (2.17+)

- **LMR1a: Replicate MDT0000 Services to Other MDS Nodes**
  - Add replication for FLDB, Quota Master, \texttt{flock()} across all MDTs

- **LMR1b: DNE Distributed Transaction Performance**
  - DNE2 Distributed Transactions have excessive ordering/sync operations
  - Optimizations improve all DNE ops, independent of LMR

- **LMR1c: Replicate Top-level Directories for improved availability**
  - \texttt{ROOT/} dir (rarely modified) replicated over MDTs, no file replication

- **Additional LMR2/3 phases to reach full MDT redundancy**
  - Full tree replication, inode replication, configurable per directory
  - Recovery, LFSCK, rebuild replicated directories after MDT loss
Thank You!
Questions?
General Discussion
Lustre at CEA

Jacques-Charles Lafoucrière, Thomas Leibovici

ISC 2022
Lustre at CEA: a long history

• In production at CEA since 2005, on the TERA10 supercomputer
  – At that time, Lustre version was ... 1.4
  – Initial use in a testbed since 2002 (0.5.16)

• A long history of contributions
  – Nearly 600 patches written or reviewed by CEA
  – CEA developed key features of Lustre like
    • OST pools
    • HSM

• CEA also develops open-source tools for Lustre
  – Shine to ease Lustre FS administration: https://github.com/cea-hpc/shine
  – Robinhood Policy Engine to monitor and manage filesystem contents:
    https://github.com/cea-hpc/robinhood
Lustre file systems at CEA: overview

11 Lustre filesystems and 100PB+ in 4 facilities

**TERA/EXA computing center (defense)**
- 2 SCRATCH filesystems
  - TERA 1000
  - EXA 1
- STORE filesystem = long-term storage based on HSM
- WORK filesystem = permanent workspace

**TGCC computing center (research & industry)**
- 2 SCRATCH filesystems
  - Joliot-Curie (Prace Tier 0)
  - Topaze (CCRT)
- STORE filesystem
- 2 WORK filesystems
  - Full HDD
  - Hybrid (SSD+HDD)

**TERA+ lab (prototyping)**
- 1 filesystem split in 2 workspaces (SCRATCH and WORK)
- 1 filesystem for long term storage (STORE)
Focus on T1K-F, a full-flash Lustre filesystem

T1K-F
• First full-flash filesystem installed at CEA, in 2019
• Configuration:
  – 16 DDN SFA 18KXe
  – 864 NVMe drives (3.2TB, 3DWPD)
  – 128 ports InfiniBand EDR 100Gb
• 2.1 PB @ 1.2 TB/s IOR
• Ranked 6th fastest filesystem at SC19’s IO500
  – 3rd of HPC sites
Focus on EXA1’s STORE: a 3-level hierarchical storage

3 storage technologies in a single Lustre namespace

- NVMe drives (670TB)
- Hard Drive Disks (45PB)
- Tapes (150TB+) -> expandable at will

- NVMe and HDD partitioning using the OST pool feature
  - `lfs migrate` operations distributed using the `celery` framework
  - Robinhood to schedule data migrations

- HPSS integration using the Lustre/HSM feature
  - Robinhood to schedule data migrations
Lustre features – in used or planed -

• Most used features in production:
  – OST pools + lfs migrate
  – HSM
  – DNE (static)

• Planning to use in the short-term
  – Project quota
  – Data-on-MDS (DoM)
  – Progressive file layout (PFL)
  – LNET Multirail
Lustre features – under evaluation -

• Other features of interest (under evaluation):
  – Pool quota
  – Kerberos support
  – Client data encryption / client namespace encryption
  – User Defined Selection Policy (network rules)
  – GPU direct
  – Persistent client cache
Lustre-related developments

**Lustre-OpenStack integration** (funded by ICEI Eu project)

- Benefits: using Lustre as a storage backend for both HPC and cloud usages
- Development includes:
  - Swift-over-Lustre
  - Cinder-over-Lustre
Lustre-related developments

Lustre-OpenStack integration

• Icing on the cake: unified view between Lustre and Swift
  – Bi-directional synchronisation of Lustre and Swift namespaces
Lustre-related developments

Lustre-OpenStack integration

• Developed in the framework of the European project « ICEI », to implement the FENIX infrastructure
  – [https://fenix-ri.eu](https://fenix-ri.eu)

• Sub-contractor: LINAGORA
  – [https://linagora.com](https://linagora.com)

• Status:
  – Swift-over-Lustre works. Unified view to be implemented.
  – Cinder-over-Lustre: base feature is done, additional features to be implemented.
  – Target: all code integrated upstream in the coming year.
Lustre-related developments

RobinHood4: next gen policy engine

- Mirrors Lustre’s filesystem metadata to a **MongoDB** database for higher performance
- CLI commands to:
  - `rbh-sync` to scan FS namespace
  - `rbh-fsevents` to read Lustre changelogs
  - `rbh-find` to query the database using a find-like command
- Flexible use and integration using the **librobinhood API**
- [https://github.com/robinhood-suite/](https://github.com/robinhood-suite/)
Last but not least...

LAD is back in person!

• Lustre Admin & Dev workshop, with key actors of the Lustre community

• Hybrid event:
  – In person in Hotel des Arts et Métiers, Paris, France
  – Live broadcast online

• Save the date: **26 & 27 September 2022** (developer summit on September 28th).

• Keep updated on [https://www.eofs.eu/events/lad22](https://www.eofs.eu/events/lad22)
Thank you! Questions?
Final Q&A
Thank You

Looking forward to meeting you at

LAD’22 (planned as a hybrid event)
in Paris, France - September 26 to 28