



# SRCC

STANFORD RESEARCH COMPUTING CENTER

## Lustre @ SRCC

*Site update*

LUG Webinar Series

September 9, 2020



**Stéphane Thiell**

Stanford Research Computing Center







# SRCC

## Stanford Research Computing Center

<https://srcc.stanford.edu/>

### Our mission

Build & support a comprehensive program and capabilities to advance **computational** and **data-intensive** research at Stanford




Lustre systems  
at the **SRCC**





# Sherlock

- ▶ shared **HPC cluster**
- ▶ available to **all**  **faculty**  
*800+ groups, 5,100+ users*
- ▶ evolving continuously  
*1,385 nodes, 30,000+ cores, 550 GPUs*
- ▶ separate IB fabrics  
*InfiniBand FDR, EDR and **HDR 200Gb/s***
- ▶ **Lustre 2.13** (clients)



Sherlock racks at the Stanford Research  
Computing Facility (SRCF)



## SCG cluster

*Stanford Genomics Center*

- ▶ shared **HTC cluster**  
operated by the SRCC  
*High Throughput Computing*
- ▶ includes a SGI UV 300  
*NIH funded, 360 cores and 10TB RAM*
- ▶ Ethernet fabric  
*up to 100Gb/s over **TCP/IP***
- ▶ **Lustre 2.12.5** (*clients*)





## Fir storage

- ▶ **Sherlock's scratch**  
*Home-grown, multiple hardware vendors*
- ▶ **fast & large**  
*16 OSS, 6 PB usable, HDD-based OSTs*
- ▶ automatically **purged**  
*temporary filesystem (3 months)*
- ▶ **Lustre 2.12.5** (servers)



## Oak storage

- ▶ **site-wide Lustre** storage system for research  
*Home-grown w/ 4-year cost-recovery*
- ▶ growing continuously  
*today ~3,000 drives and 25 PB usable*
- ▶ **Lustre 2.10.8** (servers)

Oak's SAS switches at the  
Stanford Research Computing  
Facility (SRCF)

# Lustre 2.13 on Sherlock





## Sherlock Lustre 2.13 (lustre-client)

- ▶ **December 2019:** Lustre 2.13 rolling upgrade started!
  - ▷ Big performance boost for **single-threaded workloads**
  - ▷ We quickly found out that executables segfaulted on /scratch after DLM locks were revoked, whoops!
    - ▷ Workaround was to increase `lru_max_age`
- ▶ **Lustre 2.13 + PCC patch** (January 2020)
  - ▷ **LU-13137** *“User process segfaults since 2.13 client upgrade”*
    - ▷ Patch from Whamcloud: *“llite: do not flush COW pages from mapping”*
- ▶ **No further patching required (very stable since then)**
  - ▷ Even after **MOFED 5.0 upgrade** in early June/July 2020

# Lustre 2.12 on **Fir** storage





## Fir storage changelog (1/3)

- ▶ **Feb 2019**
  - ▷ Production started with **Lustre 2.12.0**
  - ▷ Features **DNE+DoM+PFL** enabled by default
- ▶ **May 2019**
  - ▷ Presentation at [LUG'19](#): *“Lustre 2.12 In Production”*
  - ▷ Stellar support from **Whamcloud** to fix stability issues
- ▶ **Sep 2019**
  - ▷ Added **8 OSS** with **WD Data60 JBODs** (+3PB usable)



## Fir storage changelog (2/3)

### ▶ Oct 2019

- ▷ Upgrade from IB EDR to **HDR** to prepare for **Sherlock 3**
- ▷ Added the ldiskfs feature **project** to all targets (for testing) and shortly discovered that users could change project IDs

### ▶ Nov 2019

- ▷ Discovered an obvious **performance limitation of DOM** with the AERO-F code (from the Farhat Research Group)
- ▷ DoM performance problems on shared files with multiple writers reported at the **SC'19 Lustre BoF**





## Fir storage changelog (3/3)

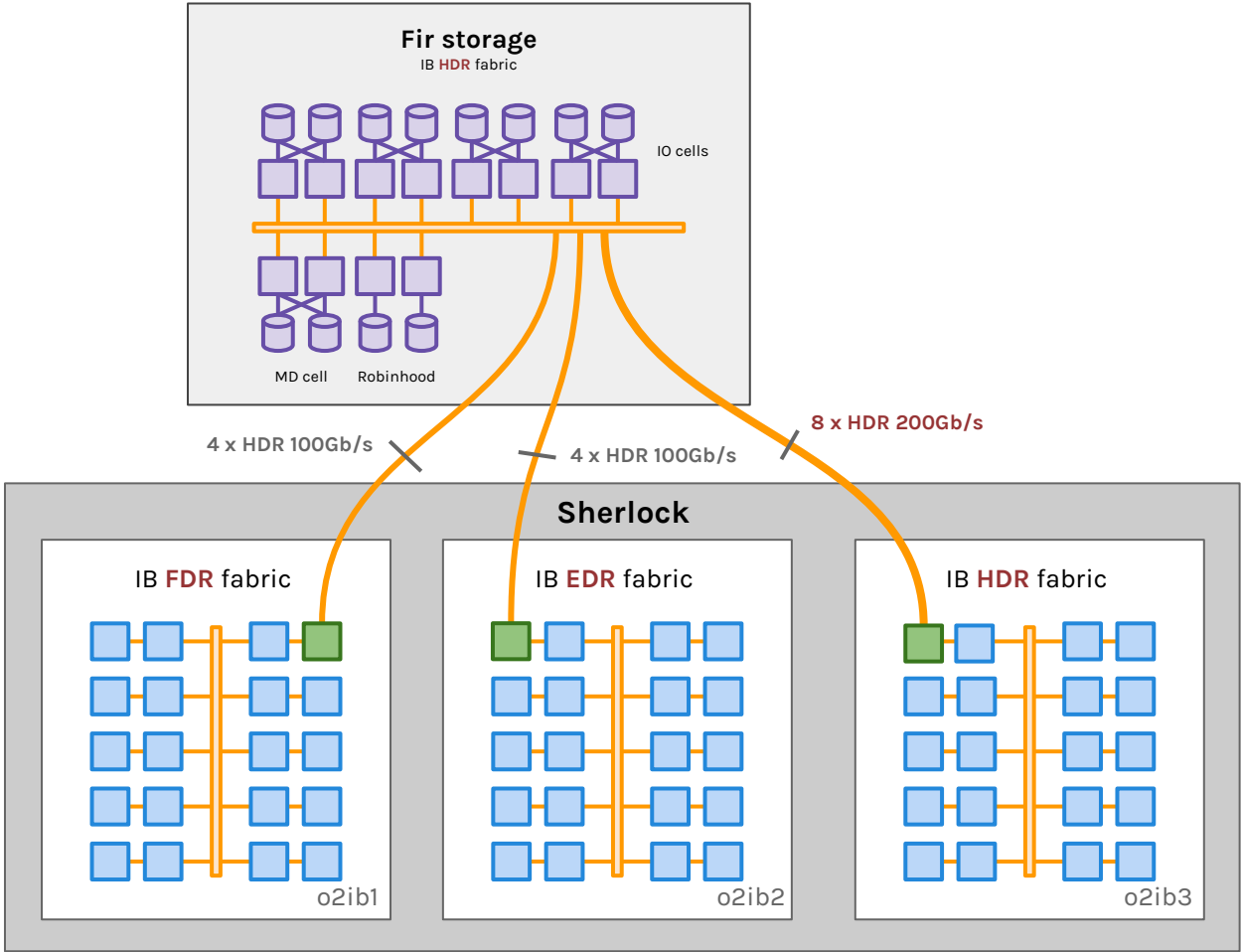
- ▶ **Dec 2019**
  - ▷ **disabled DOM** (by default) and started to **un-DOM-ify**
  - ▷ officially enforced directory quotas with **Lustre project quotas**
- ▶ **Jun 2020**
  - ▷ **Increased OSS RAM** from 256GB to 512GB (8TB total)
  - ▷ successful **backup/reformat/restore** of fir-MDT0003 with a smaller **bytes-per-inode** ratio
- ▶ **Jul 2020**
  - ▷ added second **Robinhood server** (AMD Rome) to keep up with the automatic purge

# Fir storage specs (Sep 2020)

<b>InfiniBand fabric</b>	1 x Mellanox QM8700 <b>HDR switch</b> 40 x HDR 200Gb/s -or- 80 x HDR100 100Gb/s
<b>MD cell</b>	<b>4 x MDS</b> Dell EMC R6415 256GB HDR100 <b>2 x Dell EMC MD3420</b> SSD 36TB usable
<b>IO cells</b>	<b>16 x OSS</b> Dell EMC R6415 512GB HDR100 <b>8 x QCT JBOD</b> 60 x 8TB SAS <b>8 x WD Data60 JBOD</b> 60 x 8TB SAS
<b>Policy engine</b> (Robinhood/MariaDB)	1 x Dell EMC R7425 2x7401 512GB HDR100 SSD 1 x Dell EMC <b>R7515</b> 1x7702P 512GB HDR100 <b>NVMe</b>



# Fir storage network architecture (Sep 2020)



HDR fiber cable used to connect Fir to Sherlock HDR LNet routers

- Compute nodes
- LNET routers
- Cluster Interconnect switches/links
- Storage servers/arrays/links



# Fir storage Data-On-MDT (DOM) issues

- ▶ **Stability issues in early Lustre 2.12.x**
  - ▷ AFAIK, all major DOM issues have now been **resolved by Whamcloud in Lustre 2.12.5**, for example:
    - ▷ **LU-11359** “*racer test 1 times out with client hung in dir\_create.sh, ls, ... and MDS in ldlm\_completion\_ast()*” fixed in **Lustre 2.12.3**
    - ▷ **LU-13416** “*Data corruption during IOR testing with DoM files and hard failover*” fixed in **Lustre 2.12.5**
- ▶ **Free inode issues (*ldiskfs*)**
  - ▷ Formatting MDTs for DOM with a higher bytes-per-inode ratio led to **too few inodes per MDT** and the **DOM space underutilized**
  - ▷ We should have anticipated more **very small files**



# Fir storage Data-On-MDT (DOM) issues

- ▶ **Performance issues**

- ▷ **LU-12935** “MDT deadlock on 2.12.3 with DoM”:
  - ▷ seen with up to **hundreds of writers to DoM region**
  - ▷ **MDS overwhelmed** and became slow to serve other metadata ops
  - ▷ **not enough MDS/MDTs** to sustain/spread the load!
  - ▷ same code using **many** HDD-based OSTs ran just fine

- ▶ **Possible performance improvement?**

- ▷ **LU-10664** “dom: non-blocking enqueue for DOM locks”
- ▷ Review in progress at <https://review.whamcloud.com/#/c/36903/>



## How to **un-DOM-ify** your Lustre?

- ▶ We decided to **avoid the use of DOM on Fir** until we can better understand the different problems associated with this new feature.
- ▶ Our plan to un-DOM-ify Fir:
  - ▷ **disable DOM** by default on all directories (avoid new DOM files)
  - ▷ let *most* old DOM files be automatically **purged**
  - ▷ **restripe** remaining DOM files using OST-only layout (mandatory for next step; see [LU-13691](#) “Allow for lfs migrate between MDTs to include DOM”)
  - ▷ reduce **bytes-per-inode** ratio on all MDTs
  - ▷ keep the possibility of using DOM for special cases (still TBD)





## Fir storage changing bytes-per-inode

- ▶ Migrate files off each MDT to be able to backup/restore quickly
  - ▷ Hit a few issues when using **lfs migrate -m** at scale:
    - ▷ [LU-13492](#) “*lfs migrate -m returns Operation not permitted*” **TBD**
    - ▷ [LU-13511](#) “*ASSERTION( top->loh\_hash.next == ((void \*)0) && top->loh\_hash.pprev == ((void \*)0) ) failed*” **testing patch from WC**
    - ▷ [LU-13599](#) “*LustreError: 30166:0:(service.c:189:ptlrpc\_save\_lock()) ASSERTION( rs->rs\_nlocks < 8 ) failed*” **resolved in Lustre 2.12.6**
- ▶ During a scheduled maintenance, **reformat MDT**
  - ▷ backup/restore at **Backend File System Level** (cf. Lustre Manual)
  - ▷ **reformat** ldiskfs MDT with a smaller **bytes-per-inode** ratio (for us 5120 instead of 65560)



## Fir storage and Project quotas

- ▶ To use project quotas as **directory quotas**, we needed our users to **NOT** be able to change project IDs:
  - ▷ reported in [LU-12826](#) and fixed by Whamcloud in **Lustre 2.12.4**
  - ▷ by default now, only **root** can change the projid of a file
  - ▷ server tunable was also added to control who can change projids:
    - ▷ `mdt.*.enable_chprojid_gid`



# Fir storage and OSS memory

- ▶ In March 2020, we discovered a problematic job:
  - ▷ RELION (cryo-EM) MPI job doing **random I/O read** from **288 ranks on a single 1.9TB file**
  - ▷ even with PFL, our striping didn't allow the file to spread to enough OSS to fit within OSS cache
- ▶ Solutions:
  - ▷ use different **PFL settings**
    - ▷ to ensure that enough OSTs are used to benefit from memory caching of our 16 OSS
  - ▷ **increase memory of OSS** from 256GB to 512GB, bringing the overall OSS RAM from 4TB to 8TB on Fir storage



## Fir storage and the purge policy

- ▶ Fir serves Sherlock's /scratch which is a filesystem for **temporary** files or files that are **actively modified**.
- ▶ How do we implement the purge with **Robinhood**?
  - ▷ Robinhood's **checker** module with a policy (*checkdv*) that uses a custom executable using **liblustreapi** to records all files' **data\_version** and their last **modification time**
  - ▷ files whose **content** has **not** been **modified** for **90 days** are automatically removed from the filesystem
- ▶ How could **Lustre be improved** to help us?
  - ▷ **LU-13951** to get the last time `data_version` was modified



# SRCC Lustre roadmap

# SRCC Lustre roadmap

- ▶ **Fir storage**
  - ▷ Perform remaining **MDT-to-MDT file migrations** and **reformat MDTs** to reduce the **bytes-per-inode** ratio
- ▶ **Oak storage**
  - ▷ **Upgrade** Oak servers from Lustre 2.10 to **Lustre 2.12 LTS**
  - ▷ Enable **project quotas** on Oak
    - ▷ enforced as **directory quotas** like on Fir storage
    - ▷ mitigate [LU-13172](#) (nodemap/squashed GID/quota on nobody)
  - ▷ Evaluate **Lustre NRS** with TBF per UID/GID on Oak (2.12+)

# THANKS!

Any questions?

[sthiell@stanford.edu](mailto:sthiell@stanford.edu)

<https://github.com/stanford-rc>



Stanford  
University