Lustre Filesystem - IPSL Feedback

ARGOS 16/06/2026

Philippe Dos Santos (IPSL)
Philippe Weill (LATMOS/IPSL)

Index

- Introduction
- Lustre Overview
- IPSL Lustre Servers
- Lustre File System
- Conclusion

Index

Introduction
Institut Pierre-Simon Laplace (IPSL)

- Founded in the mid-1990s by Gérard MÉGIE
- First FR CNRS and now UAR CNRS, Sorbonne Université, UVSQ

- 8 laboratories in the Paris region (CEREA, GEOPS, LATMOS, LISA, LMD, LOCEAN, LSCE, METIS) + several associated research teams in other labs
- Nearly 1,400 people

- Research Area: Ocean/Atmosphere (Terrestrial and Planetary) Climate
- Ground/Balloon/Aircraft/Satellite Observations and Modeling
- Data Distribution / Post-Processing

Introduction
Distributed Parallel File System

- Lustre : https://www.lustre.org/
- GlusterFS : https://www.gluster.org/
- BeeGFS : https://www.beegfs.io/c/enterprise-features/
- GPFS (IBM Spectrum Scale)

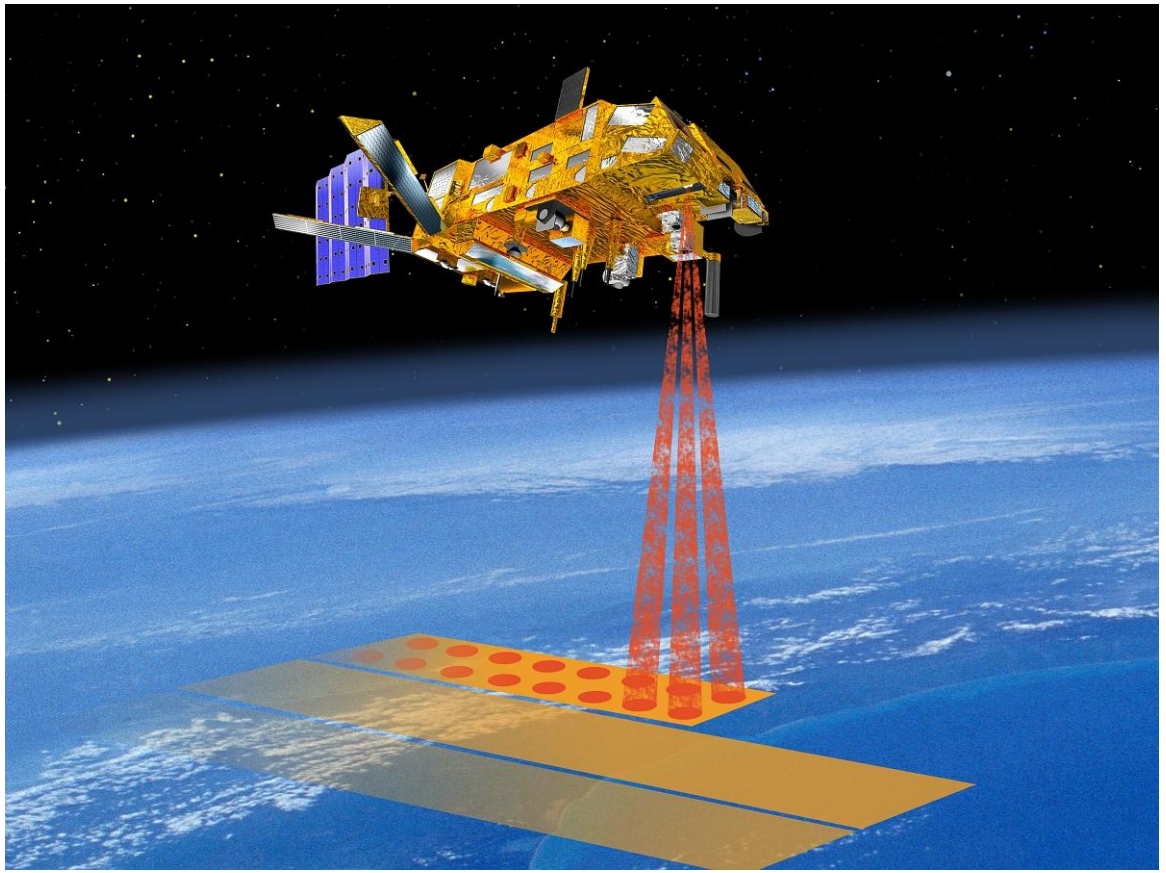
- CephFS : https://docs.ceph.com/en/reef/cephfs/
- pNFS (parallel NFS)
- Scality Ring: https://www.scality.com/fr/
- ...

Introduction
Needs and Initial Request

- IASI Data in 2007
  - Satellite launch in 10/2006
  - 100 TB of GRIB data
  - 175 TB as of today in 2026

- Early Lustre Prototype in 2007 not stable
(related to storage controller bad cache)
  - Hardware provided by transtec (120k€)
  - Infortrend storage (2x 50TB)
  - Supermicro compute
  - Lustre 1.6 then 1.8

- Production in 09/2008



Introduction
Lustre Testimony

- Mesocentre i.e. cluster
- No scale up (< 50 compute nodes per cluster)
- LUSTRE in production since 09/2008 (release 1.8 + Ethernet 1Gbps)
- LUSTRE in production since 06/2012 (release 1.8 + InfiniBand 40Gbps)
- LUSTRE release 1.8, 2.5, 2.10 and 2.15 : upgrade by data copy to new storage enclosure

- Currently Lustre Long Term Support (LTS) 2.15.8
- Older Lustre Release 2.12 LTS, 2.10 LTS

(2.12 LTS scale up > 10 000 compute nodes)

- In High Performance Computing (HPC) three common file systems
  - Network File System (NFS)

  (bottleneck and SPOF in large clusters)
  - Distributed Parallel File System (GPFS, Lustre, ...)

  (Aggregate throughput to scale up with more servers i.e. 100GB/sec ~ 10x 100Gb/sec)

Index

Lustre Overview

Key points

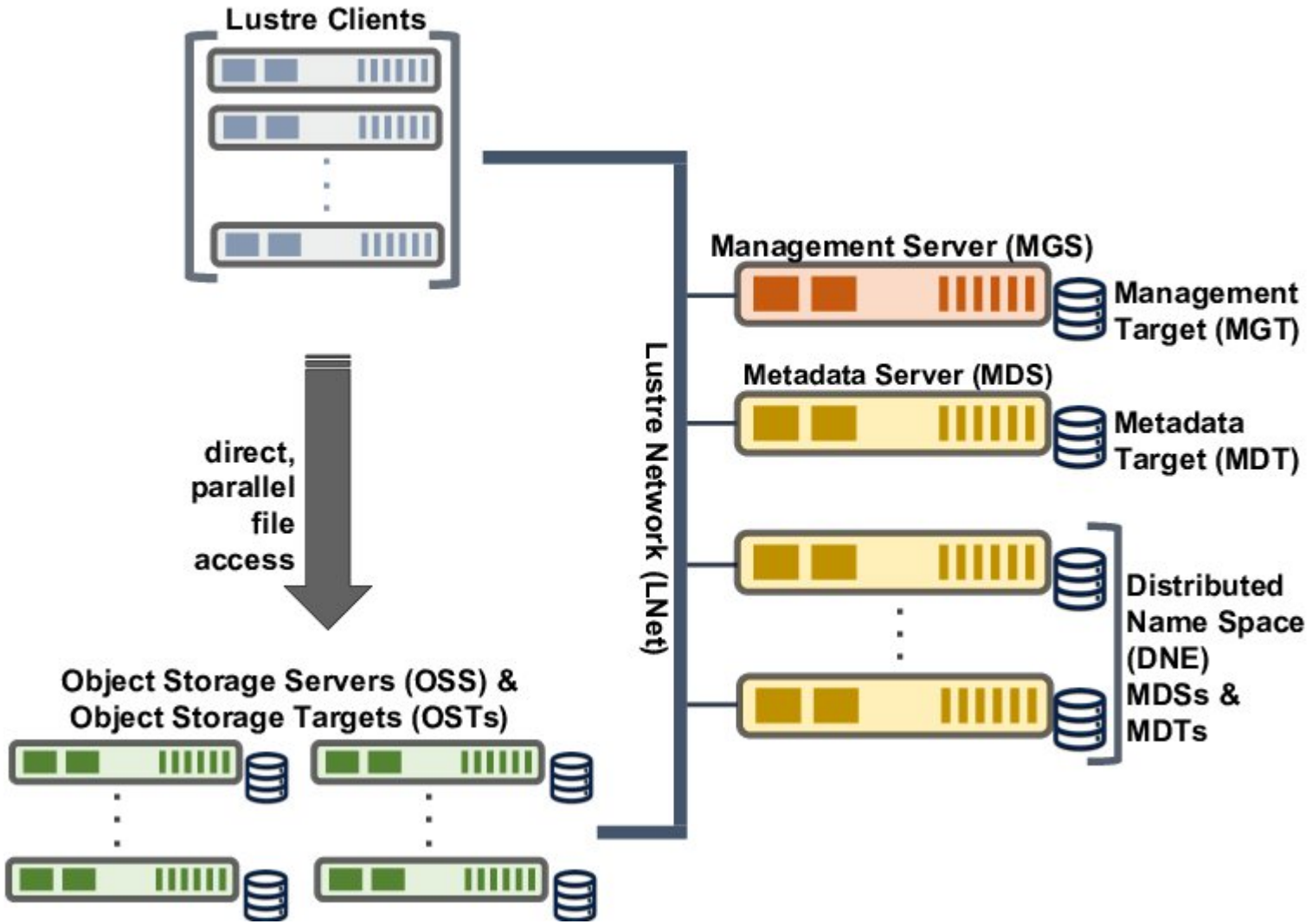
- Separate Metadata and Data
- POSIX
- Kernel space

Prerequisites

- Rock solid network
- Bandwidth > 10Gb/sec
- RDMA is better

- Reliable underlying storage (RAID)

- Not for workstations !
- Not for millions small files



Lustre Overview
Lustre Network (LNET)

Network Abstraction Layer (lnet.ko)

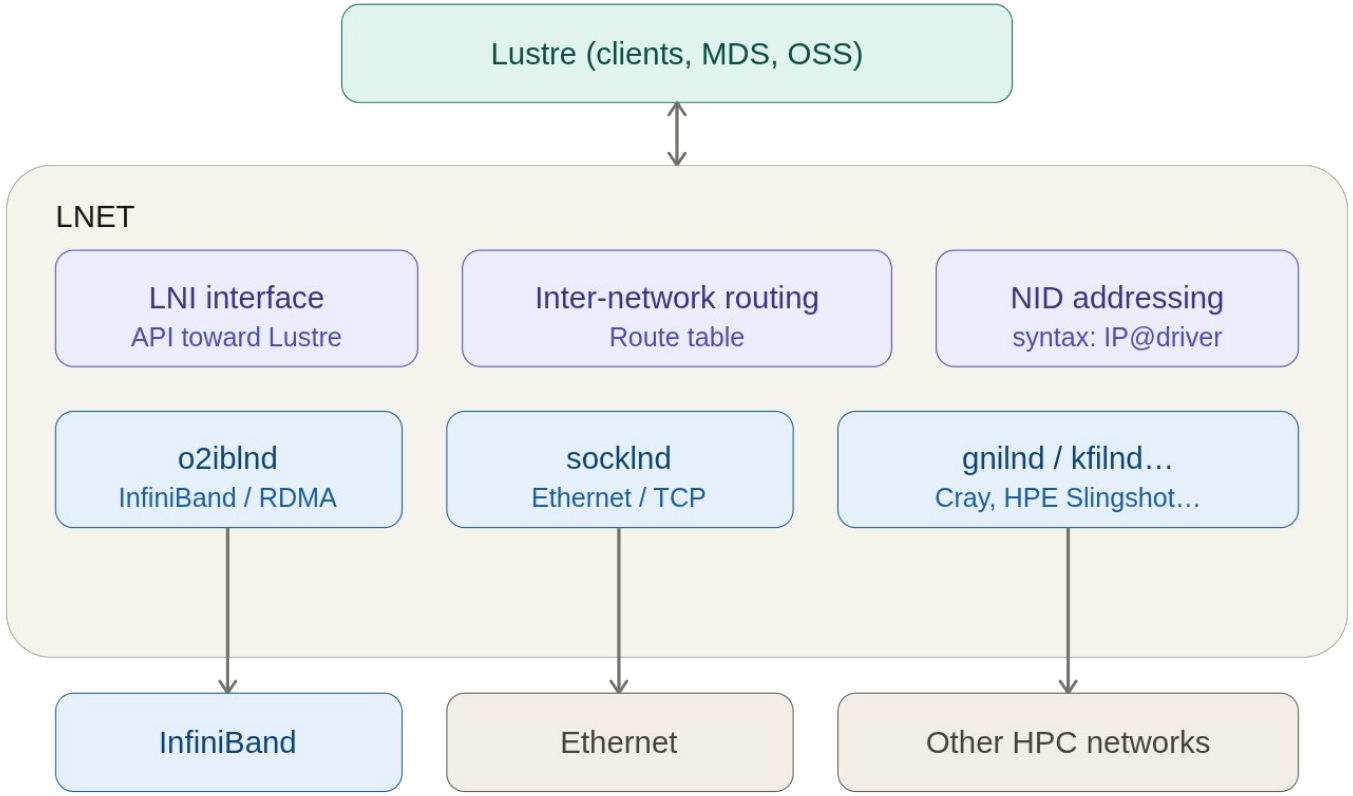
- **Lustre Network Interface (LNI)**
- **Network Identifier (NID)**

<adresse>@<driver> :

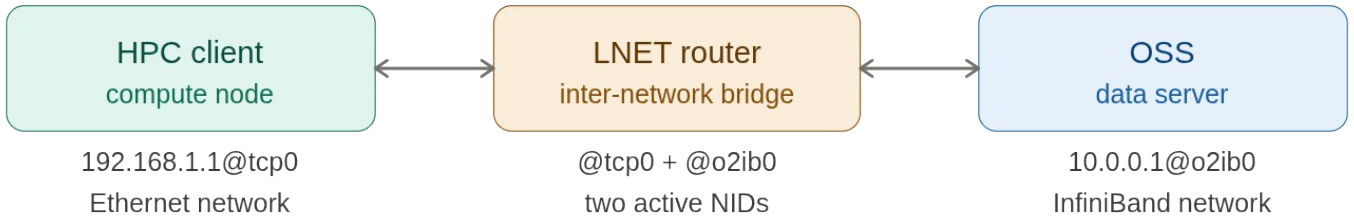
172.100.0.10@o2ib0 (o2iblnd for InfiniBand)
192.168.1.10@tcp0 (socklnd for Ethernet)

- **LNET Router**

Routing between different networks
(InfiniBand, Omnipath, BXI, Ethernet, ...)



Network layers



Driver	Network	Transfert
o2ib	**InfiniBand** (EDR, HDR, NDR)	**RDMA, no CPU load, latency** < 1us
socklnd	**Ethernet 10/25/100 GbE**	**Socket TCP, moins performant**
kfilnd	HPE **Slingshot**	**RDMA** sur **fabric Intel/HPE**
...	...	**...**

Lustre Network (LNET)

Management & Troubleshooting

- Ensure lnet is up and running
- Check lnet.conf
  - net (net type, nid)
  - route (net, gateway)
- lnetctl tool
  - LNET interactive configuration
  - LNET debugging

 # Check lnet service

```
systemctl status lnet
● lnet.service - lnet management
   Loaded: loaded (/lib/systemd/system/lnet.service; enabled; vendor preset: enabled)
```

 # Check config file

```
cat /etc/lnet.conf
net:
  - net type: o2ib
    local NI(s):
      - nid:  10.0.3.165@o2ib
        interfaces:
            0: ibs3
route:
  - net: tcp3
    gateway: 10.0.3.253@o2ib
```

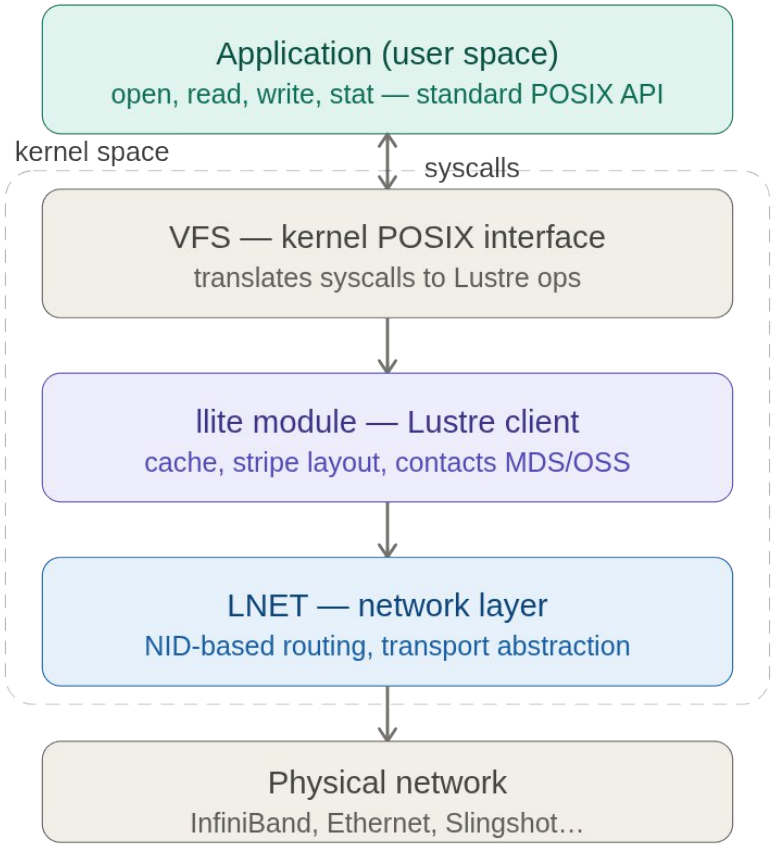
 # Check connectivity

```
lnetctl ping 10.0.3.253@o2ib
ping:
  - primary nid: 10.0.3.253@o2ib
    Multi-Rail: False
    peer ni:
      - nid: 10.0.3.253@o2ib
```

Lustre Overview
Client Node

Dynamic Kernel Module Support (DKMS)

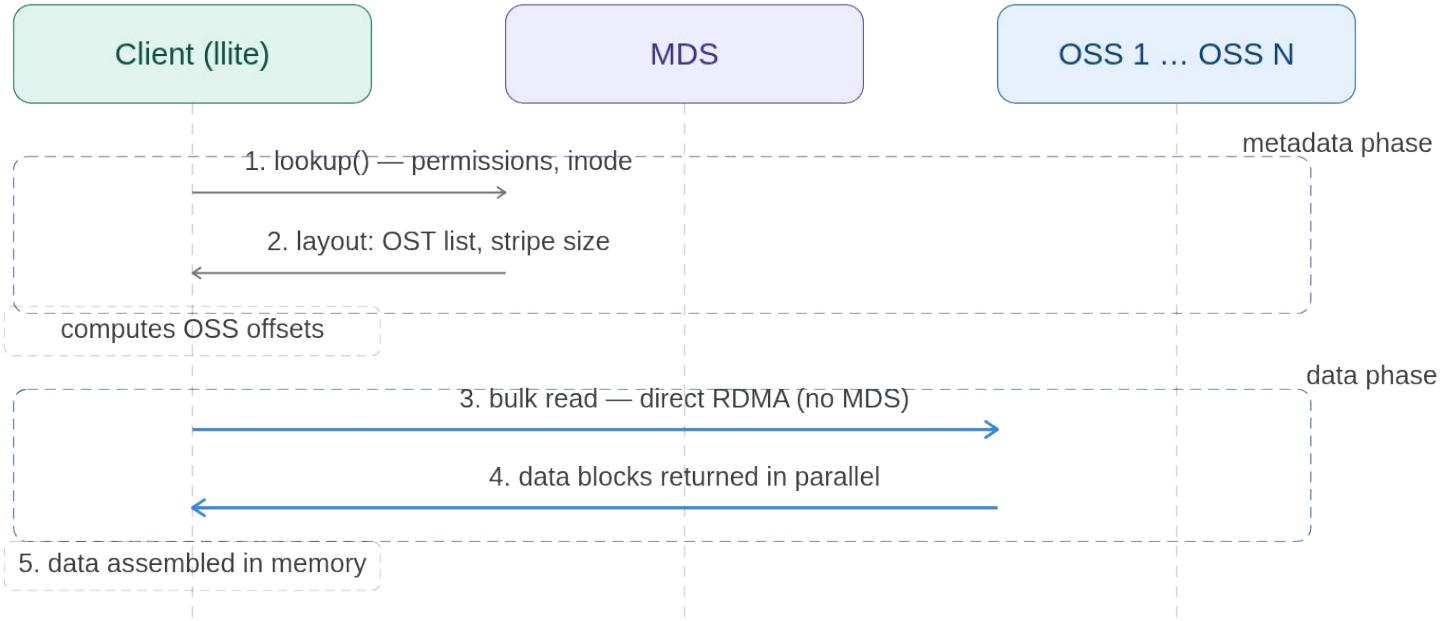
- In /lib/modules/$(uname -r)/updates/dkms/
- POSIX i.e. transparent for the users
- Lustre module registers to kernel Virtual File System (VFS)
- Lustre Lite Client (llite)
  - Contacts Lustre servers (MDS/OSS)
  - Gets file permissions and inode
  - Gets file layout (OSTs, stripe size, OSSs NIDs)
  - Keeps local cache



Lustre Overview
Client Node

What happens for open() and read() syscalls

- **Metadata Phase (MDS - open() syscall)**
  - **Client contacts MDS Lustre server**
  - **Gets file permissions, inode and layout**
  - **Keeps local cache**
- **Data Phase (OSS – read() syscall)**
  - **Client computes stripe location from layout**
  - **Connects directly to OSSs in parallel**
  - **InfiniBand RDMA (from OSS memory into client memory)**



Lustre Overview
Client Node

Client side

- **POSIX semantics for I/O access**
- **Behaves like a local file system**

- **Available on all compute nodes**

- **Data request = MDS/MDT (Low latency)**
- **Data flow = OSS/OST (High bandwidth)**

 # Load Lustre kernel module (DKMS Lustre)

```
modprobe lustre
```

 # Mount Lustre file system

```
mount -t lustre 10.0.0.1@o2ib:/lustre /mnt/lustre
```

 # Check Lustre disk space

```
lfs df /mnt/lustre
```

 # Check stripe parameter

```
lfs getstripe /mnt/lustre/my_file.dat
```

 # Set stripe count to 4 for big_file.dat

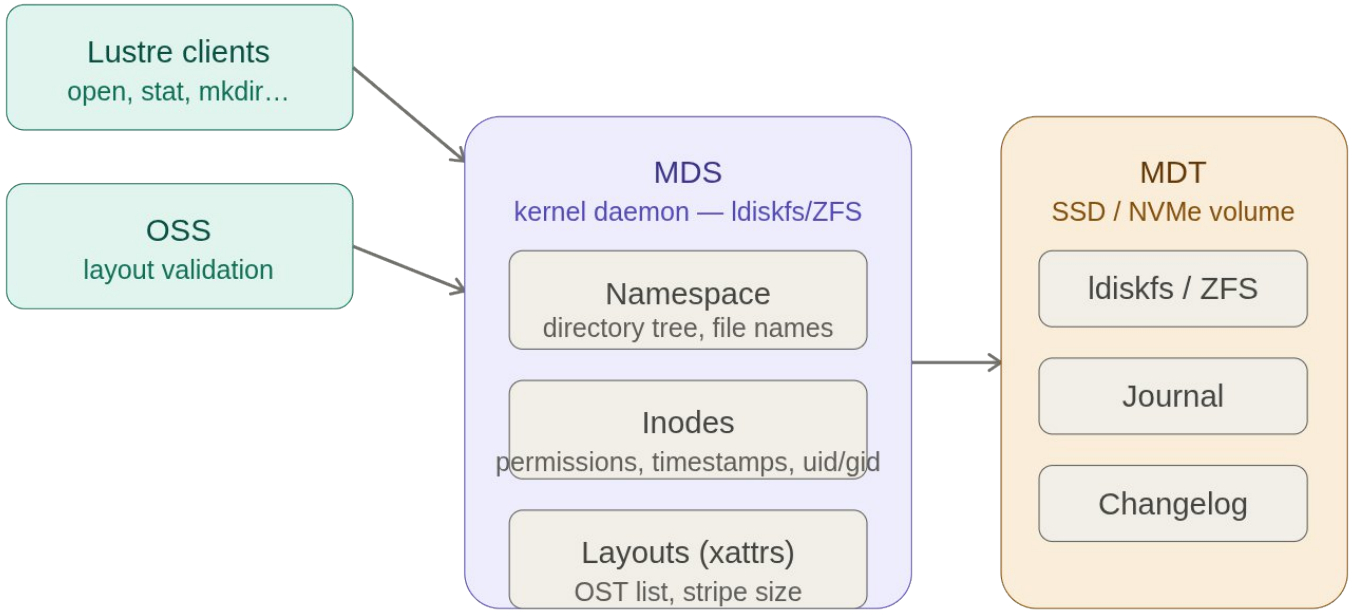
```
lfs setstripe -c 4 /mnt/lustre/big_file.dat
```

Lustre Overview
Metadata Service (MDS)

Patched kernel
(Lustre Extended Attributes (EA))

- **Metadata Server (MDS)**
  - **Kernel module (mds)**
  - **Manage requests**
  - **Quota enforcement**

- **Metadata Target (MDT)**
  - **Kernel service threads [mdt]**
  - **Storage volume**
  - **ldiskfs (ext4 based) or ZFS formatted**
  - **SSD or NVMe**
    (potential bottleneck check latency)



The MDS runs in kernel space on the server
The MDT is its dedicated storage volume

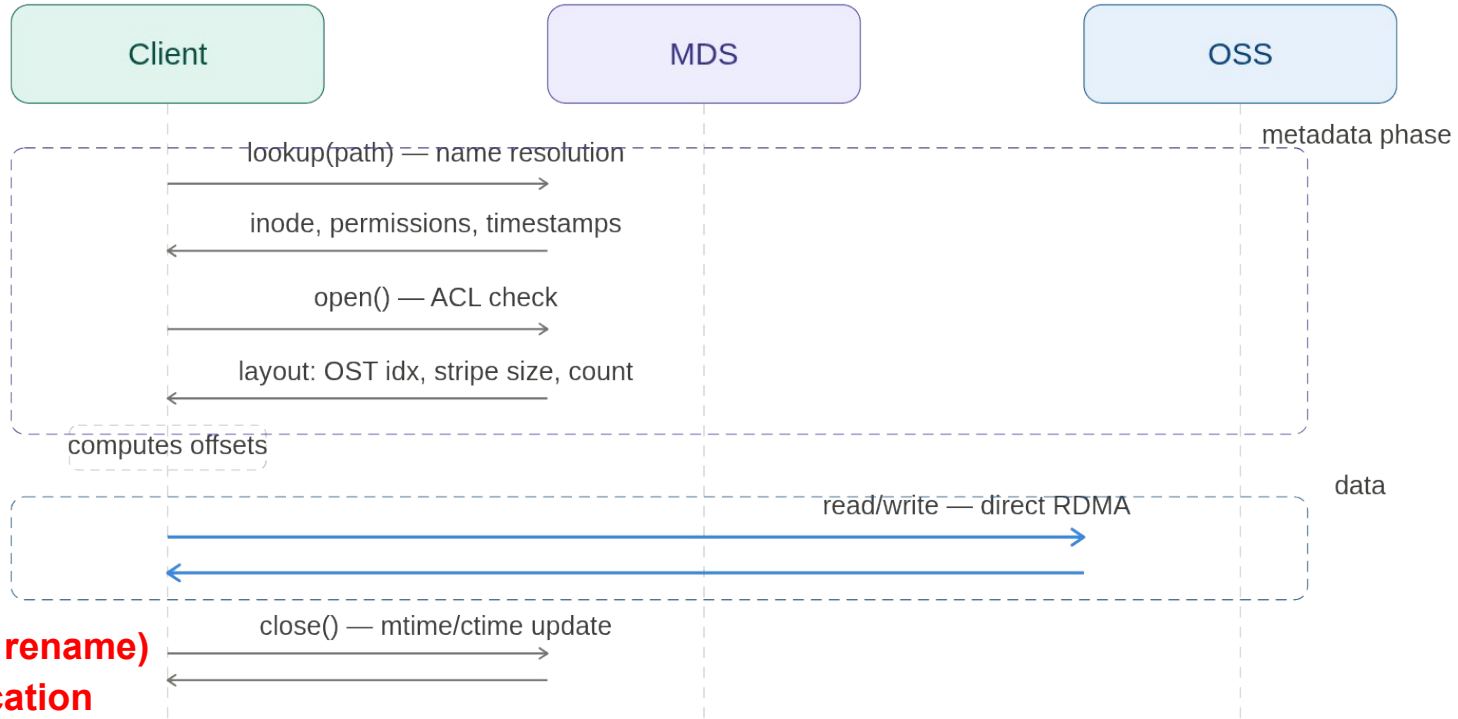
Lustre Overview
Metadata Service (MDS)

Metadata Server (MDS) operations

- **Namespace**
  (Directories)
- **File attributes**
  (open, stat, chmod, chown, mkdir, rename, unlink, symlink, getxattr)

Metadata Target (MDT)

- **File/Dir = inode on storage**
- **POSIX attributes**
  (uid, gid, mode, timestamps, size)
- **Lustre Extended Attributes (EA)**
  (related OSTs, stripe size, count)
- **Changelog**
  - **stores metadata operations (create, delete, rename)**
  - **Useful for production audit, HSM and replication**



Lustre Overview
Metadata Service (MDS)

Server side

- **POSIX**
- **Behaves like a local file system**

- **Available on all compute nodes**

- **Data request = MDS/MDT (Low latency)**
- **Data flow = OSS/OST (High bandwidth)**

 # Check metadata ops

lctl get_param mdt.*.md_stats

 # Check mds ops latency

lctl get_param mdt.*.exports.*.stats

 # Check used space on MDT (inodes !)

lfs df -i /mnt/lustre

 # Set inode quota enforcement for all users, groups, and projects

lctl set_param -P osd-*.fsname-*.quota_slave_md.enabled=ugp

 # Set data quota enforcement for all users, groups, and projects

lctl set_param -P osd-*.fsname-*.quota_slave_dt.enabled=ugp

 # Add MTIME and CTIME flags in changelog

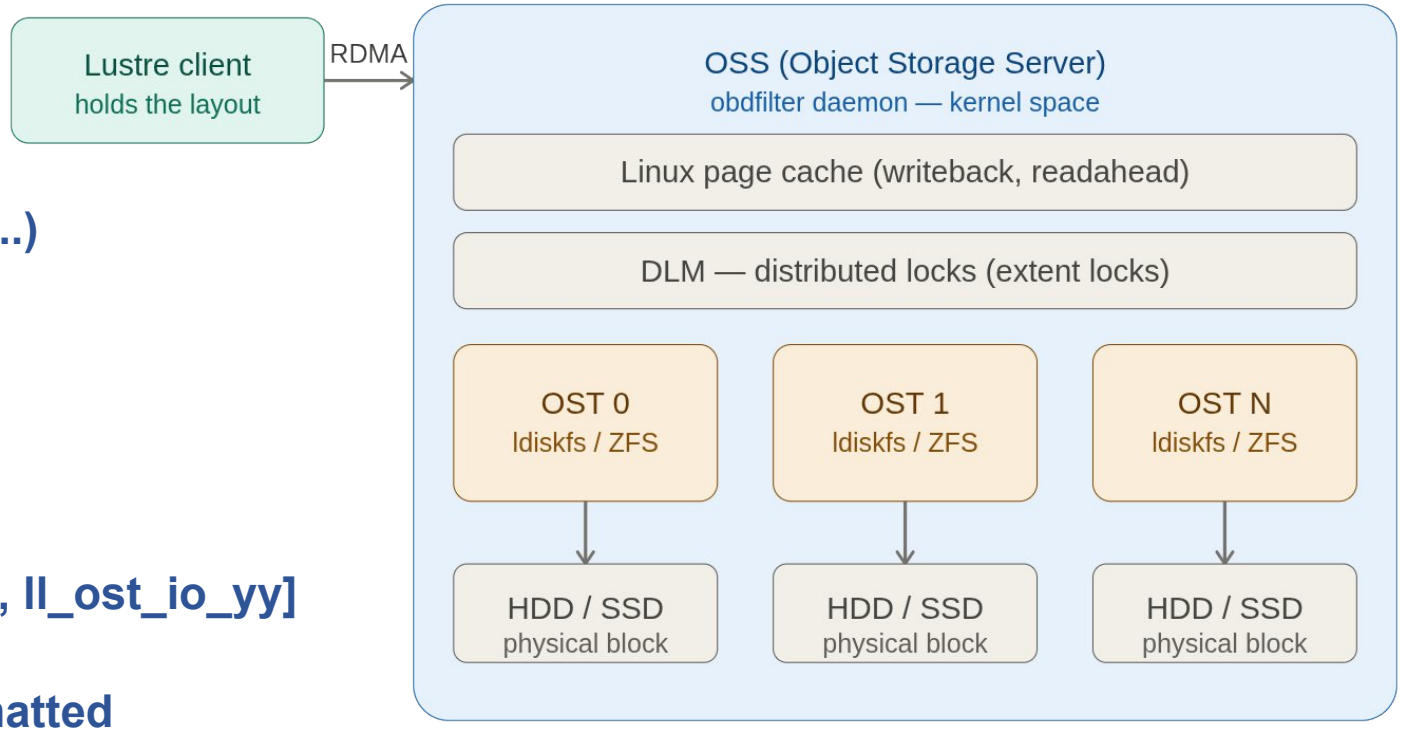
lctl set_param -P mdd.*-MDT0000.changelog_mask='+MARK +MTIME +CTIME'

Lustre Overview
Object Storage Service (OSS)

Patched kernel

- **Obect Storage Server (OSS)**
  - **Kernel modules (obdclass, ost, ...)**
  - **Manage requests**
  - **One OSS manages 4 to 8 OSTs**
  - **Quota enforcement**

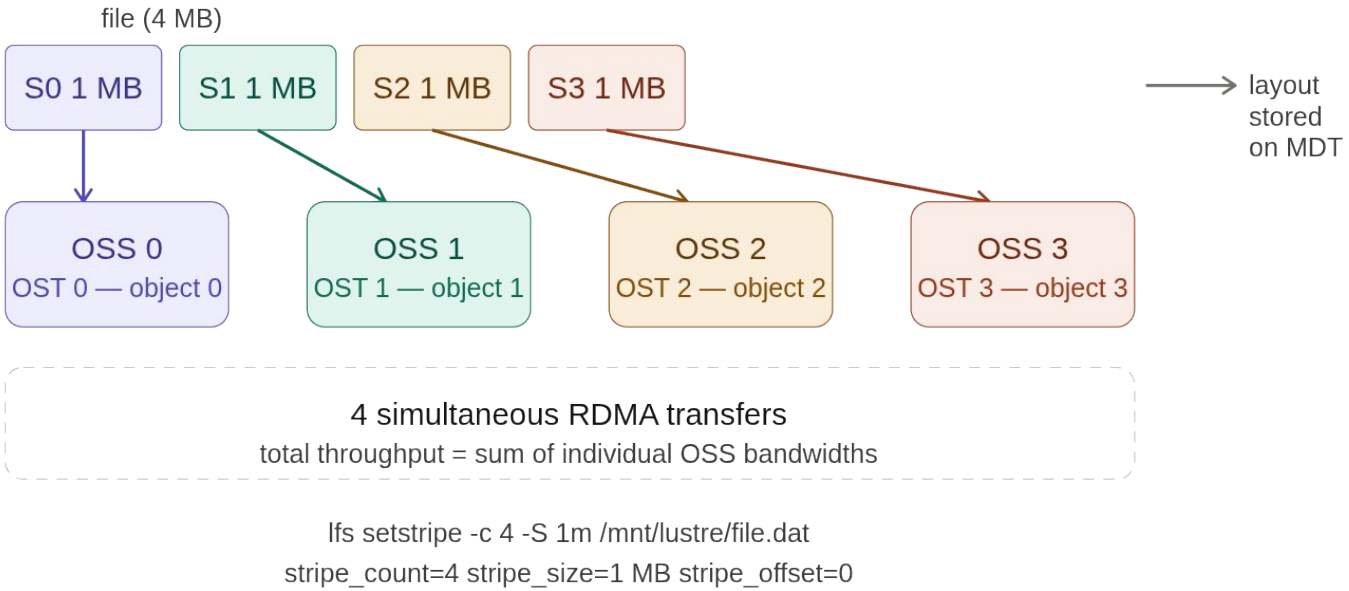
- **Object Storage Target (OST)**
  - **Kernel service threads [ll_ostxx, ll_ost_io_yy]**
  - **Storage volume**
  - **ldiskfs (ext4 based) or ZFS formatted**
  - **HDD or SSD**



Lustre Overview
Object Storage Service (OSS)

Lustre vs traditional NFS

- Files not stored on a single server
- Files divided into stripes
- Stripes spread across multiple OSTs
- OSTs often hosted on multiple OSSs
- **Total throughput = sum of OSSs bandwidths**

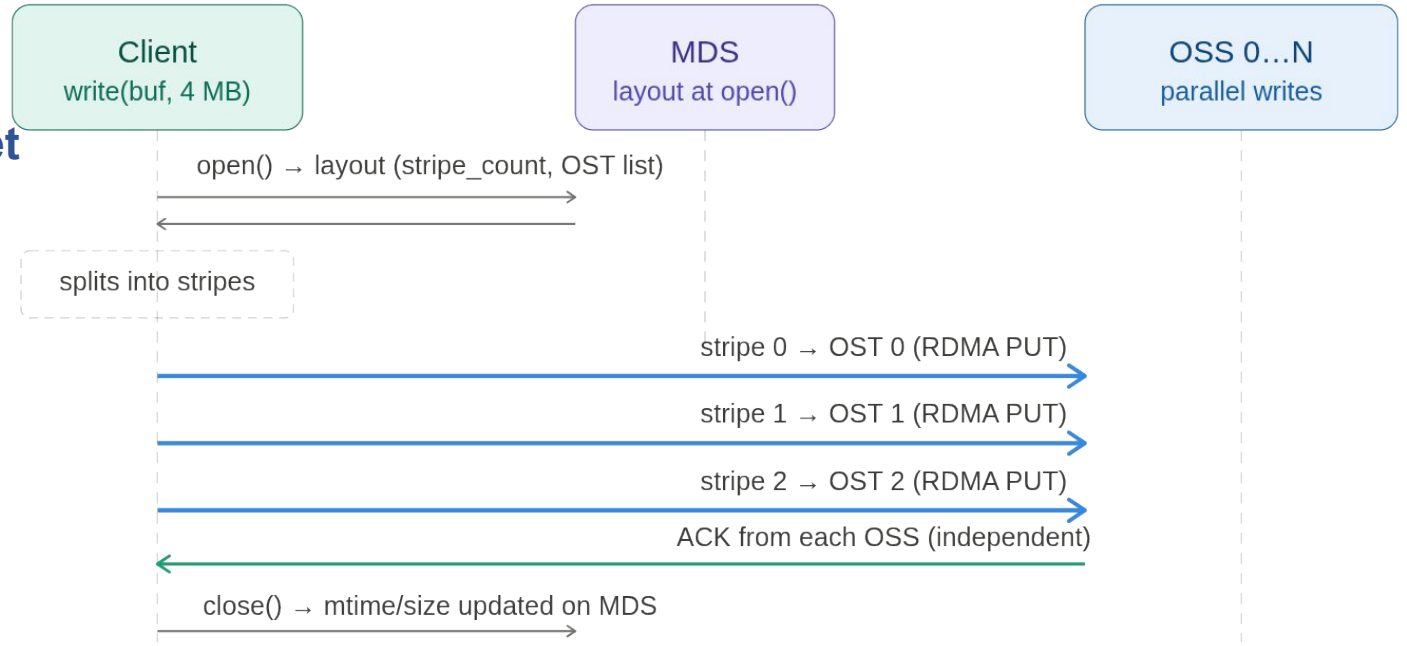


Lustre performance = striping

- MDS assigns layout on file creation
- Layout = stripe_count, stripe_size, offset
- Written on inode xattrs on MDT

- OSS don’t know the layout
- OSS only deal with IO
- IO aggregate flow (independent OSSs)
  - stripe_count x per_OSS_throughput
  - 8x 2GB/sec OSSs = 16 GB/sec total

- MDS updates mtime/size on close



Lustre Overview
Object Storage Service (OSS)

OSS side

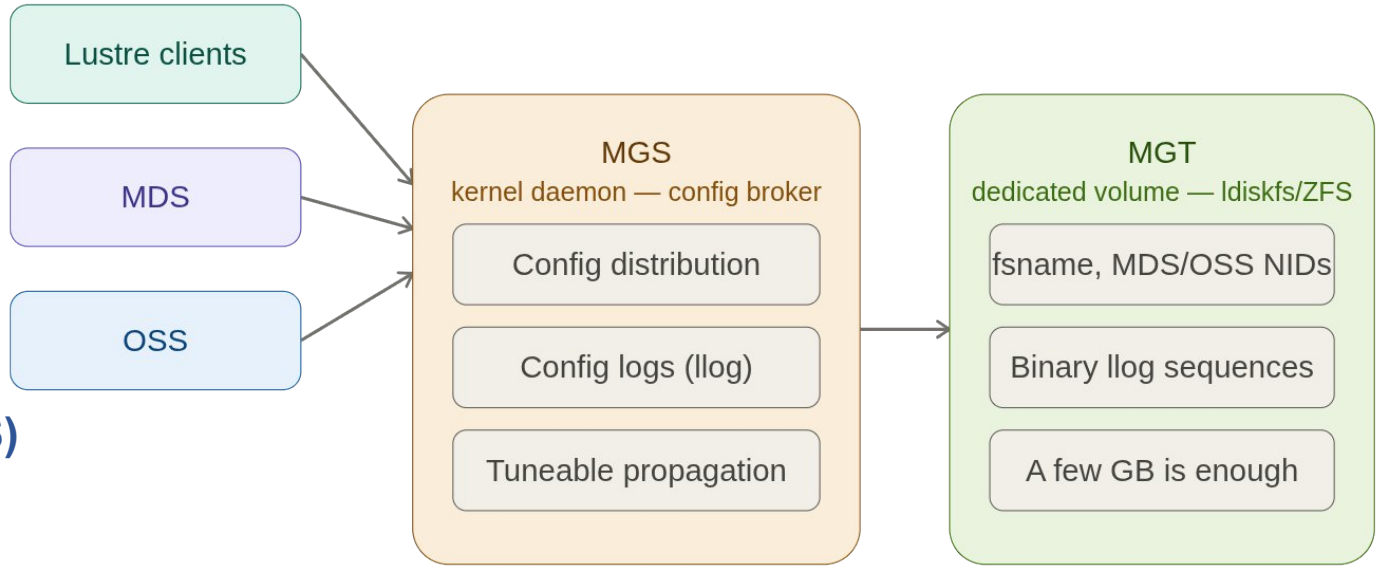
- Just serves numbered blocks
- Does not know
  - File names
  - Directories
  - Access rights
- Integrity with Distributed Lock Manager (DLM)
- Really good at moving blocks as fast as network and disks can handle

```
# Get file layout
lfs getstripe /mnt/lustre/fichier.dat

# Create file with stripe_count=8 and stripe_size=4 Mo
lfs setstripe -c 8 -S 4m /mnt/lustre/gros_fichier.dat

# Stripe on all available OSTs (-1 = auto)
lfs setstripe -c -1 /mnt/lustre/checkpoint/

# Check OST disk usage
lfs df /mnt/lustre

# I/O stats per OST (throughput, latency, RPCs)
## OSS side
lctl get_param obdfilter.*.stats
## Client side
lctl get_param osc.*.stats        # côté client
```

Lustre Overview
Management Service (MGS)

Patched kernel

- **Management Server (MGS)**
  - Distributes cluster configuration to all components (client, MDS, OSS)
  - Neither data nor metadata
- **Management Target (MGT)**
  - Dedicated storage for MGS
  - ldiskfs (ext4 based) or ZFS formatted
  - A few gigabytes are sufficient
  - Only stores configuration sequences called llogs (Lustre logs)



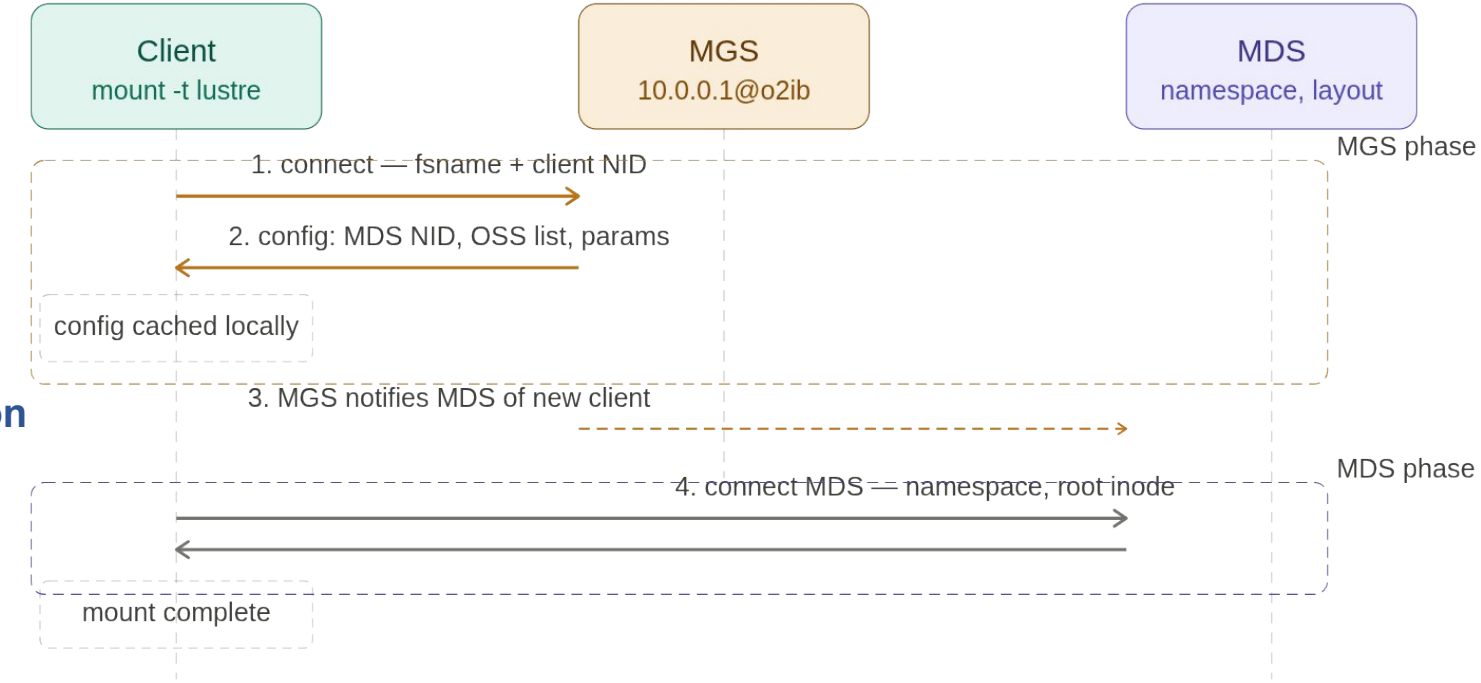
The MGS can be co-located with the MDS on the same physical server but remains a separate daemon and volume

Lustre Overview
Management Service (MGS)

1. **Wen a Client mounts a Lustre file system**
   - **MGS NID**
   - **File system**

```
mount -t lustre 10.0.0.1@o2ib:/lustre /mnt/lustre
#                ^^^^^^^^^^^^^ NID du MGS
#                              ^^^^^^ fsname
```

2. **MGS returns complete cluster configuration**
   - **MDS NID**
   - **All OSSs NIDs**
   - **Tuning parameters**
     - **Inet**
     - **...**
   - **Important property**
     - **One can add or replace an OSS**
     - **Client aware after next mount**



Lustre Overview
Management Service (MGS)

What does MGT stores ?

- Only stores Lustre logs (llogs)
- One can modify and propagate some params

MGS and MDS can share same server

- Does not use a lot of resources

What if MGS stops ?

- MGS is not a SPOF
- Clients and servers already mounted continue to work
- No more new mounts
- No cluster configuration change

 # List all config logs

lctl --device MGS llog_catlist

 # Display the log content (for instance: fs "lustre" log client)

lctl --device MGS llog_print lustre-client

 # Modify and propagate a parameter to all clients

lctl set_param -P osc.*.max_rpcs_in_flight=16

Lustre Recap
The Right Way

- MGS (Management Service) and 1 MGT (Management Target)
- MDS (Metadata Service) and 1+ MDT (Metadata Target)
- MGS and MDS can share the same server

- OSS (Object Storage Server) 1+ OST

- Client Nodes

- Lustre routers (only when needed)

Lustre Recap
Configuration and Start Order

- LNET
- MGS
- MDS
- OSS
- Client Node

```
# On MDS/MGS server start LNET
systemctl start lnet

# On MDS/MGS server start mgs (mount mgt)
mount -t lustre /dev/mapper/VGMDT-mgt /mnt/mgt/

# On MDS/MGS server start mds (mount mdt)
mount -t lustre /dev/mapper/VGMDT-proj1x_mdt0 /mnt/proj1x/mdt0/

# On OSS server start oss (mount ost)
mount -t lustre /dev/mapper/proj1x-ost-00 /mnt/proj1x/ost00/
...
# On client mount Lustre (manually or in /etc/fstab)
mount –t lustre 172.17.3.168@o2ib:/proj1x /mnt/proj1x

# Check on client
 lfs df -h /mnt/proj1x
UUID                    bytes        Used   Available Use% Mounted on
proj1x-MDT0000_UUID      353.1G       16.4G      306.6G   6% /mnt/proj1x[MDT:0]
proj1x-OST0000_UUID       93.6T       70.6T       22.2T  77% /mnt/proj1x[OST:0]
...
filesystem_summary:        1.1P      842.9T      269.5T  76% /mnt/proj1x
```

Index

- Introduction
- Lustre Overview
- IPSL Lustre Servers
  - Architectural choices (Keep It Super SImple)
  - 1x MDS/MGS Server
  - 2x OSS Servers per Filesystem
  - 1x Disk Enclosure
- Lustre File System
- Conclusion

IPSL Lustre Servers

- InfiniBand HDR Network (200Gbps)
- 2 MDS InfiniBand EDR (100Gbps) (MGS on MDS server)
- 30 OSS InfiniBand EDR (100Gbps)
- 4 Lustre Router (InfiniBand – Ethernet)

- 15 Dell Powervault MD3860/ME4084/ME5084/ME5284
  - Fiber Channel (FC) 16/32G
  - Servers Direct Attached (No FC switch)
- 15 Lustre File Systems from 300TB to 1.4PB

- Global storage capacity 11PB

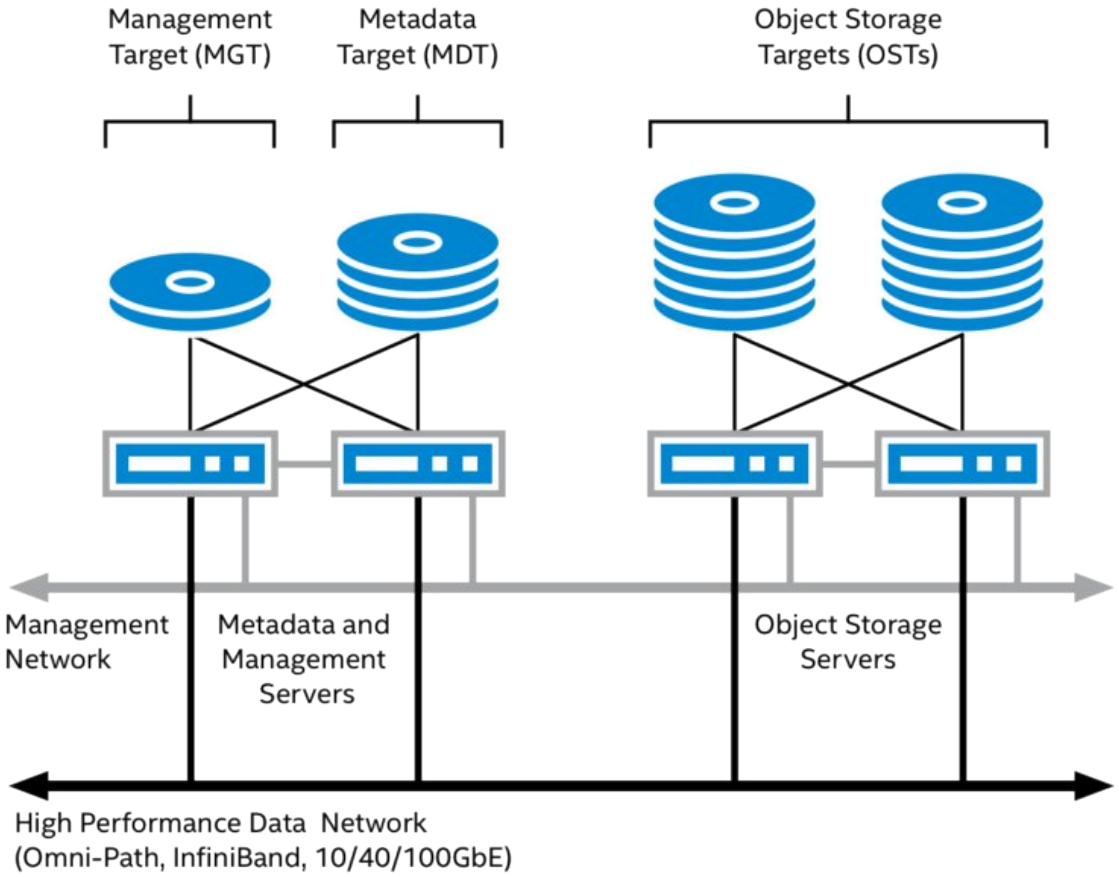
IPSL Lustre Servers Architectural Choices

Lustre is not fault tolerant
(Safeguard: High Availability (HA),RAID and backup)

- **Remember**
  - **Lustre is not Fault Tolerant (for now)**
  - **Lustre is aimed at performance**

- **MGT and MDT on different storage**
  - **Resiliant storage**
  - **MDS on HA servers**
  - **MGS does need a lot of ressources (VM)**

- **OST on dedicated hardware (DDN, PowerVault, ...)**
  - **Resiliant storage**
  - **OSS on HA servers**
  - **Striping among OSTs for performance**



IPSL Lustre Servers
1x MGS/MDS server KISS

1x HP ProLiant DL325 Gen10 Plus v2

(Remember, Lustre is not fault tolerant)

- **2 SSD SATA 480GB disks**

(RAID1 for OS)

- **4 SSD NVME 3.2To disks**

(RAID10 for MDT/MGT)

- **2 Power Supply**

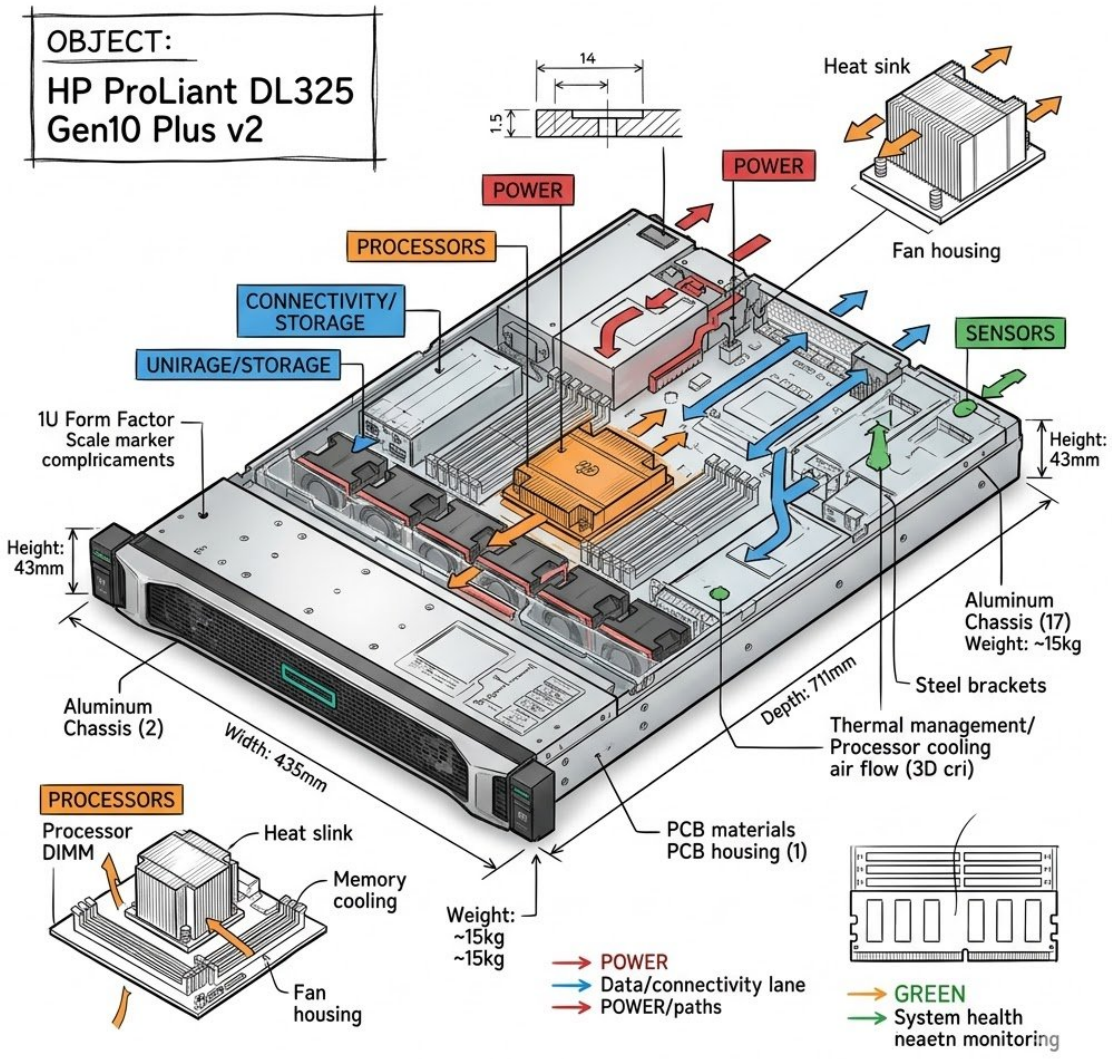
- **1 CPU / 128GB RAM**
- **InfiniBand HDR 200Gbps**
- **Rocky Linux 8.10**

(Lustre patched kernel)

- **Metadata backup**

(LVM snapshot + dd)

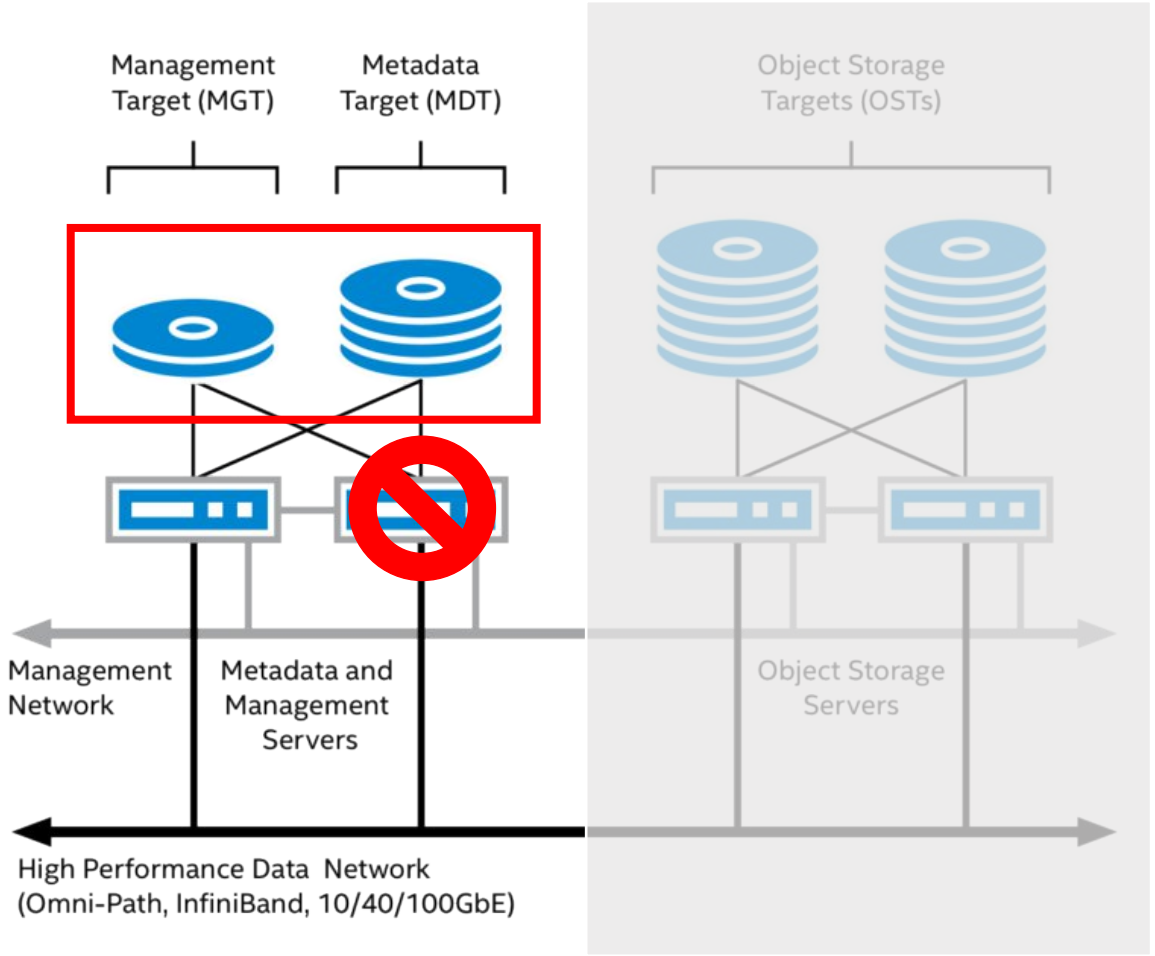
- **8 File systems for 5.4 Po**



IPSL Lustre Servers In Case of Failure

Lustre is not fault tolerant
(Safeguard: High Availability (HA), RAID and backup)

- If the MGT breaks down
  - No data loss
  - But Lustre config cannot be changed
- If the MDT breaks down
  - Data not accessible
  - If no backup, data is lost ...
- MDS/MDT and MGS/MGT
  - Share the same server
  - Backup once per day via cron
  - Keep It Super Simple (KISS) = no HA



IPSL Lustre Servers
2x OSS servers KISS

2x Dell PowerEdge R660

(Remember, Lustre is not fault tolerant)

- 2 SSD SATA 480GB disks

(RAID1 for OS)

- 2 Power Supply

- 2 CPU / 128GB RAM
- InfiniBand HDR 200Gbps
- Fiber Channel 2x 32Gbps

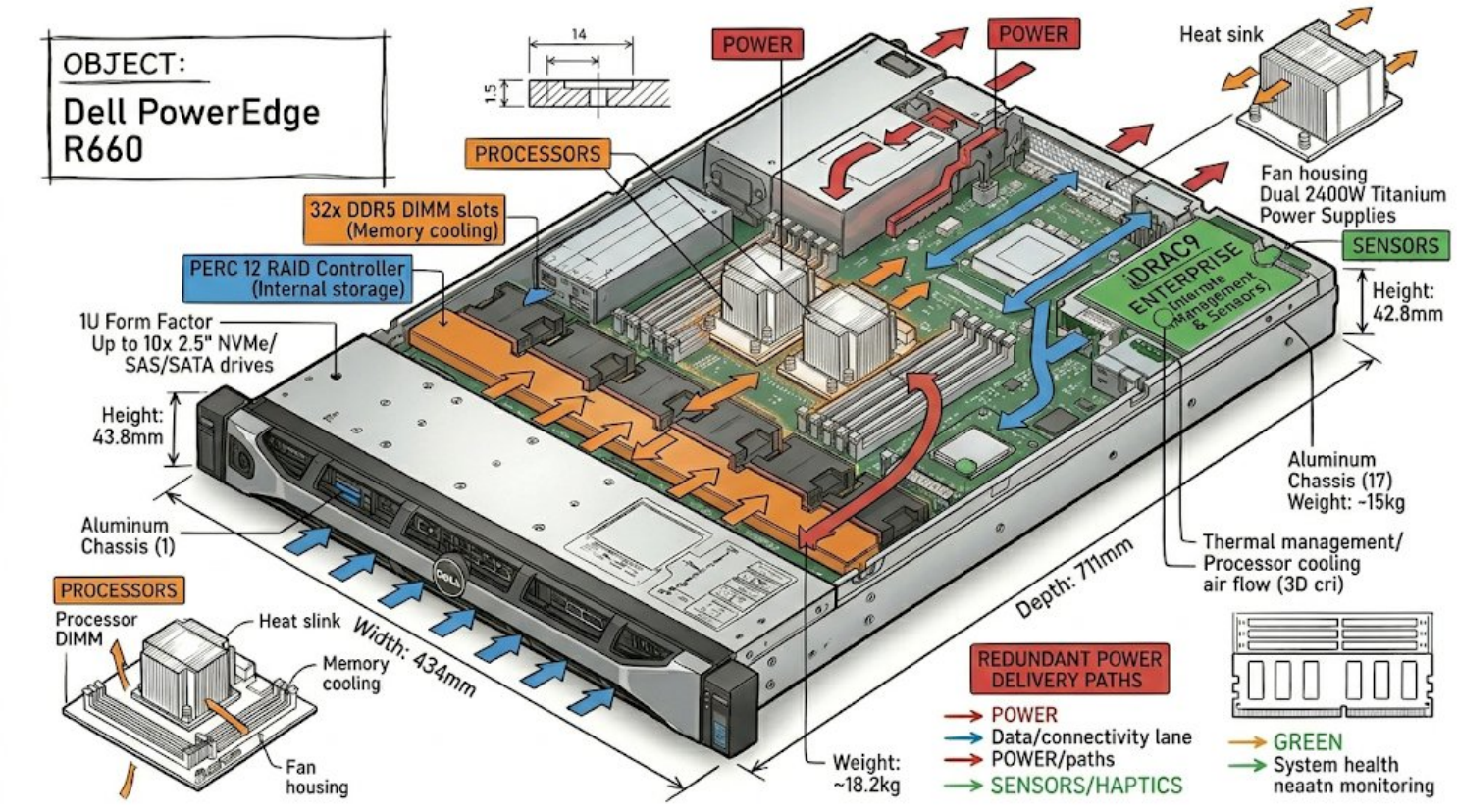
(Direct Attached and multipath)

- Rocky 8.10

(Lustre patched kernel)

- 2 servers
  - Each mounting half of the OSTs
  - Manual fallback on 1 server

- 1 File System

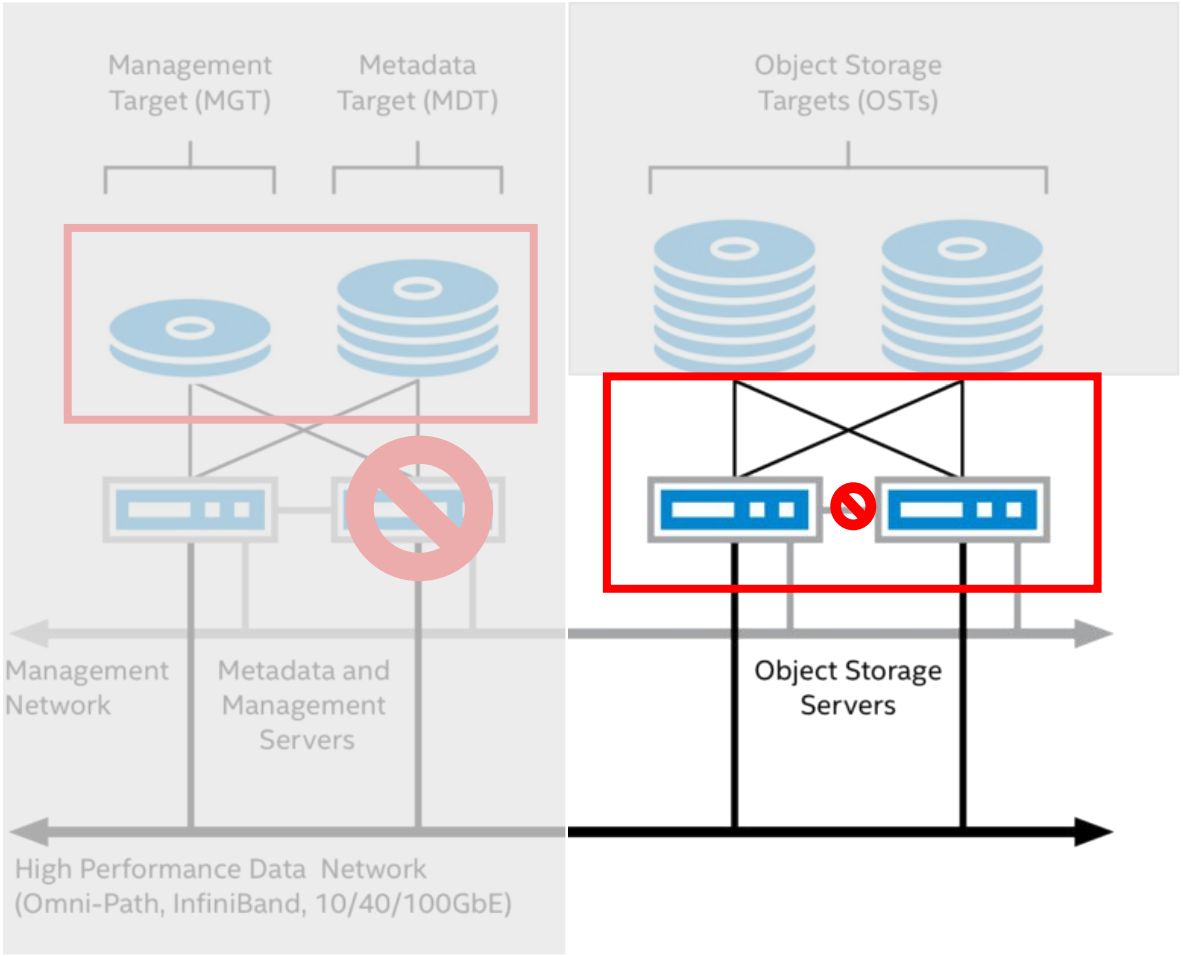


IPSL Lustre Servers In Case of Failure

Lustre is not fault tolerant
(Safeguard: High Availability (HA), RAID and backup)

- OST and OSS as recommended ?
  - Not exactly
  - Keep It Strictly Simple (KISS) = no HA
- If an OSS breaks down
  - 6x OST per OSS
  - multipath
  - Mount all OST on the remaining OSS
- 2x OSS but 1x mount point

```
df -h /mnt/proj4x
Filesystem                    Size  Used Avail Use% Mounted on
172.17.3.168@o2ib:/proj4x     1.4P  468T  868T  36% /mnt/proj4x
```

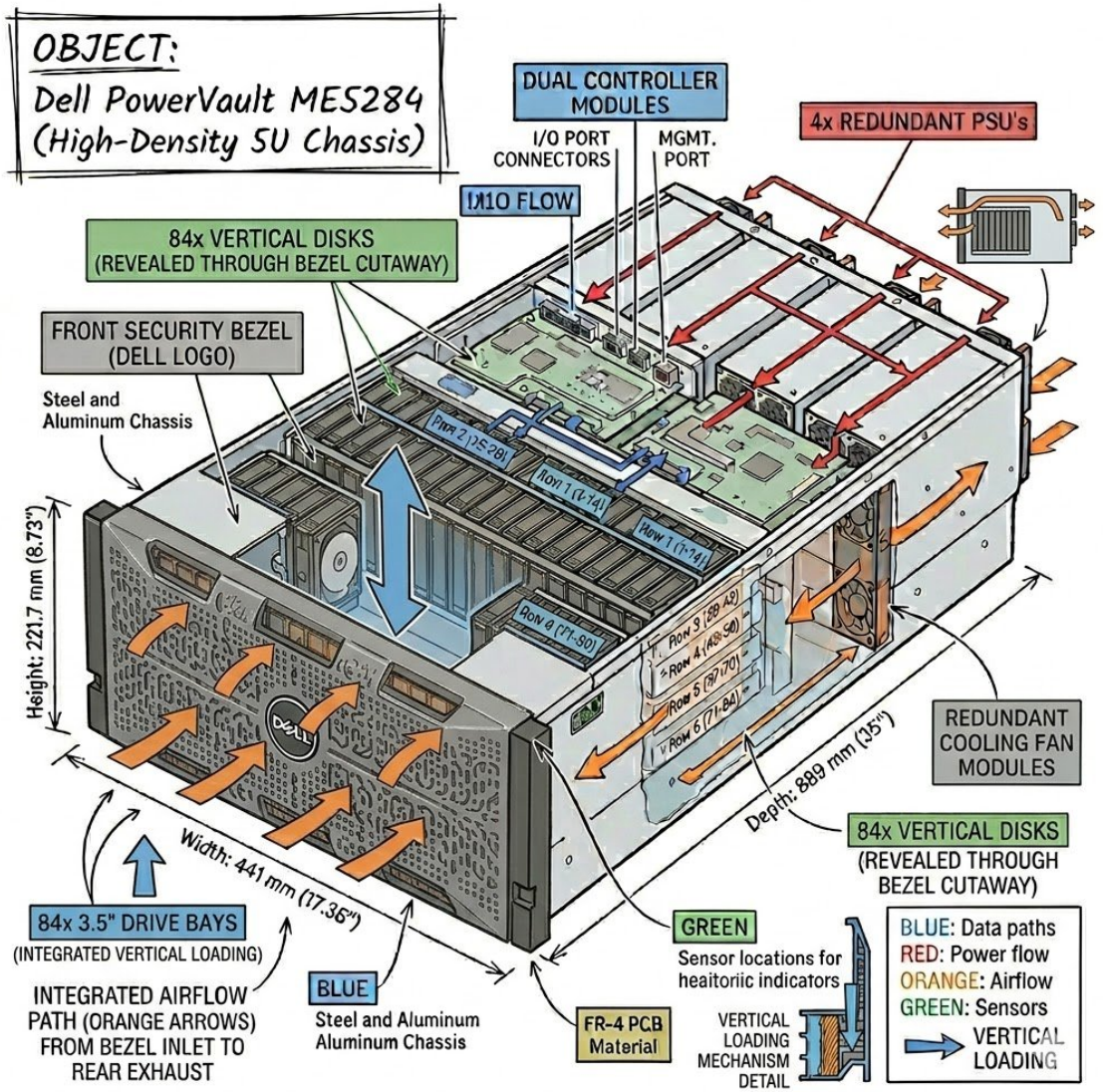


IPSL Lustre Servers
1x Disk Enclosure KISS

Dell PowerVault
(Remember, Lustre is not fault tolerant)

- Dell PowerVault disk enclosure
  - SAN Storage ready
  - Fiber Channel 4x 8/16/32Gbps
  - MD 3860 then ME 4084/5084/5284
  - Dual controller (8/16/32GB cache)
  - No extension (error prone)

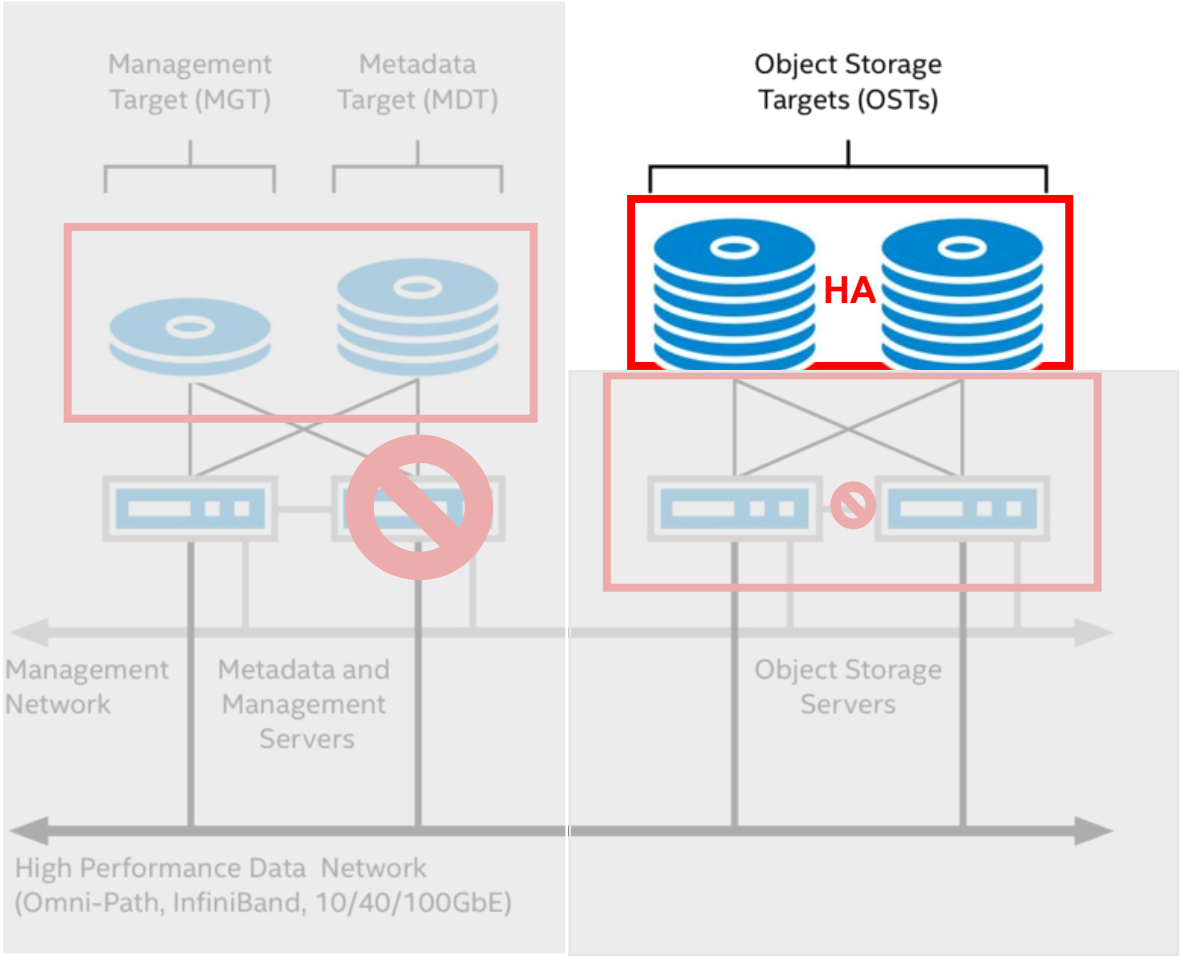
- No Fiber Channel switch (KISS)
  - Direct Attached to Server
  - multipath



IPSL Lustre Servers
1x Disk Enclosure KISS

Configuration 84x HDD of 24TB
(Safeguard : High Availability (HA), RAID ADAPT)

- 2x pools of 42 HDD in RAID ADAPT
  - One pool per controller
  - Controllers in HA
  - RAID 8+2
  - 3 to 4 HDD spare capacity
- 12x linear volumes (OST), 6x per OSS
- If an OST breaks down
  - All data having a stripe on that OST is lost
  - Is striping a good idea ?
  - No backup (too much data)
- Max Perf per controller: ~ 2GB/s
- Usable capacity: 1.4PB



Index

- Introduction
- Lustre Overview
- IPSL Lustre Servers
- Lustre File System
  - Quotas
  - Monitoring
  - Auditing
- Conclusion

Lustre File System Quotas

- **User Quotas**
- **Group Quotas**
- **Project Quotas**
- **Some issues with Lustre 2.12**
- **Fully functional with Lustre 2.15**

 # Configure data/block quota managed by OSTs

lctl set_param -P osd-*.LUSTRE_FS*.quota_slave_dt.enabled=XXX

 # Configure data/block quota managed by MDT

lctl set_param -P osd-*.proj5x*.quota_slave_md.enabled=XXX

 # Enable quota enforcement for users only

XXX=u

 # Enable quota enforcement for groups only

XXX=g

 # Enable quota enforcement for projects only

XXX=p

 # Enable quota enforcement for users and groups

XXX=ug

 # Enable quota enforcement for users and groups

XXX=ugp

 # Disable quota enforcement for users and groups

XXX=none

Lustre File System Monitoring

- Use prometheus exporter
  - Not tested yet
  - https://github.com/whamcloud/lustrefs-exporter
- Problems
  - Usually detected on client side
  - File read() and write() calls are blocking
  - Metadata ops can block longer than data I/O

  (stat(), opendir(), readdir() on dirs with millions of entries)
- Quick check with « lfs df »
- If not OK, check Linux kernel logs
  - Just sit back and stay calm – It is urgent to wait
  - On client, then on OSS and on MDS/MGS
  - RPC timeouts, evictions, mount issues, ...

 # Check Lustre File System on client

```
root@spiritx1:~# lfs df -h /mnt/proj4x
UUID                       bytes        Used   Available Use% Mounted on
proj4x-MDT0000_UUID        353.1G       17.5G      305.6G   6% /mnt/proj4x[MDT:0]
proj4x-OST0000_UUID        112.4T       40.5T       70.8T  37% /mnt/proj4x[OST:0]
proj4x-OST0001_UUID        112.4T       40.5T       70.8T  37% /mnt/proj4x[OST:1]
proj4x-OST0002_UUID        112.4T       41.0T       70.3T  37% /mnt/proj4x[OST:2]
proj4x-OST0003_UUID        112.4T       41.2T       70.1T  38% /mnt/proj4x[OST:3]
proj4x-OST0004_UUID        112.4T       41.0T       70.3T  37% /mnt/proj4x[OST:4]
proj4x-OST0005_UUID        112.4T       41.0T       70.3T  37% /mnt/proj4x[OST:5]
proj4x-OST0006_UUID        112.4T       40.8T       70.4T  37% /mnt/proj4x[OST:6]
proj4x-OST0007_UUID        112.4T       40.9T       70.4T  37% /mnt/proj4x[OST:7]
proj4x-OST0008_UUID        112.4T       40.6T       70.7T  37% /mnt/proj4x[OST:8]
proj4x-OST0009_UUID        112.4T       41.6T       69.7T  38% /mnt/proj4x[OST:9]
proj4x-OST000a_UUID        112.4T       41.5T       69.8T  38% /mnt/proj4x[OST:10]
proj4x-OST000b_UUID        112.4T       40.9T       70.4T  37% /mnt/proj4x[OST:11]

filesystem_summary:          1.3P       491.4T      844.0T  37% /mnt/proj4x
```

 # Check Linux kernel logs

```
root@spiritx1:~# dmesg -T | grep -i -E '(lnet|lustre)'
...
```

Lustre File System Auditing

- **Robinhood Policy Engine (Changelog)**
  - **Replicate of filesystem metadata**
  - **Database can be queried at will**
  - **Database is populated with changelog**
  - **https://github.com/cea-hpc/robinhood**
- **Scan Lustre Filesystem indirectly**
  - **Robinhood v3 for now**
  - **Robinhood v4 to come**
- **UI available**

 # On MDS register a user to receive changelogs

lctl --device lustre-MDT0000 changelog_register

output: lustre-MDT0000: Registered changelog userid

'cl1'

 # Enable changelog entry types MTIME and CTIME flags

lctl set_param -P mdd.*-MDT0000.changelog_mask='+MARK +MTIME +CTIME'

 # Check flags on MDS Lustre changelog

lctl get_param mdd.*-MDT*.changelog_mask

 # On OSS server start oss (mount ost)

rbh-report --class-info=small_files -f proj1x

rbh-find -f proj1x -name AMSU-1C

rbh-du -f proj2x -H /mnt/proj2x

rbh-report --top-dirs=25 -f homedata

...

Index

- Introduction
- Lustre Overview
- IPSL Lustre Servers
- Lustre File System
- Conclusion
  - Lustre 2.15 is production ready !
  - Digging into features
  - Easy prototyping with 4 VMs

Conclusion

- **Lustre 2.15 is production ready !**
  - **KISS : No High Availability / No Fibre Channel Switch**
  - **1 MDS for all filesystems**
  - **2x OSS per filesystem**
  - **Choose efficient storage solutions**
  - **One storage enclosure per filesystem**

- **Digging into features**
  - **Data On Metadata (DOM)**
  - **Progressive File Layout (PFL)**
  - **Bandwidth Limit**
  - **Roadmap to Erasure Coding (Fault Tolerant Lustre)**

- **Easy prototyping with 4 VMs**

MERCI - THANK YOU

Institut Pierre Simon Laplace IPSL