COFS
China Open File System

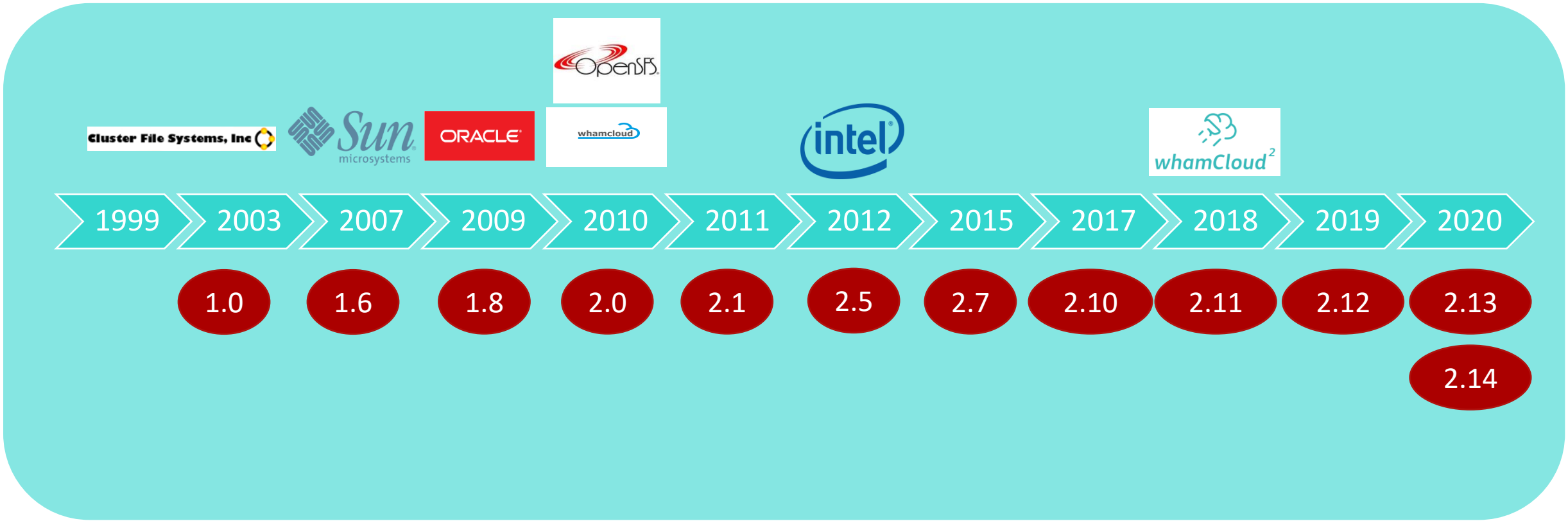
Whamcloud

# 开源Lustre文件系统技术演进及社区协作

Li Xi, Principal Engineer

DDN®
STORAGE

# History of Lustre



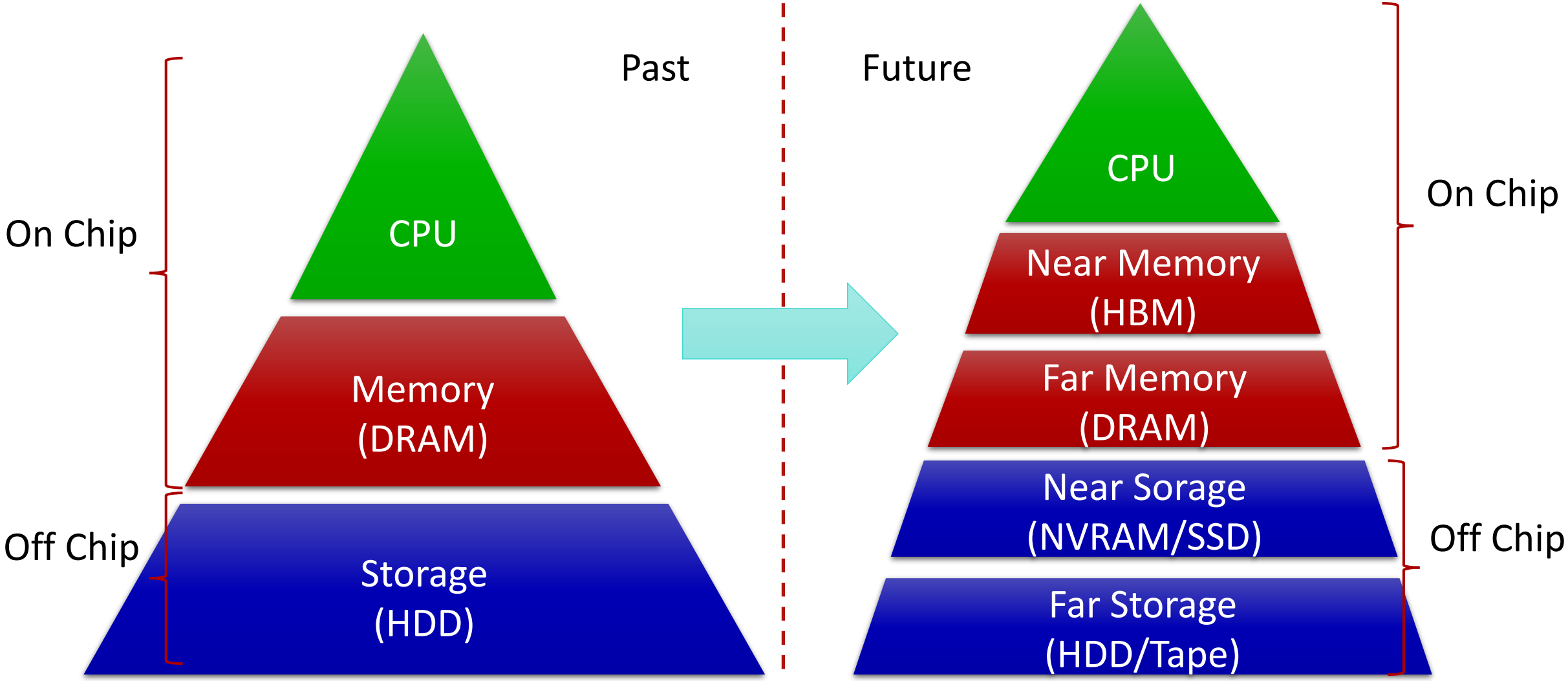

Illustrates the robustness of open source technology in the face of organizational changes

# New Trends for Lustre

Whamcloud

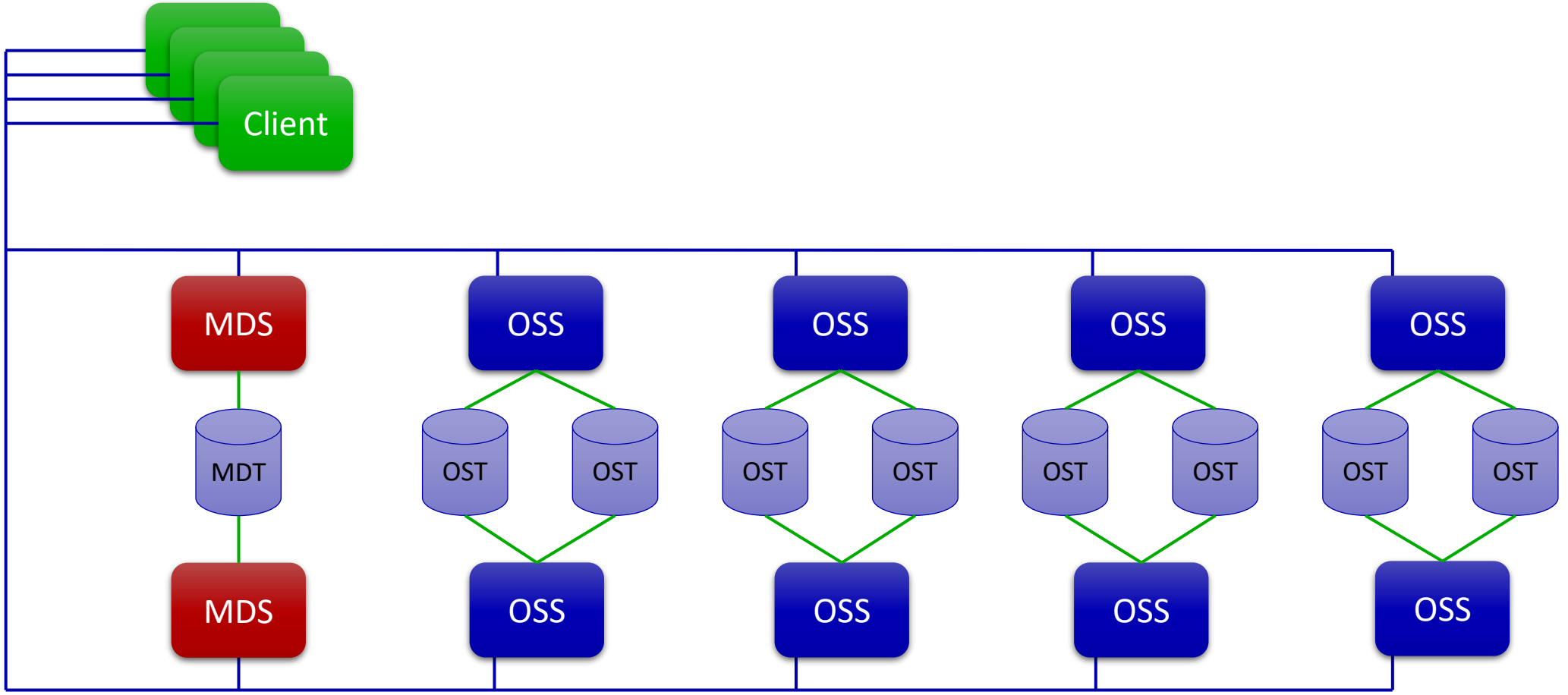
- Variable I/O patterns because of wider types of applications
  - Metadata intensive, Read intensive, Small I/O, random I/O
  - Understand the I/O patterns of applications deeply!
- New types of storage media
  - SSD, NVMe, NVRAM, Persistent Memory
  - Heterogeneous media types for multiple layers of cache
  - Smart data placement and replacement!
- Compute nodes are becoming fatter
  - Much more powerful and reliable
  - Huge memory and persistent local storage
  - Keeping or moving data closer to compute!

# Storage Hierarchy is Changing

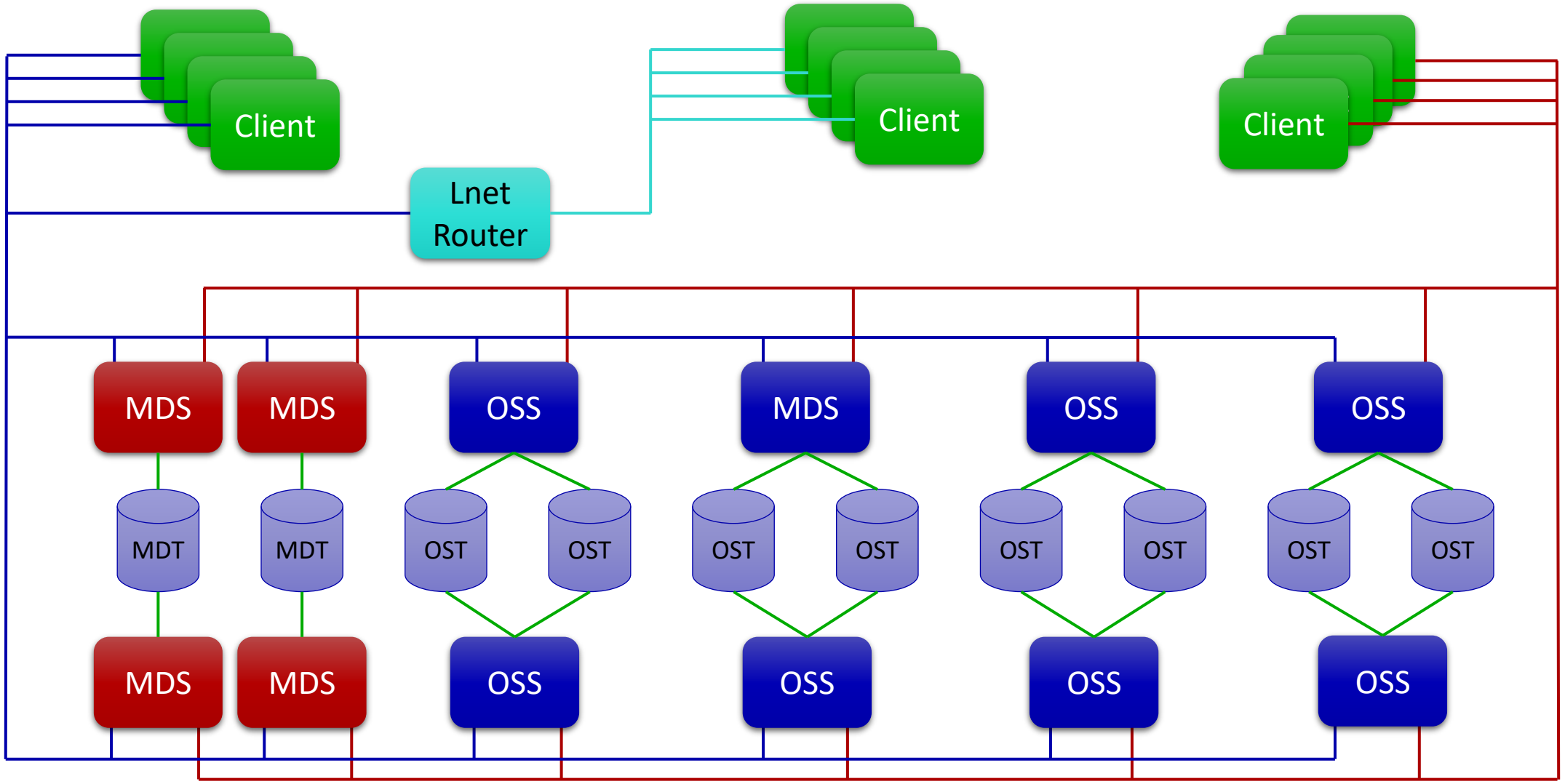


# Basic Lustre File System in Production

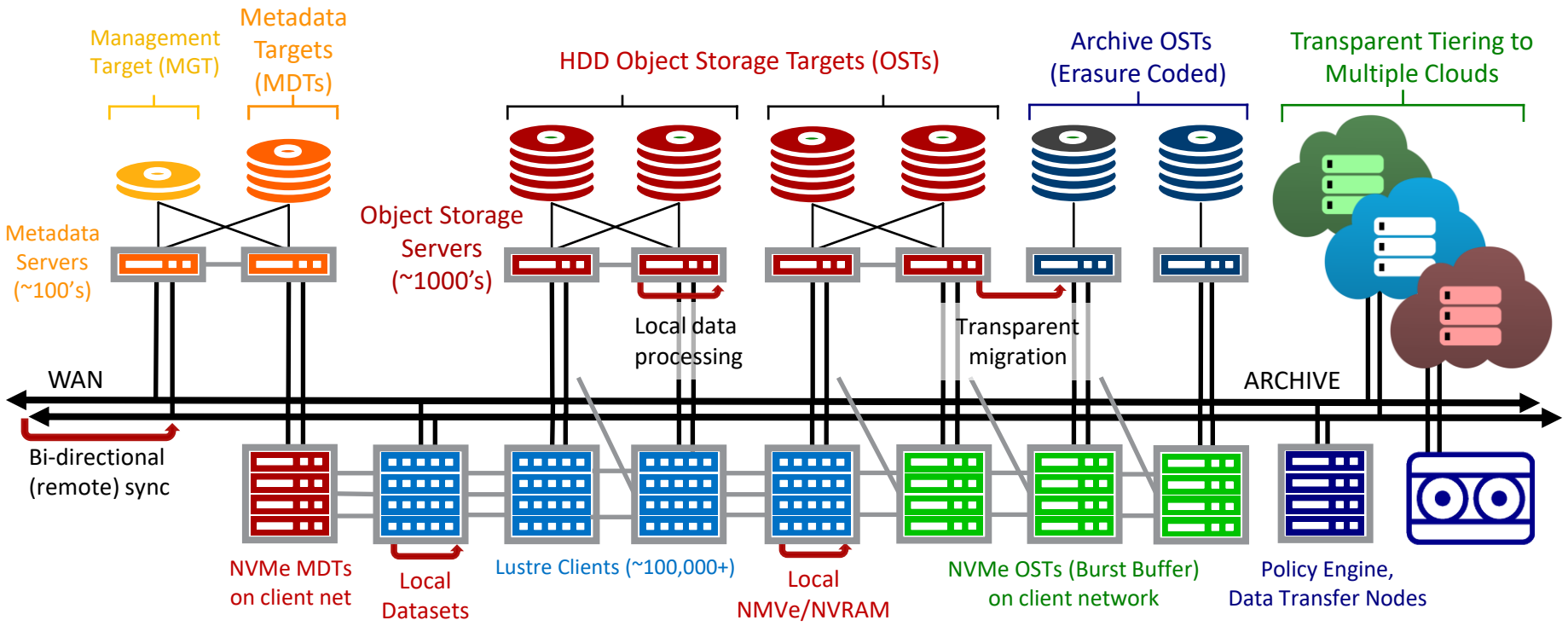
Whamcloud



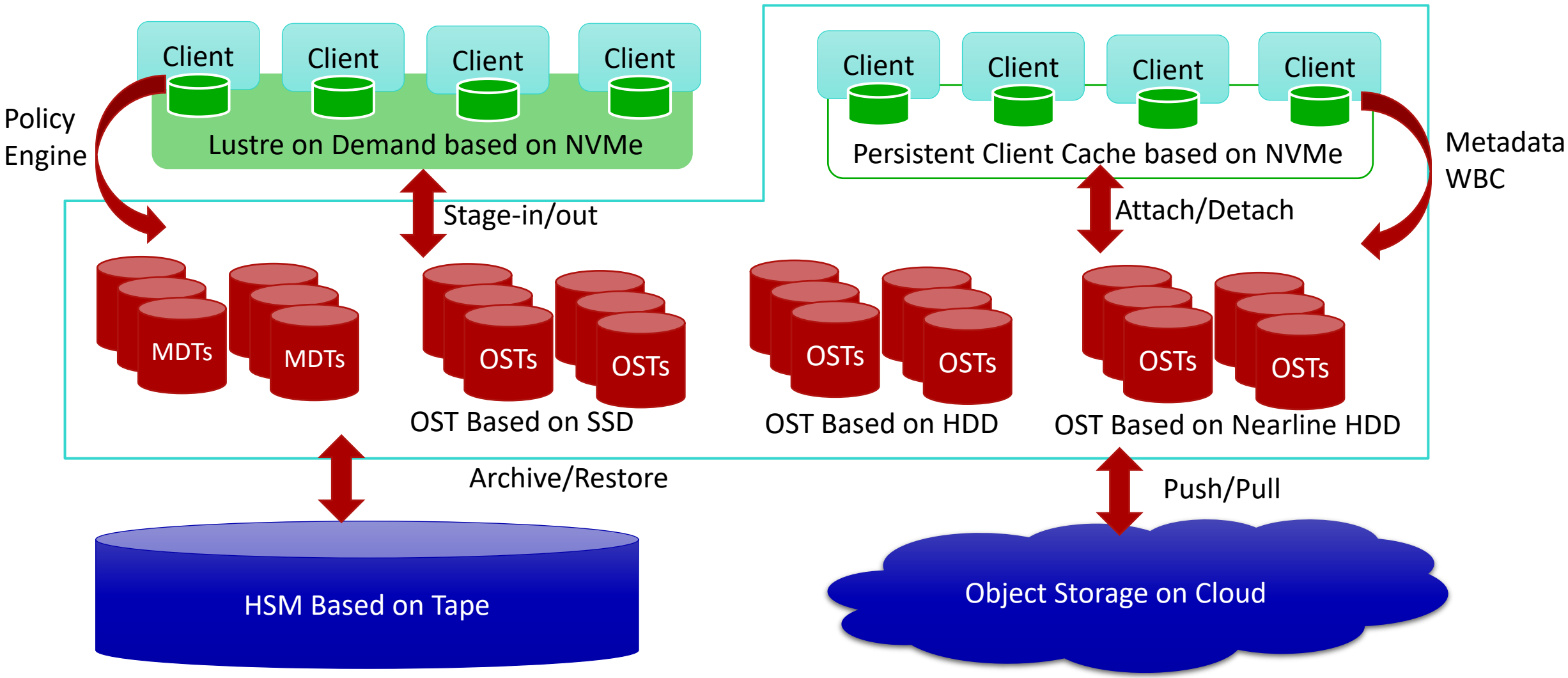
# Complex Lustre File System



# Tiered Lustre File System



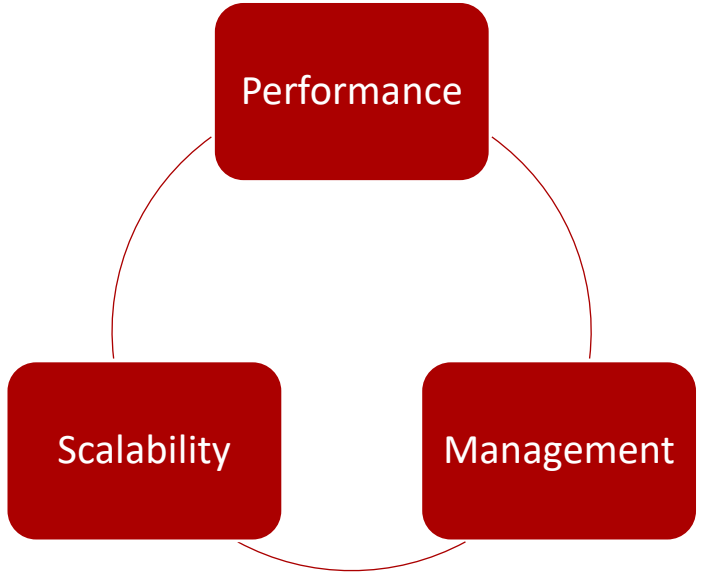
# Example Architecture of a Heterogeneous Lustre File System



# Challenges and Opportunities for Lustre File System

Whamcloud

- Performance challenges
  - Wide usage of NVMe/SSD highlights the software latency
  - Software could be the bottleneck of collective bandwidth/IOPS
- Scalability challenges
  - Both data and metadata sizes keep on enlarging
- Data management challenges
  - Heterogeneous storage types
  - Data migration between multiple storage tiers
  - Data movement for local access
  - S3/POSIX storage integration/connectivity



# Features of Lustre to Solve the Challenges


- Write Back Cache
- Parallel Readahead
- Fast Read
- Size on MDT
- Persistent Client Cache
- Data on MDT
- Lock Ahead Ladvise
- Large RPC Size
- Overstriping
- LNet Multi-Rail

Performance

- Big OST support
- Parallel e2fsck
- Distributed NamEspace
- LNet Health
- File Level Redundancy
- Large Directory on MDT
- ZFS OSD
- Large Xattr of Ext4
- Erasure Coding
- Progressive File Layout

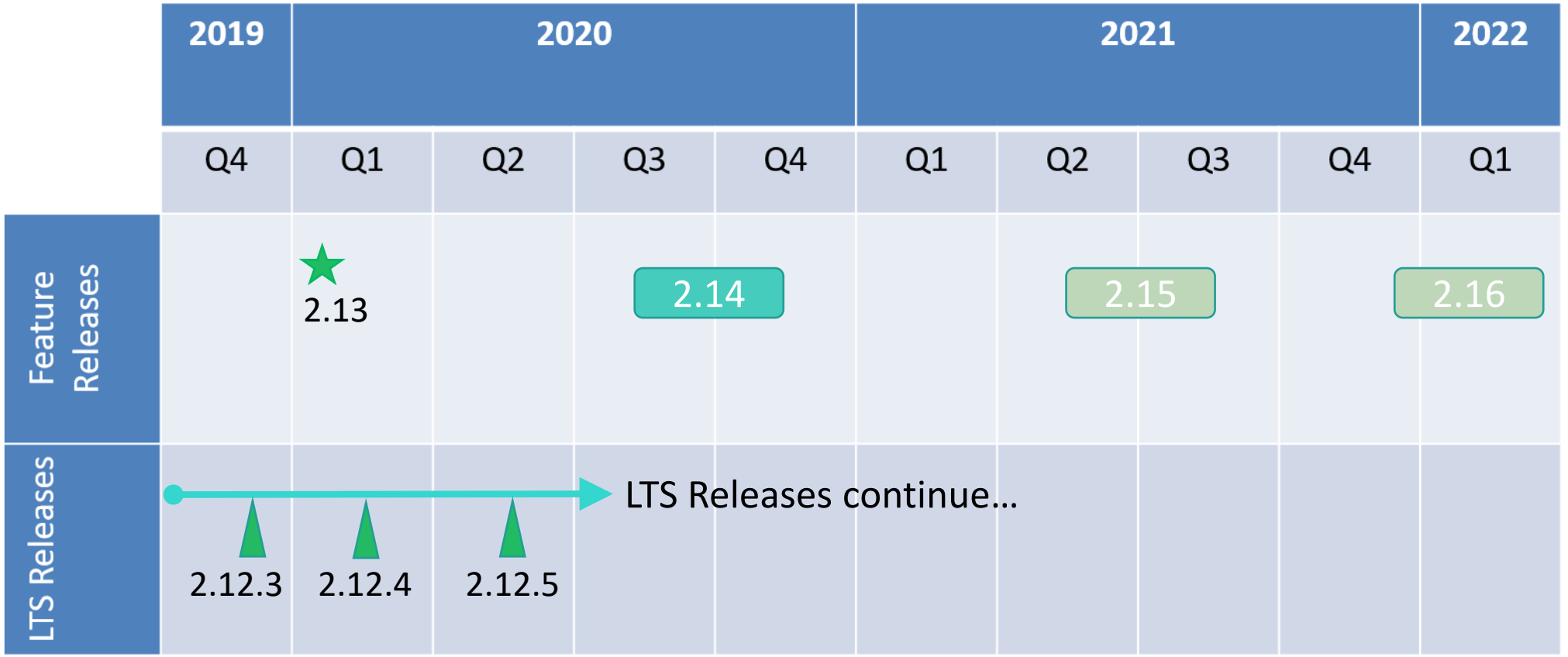
Scalability

- Data Integrity
- Project Quota
- OST Quota Pools
- Monitoring
- Data Placement Policy
- Token Bucket Filter
- Policy Engine
- Data Security
- HSM
- Data Encryption
- Changelog

Management

# Lustre Community Roadmap

Whamcloud




**2.13 (Released)**
- Persistent Client Cache
- Multi-Rail Routing
- Overstriping

**2.14 (Working on)**
- Client Data Encryption
- OST Quota Pools
- DNE Auto Restriping

**2.15 (Planning)**
- FLR Erasure Coding
- Client Directory Encryption
- LNet IPv6 Addressing

**2.16 (Planning)**
- FLR Immediate Mirror
- Metadata Writeback Cache

# Planned Feature Release Highlights

Whamcloud

- **2.14** at feature freeze, with several important additions
  - DNE directory auto-split – improve usability and performance with multiple MDTs
  - OST Quota Pools – manage space on tiered storage targets with OST pools
  - Client-side *data* encryption – persistent encryption of data from client to disk
- **2.15** feature development already well underway
  - Client-side *directory* encryption – encrypt filenames on disk on MDT
  - File Level Redundancy - Erasure Coding (EC) – efficiently store striped file redundancy
  - LNet IPv6 addressing - allow over 32-bit addresses, more flexible server configuration
- **2.16** plans continued functional and performance improvements
  - Metadata Writeback Cache (WBC) – low latency file operations in client RAM
  - File Level Redundancy - Immediate Write – write to mirrors directly from client
  - Dynamic inode allocation for ldiskfs - improve flexibility for DoM and large OSTs

# DNE Usability Improvements (WC) (2.13+)

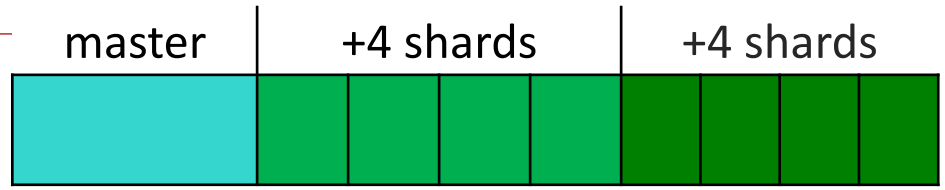
Whamcloud

- **Space balance new directories** on "best" MDT based on available inodes/space
  - Transparently select "best" MDT for normal `mkdir()` based on parent policy (LU-10784)
  - Set default policy on parent via "`lfs setdirstripe -D -i -1 dir`" (LU-11213)
  - Most useful for root and top-level user directories

2.13

2.14

- **New `crush` directory hash type** (LU-11025)
  - Minimize number of directory entries migrated by restripe




- **Automatic directory restriping** as directory size grows (LU-11025)
  - Create one-stripe directory for low overhead, increase shards/capacity/performance with size
  - Add `mdt.*.dir_split_delta=4` shards if shard over `mdt.*.dir_split_count=50000` entries
  - Move fraction of existing **directory entries** to new directory shards (names only, not inodes)
  - New directory entries and inodes created directly on new MDTs

2.15

- Improve MDT usage/space balancing for new filesystems (LU-13417)
- Select closest network-local MDT(s) for mkdir for tiered/distributed configs (LU-12909)

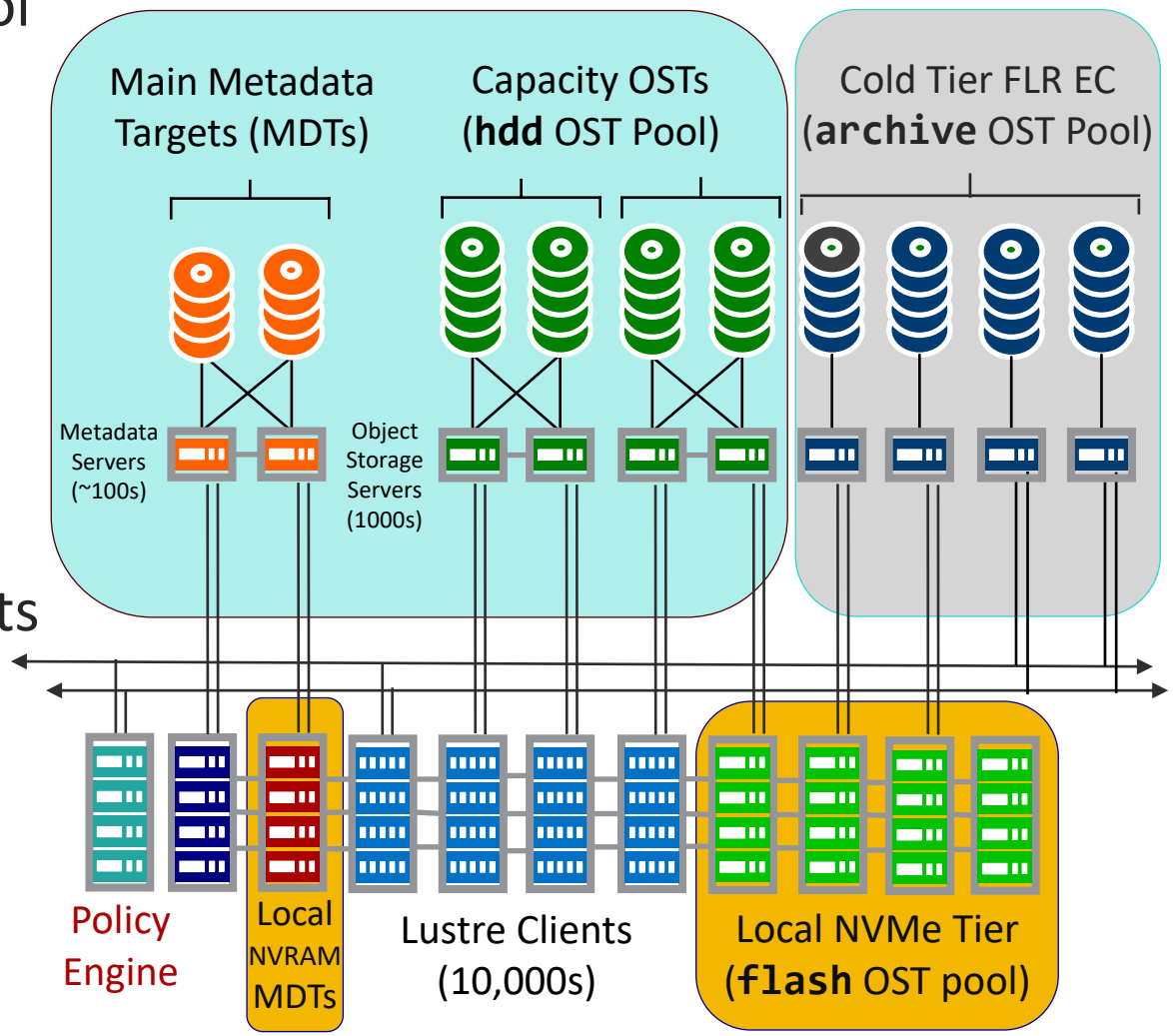
# OST Quota Pools (LU-11023, Cray/HPE) (2.14+)

- Account/limit space for OSTs in a specific pool
  - Control usage of small flash OSTs in tiered config
- Use existing Lustre quota infrastructure
  - OST already tracks space per UID/GID/ProjID
  - Pool usage based on sum of current OSTs in pool
- Add quota pool limits per UID/GID/ProjID
  - No extra accounting on the OSTs
  - Only new aggregation/reporting by MDS

2.14

TBD

- Check quota limit when allocating OST objects
  - Avoid OSTs with little/no quota available
- **MDT pools** to allow MDT tiering
  - Manage/balance DoM MDT space usage
  - Handle MDT storage classes (e.g. NVRAM vs. NAND)



# Client *Data* Encryption to Disk (LU-12755, WC) (2.14+)

Whamcloud

- Protect from storage theft or loss, and network or malicious client snooping
- **Encryption on Lustre client** down to storage
  - **Securely store user crypto keys** in client kernel keyring
  - **Data encrypted before sending** to servers
  - **Data decrypted after receiving** from servers
  - Servers/storage only see encrypted data
  - **Transparent to backend** filesystem/storage (ldiskfs/ZFS)
  - Use larger client CPU capacity to encrypt/decrypt data
- **Use existing ext4/f2fs `fscrypt` library/tools**
  - Inventing your own encryption is a fool's errand
  - **Per-directory tree** tunable encryption setting/user master key
  - **Per-file encryption key**, itself encrypted by user master key
    - **Fast and secure deletion** of file once per-file key is erased
    - Decrypted data dropped from client cache when user master key removed

2.14

2.15

- ***Filenames* encrypted on client** for MDT directory entries

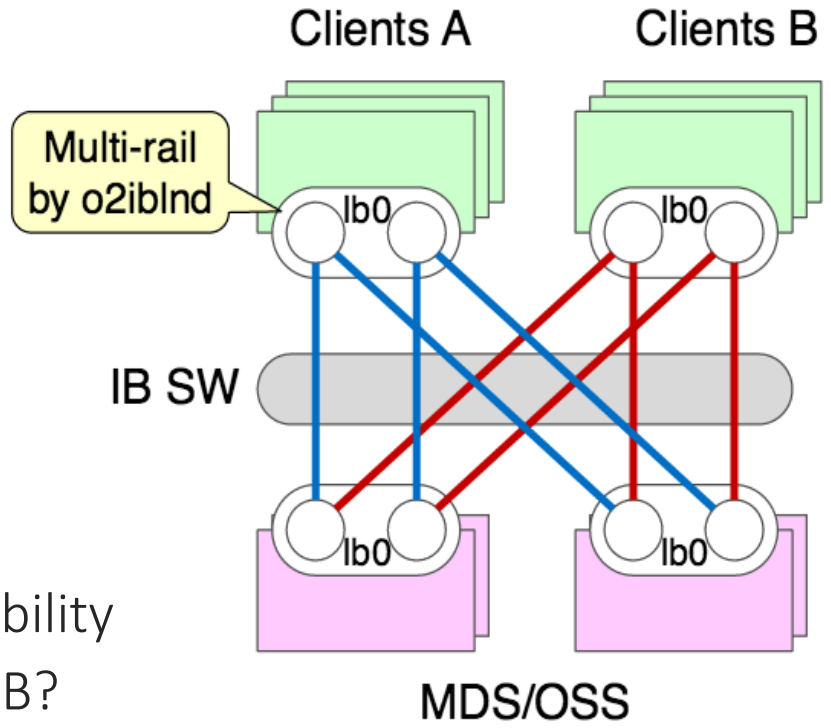
# LNet Improvements (2.14+)

Whamcloud

2.14 ▶ MR Router Health improvements (LU-12941, LU-13510, LU-13025, …, HPE, WC)

2.14 ▶ User Defined Selection Policy (LU-9121, WC)

- Fine grained control of interface selection
  - TCP vs. IB networks, primary vs. backup, local vs. remote
- Optimize internal RAM/CPU/PCI data transfers
- Useful for large NUMA machines with multi-rail




2.15 ▶ IPv6 Node Addressing (LU-10391, WC, SuSE)

- Allow NIDs larger than 32+32bits in TCP and IB
  - New sockv6lnd, o6iblnd nettypes for protocol compatibility
  - Allow direct IB GUID addressing, to avoid need for IPoIB?
- Use Imperative Recovery log for MDT/OST addressing on clients (LU-10360, WC)
  - Allow OSTs and MDTs to mount on any server, no *requirement* for failover addresses
- NIDs no longer needed in Lustre config (LU-13306, WC)

# Data-on-MDT (DoM) Improvements (WC) (2.14+)

Whamcloud

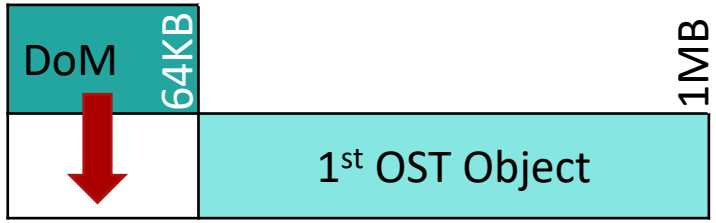
- Shrink DoM component size if MDT free space running out too quickly (LU-12785)

2.14

- Early lock cancel for DoM, +28% IOPS on IO500 `mdtest-easy-delete` (LU-12321)

2.15

- Optimized DoM->OST component removal (LU-13612)
  - Avoid whole-file copy when freeing space from MDT
- Merge data write with MDS_CLOSE RPC (LU-11428)
  - Reduce RPC count by half for `mdtest-hard-write`
- Cross-file data prefetch via statahead (LU-10280)




2.16

- Store very small files (< 600 bytes) directly in ldiskfs inode (**`inline_data`**, LU-5603)
- Dynamic inode allocation for ldiskfs (LU-12099)
  - Simplify initial MDT setup, less need for up-front decision about bytes-per-inode ratio
  - Also improves flexibility for OSTs as they become larger

# Persistent Client Cache (PCC) (LU-10092, WC) (2.13+)

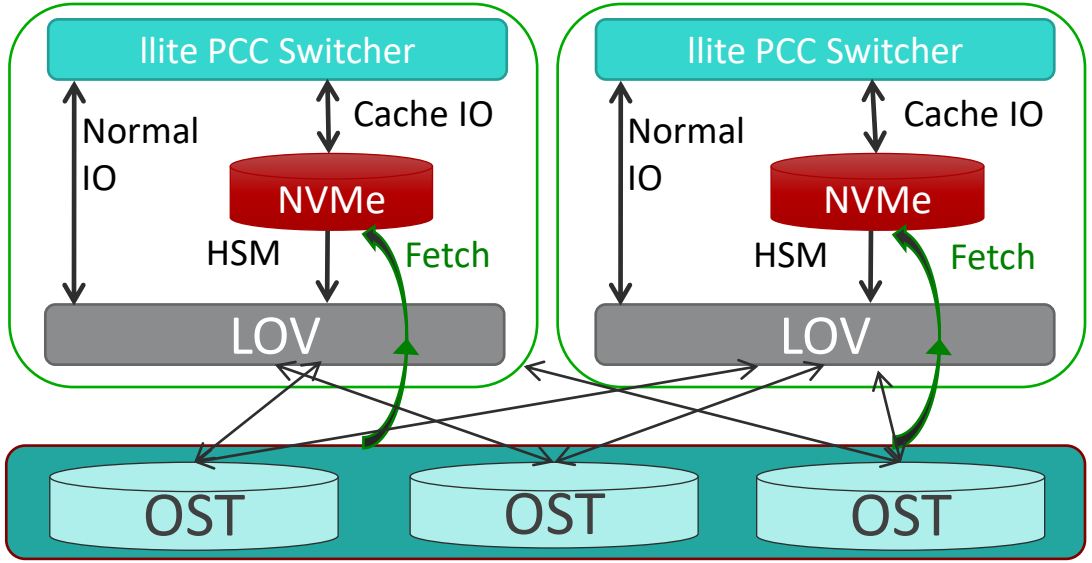
Whamcloud

- **Reduce latency**, improve small/unaligned IOPS, reduce network traffic
- PCC integrates Lustre with a persistent per-client **local cache storage**
  - A local filesystem (e.g. `ext4` or `ldiskfs`) is created on client device (SSD/NVMe/NVRAM)
  - New files created in PCC are *also* created on Lustre MDS

2.13

2.15

- Integrate PCC, HSM, FLR to manage layouts (LU-13637)
  - Simplify code, improve functionality
- Add **shared read** vs. **exclusive write** cache
- Integrate with DAX for NVRAM cache device
  - Use dedicated NVRAM filesystem (e.g. NOVA) for speed



# File Level Redundancy (FLR) Enhancements (WC) (2.15+)

Whamcloud

- **Erasure coding** adds redundancy without 2x/3x mirror overhead (LU-10911)
  - Delayed erasure coding to new/existing striped files *after* normal write
  - For striped files - add N parity per M data *stripes* (e.g. 16d+3p)
  - Leverage CPU-optimized EC code (Intel ISA-L) for best performance
  - **Fixed RAID-4 parity** layout *per file,* **declustered Parity** across files to avoid OST bottlenecks

2.15

2.15

- HSM in composite layout xattr like FLR mirror (LU-10606, WC)
  - Allow multiple archives per file (POSIX, S3, tape, ...)
  - Allow partial HSM file copy/restore to/from archive

2.16

- **Immediate file write** mirroring (LU-13643)
  - Client writes both copies of mirror directly
    - Reduces available bandwidth on client
  - Mirrors kept in sync unless client write fails
  - Delayed resync if mirror goes stale, like before

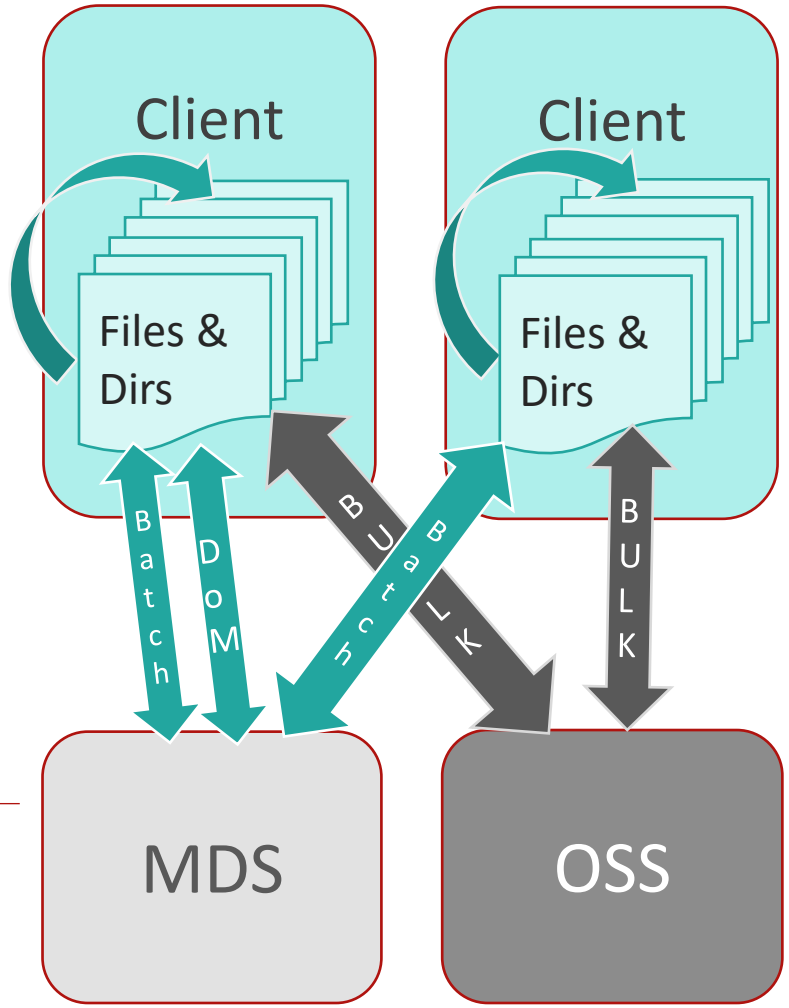
# Metadata Writeback Cache (WBC) (LU-10983, WC) (2.16+)

Whamcloud

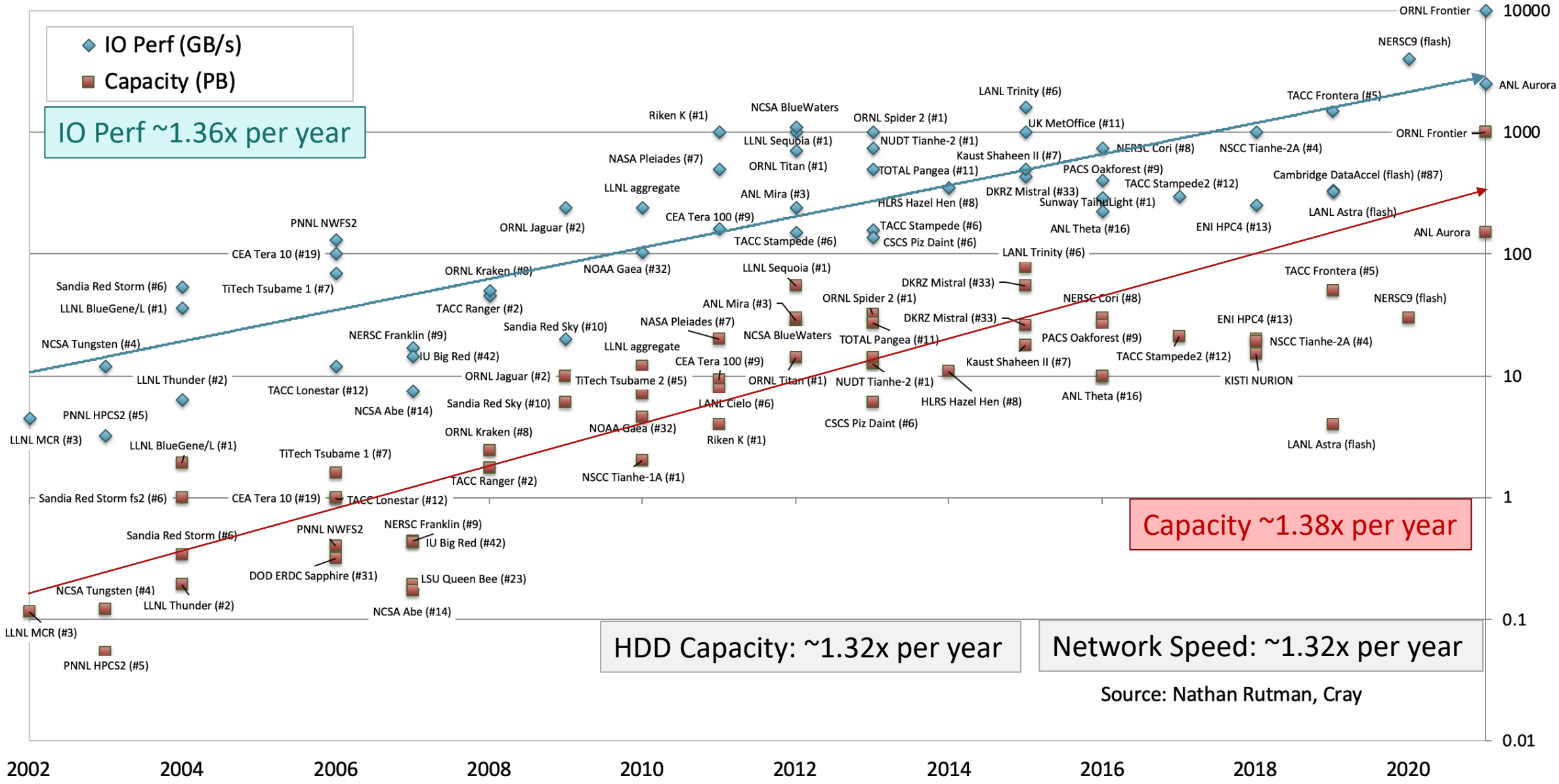
- Create new dirs/files **in client RAM without RPCs**
  - Lock new directory exclusively at `mkdir` time
  - Cache new files/dirs/data in RAM until cache flush or remote access
- **No RPC round-trips** for file modifications in new directory
- **Files globally visible on MDS flush**, *normal use afterward*
  - 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
- WBC prototype developed to test concept
  - 10-20x *single-client* speedup in early testing (`untar`, `make`, …)
- Productization of WBC code well underway

2.16
  - Complexity handling partially-cached directories, space usage

2.17
- **Aggregate operations to server** to improve performance
  - Batch operations in one RCP to reduce network traffic/handling
  - Batch operations to disk filesystem to reduce disk IOPS?



# Lustre Performance and Capacity Growth



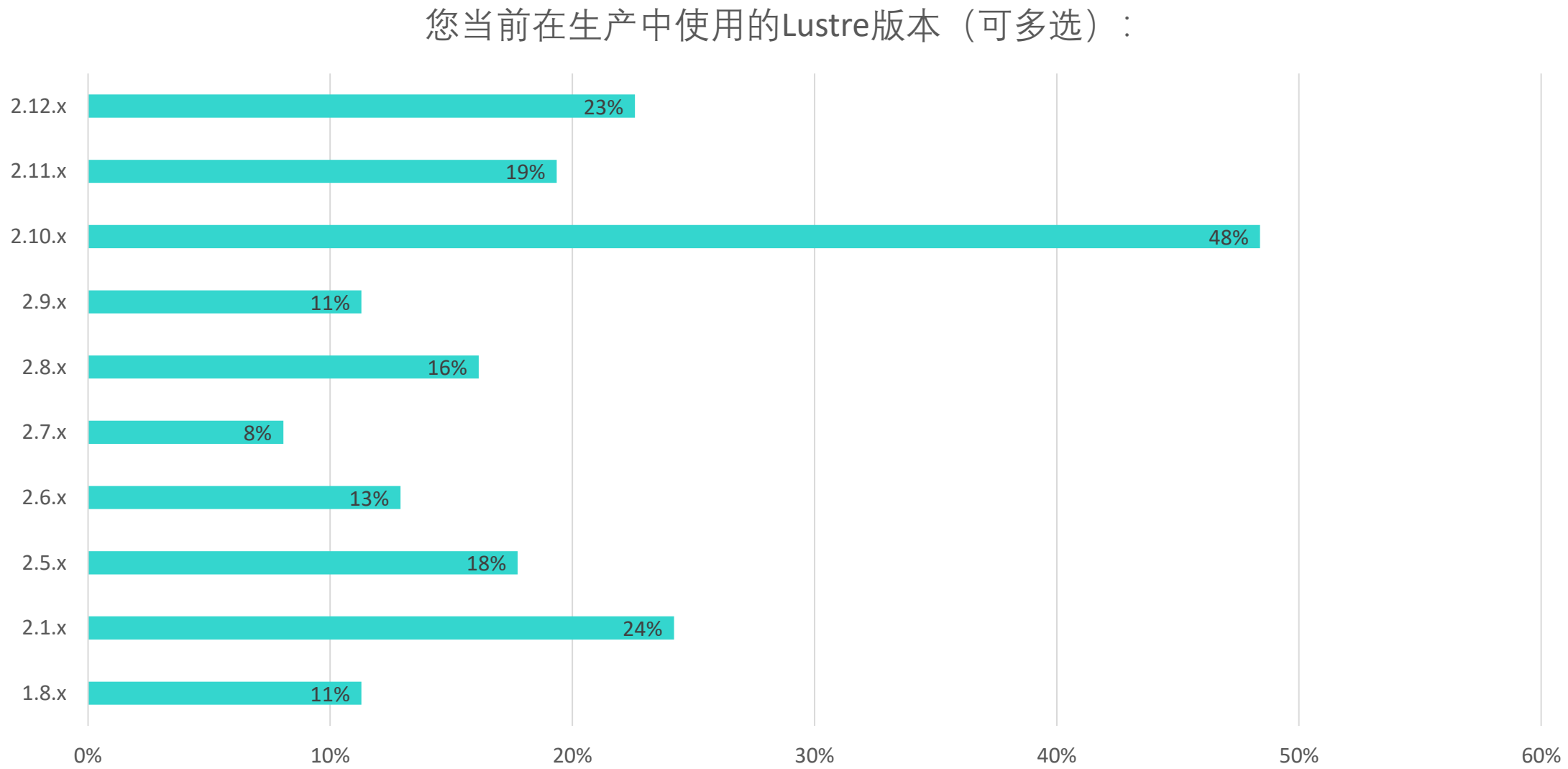

Source: *Rock Hard Lustre*, Nathan Rutman, Cray (with updates for recent years); *Disk Drive Prices (1955-2019)*, John C. McCallum

# 中国Lustre用户社区

- Lustre中国社区网站：http://lustrefs.cn
- 中文Wiki：http://wiki.lustrefs.cn/index.php
- 微信公众号“Lustre文件系统”
- 微信讨论群“Lustre开源用户社区”
- COFS委员会

# CLUG2019 Survey

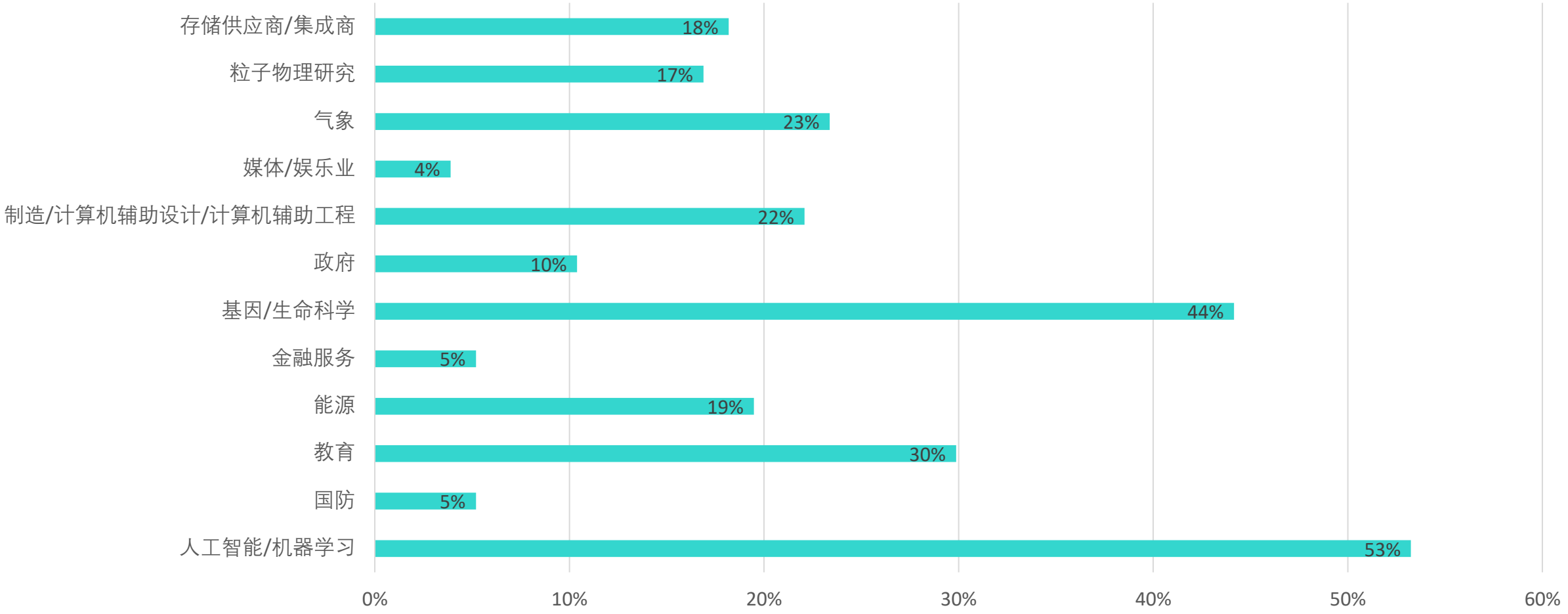
Whamcloud



# CLUG2019 Survey

Whamcloud

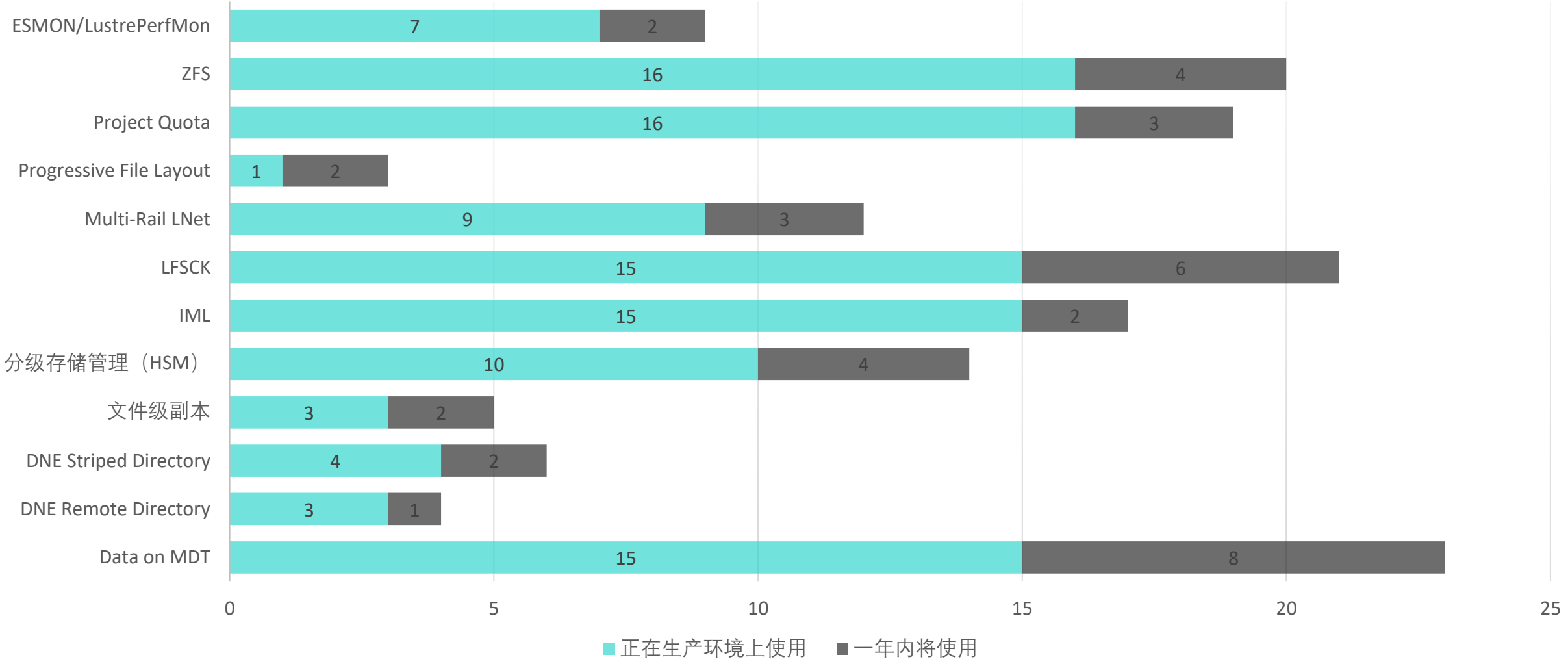
下列选项中哪些描述最接近您的公司/组织对Lustre的使用（可多选）：



# CLUG2019 Survey

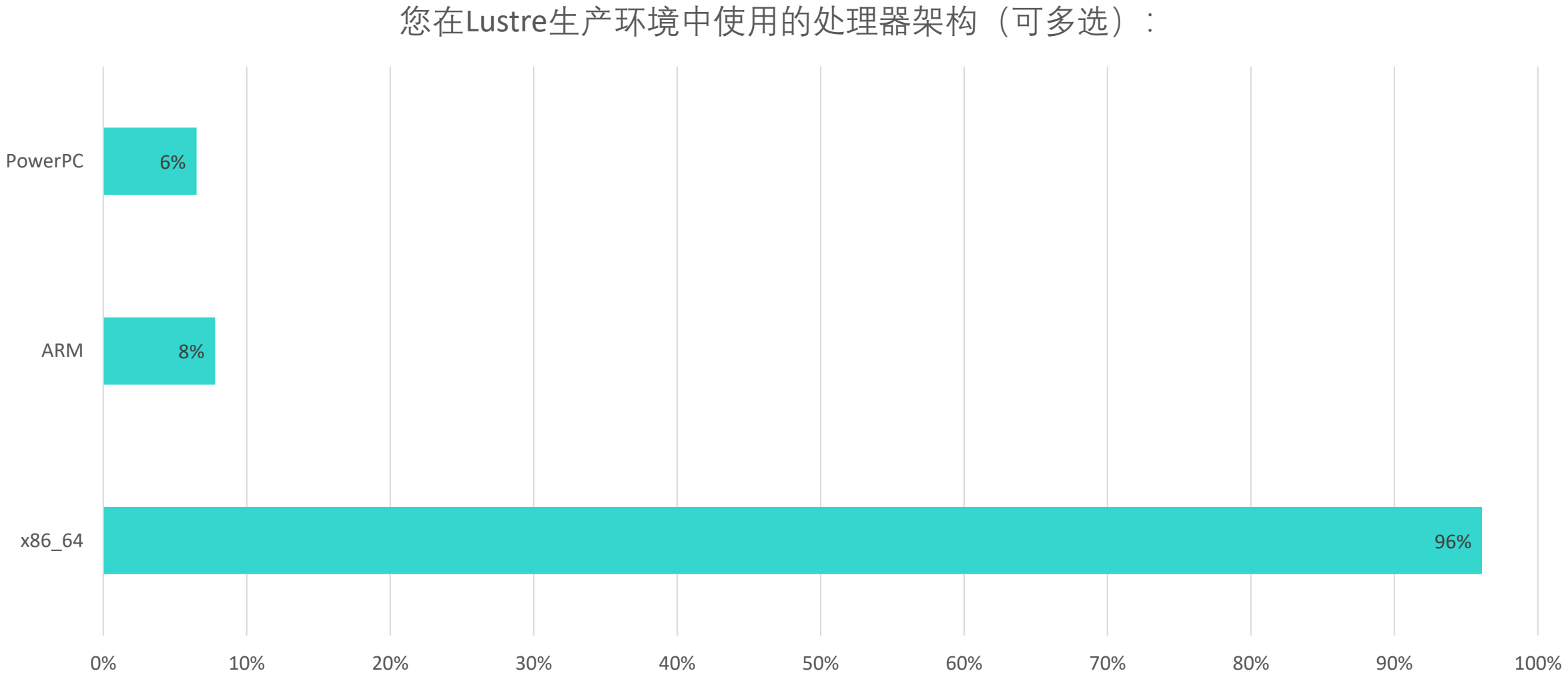
Whamcloud

## 针对下列LUSTRE功能，请选择最接近您感受的选项：



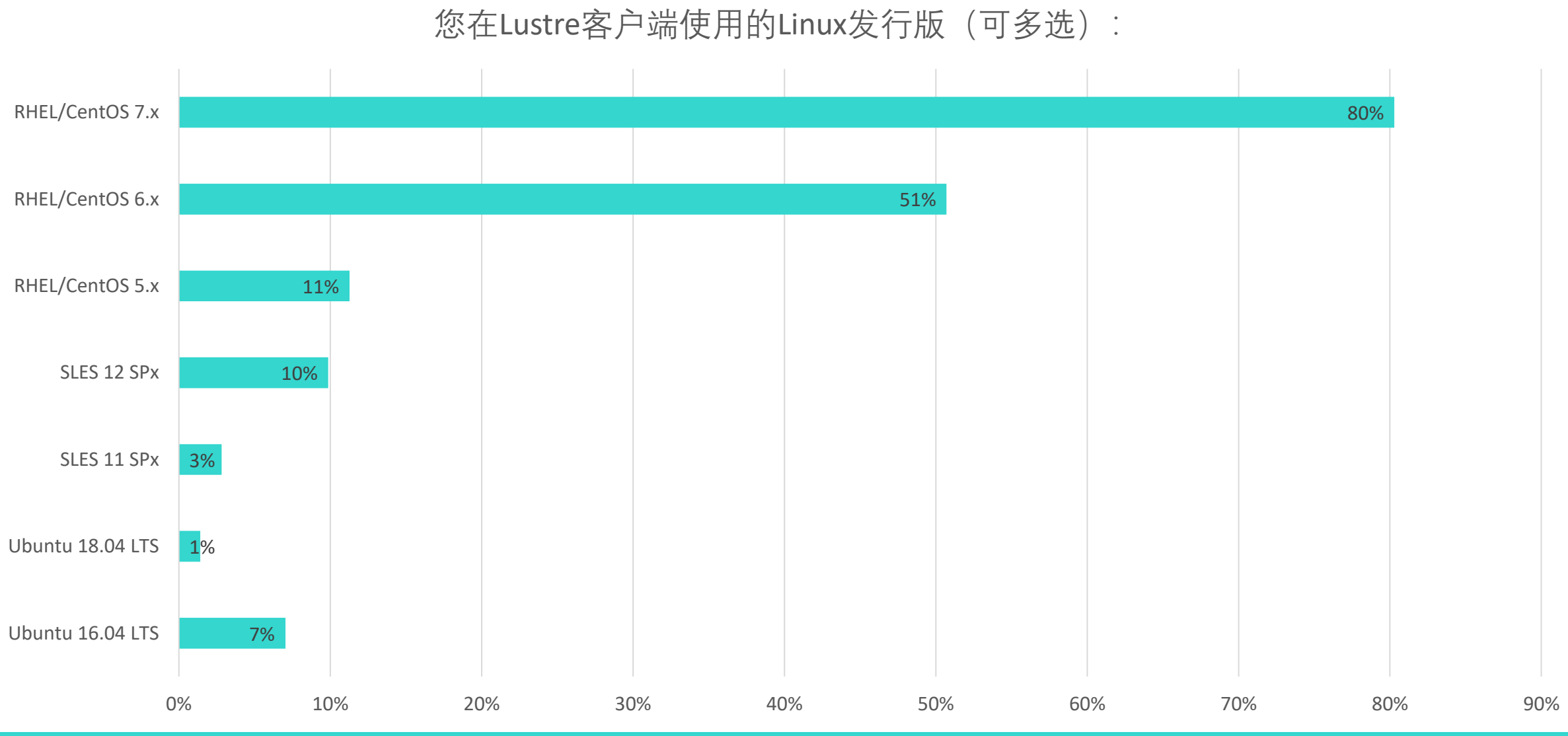
# CLUG2019 Survey

Whamcloud



# CLUG2019 Survey

Whamcloud



# CLUG2011 to CLUG 2020

Whamcloud

CLUG2011

CLUG2013

CLUG2014

CLUG2015

CLUG2016

CLUG2017

CLUG2018

CLUG2019

NOW!