Hewlett Packard
Enterprise

Tiered Data Management for Lustre

Enabling New Use Cases & Deployment Models

Olaf Weber
Master Technologist
HPC Data Management & Storage

Disclaimer

- The information contained in this presentation is proprietary to Hewlett Packard Enterprise (HPE) and may contain forward-looking information regarding products or services that are not yet available.
- Do not remove this slide from the presentation
- HPE does not warrant or represent that it will introduce any product to which the information relates
- The information contained herein is subject to change without notice
- HPE makes no warranties regarding the accuracy of this information
- The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services
- Nothing herein should be construed as constituting an additional warranty
- HPE shall not be liable for technical or editorial errors or omissions contained herein

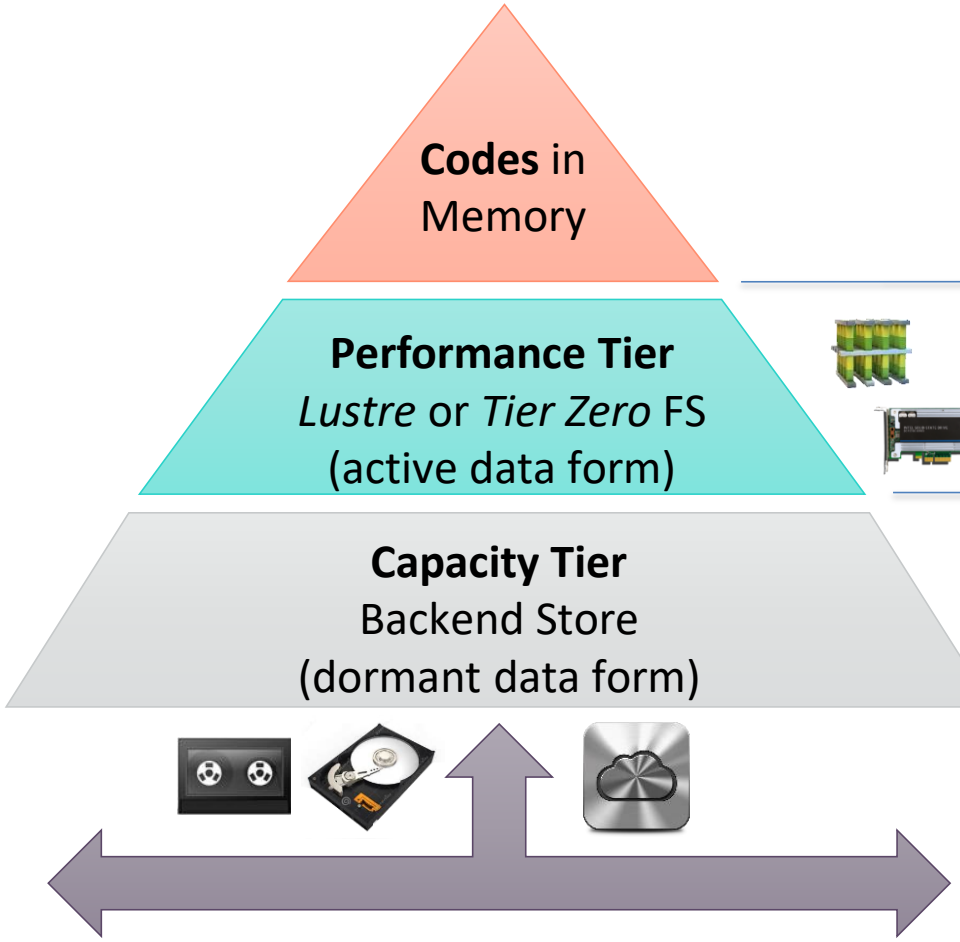
The four Commandments of Data Management

1. Data remains useful far longer than expected.
2. Data must outlive the hardware on which it is stored.
3. Data must outlive the software that manages it.
4. Forward migration to new technology should always be an option.

What is Tiered Data Management?

Core Concepts

Data Tiers



HPC codes require POSIX
Buffered or Direct I/O
MPI-IO libraries

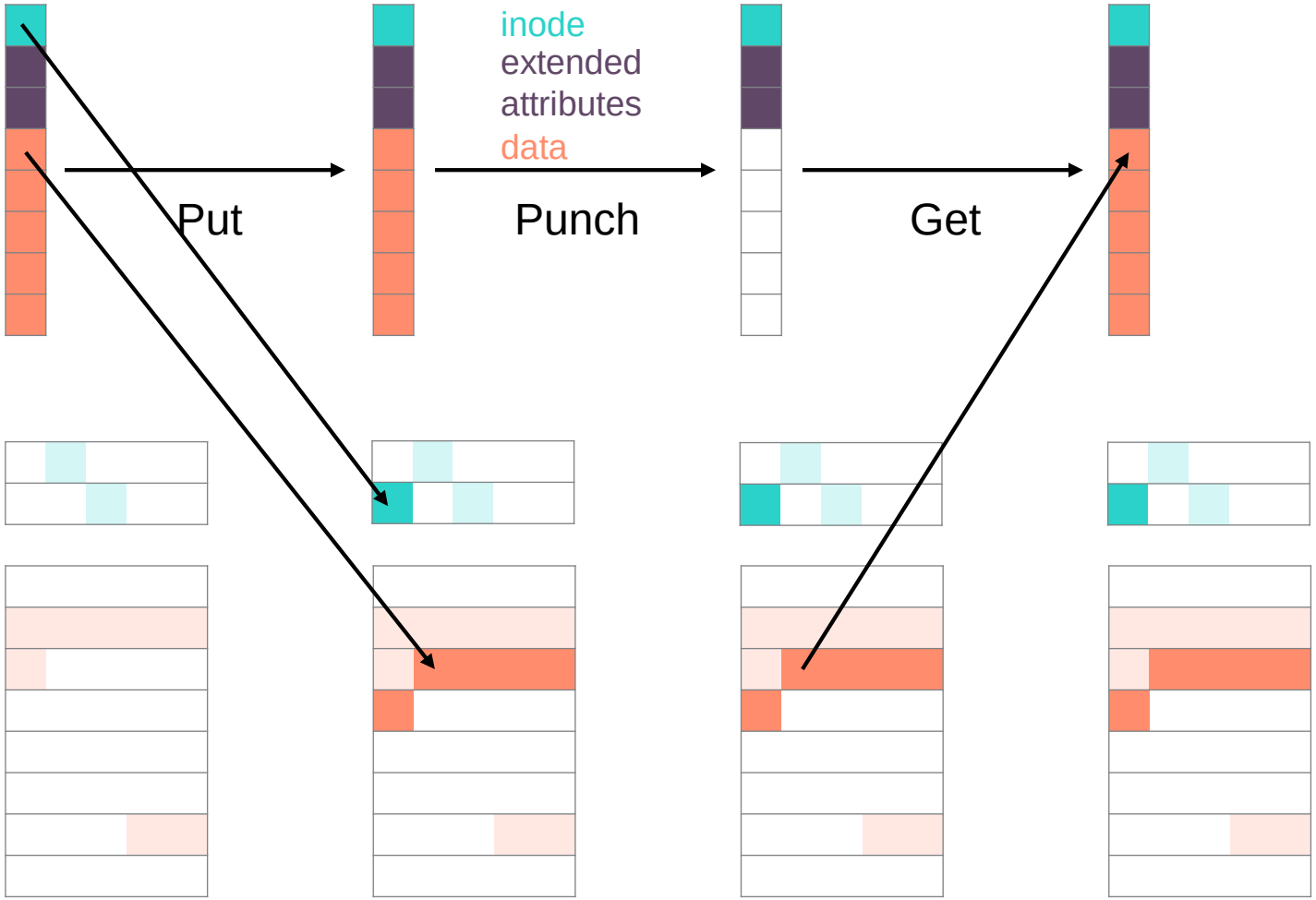
Highest random I/O performance
Limited namespace (up to 100M)
Flexible or no data redundancy

Good streaming performance
Very large namespace (over 10B)
Strong data protection

Multi-site support
Geo replication & Backup
Cloud integration

Core Concepts

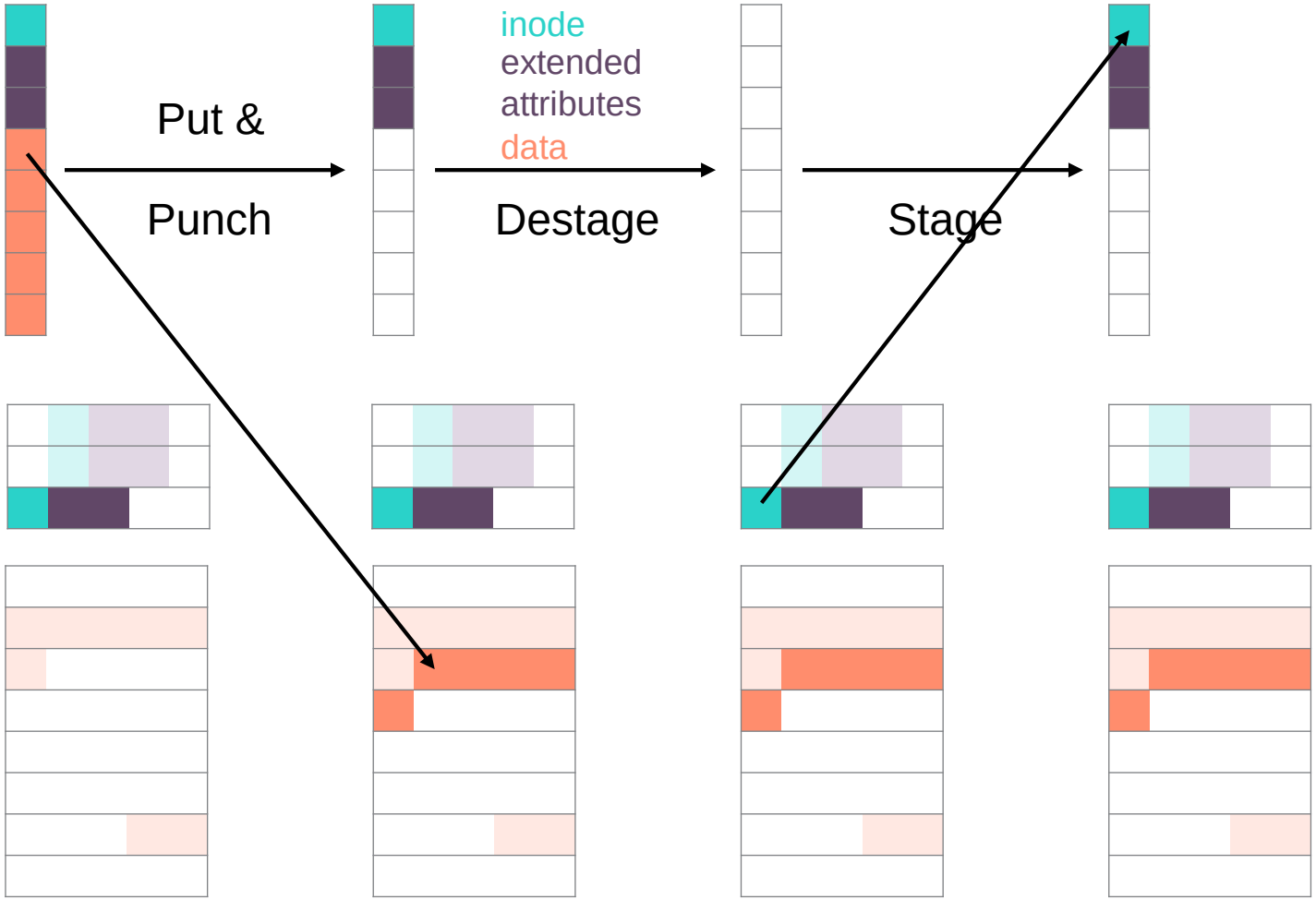
Hierarchical Storage Management



- Data moves between Performance Tier and Backend Store
- Three operations on Performance Tier
  - Put: copy data to Backend Store
  - Punch: remove data from Performance Tier
  - Get: restore data from Backend Store
- A database keeps track of data on Backend Store
- Policy Engine decides when data can be removed from Backend Store

Core Concepts

Tiered Data Management



- Track all file metadata in a Metadata Database
  - This includes file names
  - This includes directory info
- *Five* operations on Performance Tier
  - Put: copy data to Backend Store
  - Punch: remove data from Performance Tier
  - Get: restore data from Backend Store
  - Destage: remove metadata from Performance Tier
  - Stage: restore metadata to Performance Tier

Hewlett Packard Enterprise

Hierarchical Storage Management vs Tiered Data Management

Data Migration Facility

- Filesystem *is* the metadata database
- Entire namespace is in filesystem
  - Database does not have directory info
- File data is migrated transparently
  - Policy engine drives put/punch/get
  - Access drives get
- Migration leaves inodes in place
- Migration leaves extended attributes in place

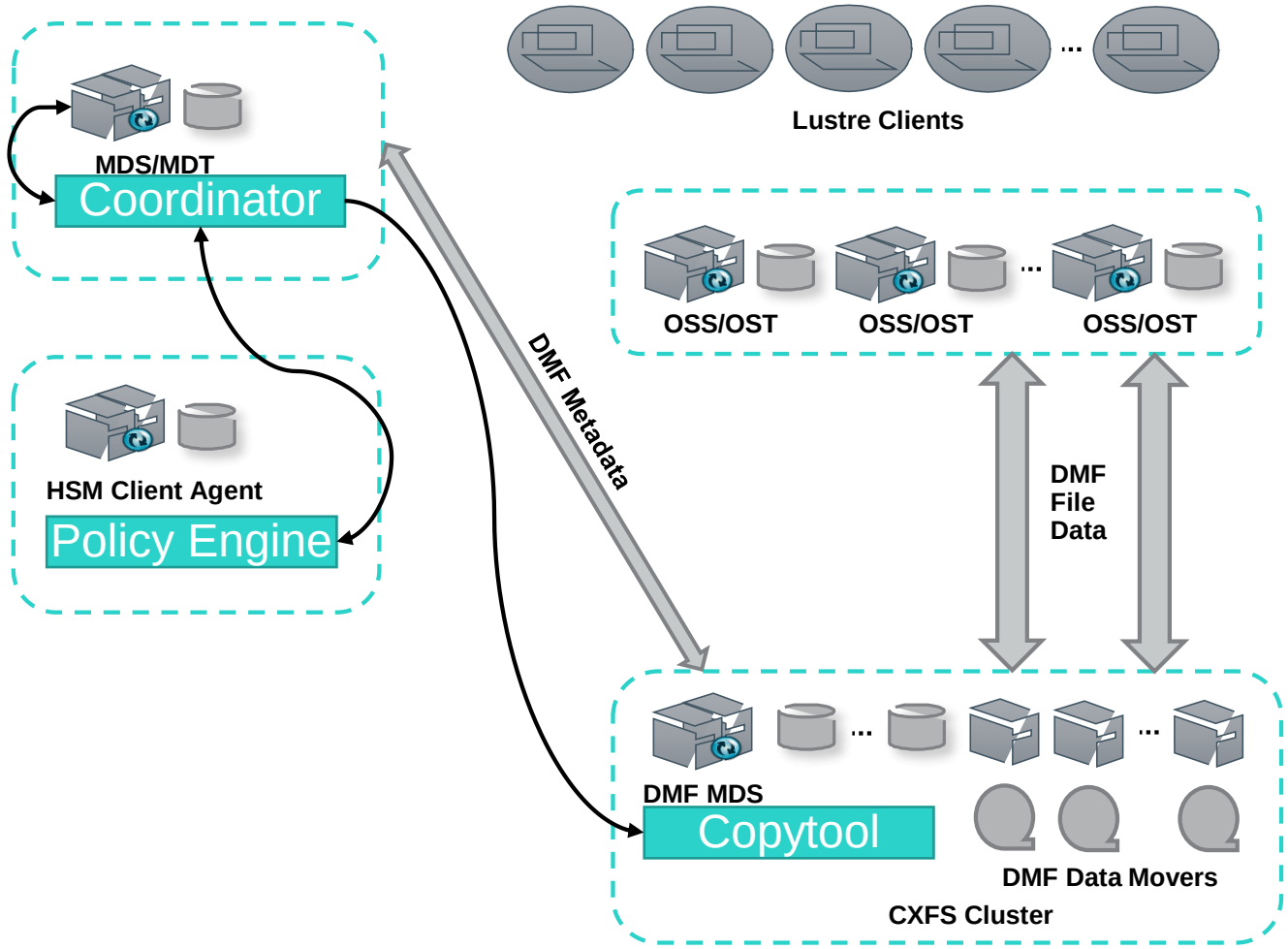
Data Management Framework

- Separate Metadata Database for a filesystem
- Entire namespace is in Metadata Database
  - Metadata Database does have directory info
- Object Database tracks *all* known objects
- File data is migrated transparently
  - Policy engine drives put/punch/get
  - Access drives get
  - But only for Staged files
  - Policy engine drives destage/stage
  - Other processes can also drive destage/stage
- Destaging removes inodes
- Destaging removes extended attributes

Using Lustre as Performance Tier

Hierarchical Storage Management

Data Migration Facility Architecture

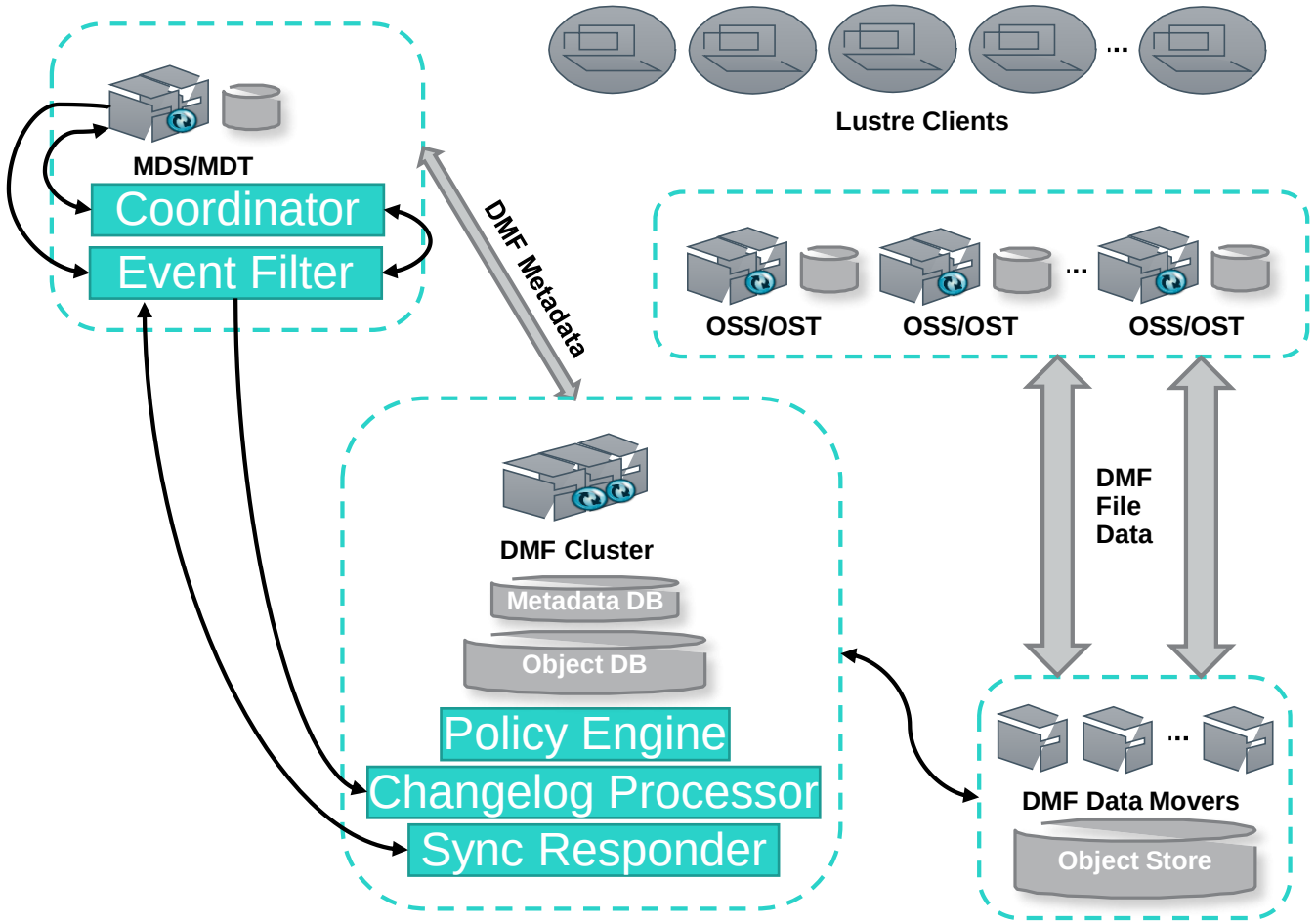


Current Implementation

- Robinhood is the Policy Engine
- A Coordinator runs on the Lustre MDS
- The Coordinator drives the Copytool
- The DMF MDS and Data Movers are part of a CXFS cluster and Lustre clients
- DMF FID mapping database lives on the CXFS filesystem
- DMF Data Movers copy directly between the OSS nodes and the backend storage
- Put stages small files on the CXFS filesystem for performance

Tiered Data Management

Data Management Framework Architecture



Integrate more tightly with Lustre

- DMF Event Filter
  - Consumes Lustre Changelog
  - Handles HSM Coordinator requests
  - Populates Asynchronous Events Queue
  - Handles Synchronous Events Queue
- DMF Cluster nodes run
  - Policy Engine
  - Changelog Processor
  - Sync Responder
- DMF Changelog Processor handles Asynchronous Events Queue
- DMF Sync Responder handles Synchronous Events Queue
- DMF cluster nodes direct the Data Movers
- DMF Data Movers are simple Lustre clients

Optimizing Lustre for use as Performance Tier

Speed over Size

MDS / MDT

MDS

– Also runs the Event Filter

– Metadata performance is limited by speed of Changelog Consumption

– Use DNE2 to provide multiple MDSs

– Use Multi-Rail

– Single-socket with fast CPU may be best

MDT

– Aggressive destaging of inodes saves MDT space

– Use fastest affordable hardware: (NVMe) SSD

OSS / OST

OSS

– Use Multi-Rail

– Single-socket with fast CPU may be best

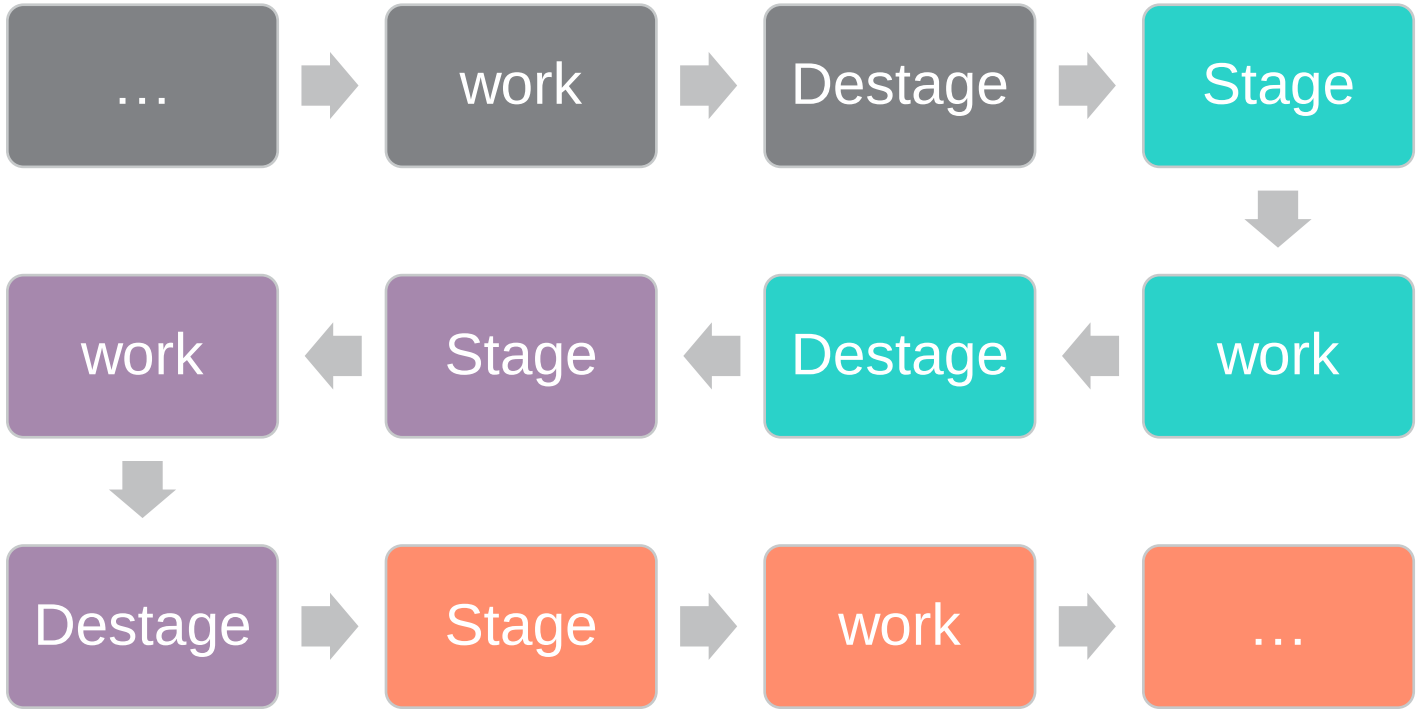
OST

– Use fastest affordable hardware: (NVMe) SSD

– Redundancy is less important

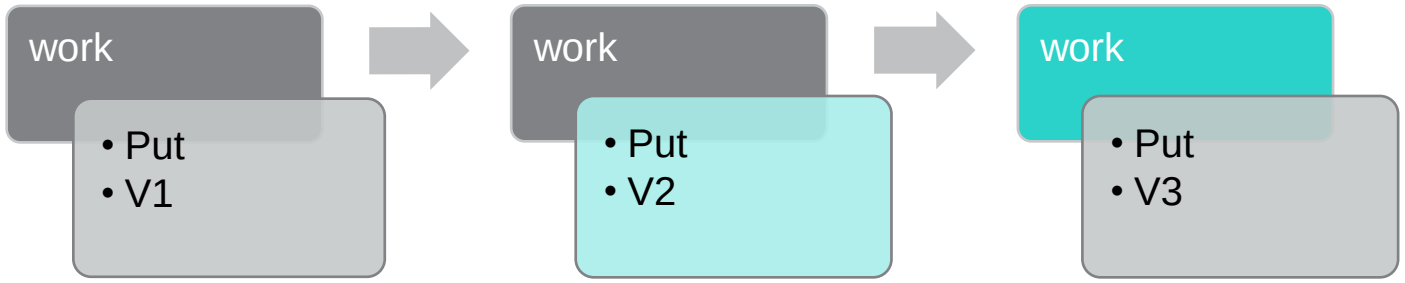
Using Tiered Data Management

Hierarchical Storage Management with Fewer Inodes



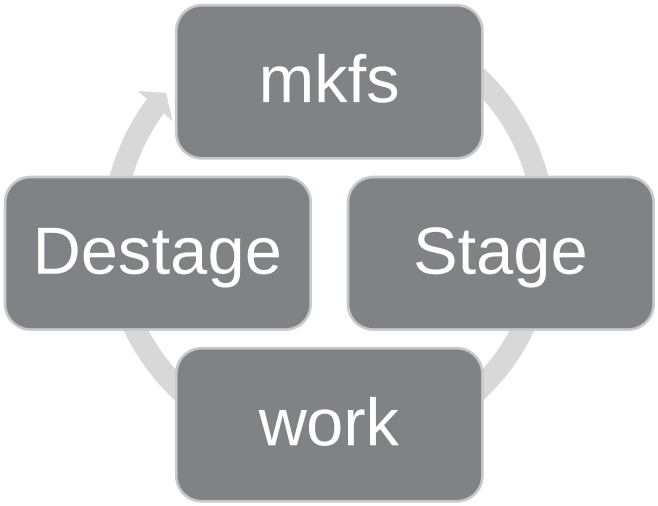
- Advanced HSM
- Number of inodes is an issue on Performance Tier
  - Full filesystem traversal is expensive
- Destage unneeded files
- Stage required files
- Backend Storage serves as an archive of Performance Tier

Versioned Files and File sets



- The Object Database supports versioning of files
- Each Put creates a new version
- Create file sets with matching versions
- Stage specific versions of files or a file set
- Applications need not be versioning-aware

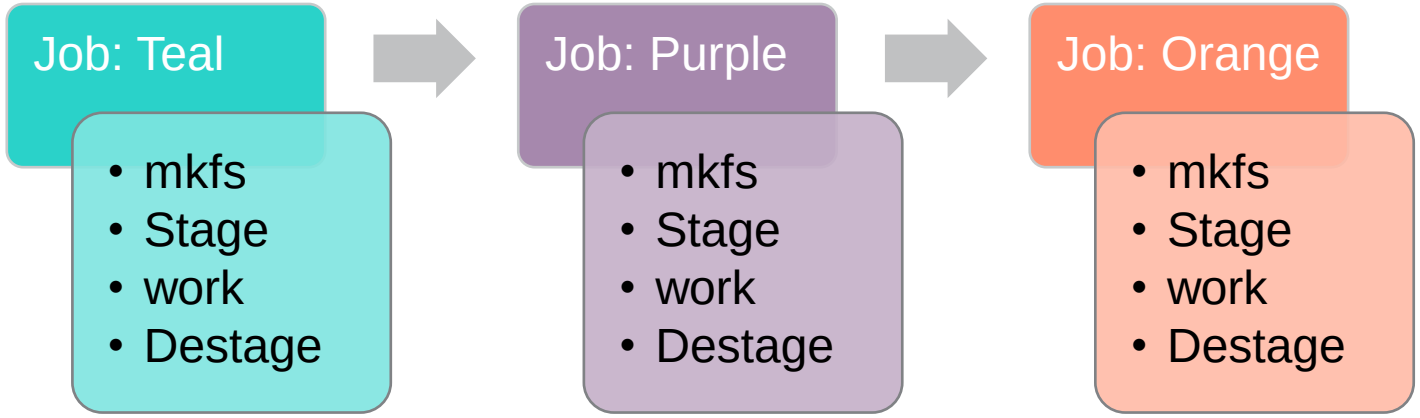
Dynamic Filesystems



Create filesystems on an as-needed basis

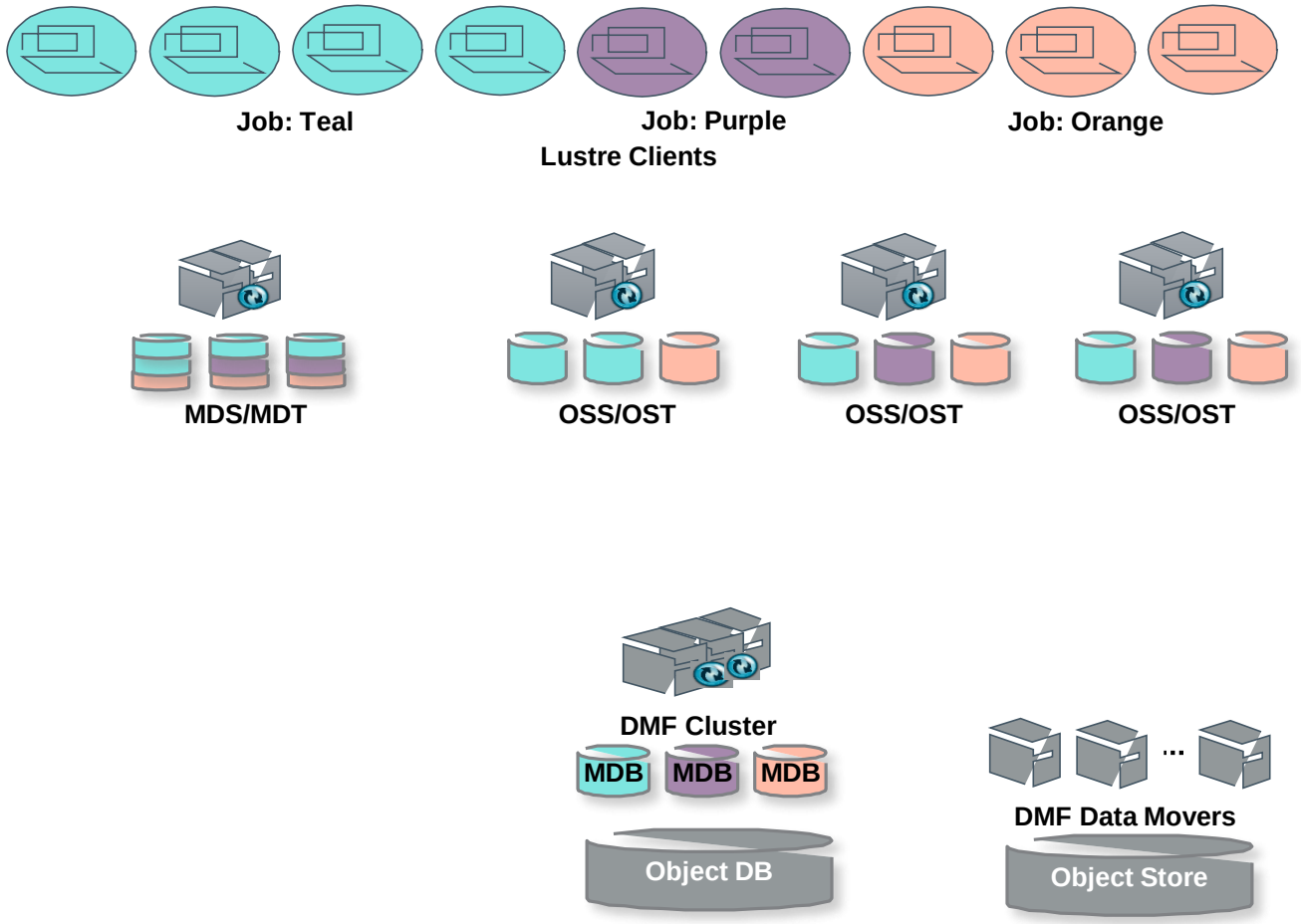
- Destage entire Performance Tier filesystem
- Filesystem can be rebuilt from scratch
- Query Object Database to populate the filesystem's Metadata Database
- Any object metadata is usable to select populace
  - Add metadata tags to taste
- Multiple filesystems
  - Centralized object database tracks data
  - Per-filesystem metadata database tracks migration status

Per-Job Filesystems



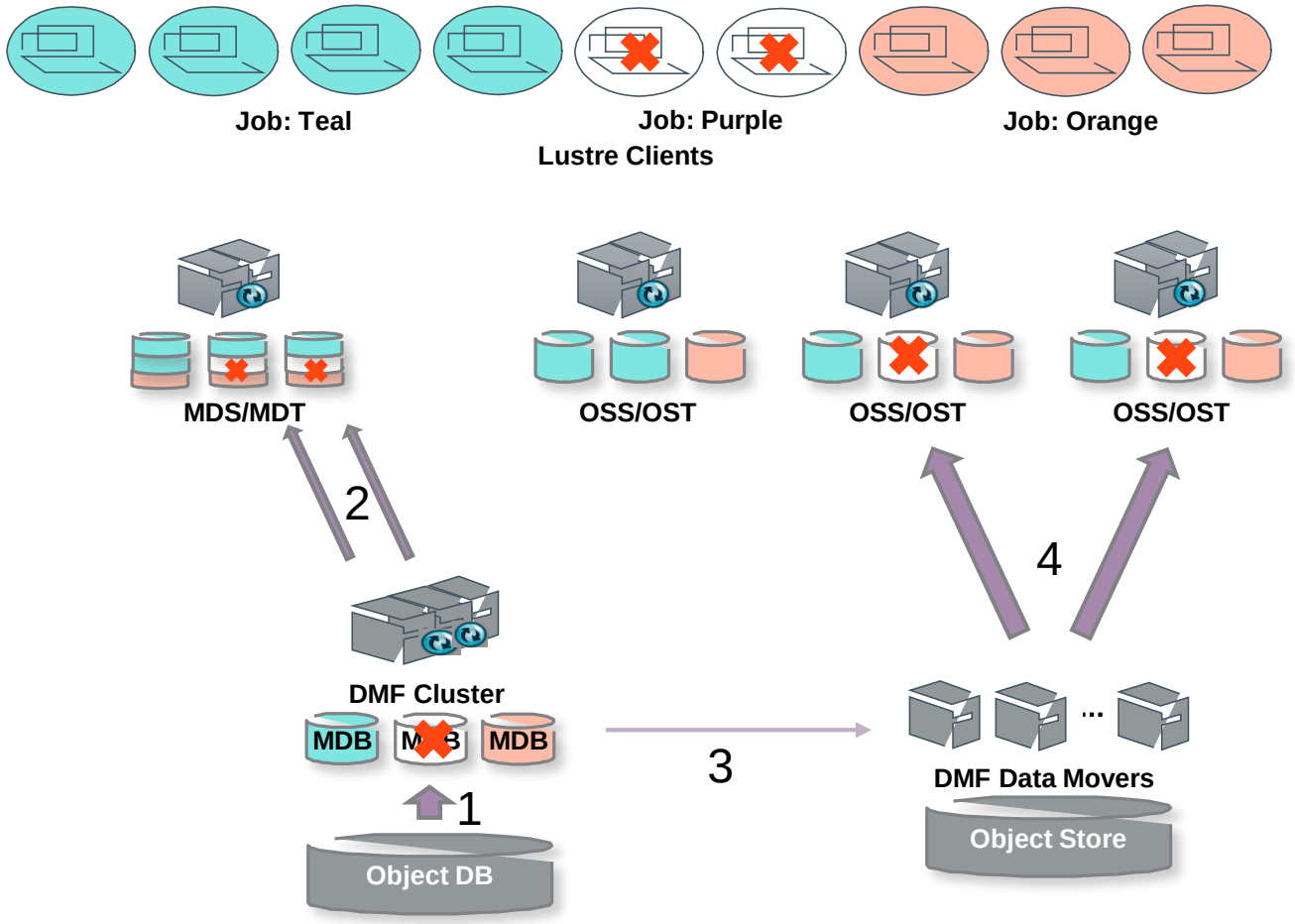
- Batch scheduler creates new a Dynamic Filesystem for each job
- Only files relevant for the job are Staged
- No accidental access to unrelated data
- No changes needed in Lustre, Linux, or applications

Simultaneous Per-Job Filesystems



- Multiple jobs run at the same time
- Multiple Dynamic Filesystems
- MDT/OST space is a schedulable resource
- Job scheduler manages
  - CPU cores
  - Memory
  - MDT space
  - OST space
- A way to simplify scheduling
  - Each job gets one or more nodes
  - Tie each MDT and OST space to a node
- Scheduling issues similar to NUMA-aware scheduling on big iron

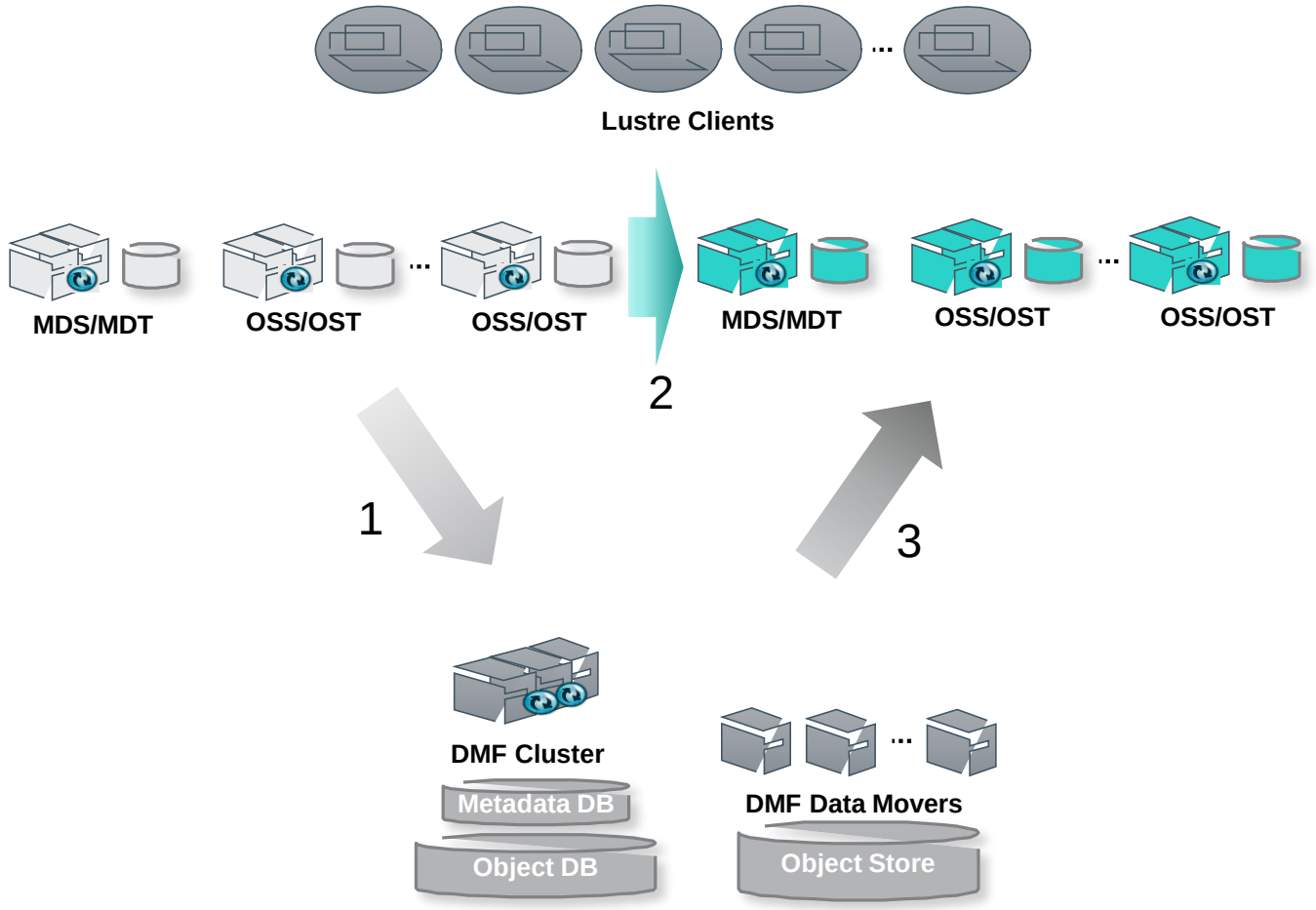
Fault Isolation



Each Dynamic Filesystem has its own

- Event Queues
- Daemons
- Metadata Database
  - Tracks directory tree of *this* filesystem
  - Tracks migration state of files in *this* filesystem
  - As opposed to the global
    - Object Database
    - Object Store
- Recreating Purple Dynamic Filesystem
  1. Object DB query creates Purple MDB
  2. DMF creates Purple MDT, stages files
  3. DMF directs Data Movers
  4. DMF Data Movers get files into Purple OSTs

Replacing Performance Tier



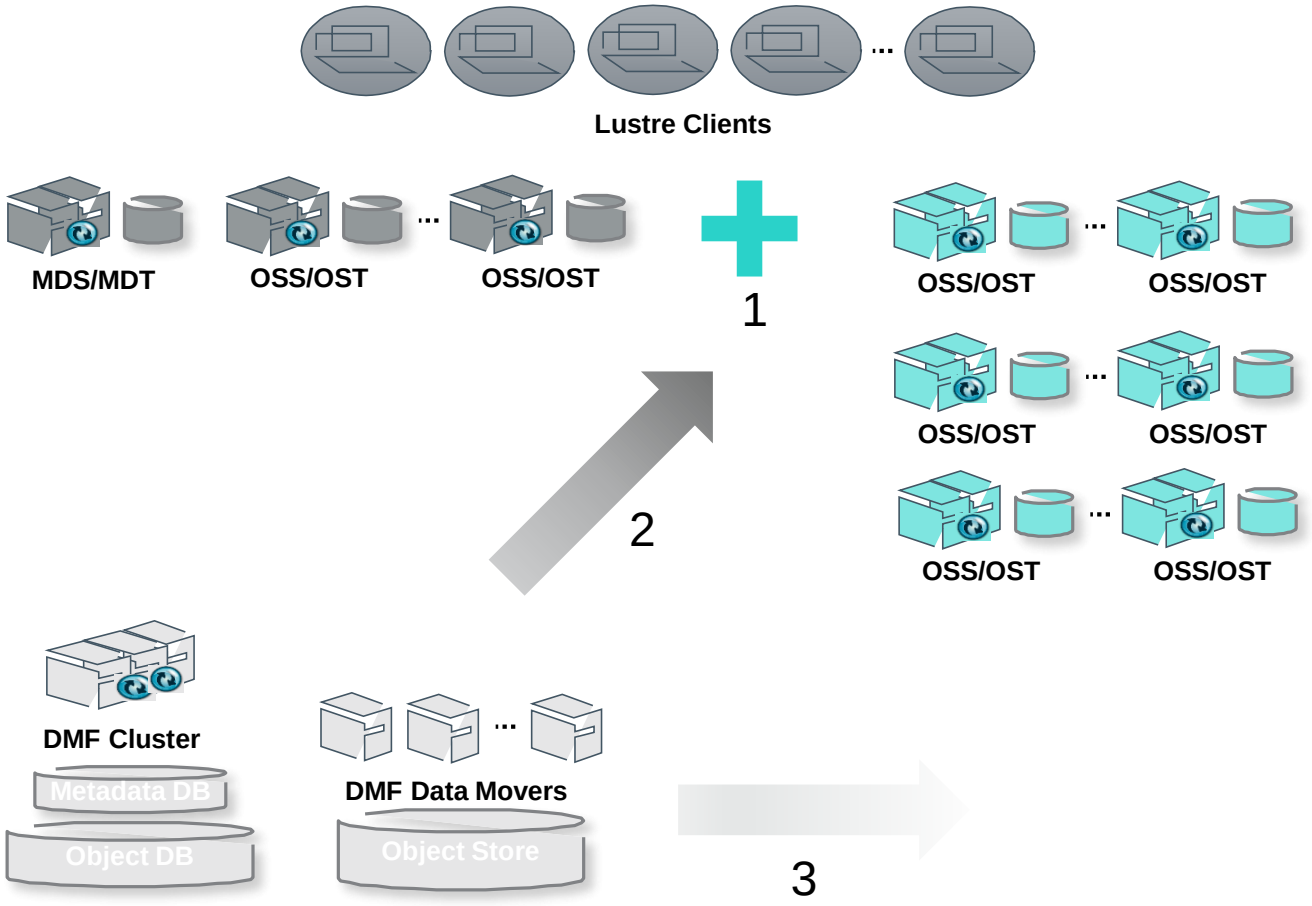
Performance Tier can be replaced

– Change to new supported filesystem type

– Migrating to new hardware

1. Destage all filesystems
2. Replace hardware
3. Stage on new hardware

Lustre Prevents Lock-in to Proprietary Software



Lustre scales really well to large filesystems

– Migrating into Lustre filesystem
  1. Add OSTs to obtain required capacity
  2. Stage everything in the Lustre filesystems
  3. Disconnect DMF systems
– Data remains available for use during and after this process

Conclusion

Status of Work

– Implementation on top of CXFS in Data Management Framework 7

– Implementation on top of Lustre planned for DMF 7.1

– There is an issue with FIDs changing when using DNE2 with multiple MDSs that needs to be sorted out

Summary

– Tiered Data Management is an evolutionary change from Hierarchical Storage Management

– It reduces the size of the managed filesystem

– This allows for faster hardware to be used

– Dynamic Filesystems enable new ways to use the stored data without changing applications or OS

– With its built-in scalability and parallelism Lustre is an excellent match for Performance Tier

Hewlett Packard
Enterprise

Q&A

Hewlett Packard
Enterprise

Thank you

Olaf Weber
olaf.weber@hpe.com