Intel® Lustre* system and network administration

Data lifecycle management with Lustre

11.2016

Backup and restore

Options for backing up Lustre

Enterprise backup utilities - POSIX file system level

- Amanda, NetBackup, Tivoli Storage Manager (TSM), HP Dataprotector
- Have support for EA, but not for Lustre distributed EA

Familiar utilities that support EA

- GNU tar (1.25+) and rsync each support EA

File level backup - target mounted type ldiskfs

- Use either tar -xattrs command or [get|set]attr to backup/restore EA

Backup targets at the raw device level

- Use a utility like dd

Just backup the target configuration

- Used for Disaster Recovery purposes for lost OSTs
- Collect small amount of configuration data once, but no "file" data

Using Enterprise Backup Utilities

Run backups from:

- Lustre client
- NFS / SMB client, if Lustre file system is re-exported

Must support backing up Extended Attributes

- Otherwise, restored files will use default striping policies

Consider backing up in parallel from 2+ clients

Consider specific strategies for full backup and incremental backup for a multi-petabyte file system

Changelog based backup

Scanning a multi-million lustre file system could be a challenge for an Enterprise Backup software.

The changelogs feature records events that change the file system namespace or file metadata.

Changes such as file creation, deletion, renaming, attribute changes, etc. are recorded with the target and parent file identifiers (FIDs), the name of the target, and a timestamp.

Fid2path to identify the path of the single object modified

Using Familiar Backup Utilities

Both tar and rsync support EA

- Tar
  - Codebase accepted into FC13 (RHEL6)
  - Upgrade packages available for RHEL5
- Rsync
  - EA support added in even earlier
- Dump
  - May also support EA

Restores:

- Stripe count, size and pool attributes on files
- Striping policies on directories

Extended Attributes (EA)

Recall that Lustre files contain Eas

EAs on MDT provide:

- Stripe count, stripe size, OST striped across, pool attributes on files
- Striping policies on directories
- EA parameter is trusted.lov

EAs on OST provide:

- MDT inode number and stripe index of Lustre files
- EA parameter is trusted.fid
  - Can be examined with: /usr/sbin/ll_decode_filter_fid

No EA == no Lustre metadata

Performing file-level backups

Backup / restore objects stored on MDT / OST file systems

- Lustre must be stopped (or a snapshot must be used)
- EA may be backed up using `tar -xattrs` if trusted
- This example uses commands from the attr package:

```
# umount /mnt/mdt
# mount -t ldiskfs /dev/vdb /mnt/mdt
# cd /mnt/mdt
( # getfattr -R -d -m '.*' -P > /tmp/ea.bak ) # not necessary with modern tar
# tar zcf --xattrs /tmp/mdt-backup.tgz *
```

- To restore from the backup file and remove old logs, run:

```
# mount -t ldiskfs /dev/vdb /mnt/mdt
# cd /mnt/mdt
# tar zxf /tmp/mdt-backup.tgz
( # setfattr -restore=/tmp/ea.bak )
# rm OBJECT/* CATALOGS   <--- Not to be run on OST "O" directory!
```

Mounting Storage Targets

Lustre storage targets (MGT, MDT, OST) can be mounted more than one way:

1. As a component of an active Lustre file system
   - `mount -t lustre`
   - Contents become hidden from browsing (mode: 0000)
   - Lustre kernel threads (services) start up
2. Separate from the active Lustre file system
   - `mount -t ldiskfs`
   - The contents are available for browsing on the server
     - This is typically done for backup or maintenance
   - No services start
   - All Linux disk tools work when mounted ldiskfs
     - Lustre provides a lustre-patched e2fsprogs

Can you guess what this command does?

```
mount -t lustre -o nosvc
```

Backup using *dd* command

Important Note:

- Using '`dd`' is the only supported method for Lustre 2.0–2.1
- Requires the target to be unmounted
- See Lustre Operations Guide for details
- Performs a raw device level backup
  - Result is an image the size of the LUN

Backup target (OST) configuration

Background

- What happens when replacing a failed OST with a new OST ?
  - Adding an new OST leaves gap in the OST index sequence
  - MDS unable to direct client to correct OST – files lost

Without backup of OST configuration

- Still possible to replace failed OST with new OST using same index
- Requires some Lustre magic
- Process can be simplified with backup of the original (and static) OST configuration

With a backup of the OST configuration

- The process to replace the OST is very simple

Details of the process can be found in the Lustre Operations Manual

Reasons not to backup data

Only scratch data

Don’t have enough disk or tape media

- Identify critical areas and perform limited, incremental backup

Data regeneration window is shorter than restoration time from backup

Restore window for single files is shorter than full-file-restore

Consider disk-based incremental backup from direct-attached client (or server)

HSM

What is HSM?

HSM = Hierarchical Storage Management

General concept is to move data between high-cost storage (and usually high-performance) and low-cost storage (usually lower-speed, higher capacity)

Goal is to increase overall capacity at a lower total cost

Beyond the concept and goal there are many details

- Does the data movement happen automatically?
- Can the user still access the file? If so, how?
- How are policies created/managed to move data?
- How is the data moved?

Lustre* gained HSM capability in release 2.5.0

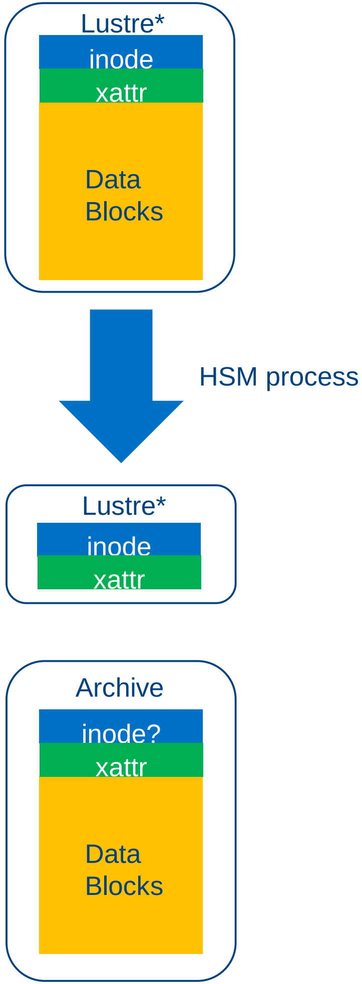
Anatomy of a file and HSM

Three basic components to a Lustre* file:

- inode (metadata – permissions, times)
- xattr (extended attributes – striping layout (lov))
- Data blocks (data)

At a minimum Lustre* HSM archives the data blocks and the xattr information

- The POSIX Copytool that comes with IEEL archives the data and the xattr information
- Other Copytools may also archive the inode (metadata)
  - In the diagram to the right this is listed with a "?" This indicates that archiving the inode is optional (depends upon the copytool)



Anatomy of a file and HSM

A file stub and the xattr information stays in place on Lustre*

- inode stays on primary storage (about 2K in size)
- Typically this is called a stub
- After a period of time, the data blocks on primary storage are erased

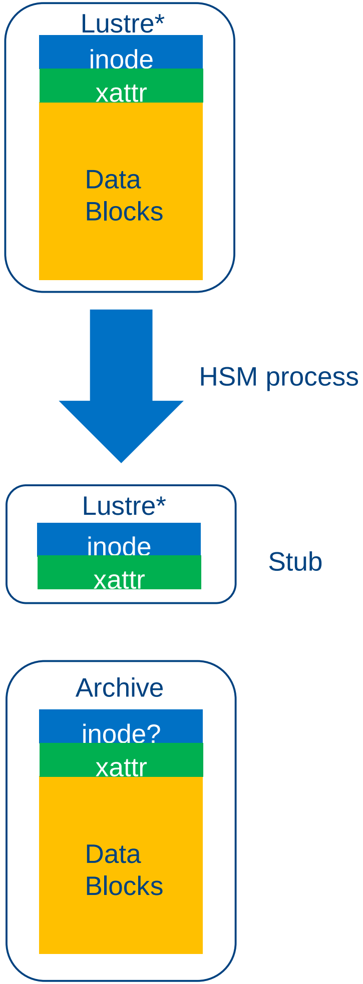
Files can be “restored” when needed (automatic)

Some HSM systems allow variable size stubs

- Lustre* has a fixed stub size

Some HSM systems could also copy the inode to the archive storage

- Adds data protection
- Can be problematic because of metadata updates
- Depends upon the copytool



How HSM works in Lustre*

At a high-level, data+xattr is migrated from primary Lustre* storage to archive storage

- This is done asynchronously
- Can include the inode (depends upon the copytool)

The file stub (inode) and xattr information remains on Lustre*

- Initially, two copies exist: the original file on Lustre* and the replica on the archive
- As data is written to the Lustre* file system, available capacity is consumed, reducing the amount of free space for new data
- When Lustre* runs out of space, or when a specified threshold is exceeded, older files are "released“
  - This means the original, local, copy is deleted from Lustre* and replaced with a stub file that points to the file on the archive.

Lustre* HSM observations

HSM does **not** represent a backup solution

- In general, the archive tier may not version data (backups are about versioning)

HSM does not guarantee that a file that has been permanently deleted from the high-performance tier can be retrieved from the archive tier

- Behavior is policy dependent
- In general if a file has been purged from Lustre* it will be *eventually* purged from the archive

Can be used for disaster recovery (DR) but requires careful design and process creation

- Need a copytool that archives the metadata (and keeps it in synch)

Lustre* HSM observations – cont'd

The user can interact with the file stub using normal commands such as ls, rm, cp, chmod, etc.

- If the command only requires metadata information then the data blocks and xattr information do not migrate back from the archive (restore)
- If the command requires access to the data blocks, the data blocks are copied back from the archive storage to the primary storage (Lustre*)
  - It will appear that the file system is "sluggish" because the data blocks have to be migrated back (i.e. increased latency and reduced throughput)
  - This is to be expected since the archive target is slower storage

Applications do not need to be rewritten to take advantage of HSM

- Happens "behind the scenes"

What does Intel EE Lustre provide?

IEEL provides and supports tools to enable HSM solutions but **does not** provide a complete HSM solution

- Lustre 2.5.x
- POSIX Copytool
- Robinhood (policy engine)
- HSM Monitoring is integrated into Intel Manager for Lustre* (IML)

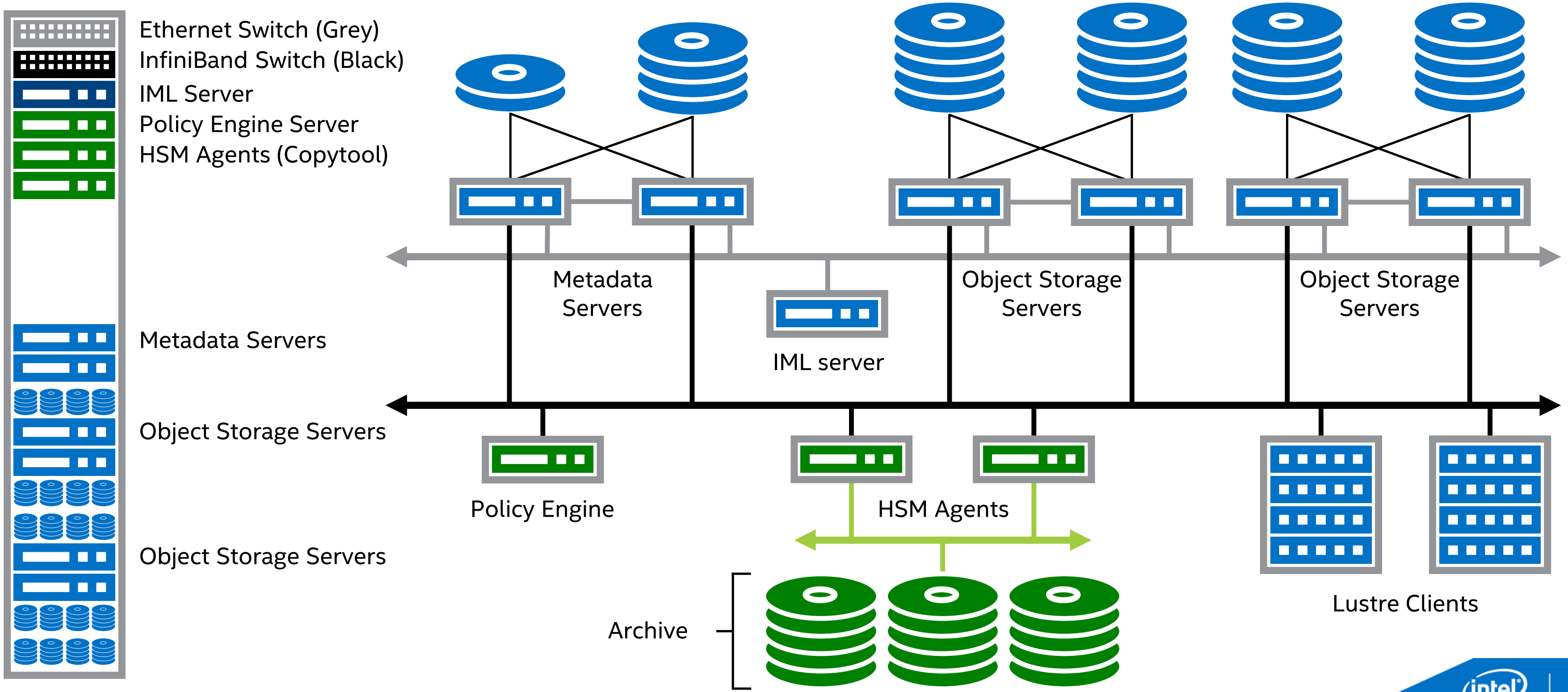
The POSIX copytool only works with POSIX archive storage

- The POSIX copytool does not copy inode data to the archive
- The POSIX copytool is an example not designed for production

The archive storage is **not** provided

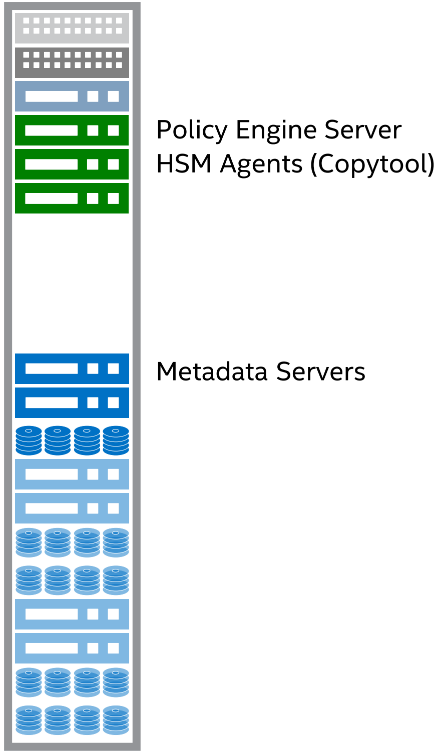
- Vendor specific (possibly including the copytool). Several possible solutions

Example: Intel EE Lustre file system with HSM



Hierarchical storage management

COTS appliance: single rack lustre file system with HSM



HSM Coordinator

- Service thread running on Metadata Servers that manages the queuing and dispatch of HSM requests
- Sends commands to Copytools on the HSM Agents

HSM Agents (Copytool)

- Interface between Lustre file system and HSM archive storage tier
- Closely-coupled with the Archive – Copytools must be able to interact using the Archive API.
  - Archive vendors may supply their own copytool
- Lustre supplies a POSIX-compatible Copytool reference implementation

Archive (not pictured in rack)

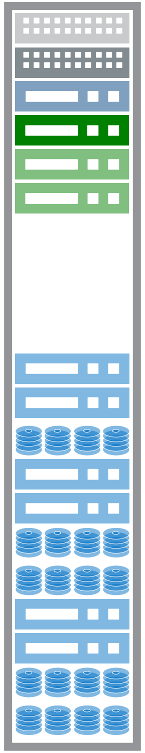
- High capacity storage tier for long term storage of non-volatile data. Common options are tape, disk, cloud

Policy Engine

- HSM management and monitoring software, provides automation of HSM tasks
- Robinhood provides Lustre policy engine reference implementation

Robinhood Policy Engine

COTS appliance: single rack lustre file system with HSM



Policy Engine Server

Robinhood policy engine provides task automation for file system archiving

Reference implementation for an HSM policy engine on Lustre

Consumes MDT Changelogs to monitor changes on the Lustre file system

User-defined policies describe the sets of files to manage, the conditions under which to initiate actions (archive policies, purge triggers)

Supplied with tools to generate reports

Runs on a Lustre client

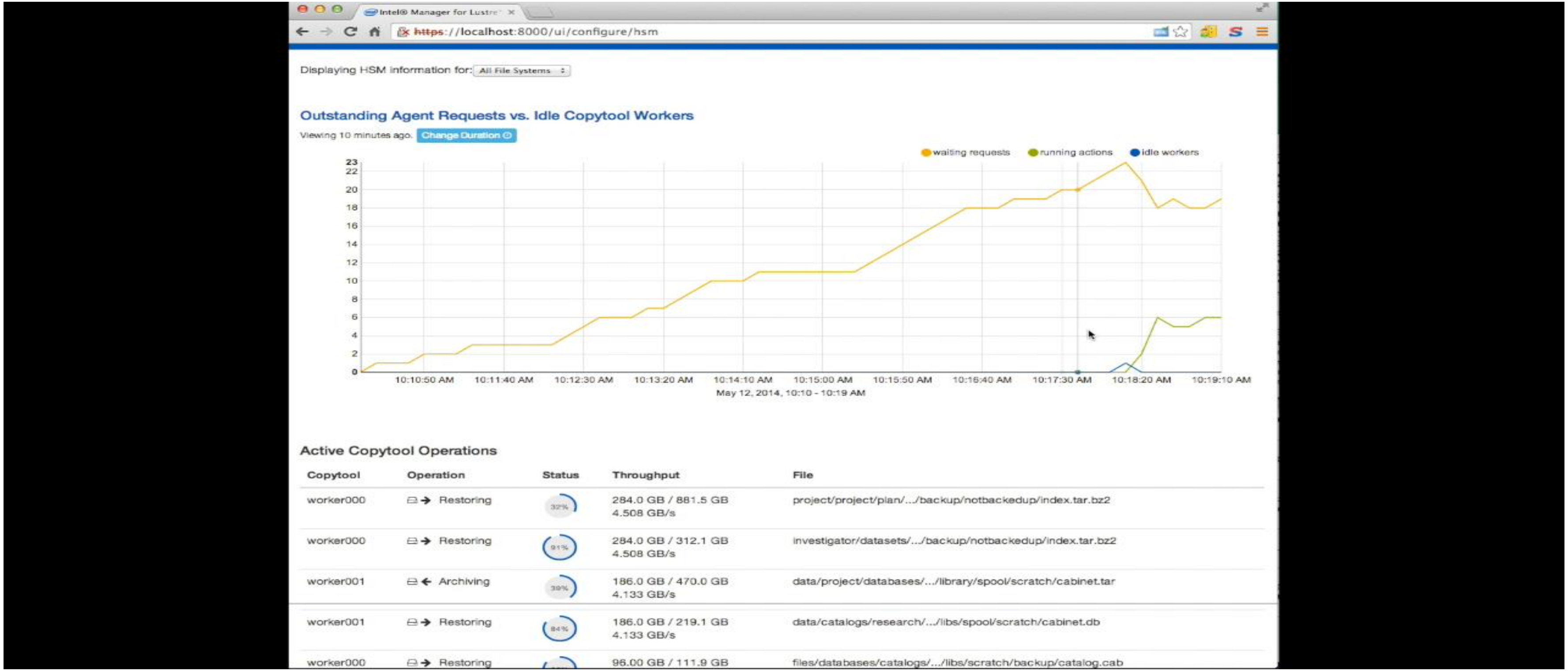
IML HSM Monitoring

IML has the ability to monitor the throughput of the copytools

- Pulls data from MDT Coordinator

Simple monitor without any statistics

IML HSM Monitoring



Displaying HSM information for: All File Systems

Outstanding Agent Requests vs. Idle Copytool Workers

Viewing 10 minutes ago. Change Duration

Active Copytool Operations

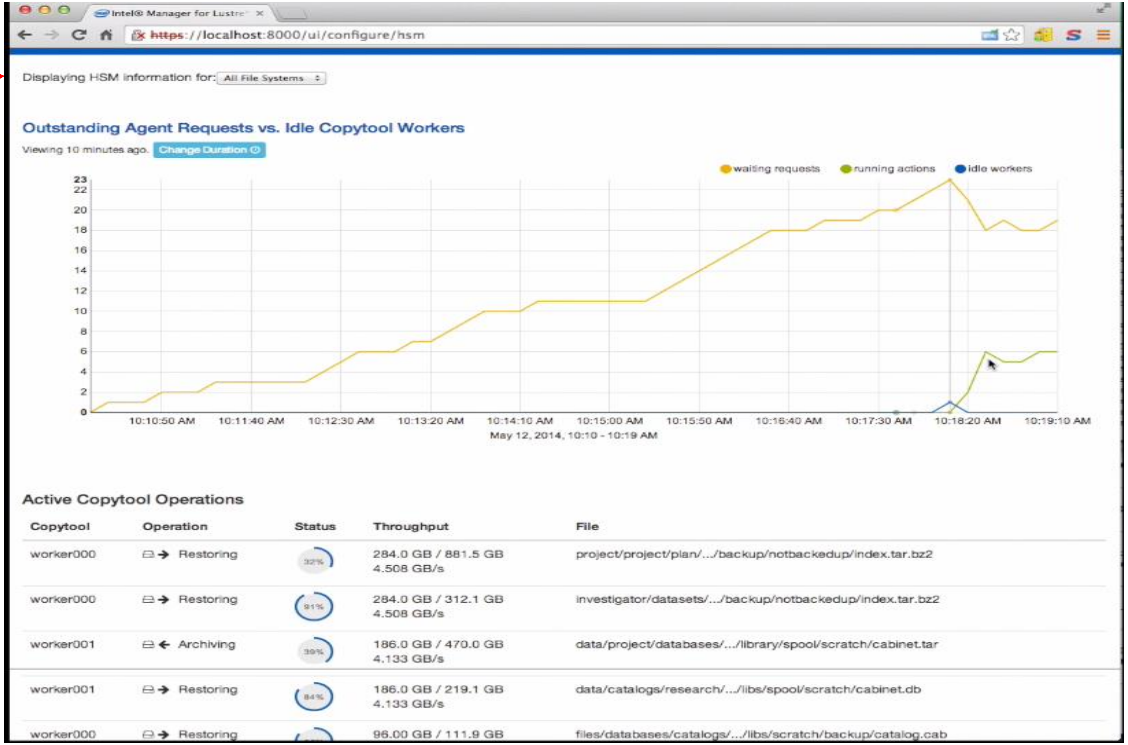
Copytool	Operation	Status	Throughput	File
worker000	→ Restoring	32%	284.0 GB / 881.5 GB 4.508 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker000	→ Restoring	91%	284.0 GB / 312.1 GB 4.508 GB/s	investigator/datasets/.../backup/notbackedup/index.tar.bz2
worker001	← Archiving	39%	186.0 GB / 470.0 GB 4.133 GB/s	data/project/databases/.../library/spool/scratch/cabinet.tar
worker001	→ Restoring	84%	186.0 GB / 219.1 GB 4.133 GB/s	data/catalogs/research/.../libs/spool/scratch/cabinet.db
worker000	→ Restoring		96.00 GB / 111.9 GB	files/databases/catalogs/.../libs/scratch/backup/catalog.cab

IML HSM Monitoring

Monitoring all File systems

- Number of requests (y-axis)
- Yellow are requests waiting
- Green line is number of requests running
- Blue is number of idle workers (copytool)

- Details of active Copytool operations:
    - Specific Copytool
    - Operation (restore/archive)
    - Status (how complete)
    - Throughput
    - Specific file



Displaying HSM information for: All File Systems

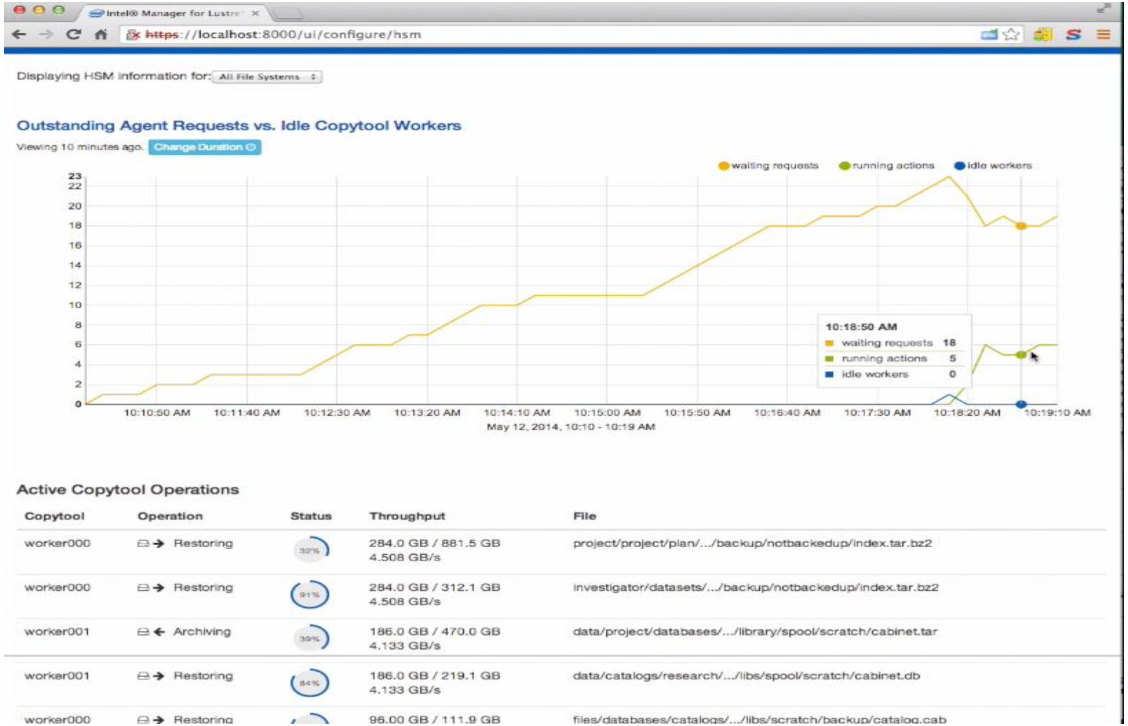
Outstanding Agent Requests vs. Idle Copytool Workers

Viewing 10 minutes ago.

Active Copytool Operations

Copytool	Operation	Status	Throughput	File
worker000	→ Restoring	32%	284.0 GB / 881.5 GB 4.508 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker000	→ Restoring	91%	284.0 GB / 312.1 GB 4.508 GB/s	investigator/datasets/.../backup/notbackedup/index.tar.bz2
worker001	← Archiving	39%	186.0 GB / 470.0 GB 4.133 GB/s	data/project/databases/.../library/spool/scratch/cabinet.tar
worker001	→ Restoring	84%	186.0 GB / 219.1 GB 4.133 GB/s	data/catalogs/research/.../libs/spool/scratch/cabinet.db
worker000	→ Restoring	[illegible]	96.00 GB / 111.9 GB	files/databases/catalogs/.../libs/scratch/backup/catalog.cab

IML HSM Monitoring – hover for details



Displaying HSM information for: All File Systems

Outstanding Agent Requests vs. Idle Copytool Workers

Viewing 10 minutes ago. Change Duration

10:18:50 AM

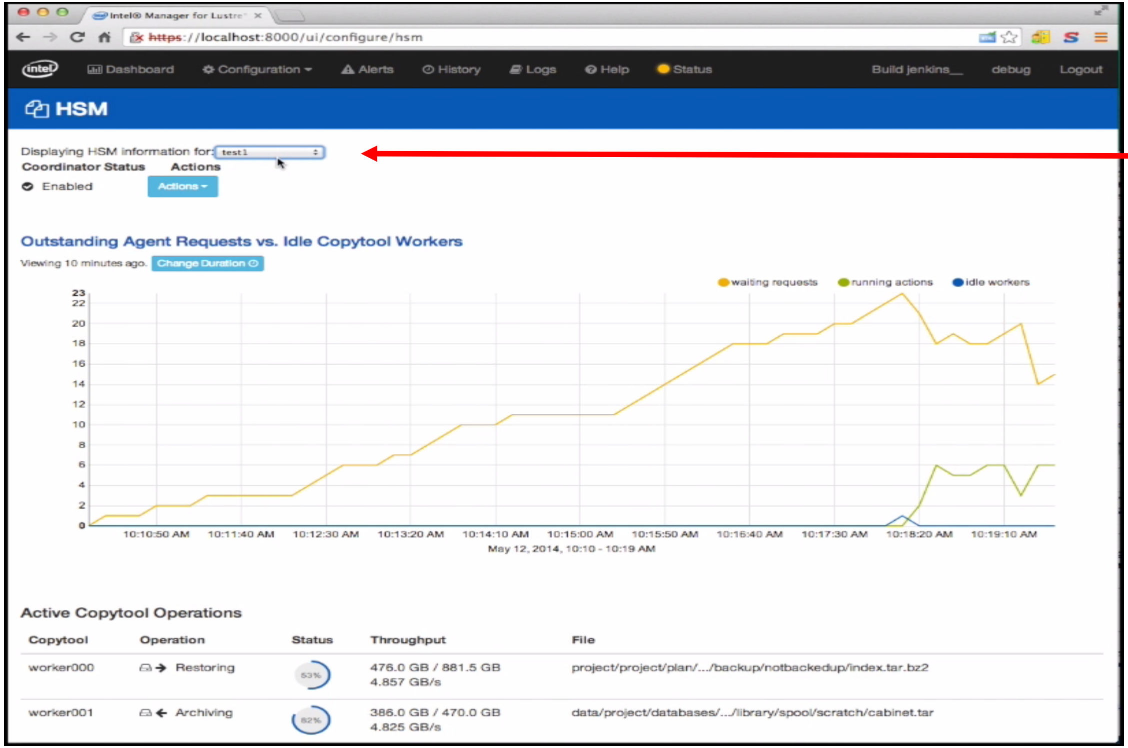
waiting requests	18
running actions	5
idle workers	0

Active Copytool Operations

Copytool	Operation	Status	Throughput	File
worker000	Restoring	32%	284.0 GB / 881.5 GB 4.508 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker000	Restoring	91%	284.0 GB / 312.1 GB 4.508 GB/s	investigator/datasets/.../backup/notbackedup/index.tar.bz2
worker001	Archiving	39%	186.0 GB / 470.0 GB 4.133 GB/s	data/project/databases/.../library/spool/scratch/cabinet.tar
worker001	Restoring	84%	186.0 GB / 219.1 GB 4.133 GB/s	data/catalogs/research/.../libs/spool/scratch/cabinet.db
worker000	Restoring		96.00 GB / 111.9 GB	files/databases/catalogs/.../libs/scratch/backup/catalog.cab

- Hover for details
    - How many waiting requests
    - How many running requests
    - How many idle workers

IML HSM Monitoring – File System Change



HSM

Displaying HSM information for: test1

Coordinator Status	Actions
Enabled	Actions

Outstanding Agent Requests vs. Idle Copytool Workers

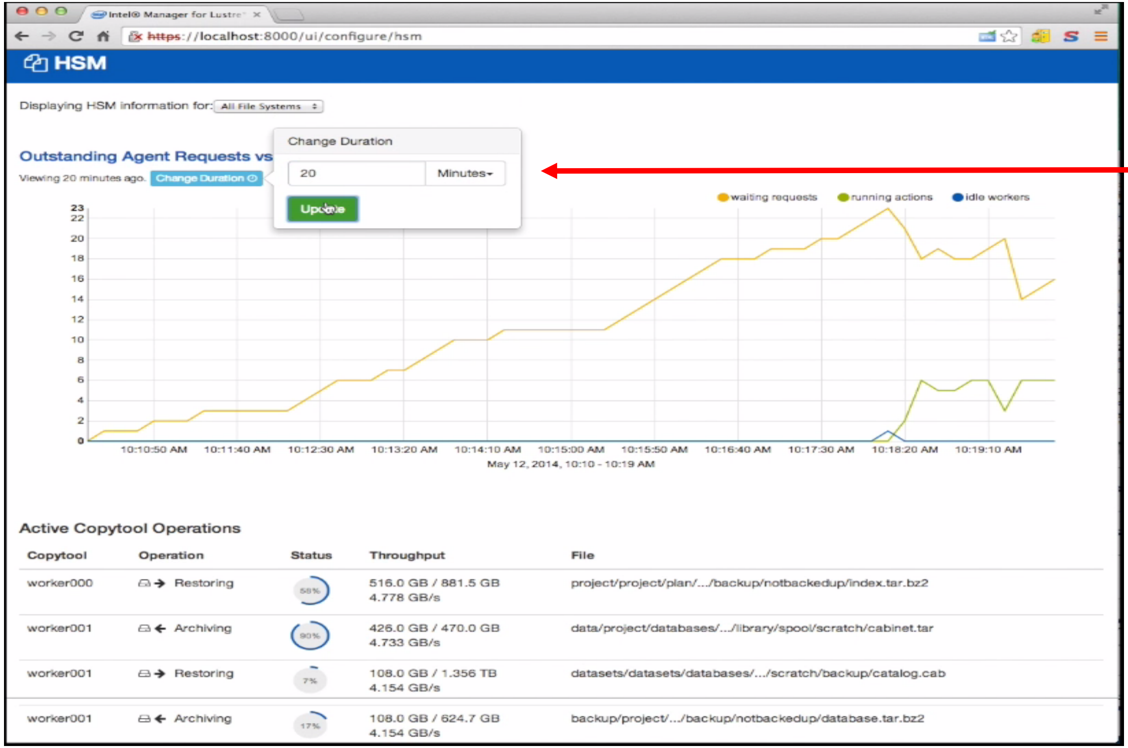
Viewing 10 minutes ago.

Active Copytool Operations

Copytool	Operation	Status	Throughput	File
worker000	→ Restoring	53%	476.0 GB / 881.5 GB 4.857 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker001	← Archiving	82%	386.0 GB / 470.0 GB 4.825 GB/s	data/project/databases/.../library/spool/scratch/cabinet.tar

- Switch file systems and get updated HSM information

IML HSM Monitoring – Time Scale (Duration)



Displaying HSM information for: All File Systems

Viewing 20 minutes ago.

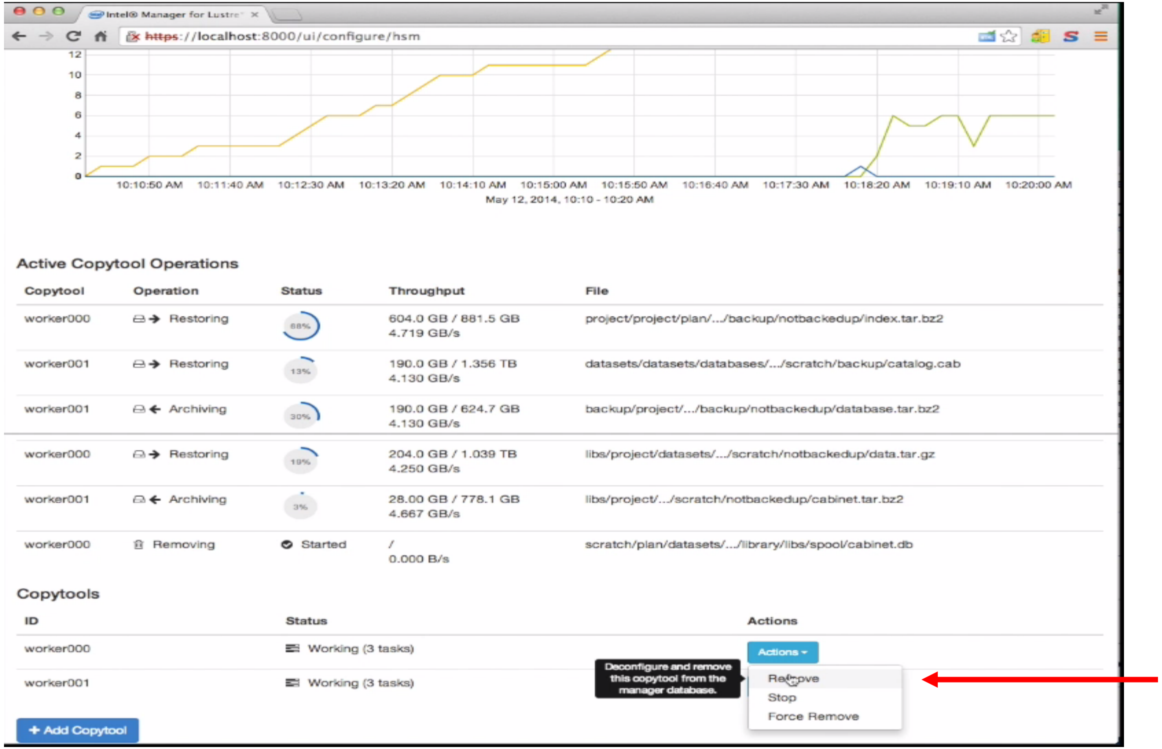
20

Active Copytool Operations

Copytool	Operation	Status	Throughput	File
worker000	→ Restoring	58%	516.0 GB / 881.5 GB 4.778 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker001	← Archiving	90%	426.0 GB / 470.0 GB 4.733 GB/s	data/project/databases/.../library/spool/scratch/cabinet.tar
worker001	→ Restoring	7%	108.0 GB / 1.356 TB 4.154 GB/s	datasets/datasets/databases/.../scratch/backup/catalog.cab
worker001	← Archiving	17%	108.0 GB / 624.7 GB 4.154 GB/s	backup/project/.../backup/notbackedup/database.tar.bz2

Change the time scale of what is shown
from 10 minutes to 20 minutes.

IML HSM Monitoring – Modify Copytool



Active Copytool Operations

Copytool	Operation	Status	Throughput	File
worker000	Restoring	68%	604.0 GB / 881.5 GB 4.719 GB/s	project/project/plan/.../backup/notbackedup/index.tar.bz2
worker001	Restoring	13%	190.0 GB / 1.356 TB 4.130 GB/s	datasets/datasets/databases/.../scratch/backup/catalog.cab
worker001	Archiving	30%	190.0 GB / 624.7 GB 4.130 GB/s	backup/project/.../backup/notbackedup/database.tar.bz2
worker000	Restoring	19%	204.0 GB / 1.039 TB 4.250 GB/s	libs/project/datasets/.../scratch/notbackedup/data.tar.gz
worker001	Archiving	3%	28.00 GB / 778.1 GB 4.667 GB/s	libs/project/.../scratch/notbackedup/cabinet.tar.bz2
worker000	Removing	✓ Started	/ 0.000 B/s	scratch/plan/datasets/.../library/libs/spool/cabinet.db

Copytools

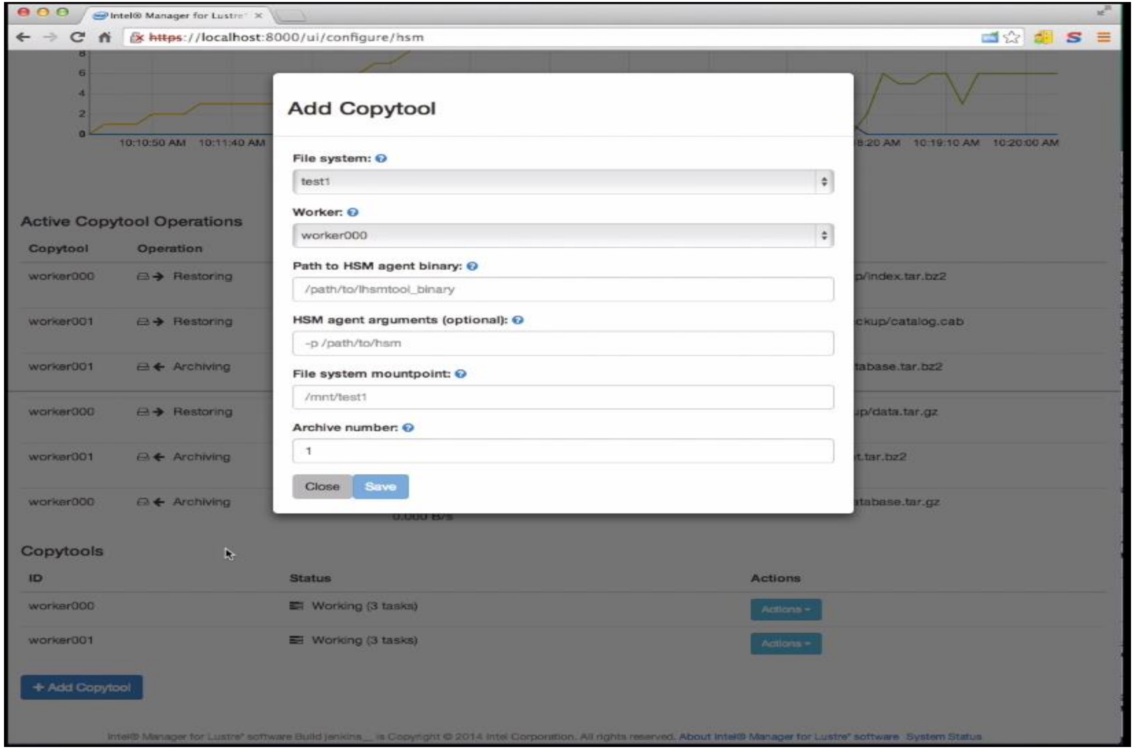
ID	Status	Actions
worker000	Working (3 tasks)	Actions ▾
worker001	Working (3 tasks)	

Deconfigure and remove this copytool from the manager database.
Remove
Stop
Force Remove
+ Add Copytool

Actions for various copytools.

- Remove, Stop, Force Remove
- Context sensitive help

IML HSM Monitoring – Add Copytool



Intel® Manager for Lustre

https://localhost:8000/ui/configure/hsm

Add Copytool

File system: test1

Worker: worker000

Path to HSM agent binary: /path/to/lhsmtool_binary

HSM agent arguments (optional): -p /path/to/hsm

File system mountpoint: /mnt/test1

Archive number: 1

Active Copytool Operations

Copytool	Operation	
worker000	Restoring	p/index.tar.bz2
worker001	Restoring	ckup/catalog.cab
worker001	Archiving	tabase.tar.bz2
worker000	Restoring	up/data.tar.gz
worker001	Archiving	t.tar.bz2
worker000	Archiving	atabase.tar.gz

0.000 B/s

Copytools

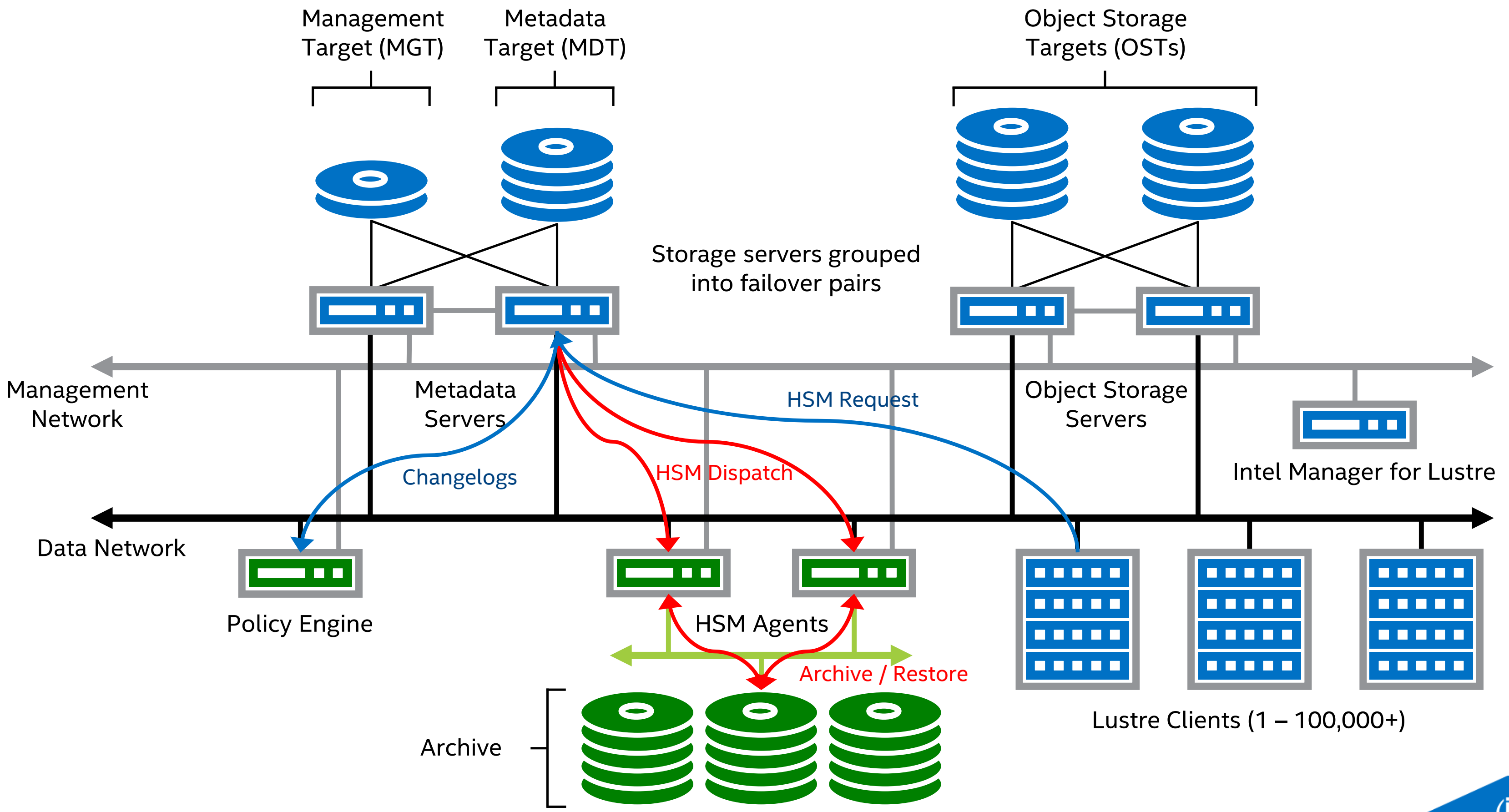
ID	Status	Actions
worker000	Working (3 tasks)	Actions ▾
worker001	Working (3 tasks)	Actions ▾

Intel® Manager for Lustre* software Build jenkins__ is Copyright © 2014 Intel Corporation. All rights reserved. About Intel® Manager for Lustre* software System Status

Add copytool

- Define the file system
- Worker node
- Path to HSM agent
- Specific arguments
- Where the file system is mounted
- Archive number (assigned)

Lustre HSM system architecture



Fundamental HSM components

Lustre* clients

- Issue HSM commands

HSM Coordinator

- Thread on MDS

Copytool

- Agent running on HSM server (data mover)
- Speaks both “Lustre*” and “Archive”

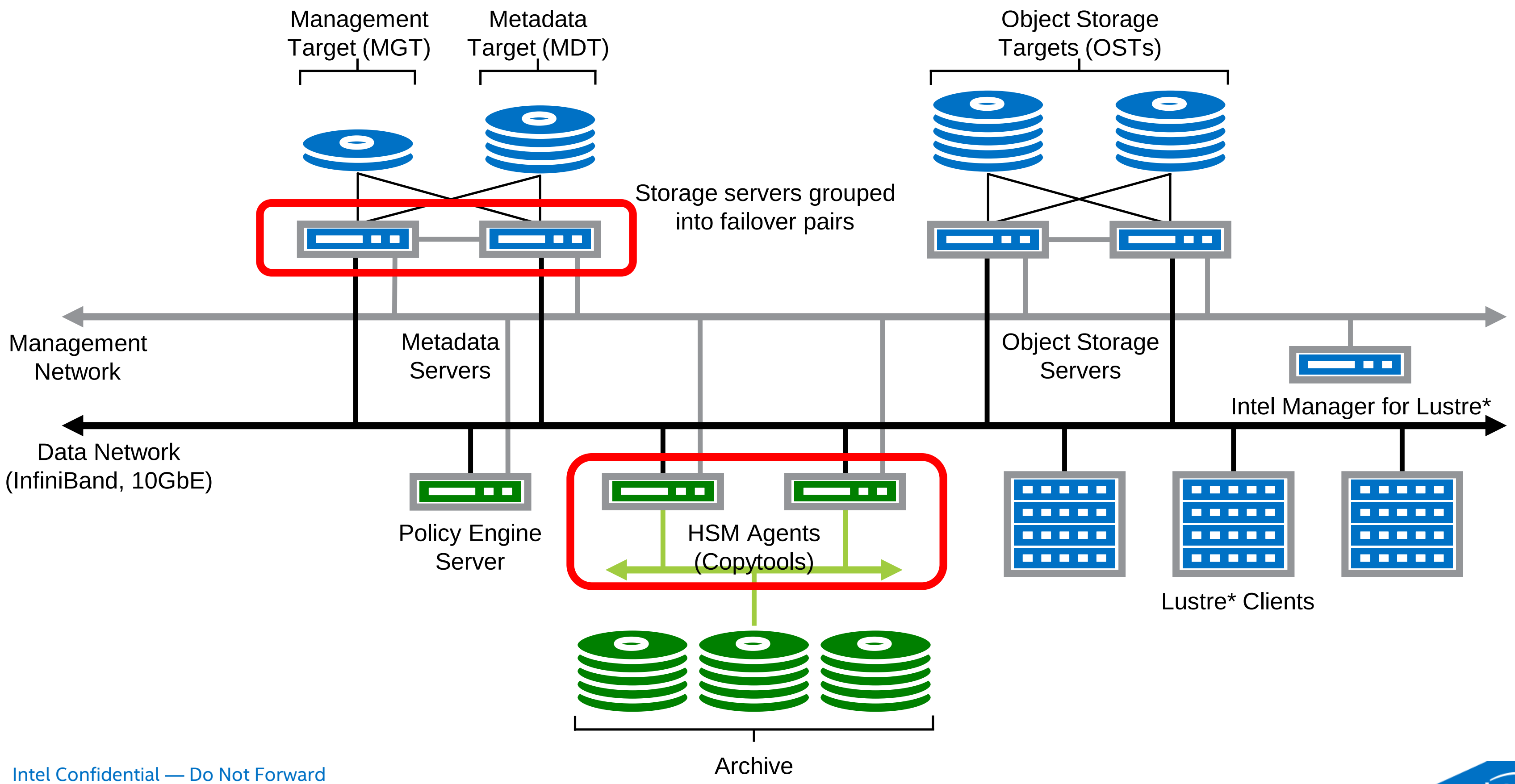
Policy Engine

- Allows policies to be created and automated for HSM

Archive Storage

- Where the data blocks are to be located
- Knows nothing about Lustre*
- Vendor specific

HSM Coordinator



HSM Coordinator

HSM support in the Lustre* file system is centered on two principal features

- Coordinator
- One or more copytool services

Coordinator is a thread running on the MDS (metadata server)

- Comes with Lustre* 2.5

Users and applications issue HSM requests to the MDT Coordinator

- User commands and policy engine commands (from Lustre* client)

Coordinator queues and dispatches the requests to the copytool server to move data to the archive

One Coordinator in each Lustre* file system

The Coordinator provides flow control, removes duplicate requests, and ensures that incomplete requests can be replayed in the event of a request failure.

Copytool

The copytool is an intermediary between the Lustre* file system and the archive storage

- Executes commands in response to requests from the MDT Coordinator

Copytool service is a daemon running on the HSM Agent server

- Lustre* client runs on HSM Agent server as well

HSM Agents are synonymous with Data Movers in other storage management environments

There can be many copytools

- Multiple copytools can improve the responsiveness of HSM requests and provide greater aggregate I/O
- Each copytool typically runs on a separate Agent host and registers with the MDT Coordinator when it is launched
- The MDT Coordinator maintains a list of registered copytools

It is possible for a single Lustre* file system to be served by multiple storage archives, each with multiple copytools

Copytool – cont'd

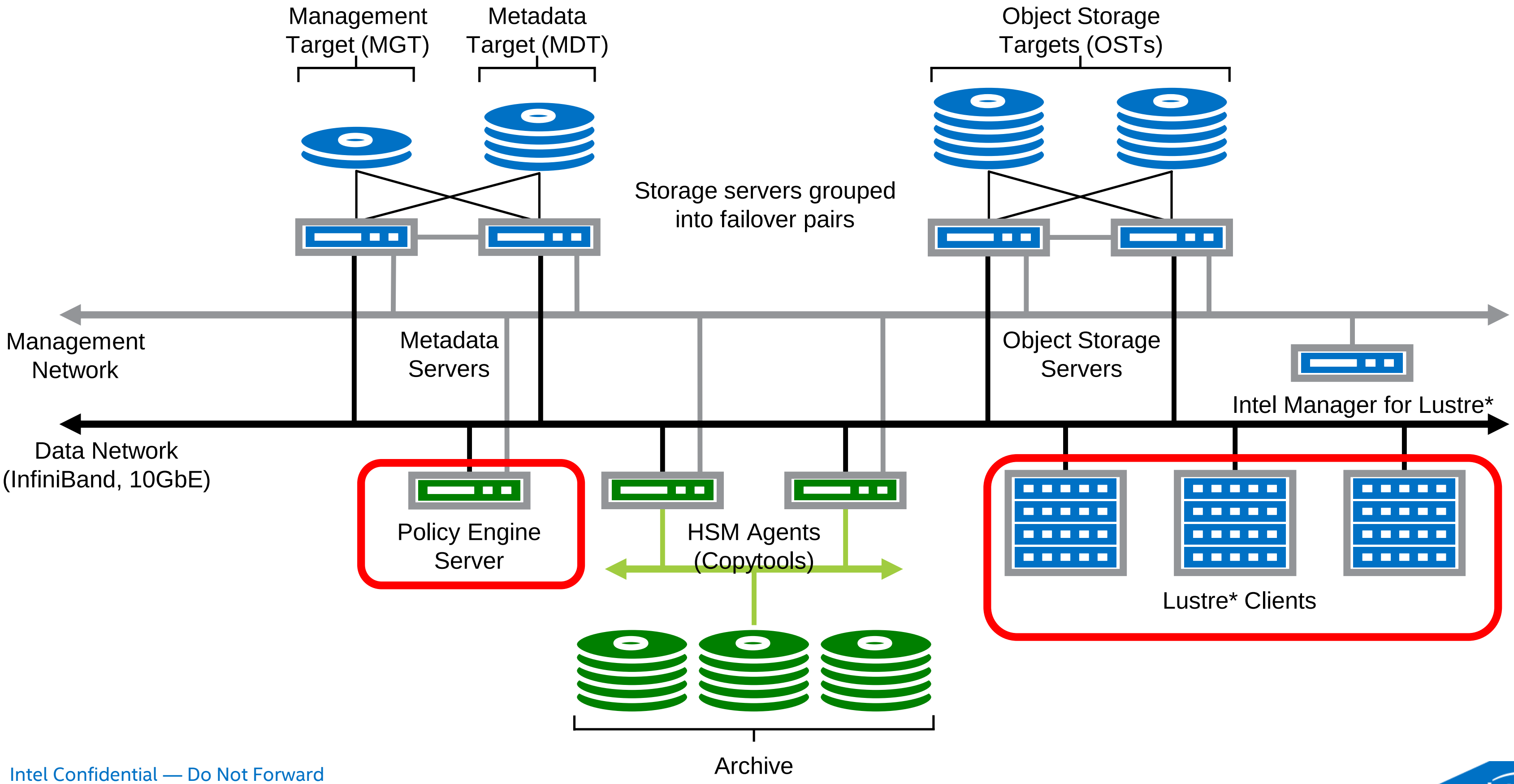
Copytool transfers data from the archive to Lustre* and vice versa

Therefore the copytool has to map files from Lustre* to the archive storage and back again

- **Each copytool is specific to the target archive and must implement the storage protocol appropriate to that archive system**
  - There is no universal copytool!
  - Lustre* comes with a sample POSIX copytool

Copytools are specific to the archive and come from the archive vendor

Lustre Clients



Lustre* clients

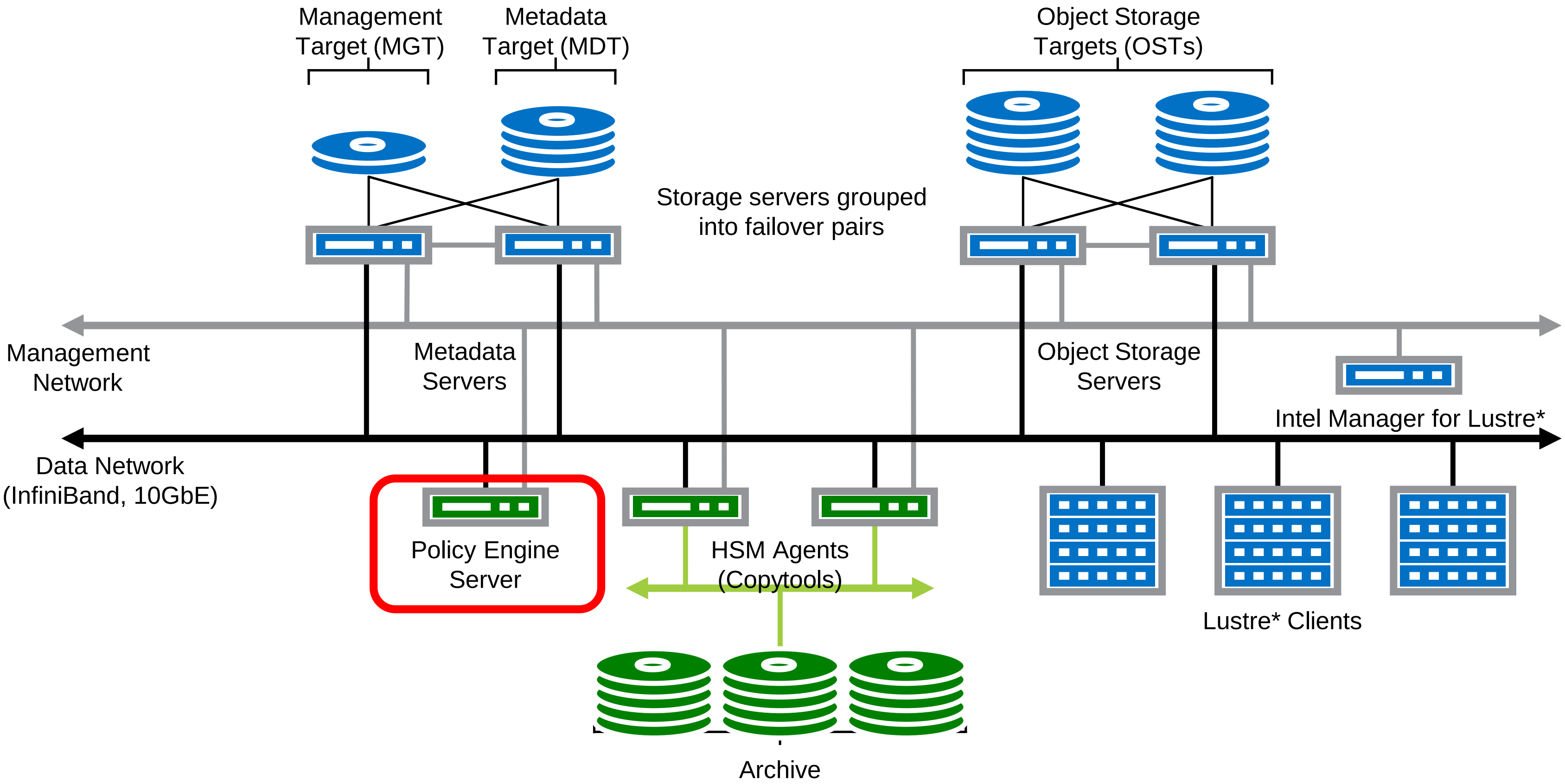
Lustre* clients have software to manage interaction with the HSM platform

- “lfs” command (user accessible)
- Commands allow users to archive, release and restore files, determine status
- A complete description of the command interface is available in the Lustre Operations manual

Applications running on Lustre* clients do not need to be modified to run on an HSM platform

Note: the Intel Manager for Lustre* (IML) is not a Lustre* client so it cannot issue the “lfs” commands

Policy Engine



Policy Engine

Strictly speaking the policy engine is optional

- Users can employ the command-line tools supplied with Lustre* to mark the files that they want to archive and can send commands to the MDT Coordinator to manage capacity by releasing files on a per-file basis
- This will quickly become cumbersome, particularly when working at scale

The Policy Engine is just a software tool where admins can define policies for Lustre* HSM

- How often files are archived
- The amount of active Lustre* storage capacity that a single user or group of users may consume
- Thresholds for file system or OST capacity that when exceeded, will trigger a purge of least frequently accessed data

The Policy Engine executes these policies by sending commands (as a user would) to the MDT Coordinator

Policy Engine – cont'd

The Policy Engine is not part of Lustre*

IEEL includes Robinhood

- 3rd-party, open source policy engine that has comprehensive Lustre* HSM support
- Project started by CEA in France
- Supported by Intel

Some archive storage vendors may instead provide their own policy engine software

- Only run one policy engine at a time

Robinhood Policy Engine Architecture

Robinhood goes beyond a simple HSM policy engine

- Has data management features such as querying the file system

Interacts with Lustre* via the changelogs

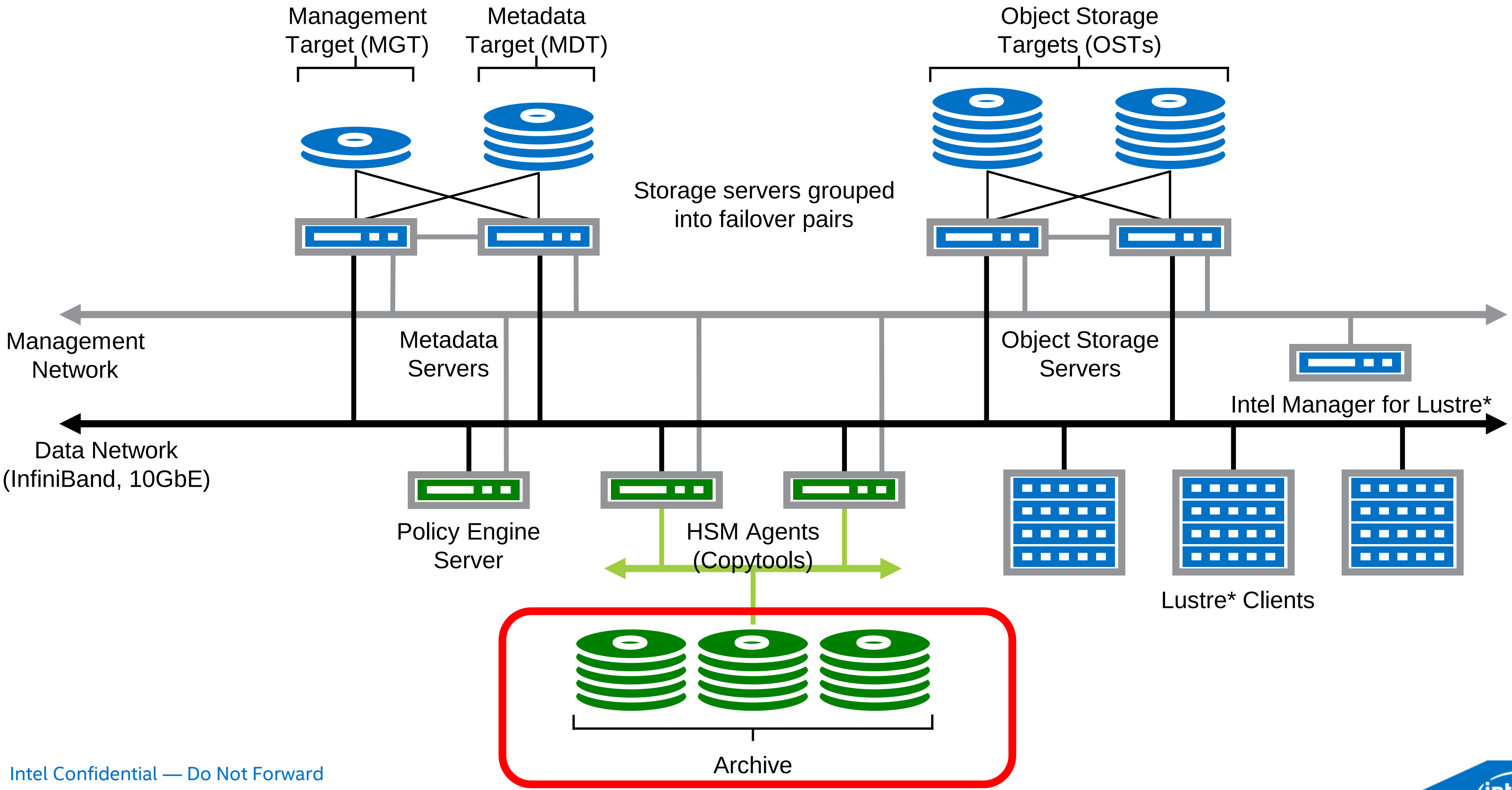
- Gathers information about the files in Lustre* by reading the changelogs

Uses MySQL to store data

- Need to architect the node configuration, especially storage, for desired performance
  - Back-end storage affects the number of files-per-second Robinhood can handle

Recommended configuration is some dedicated storage for the MySQL DB

Archive Storage



Management Target (MGT)

Metadata Target (MDT)

Object Storage Targets (OSTs)

Storage servers grouped into failover pairs

Management Network

Metadata Servers

Object Storage Servers

Intel Manager for Lustre*

Data Network (InfiniBand, 10GbE)

Policy Engine Server

HSM Agents (Copytools)

Lustre* Clients

Archive

Archive Storage

Archive systems provide the highest levels of capacity and data resilience of any storage system

- Typically much slower than primary (Lustre*) storage
- In a Lustre* HSM system, the archive provides the storage target when migrating data from the high performance Lustre file system

The copytool writes data from Lustre* to the archive storage system and retrieves data from the archive if it has been released

- Therefore the copytool needs to "speak" the archive system(s) language
- This is why there is no universal copytool
- IEEL includes the Lustre* POSIX Copytool (also provides support)
    - "Speaks" Lustre* and POSIX

Remote Replication

lustre_rsync

The lustre_rsync feature keeps the entire file system in sync on a backup by replicating the file system's changes to a second file system.

lustre_rsync uses Lustre changelogs to efficiently synchronize the file systems without having to scan (directory walk) the Lustre file system.

The lustre_rsync feature works by periodically running lustre_rsync, a userspace program used to synchronize changes in the Lustre file system onto the target file system.

The lustre_rsync utility keeps a status file, which enables it to be safely interrupted and restarted without losing synchronization between the file systems.

Legal Information

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://www.intel.com/content/www/us/en/software/intel-solutions-for-lustre-software.html.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.