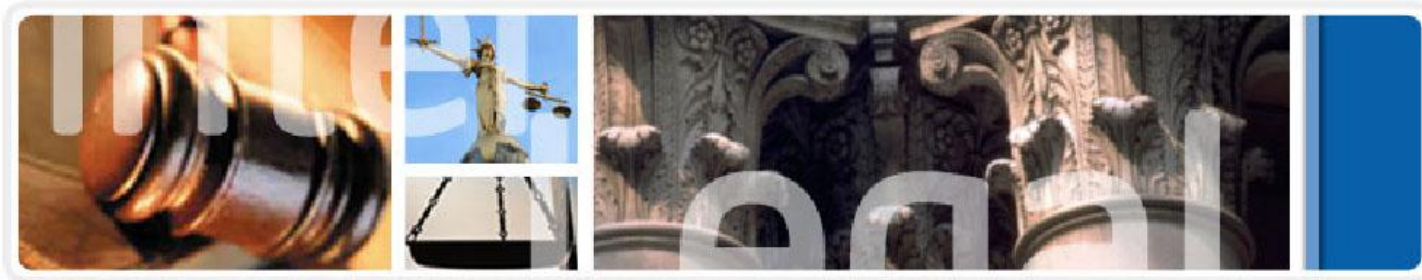




Lustre\* OST Pools



#### Legal Disclaimer

- THIS DOCUMENT AND RELATED MATERIALS AND INFORMATION ARE PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE. INTEL ASSUMES NO RESPONSIBILITY FOR ANY ERRORS CONTAINED IN THIS DOCUMENT AND HAS NO LIABILITIES OR OBLIGATIONS FOR ANY DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OF THIS DOCUMENT.
- All products, product descriptions, plans, dates, and figures are preliminary based on current expectations and subject to change without notice. Availability in different channels may vary.
- Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation in the United States and other countries.
- \*Other names and brands may be claimed as the property of others.
- Copyright 2016 © Intel Corporation. All rights reserved.



## Module Overview

# Module Overview

Introduction to OST Pools

Understanding OST Pools

OST Pools Administration

OST Pools - Deeper Waters

OST Pools Summary



## Introduction to OST Pools

# OST Pools - Introduction

## Pooling OSTs provides powerful functionality

- Restrict end users to a subset of OSTs with proper configuration / directory permissions
- Provide different SLA's to different parts of the file system
- Subdivide a Lustre\* file system into virtual Lustre\* file systems

## OST Pools are not commonly used

- Have existed for years, but less frequently implemented

## Why not?

- Lustre\* most frequently used for HPC deployments - not Enterprise
- Pooling OSTs not needed - Goal is large, shared storage

## How might this change in the near future?

- Traditional HPC vs. emerging markets



## Understanding OST Pools

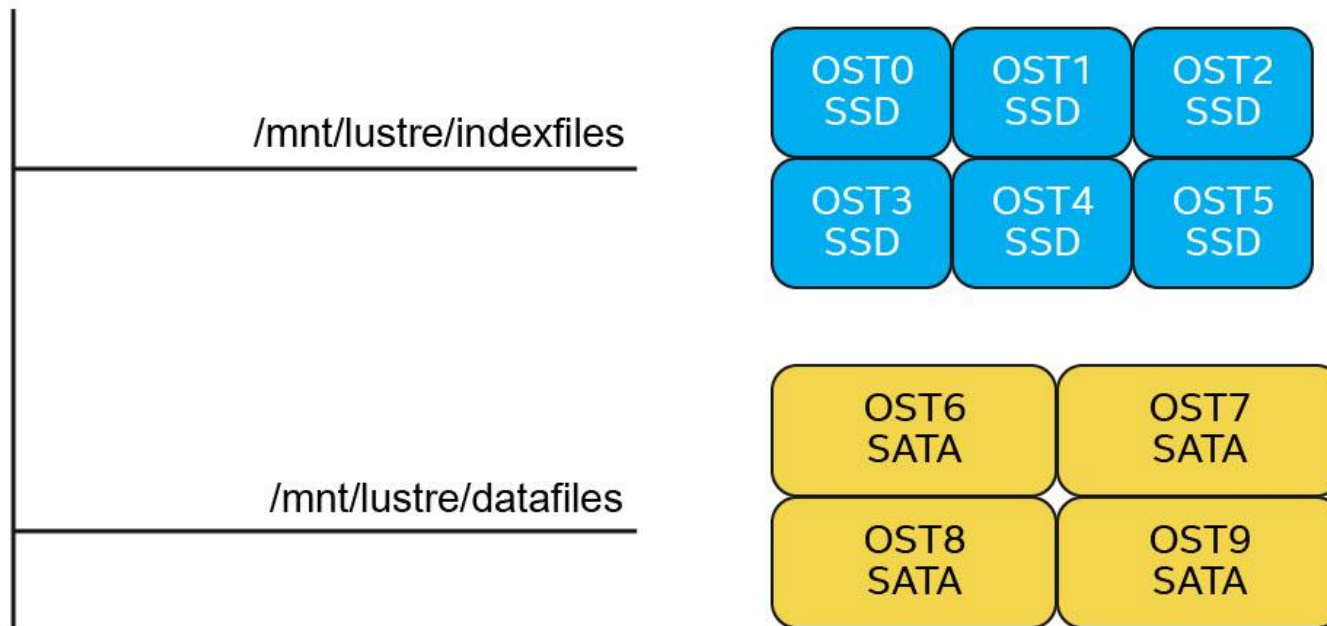
# Why Pool OSTs?

Will begin with two examples (with diagrams to follow):

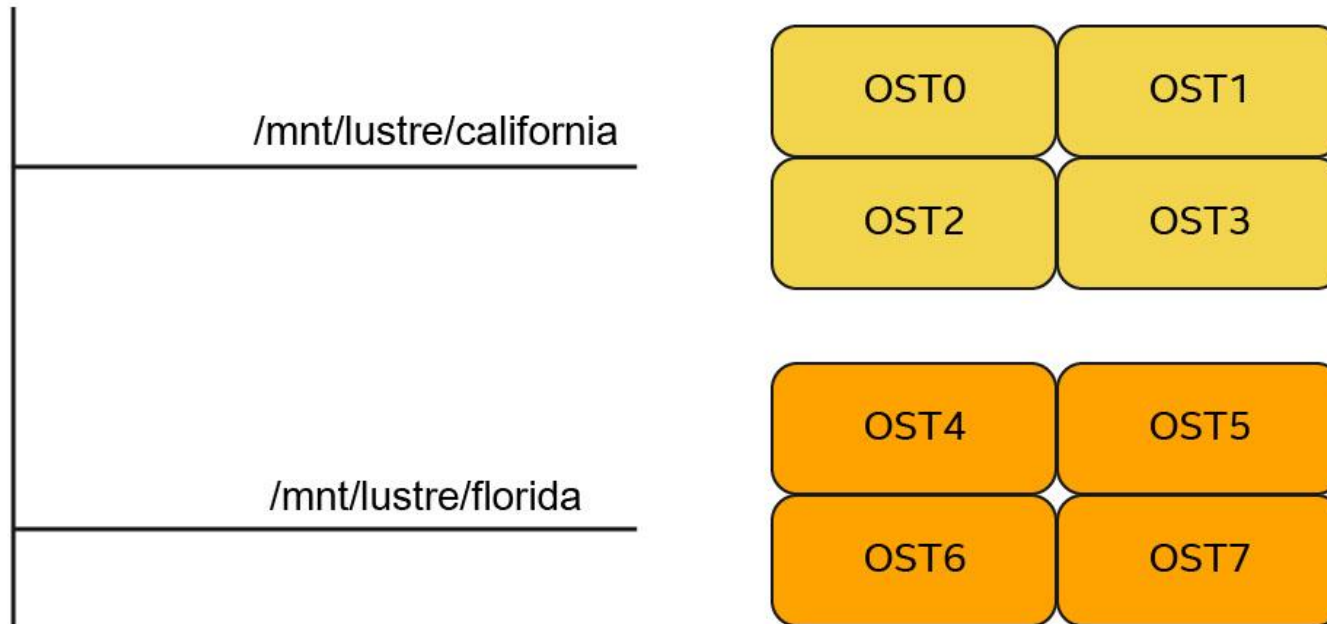
1. To group different types of storage
  - High throughput disks vs. high IOPS disks
2. To group different locations of storage
  - In a geographically distributed file system, some OSTs may be more “local” than other OSTs



# Storage Pools Example - File Types



# Storage Pools Example - Locations



# Additional Use Cases

Groups that own / support their own

OSTs that are attached to faster network (IB)

High end (or low end) storage

Encrypted (vs. not) storage

Storage that will be tiered (HSM) or archived

# Using OST Pools to Enforce Isolation

Example 1: File system with two new OST pools added

```
root group0 rwxrwxr-- /mnt/lustre
root group1 rwxrwxr-- /mnt/lustre/dir-for-pool-1
root group2 rwxrwxr-- /mnt/lustre/dir-for-pool-2
```

- Will **\*NOT\*** work as might be expected - group0 can write to all OSTs

Example 2: File system with two new OST pools added

```
root group0 rwxrwxr-- /mnt/lustre/dir-for-pool-0
root group1 rwxrwxr-- /mnt/lustre/dir-for-pool-1
root group2 rwxrwxr-- /mnt/lustre/dir-for-pool-2
```

Therefore, to enforce FULL isolation, must FULLY use OST Pools

New files created in subdirectories will use the pool by default

# Pool Rules

Every pool has its rules - even OST Pools

What are those rules?

- OST membership in a pool may be changed
- An OST can be a member of multiple pools
- No ordering of OSTs in a pool is defined
- Striping (object allocation) within pools follows normal rules
- OST pool settings are effective only at file creation



## OST Pools Administration

# Creating an OST Pool

Create a pool with: `lctl pool_new`

Example:

```
# lctl pool_new
add a new pool
usage pool_new <fsname>.<poolname>

# lctl pool_new myfs.mypool
Pool myfs.mypool created

# lctl pool_list myfs
Pools from myfs:
myfs.mypool
```

# Adding OSTs to a Pool

Add OSTs to a pool with: `lctl pool_add`

Examples:

```
# lctl pool_add myfs.mypool myfs-OST0000
OST myfs-OST0000_UUID added to pool myfs.mypool
```

```
# lctl pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0000_UUID
```

```
# lctl pool_add myfs.mypool myfs-OST0001 myfs-OST0002
OST myfs-OST0001_UUID added to pool myfs.mypool
OST myfs-OST0002_UUID added to pool myfs.mypool
```

```
# lctl pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0000_UUID
myfs-OST0001_UUID
myfs-OST0002_UUID
```



# Removing OSTs from a Pool

Remove OSTs with: `lctl pool_remove`

Example:

```
# lctl pool_remove myfs.mypool myfs-OST0001
OST myfs-OST0001_UUID removed from pool myfs.mypool

# lctl pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0000_UUID
myfs-OST0002_UUID
```

# Removing an OST Pool

Remove a pool with: `lctl pool_destroy`

Note that the pool must not contain any OSTs

Example:

```
# lctl pool_destroy myfs.mypool
Pool myfs.mypool not empty, please remove all members
pool_destroy: Directory not empty

# lctl pool_remove myfs.mypool myfs-OST0000 myfs-OST0002
OST myfs-OST0000_UUID removed from pool myfs.mypool
OST myfs-OST0002_UUID removed from pool myfs.mypool

# lctl pool_destroy myfs.mypool
Pool myfs.mypool destroyed

# lctl pool_list myfs
Pools from myfs:
```

# After Deleting an OST Pool - What then...?

After deleting an OST pool

- What happens to the files and directories in that pool?

Files:

- Retain their current stripe settings
- Retain the pool name in the layout

Directories:

- Return to the file system's default stripe settings



OST Pools - Deeper Waters

# Regular Expressions Simplifies Syntax

Regular expressions (regex's) in Lustre\* simplify commands:

Without a regex: myfs-OST0000 myfs-OST0001 myfs-OST0002 myfs-OST0003

With a regex: myfs-OST[0-3]

The regex command is now:

```
# lctl pool_add myfs.mypool OST[0-3]
OST myfs-OST0000_UUID added to pool myfs.mypool
OST myfs-OST0001_UUID added to pool myfs.mypool
OST myfs-OST0002_UUID added to pool myfs.mypool
OST myfs-OST0003_UUID added to pool myfs.mypool

# lctl pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0000_UUID
myfs-OST0001_UUID
myfs-OST0002_UUID
myfs-OST0003_UUID
```

Also, Lustre\* prepends the file system name, if it is not provided

## Regular Expressions Syntax (cont.)

Removing OSTs can also be simplified:

```
# lctl pool_remove myfs.mypool OST[0,3]
OST myfs-OST0000_UUID removed from pool myfs.mypool
OST myfs-OST0003_UUID removed from pool myfs.mypool

# lctl pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0001_UUID
myfs-OST0002_UUID
```

# Administrative vs. User Commands

Previously, commands have been demonstrated using “lctl”

Administrative commands use “lctl”

- Non-Administrative commands use “lfs”
  - Example: Get information about the OST pools

```
$ lfs pool_list /myfs
Pools from myfs:
myfs.mypool
```

... and then follow up getting a list of the OSTs in a particular pool

```
$ lfs pool_list myfs.mypool
Pool: myfs.mypool
myfs-OST0001_UUID
myfs-OST0002_UUID
```

# Using Pools to Manage Layouts

Use the `lfs setstripe` command to assign a directory to an OST pool

```
mkdir -p /myfs/projectX
```

```
lfs setstripe -p myfs.mypool -a /myfs/projectX
```



# Pool Support in Other Commands

Many Lustre\* tools are pool aware

For example, lfs:

```
$ lfs df -p myfs.mypool
```

UUID	1K-blocks	Used	Available	Use%	Mounted on
myfs-MDT0000	9174328	178572	8471468	1%	/myfs [MDT:0]
myfs-OST0001	721984264	653299296	68684904	90%	/myfs [OST:1]
myfs-OST0002	721984264	653299296	68684904	90%	/myfs [OST:2]
Summary:	1443968528	1306598592	137369808	90%	/myfs

```
$ lfs find -p myfs.mypool -uid 501
```

- Returns files created in myfs.mypool, owned by id 501



## Module Summary

# Module Summary

Introduction to OST Pools

Understanding OST Pools

OST Pool Administration

OST Pool Summary

Congratulations! You have completed:  
Lustre\* OST Pools

