whamcloud

CUG 2012
Stuttgart, Germany
April 2012

Tutorial: Lustre 2.x Architecture

- Johann Lombardi

Why a new stack?

- Add support for new backend filesystems
  - e.g. ZFS, btrfs
- Introduce new File IDentifier (FID) abstraction
- Better layering separation
- Portability to other operating systems
- Foundations for new features
  - Distrisbuted NamespacE (DNE), new network RAID type, metadata writeback cache, …

Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

File Identifiers (FIDs)

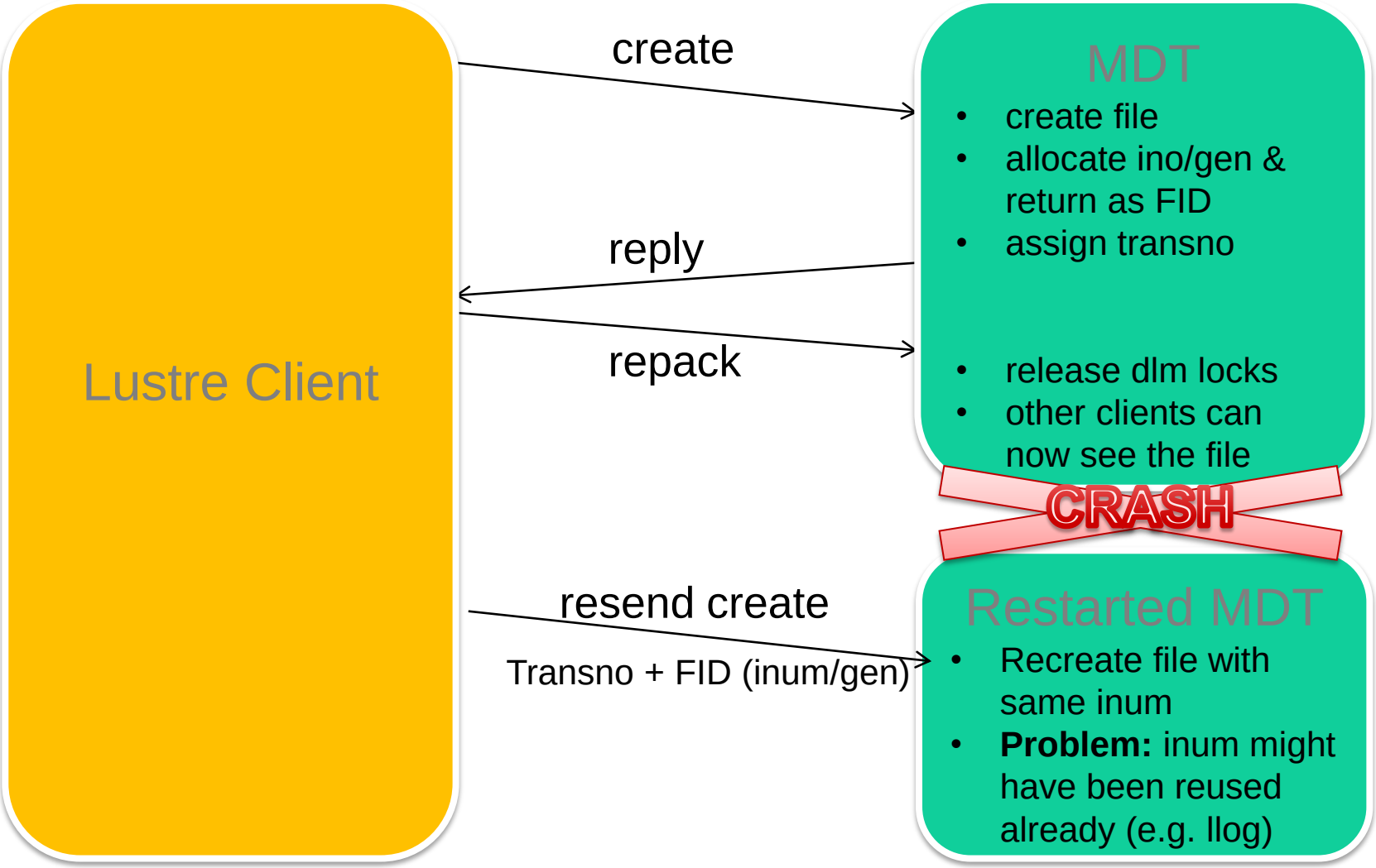
- All network filesystems rely on a file identifier
- Used to uniquely identify file/object in network request
- NFS uses a 64-bit file handle

FIDs in Lustre 1.8

- On the MDS, files are identified by:
  - 32-bit inode number allocated by underlying ldiskfs filesystem
  - 32-bit generation number also maintained by ldiskfs
- On the OSTs, objects are identified by:
  - 64-bit object id allocated sequentially starting from 1
  - 32-bit index which is the OST index in the LOV

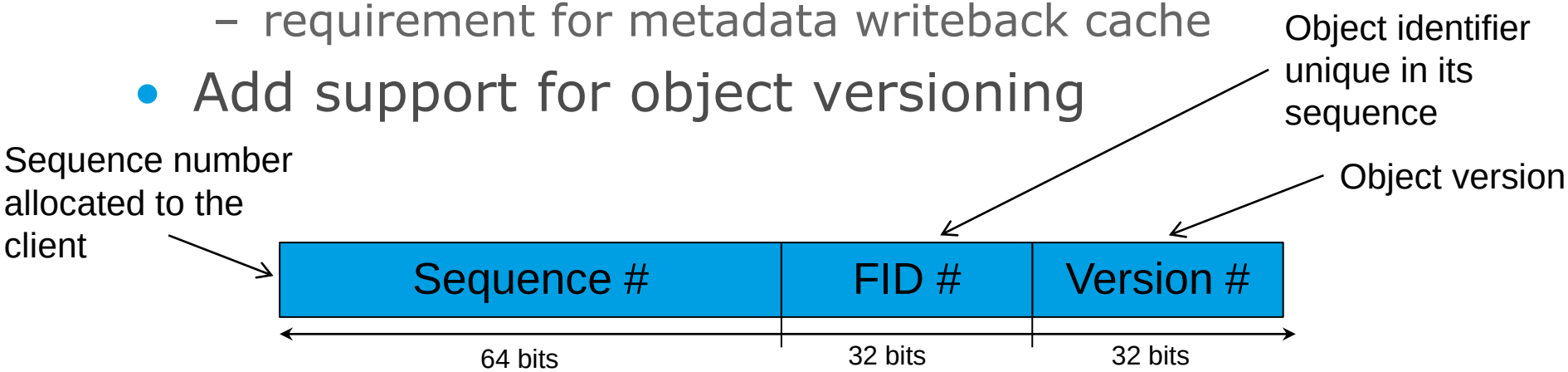
```
[client]# lfs getstripe foo
foo
lmm_stripe_count:   2
lmm_stripe_size:    1048576
lmm_stripe_offset:  0
     obdidx       objid      objid        group
        0           3        0x3            0
        1           3        0x3            0
```

Replay Issue



New FID Scheme in Lustre 2.x

- Independent of MDS backend filesystem
- Simplify recovery
  - e.g. no need to regenerate inode with specific inode number during replay
- Get rid of the infamous iopen patch
- Can be generated on the client
  - requirement for metadata writeback cache
- Add support for object versioning



FIDs in Practice

```
[client]# touch foo
[client]# lfs path2fid foo
[0x200000400:0x1:0x0]
```

Sequence Object ID Version

```
[client]# ln foo bar
[client]# lfs fid2path /mnt/lustre [0x200000400:0x1:0x0]
/mnt/lustre/foo
/mnt/lustre/bar
```

Sequence

- Sequences are granted to clients by servers
- When a client connects, a new FID sequence is allocated
  - upon disconnect, new sequence is allocated to client when it comes back
- Each sequence has a limited number of objects which may be created in it
  - on exhaustion, a new sequence should be started
- Sequences are cluster-wide and prevent FID collision

Sequence in Practice

last allocated FID

```
[client]# lctl get_param seq.cli-srv*.*
seq.cli-srv-xxxxx.fid=[0x200000400:0x1:0x0]
seq.cli-srv-xxxxx.server=lustre-MDT0000_UUID
seq.cli-srv-xxxxx.space=[0x200000401 - 0x200000401]:0:0
seq.cli-srv-xxxxxx.width=131072

[client]# touch foobar
[client]# lfs getstripe -v ./foobar
./foobar
lmm_magic:          0x0BD10BD0
lmm_seq:            0x200000400
lmm_object_id:      0x2
lmm_stripe_count:   1
lmm_stripe_size:    1048576
lmm_stripe_pattern: 1
lmm_stripe_offset:  1
     obdidx       objid       objid         group
        1           3         0x3             0
```

unused sequence allocated to this client

number of FIDs that can be generated out of a sequence

Where are FIDs stored? (1/2)

- The underlying filesystem still operates on inodes
- An object index is stored on disk to handle FID/on-disk inode mapping
- For ldiskfs, the object index is an IAM lookup table (namely oi.16)

```
debugfs:  ls
    2(12)  .            2(12) ..            11(20) lost+found
   12(16)  CONFIGS 25001(16) OBJECTS       19(20) lov_objid
   22(16)  oi.16      23(12) fld           24(16) seq_srv
   25(16)  seq_ctl    26(20) capa_keys 25002(16) PENDING
25003(12)  ROOT       27(20) last_rcvd 25004(20) REM_OBJ_DIR
   31(3852)CATALOGS
```

Where are FIDs stored? (2/2)

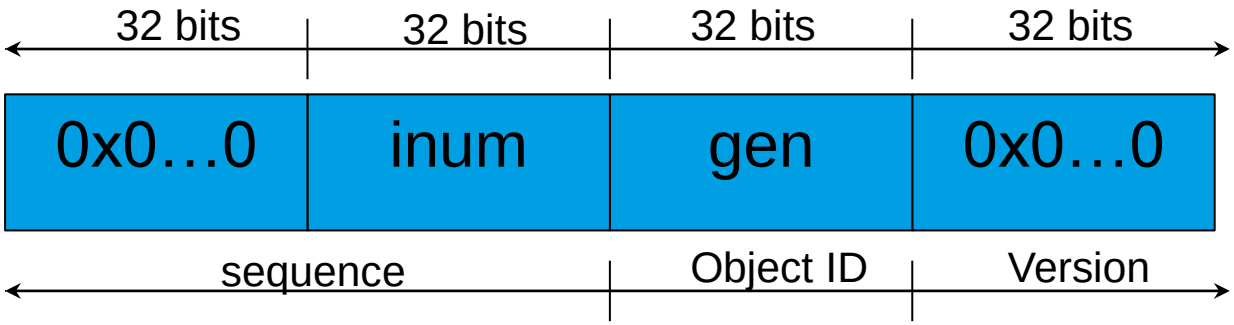
- The FID is also stored:
  - in an extended attribute called LMA
    - stands for Lustre Metadata Attributes
    - also stores SOM/HSM states
    - see struct lustre_mdt_attrs for the format
  - in the directory entry, along with the filename
    - path->FID translation does not require accessing LMA XATTR
    - ext4 & e2fsprogs patch to support this feature

Link Extended Attribute

- New XATTR storing list of parent FIDs and names
- Useful for:
  - verifying directory hierarchy
  - FID to path translation
    - `lfs fid2path`
  - updating parent directory entries when migrating files
  - POSIX lookup-by-FID path permission checks

Compatibility Mode: IGIF

- Filesystems upgraded from 1.8 don't have fid stored in EA or in directory entry
- Name/fid mapping handled by IGIF
- IGIF allows to reversibly map inode/generation into FID
- Special sequence range reserved
  - up to ~4B of inodes



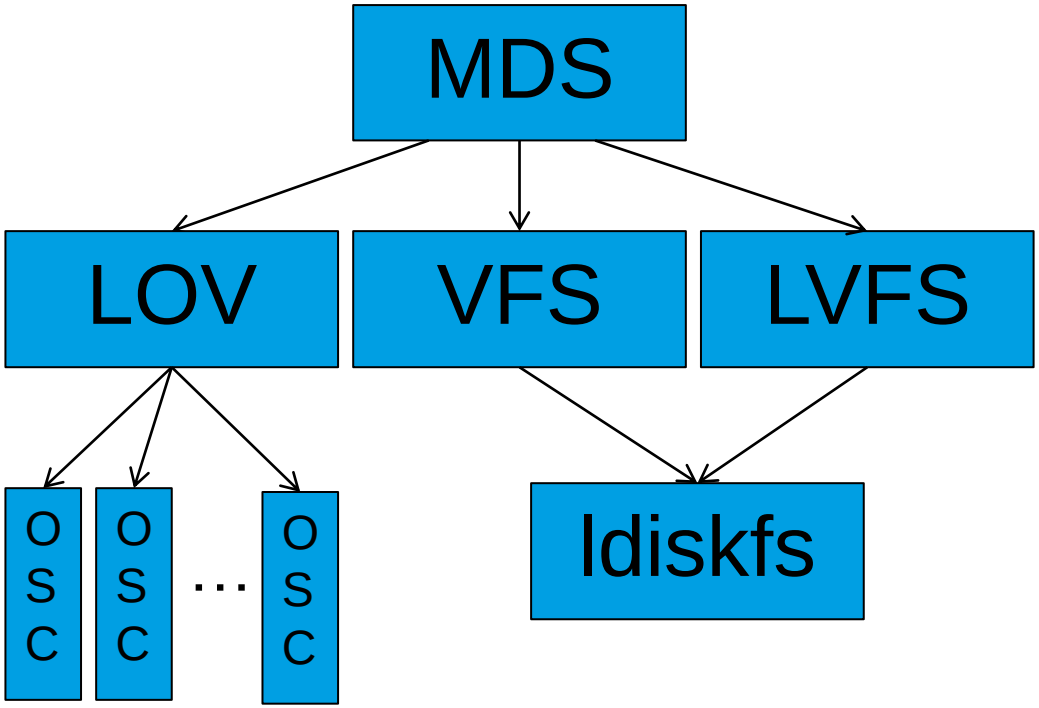
What about OST objects …

- In 2.2, OST objects are still identified with an object id local to the OST
  - Not unique across the cluster
- FIDonOST is going to change that
  - Requirement for Distributed NamespacE (DNE) support
  - Multiple MDTs will now pre-create objects on OSTs
  - Each MDT is granted an unique sequence to allocate OST objects from
  - OST should be able to request super sequence from MDT0
    - OSTs to set up a connection to MDT0
- Sequence already reserved for compatibility
  - Called IDIF (IGIF for data)

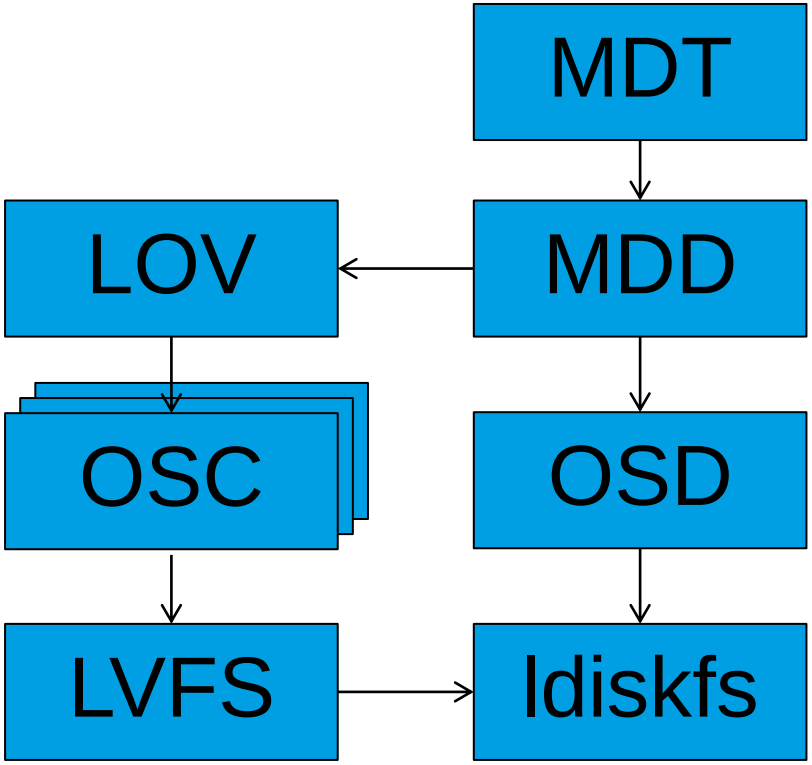
Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

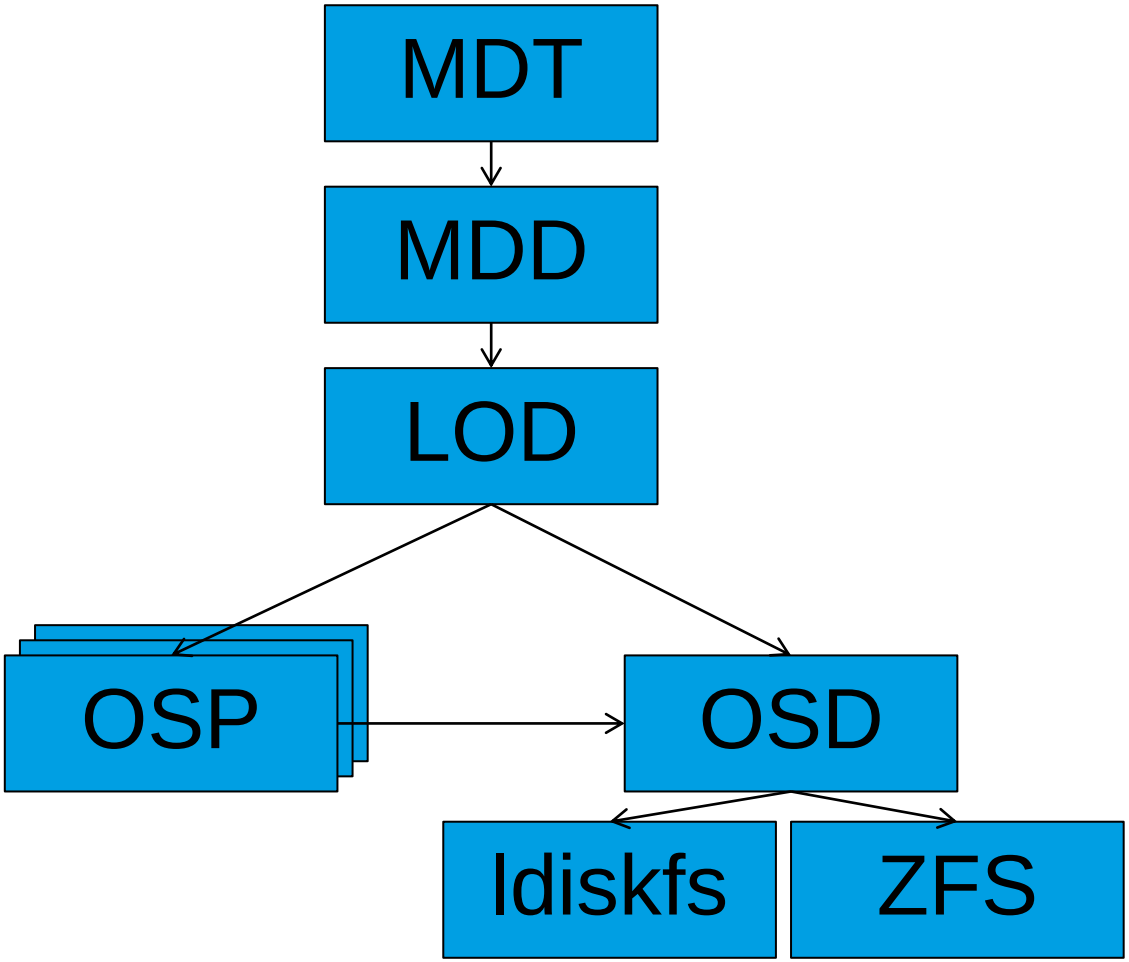
1.x MDS Layering



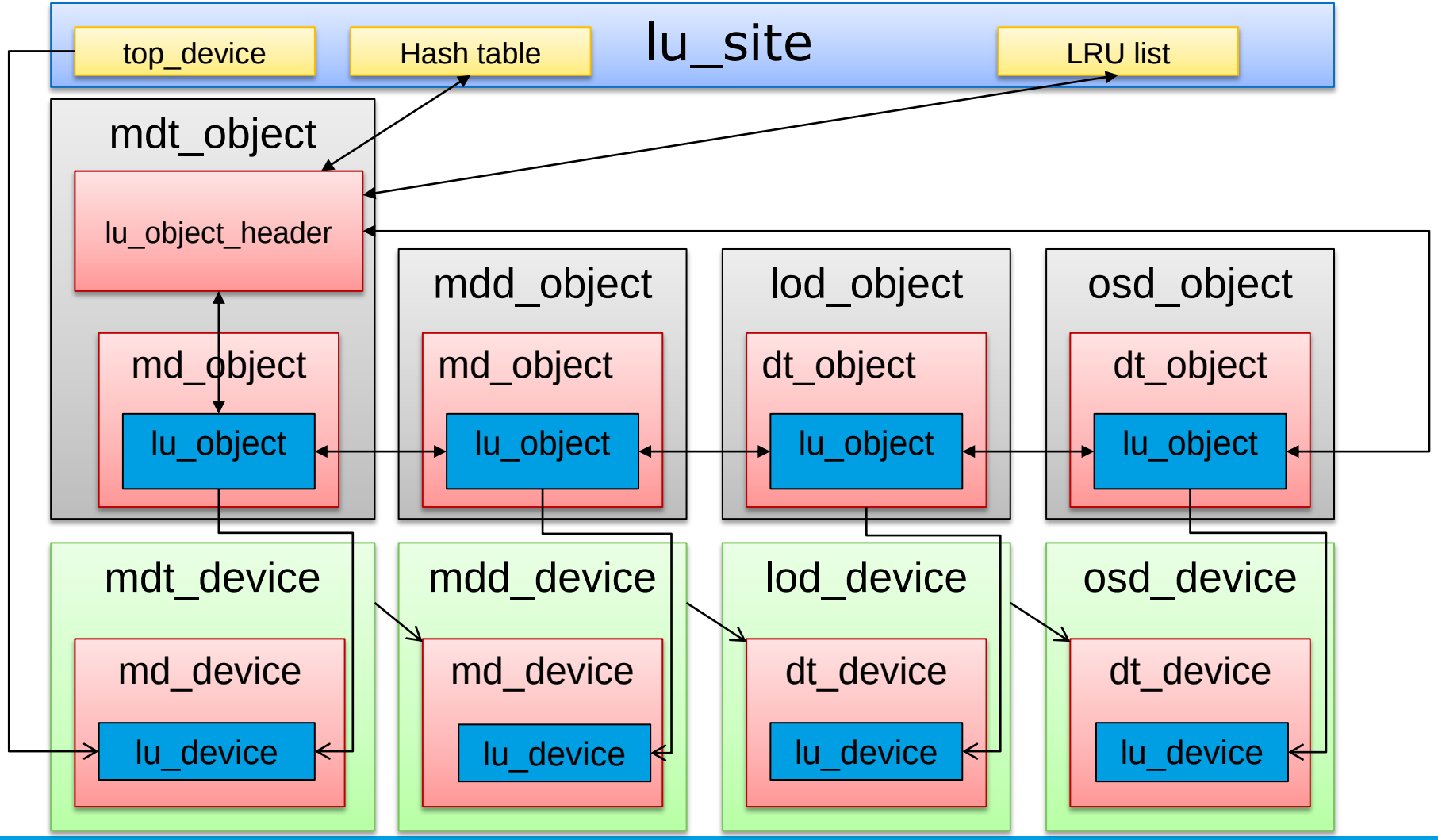
2.0-2.3 MDS Layering



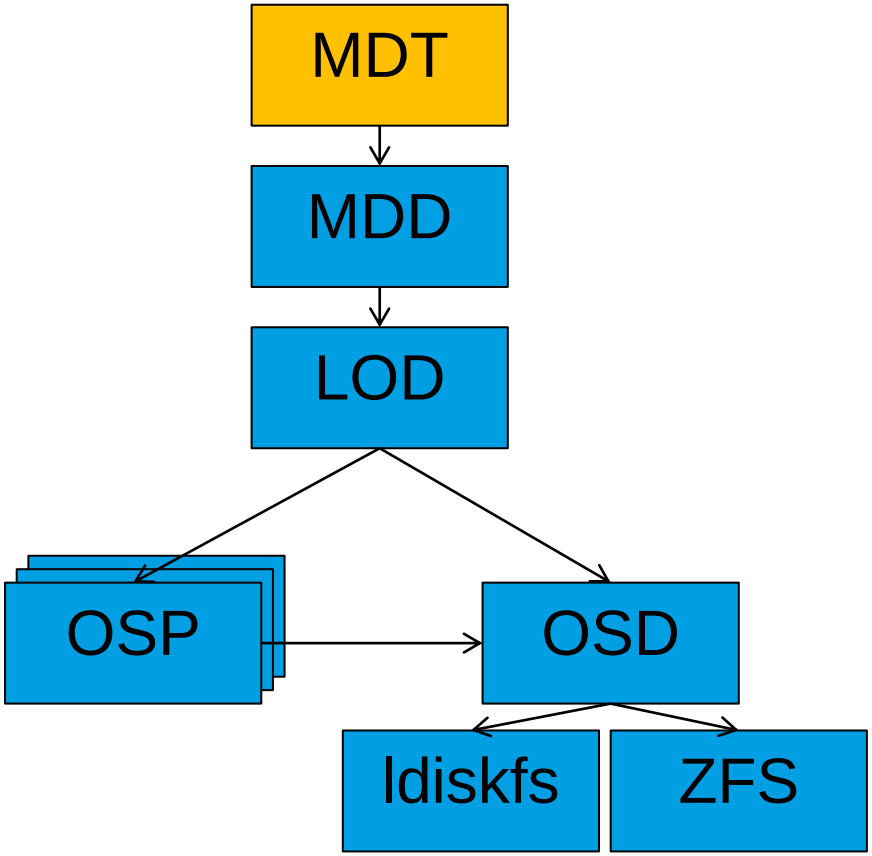
2.4 MDS Layering



New Device & Object Stacking

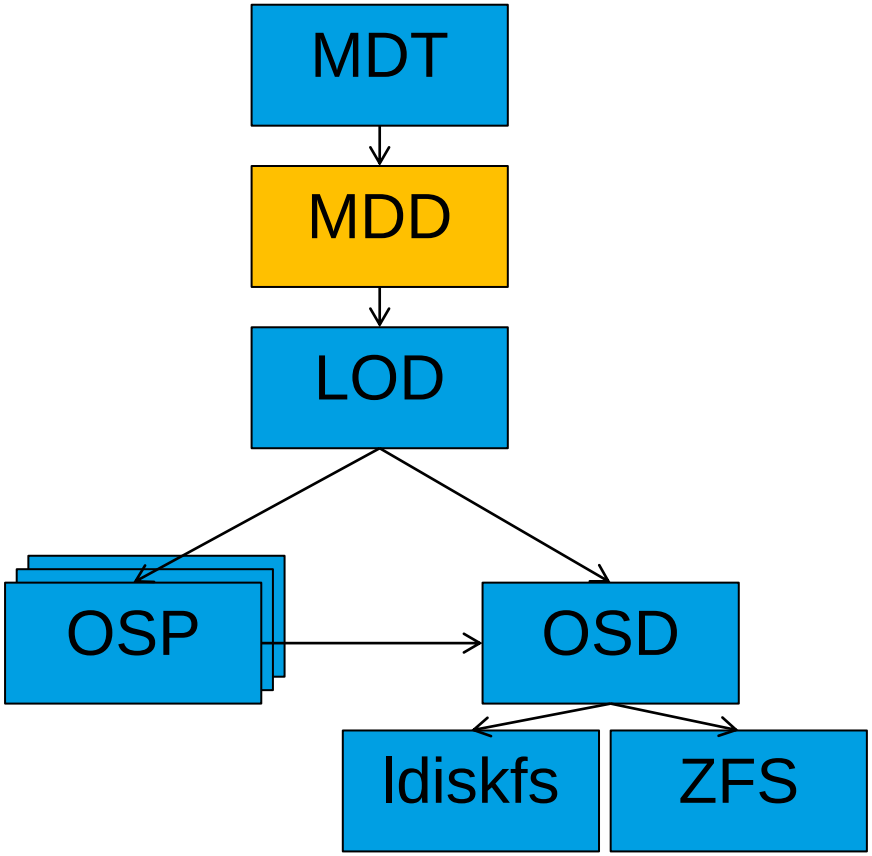


MetaData Target (MDT)



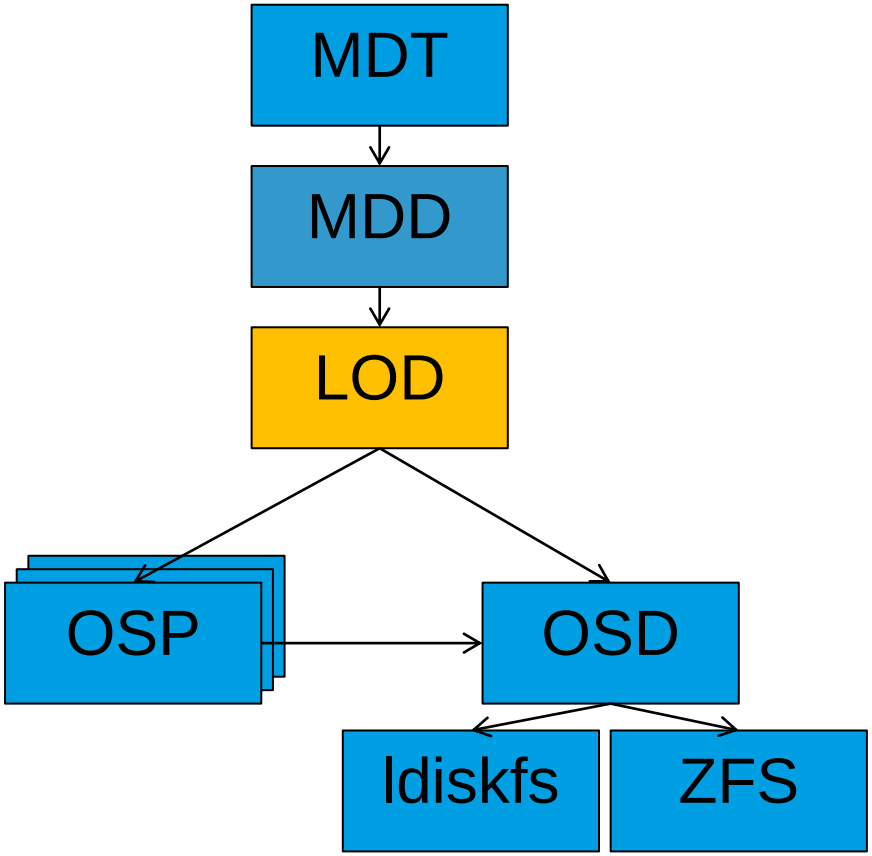
- In charge of all network operations
  - Request packing/unpacking
  - Replies
  - Resent
  - Recovery
  - Ptlrpc services
- Take DLM locks
- Intent & Reintegration handling

Metadata Device Driver (MDD)



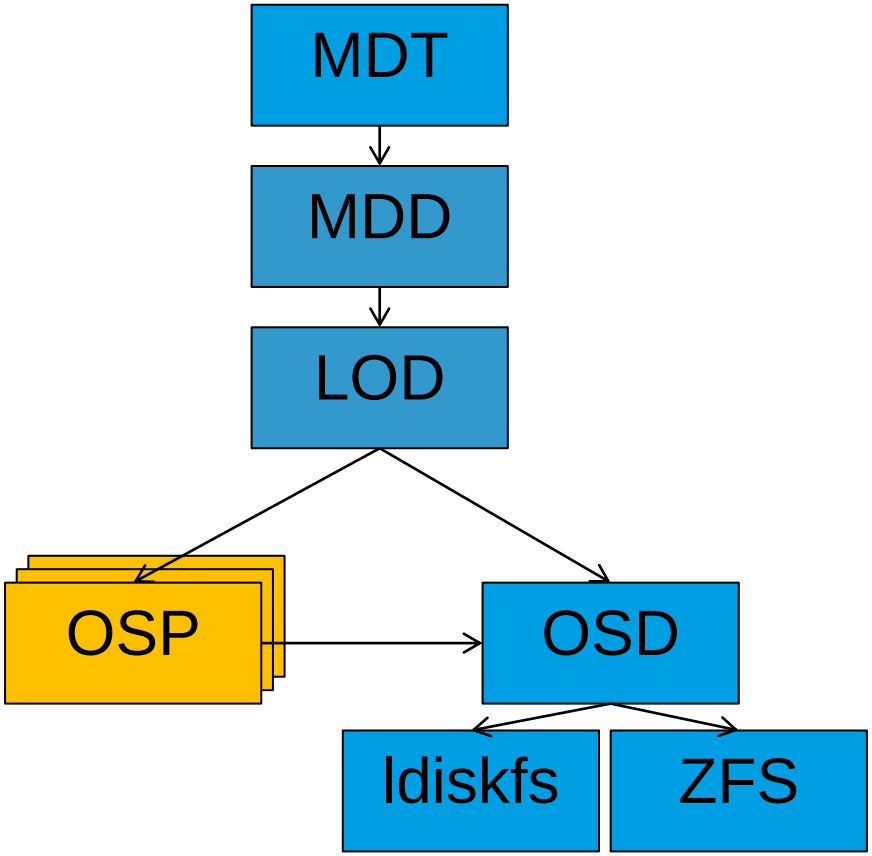
- Directory handling
  - lookup, link/unlink, readdir
- Split metadata operation into a series of OSD operations
  - E.g. mdd_create() creates the new objects and insert it into a parent index
- Permission checks / ACLs

Logical Object Device (LOD)



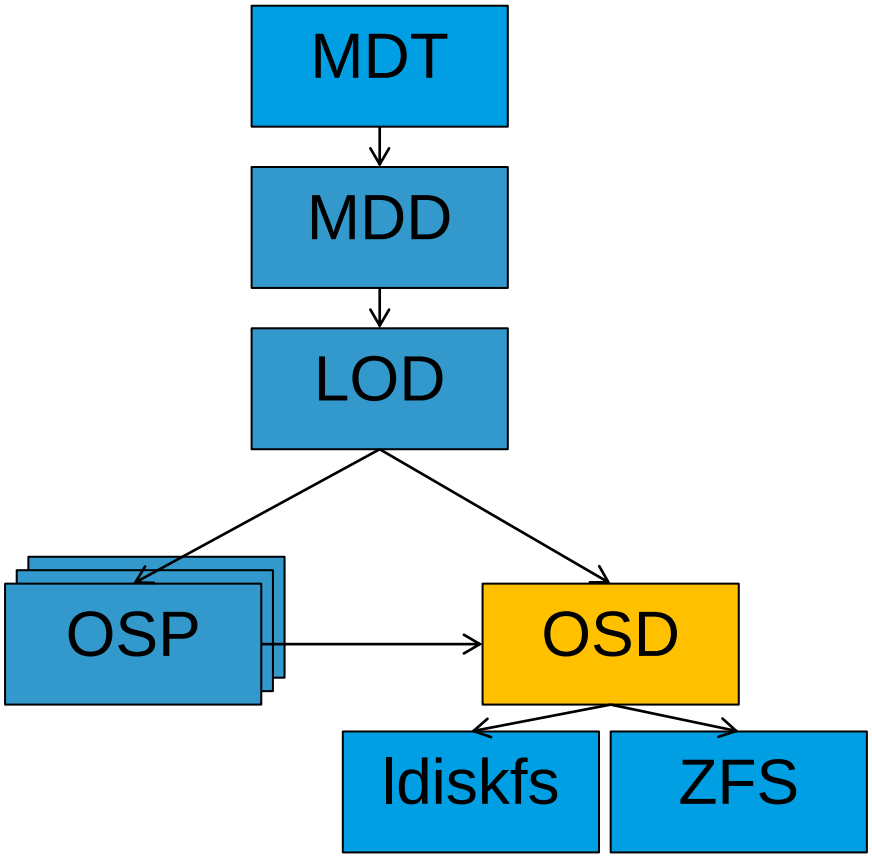
- Replacement for LOV on the MDS
- Take care of striping
  - Maintain LOV EA on disk
- Call OSPs to manipulate OST objects

Object Storage Proxy (OSP)



- Replacement for OSC
- Hide object pre-creation logic
- Handle cleanup of orphan OST objects
- Destroy OST objects on file unlink
  - No longer done by clients
  - Address vulnerability to files w/o objects on double failures
  - Can be batched in the future

Object Storage Device (OSD)



- Objects identified by FID
  - Each OSD has to implement its own object index
    - zfs-osd uses a dedicated ZAP
    - ldiskfs-osd uses an IAM index file, namely oi.16
- Attributes & Extended attributes
- 2 access method types:
  - Body (read/write/truncate)
  - Index (key/record pair)

Wide-striping Support

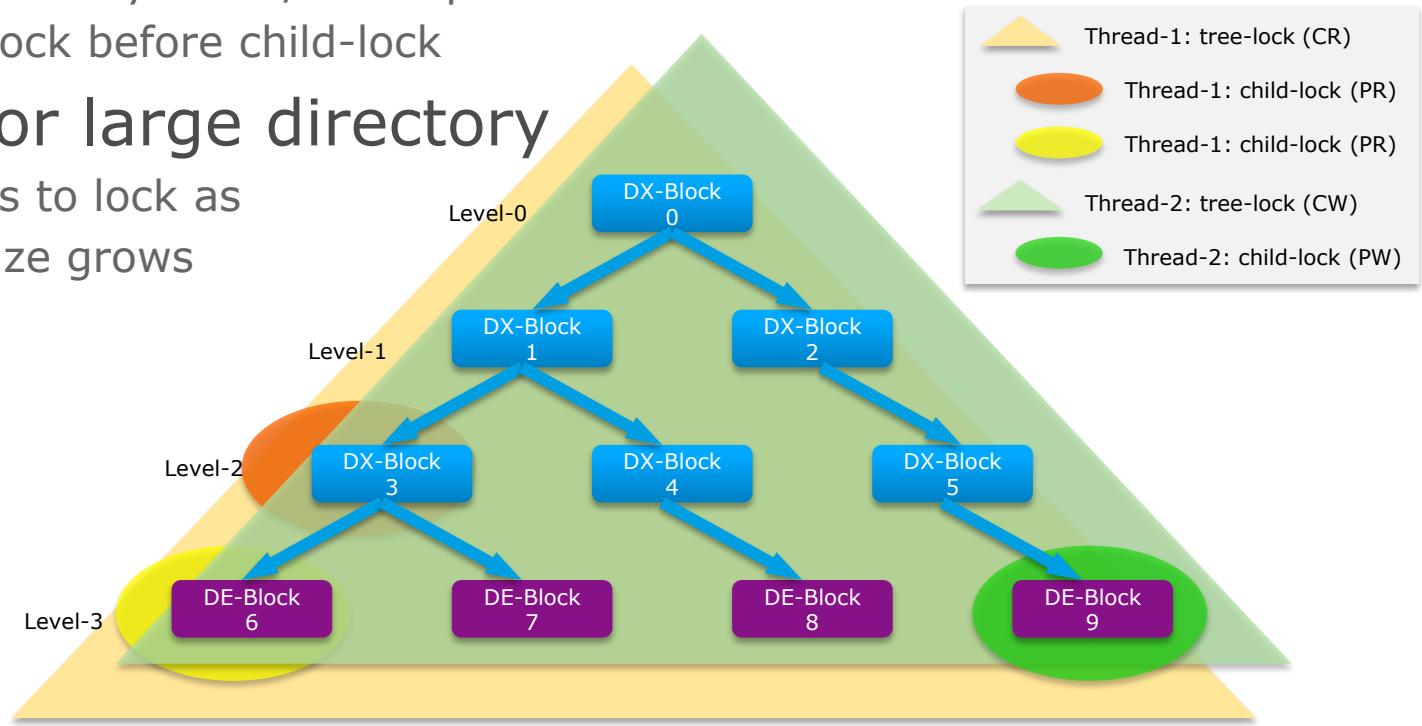
- Increase to 2000 stripes per file
  - Layout stored in extended attribute (xattr)
  - Can allocate single file over all OSTs
- Add large xattr support to ldiskfs
  - Allocate new xattr inode
  - Store large xattr data as file body of that inode
  - Original file inode points to this new xattr inode
  - Not backward compatible with older ext4 code
- Require larger network buffers
  - Return –EFBIG for old clients with smaller buffers
  - Old clients can still unlink such files (done by MDS)

Improving Metadata Server Throughput

- Network/RPC/thread efficiency
  - Better CPU affinity (less cache/thread pinging between CPUs)
  - Service threads awoken in MRU order (newest first)
  - Multiple RPC request arrival/queues from network
  - Improved internal hashing functions for balance
- Parallel directory locking
  - Parallel directory DLM locking was implemented in Lustre 2.0
  - Testing showed ext4 directory was primary bottleneck
  - Add parallel locking for directory operations, per disk block
  - Allow concurrent lookup/create/unlink in one directory
  - Improves most common use case for applications

Parallel Directory Locking

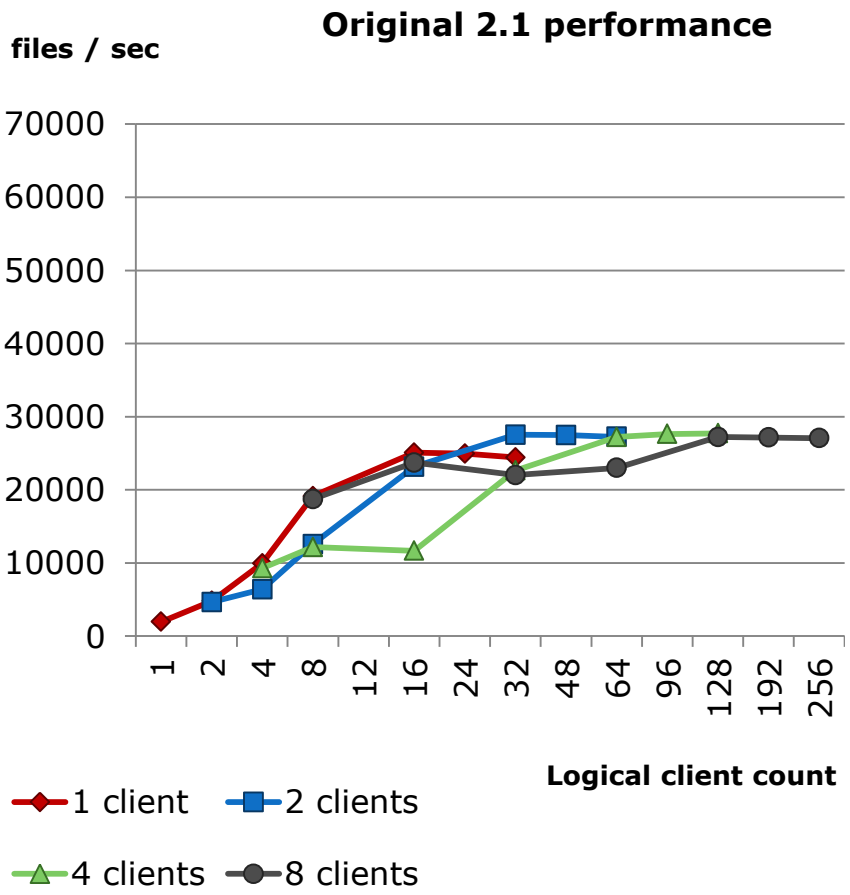
- Protect tree topology
  - Optimistically lock top levels of tree
  - Lock bottom level(s) as needed for operation (read/write)
  - Backout and retry if leaf/node split is needed
  - Take tree-lock before child-lock
- Scalable for large directory
  - More leaves to lock as directory size grows

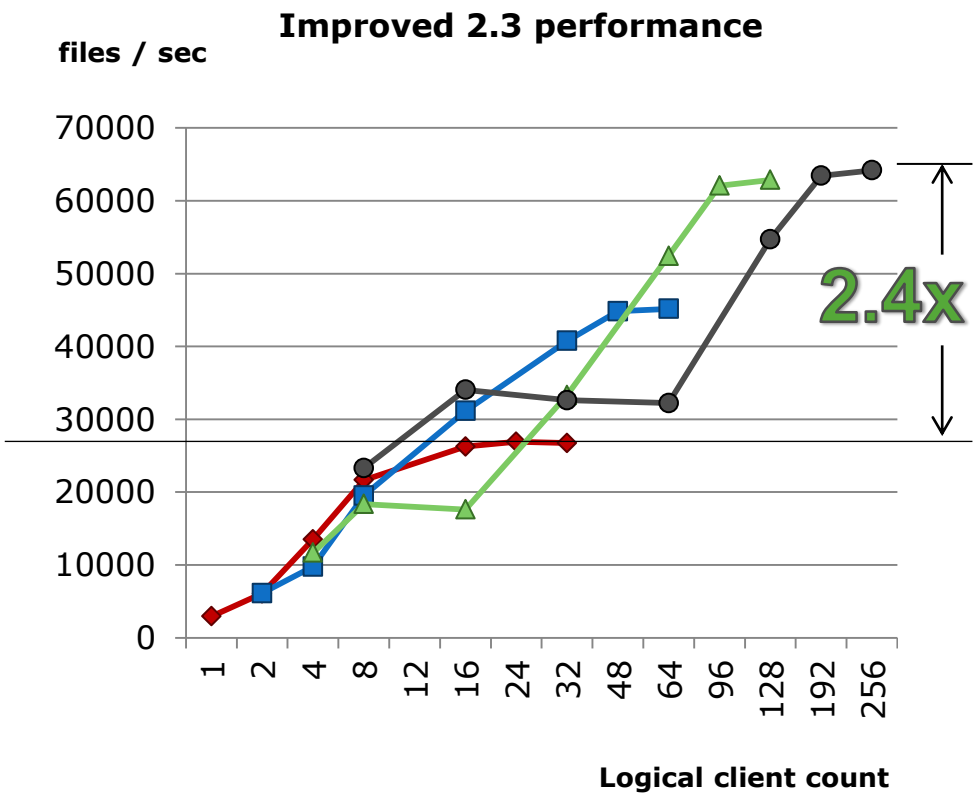


Graph-2 : htree and htree-lock

Single Shared Directory Open+Create

1M files, up to 32 mounts/client





Metadata Performance Testing

- New tool to measure metadata performance called mds-survey
- Run directly on the MDS
  - Doesn't require any lustre clients
  - Similar to obdfilter-survey, but for metadata
- Rely on extensions made to the echo-client to support metadata operations
- Support create/lookup/getattr/setxattr/destroy operations

Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

New Data Stack

- Lustre 2.2 still use same OST stack as 1.6/1.8
  - Rely on LVFS layer instead of OSD API
- Port OST to new OSD API
- Object Filter Device (OFD) replaces obdfilter
  - Runs on top of the OSD API
  - Allow to use zfs-based OST
- All low level I/O is moved to OSD layer

Object Filter Device (OFD)

- Rely on FIDs
  - IDIF now
  - Will support FIDonOST soon for DNE
- Propagate changes to OSD properly
- Handle pre-creation and orphan cleanup
- Grant management is no longer ldiskfs specific
  - ZFS, btrfs & ext4 (soon) support large block size (>4KB)
  - Introduce changes to the client & OST to understand series of pages from single block (i.e. extent)

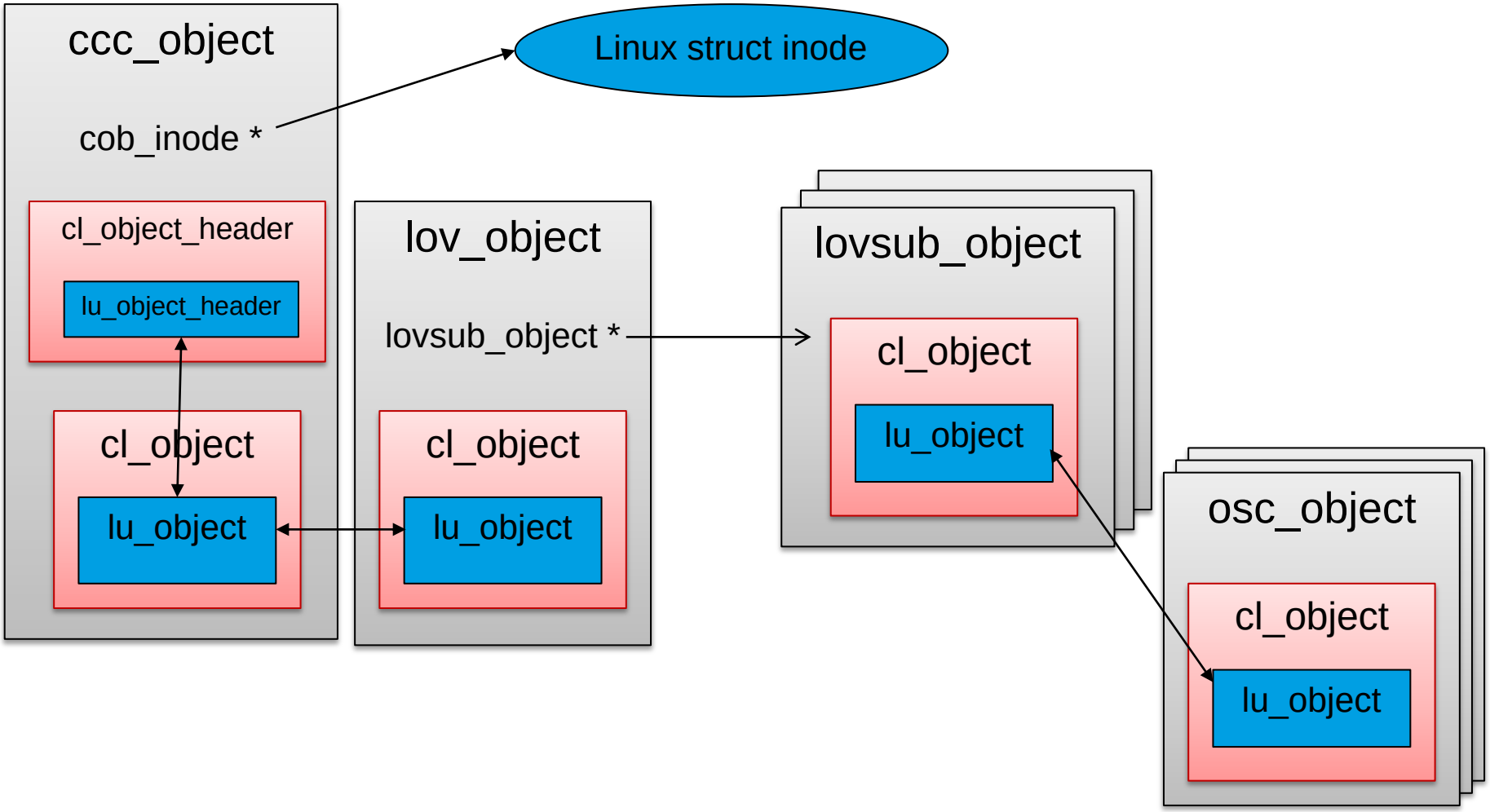
Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

New Client I/O Stack

- Reuse infrastructure developed for new MD stack
  - e.g. lu_device, lu_object, …
- llite/VVP/SLP
  - OS-dependant layer used to convert file operations into CLIO operations
  - sits on top of CLIO stack, start point of clio operations
- LOV
  - Manages stripe data
  - Parses I/Os and distributes them to OSC
- LOVSUB
  - Mostly dummy layer
  - Refers back to the LOV layer
- OSC
  - RPC layer & DLM locks

CLIO Object Stacking



Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

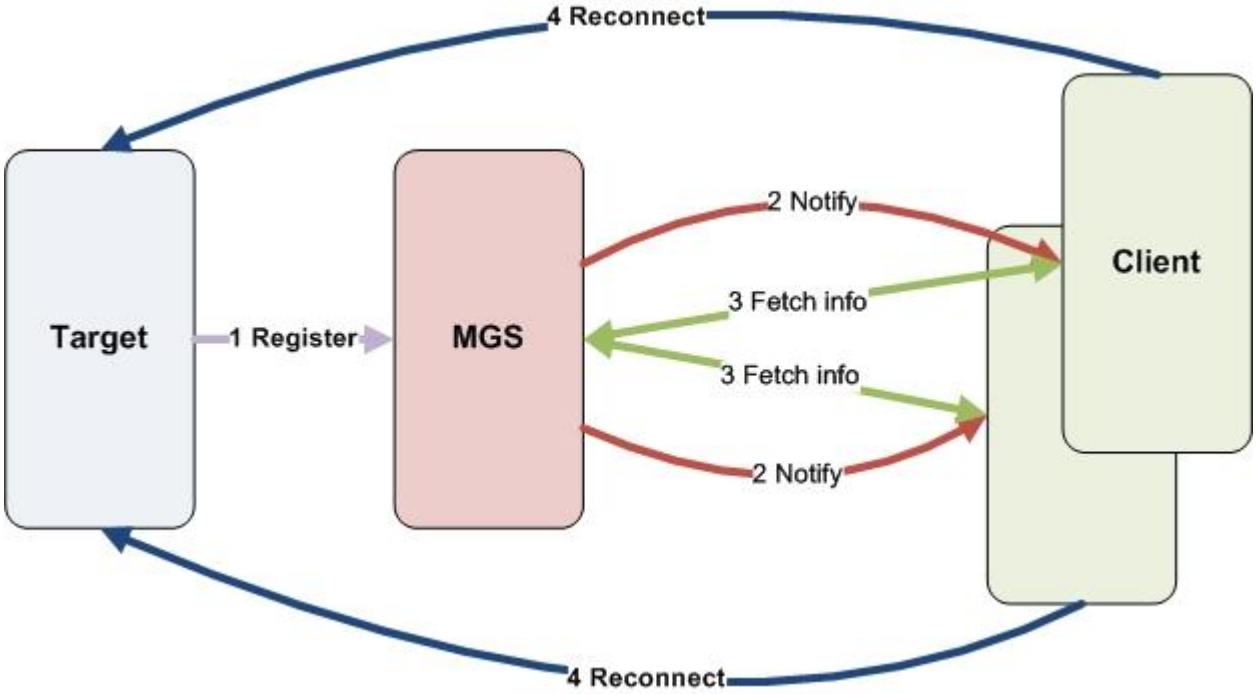
What makes recovery slow?

- Server must wait for all clients to reconnect
  - Recovery replays uncommitted client transactions
    - Must be executed in original order – transno
  - No new transactions until recovery completes
    - Could invalidate recovery transactions
- Clients slow to detect server death
  - Only fault detection is in-band RPC timeout
    - Includes both network and service latency
    - Server under heavy load hard to distinguish from dead server
  - Ping not scalable
    - Ping overhead O(#servers * #clients / ping_interval)
    - Ping interval must increase with system size
  - A client may know the server failure after ping interval + RPC timeout

Introduction of Imperative Recovery

- Accelerate reconnection by notifying clients of server restarts, no longer use timeout
- MGS is used to reflect server failure event to clients
  - Notify clients when a restarted target registers itself to MGS
  - Clients will do reconnection
- Imperative recovery depends on MGS, it's a best-effort service
  - Not impede normal recovery from happening
  - It's important to identify which instance of targets the clients are connecting
- Failover server support

Implementation - overall



Performance

- A restarting target is able to finish recovery within 66 seconds
  - 125 client nodes, 600 mountpoints on each node, 75K clients in total
  - No workload in the cluster
- As a comparison, it took ~300 seconds w/o IR

Agenda

- New FID Abstraction
- New MDS Stack
- New OSS Stack
- New Client I/O Stack
- Recovery Improvements
- Development and Process Guidelines

The Tools We Use Today

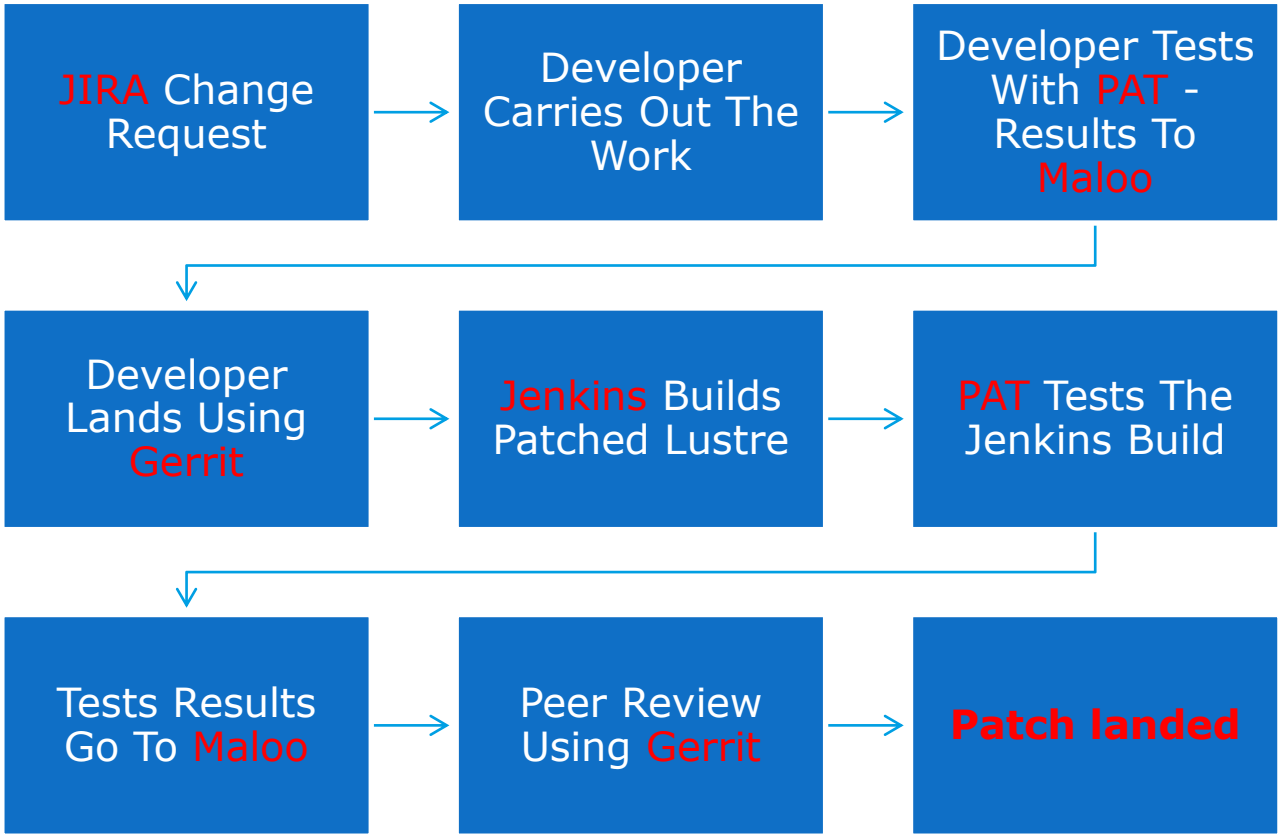
Jira, Jenkins
Git and Gerrit

Tools Live Today

jira.whamcloud.com
build.whamcloud.com
review.whamcloud.com

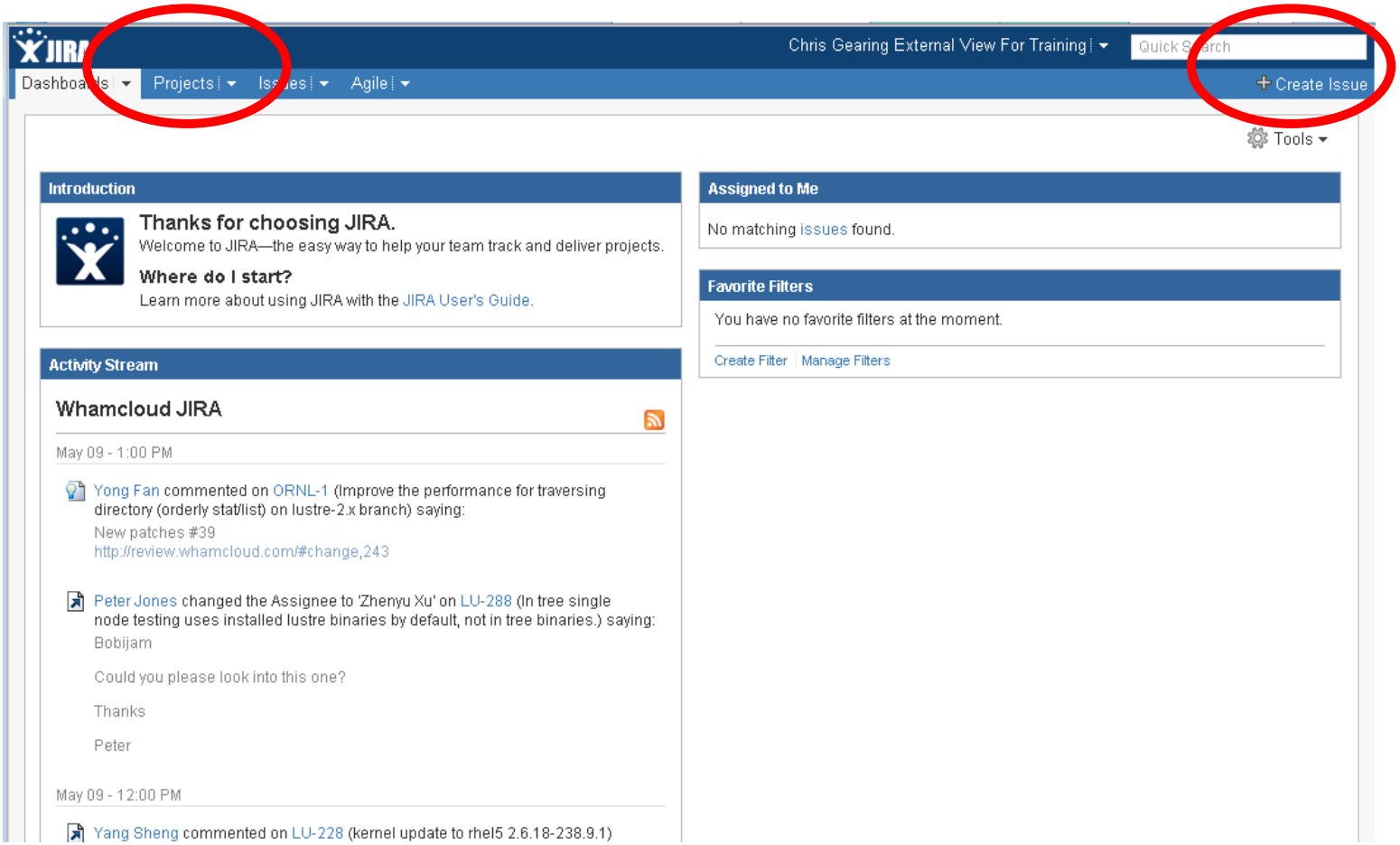
- **JIRA** is Whamcloud's Issue and Agile management tool
- **JENKINS** is the build tool that continuously builds mainstream branches and all patches submitted by the community
- **GIT** is source code tool used for managing the Lustre canonical tree
- **GERRIT** is code review tool that allows the whole community to be part of the code review process

Work Flow



http://wiki.whamcloud.com/display/PUB/Submitting+Changes

Front screen of Jira



This is the front screen of jira that you'll see after you login.

It contains the last few issues entered on the right and provide some useful links.

Top left link to projects.

Top right the Create-Issue button that is present on every Jira screen

JIRA

Chris Gearing External View For Training | Quick Search

Dashboards | Projects | Issues | Agile | + Create Issue

Create Issue

Enter the details of the issue...

Project: Lustre

Issue Type: Bug

* Summary:

Story Points:

Measurement of complexity and/or size of a requirement.

Priority: Minor

Severity: 3

Component/s: Unknown

Affects Version/s:

- Unknown
- *Unreleased Versions*
  - Lustre 1.8.x
  - Lustre 1.8.6
  - Lustre 2.0.0
  - Lustre 2.1.0

Environment:

For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).

Description:

Attachment: Browse...

The maximum file upload size is 10.00 MB. Please zip files larger than this.

Bugzilla ID:

Epic:

Suggested Labels: build c99 client cmd_phase1 connect eviction iam ldlm multirail obdecho quota results test

Use whitespace to separate labels and combine words with an underscore.

Link epics to child stories

Project: None

Project that action item is tied to.

Create | Cancel

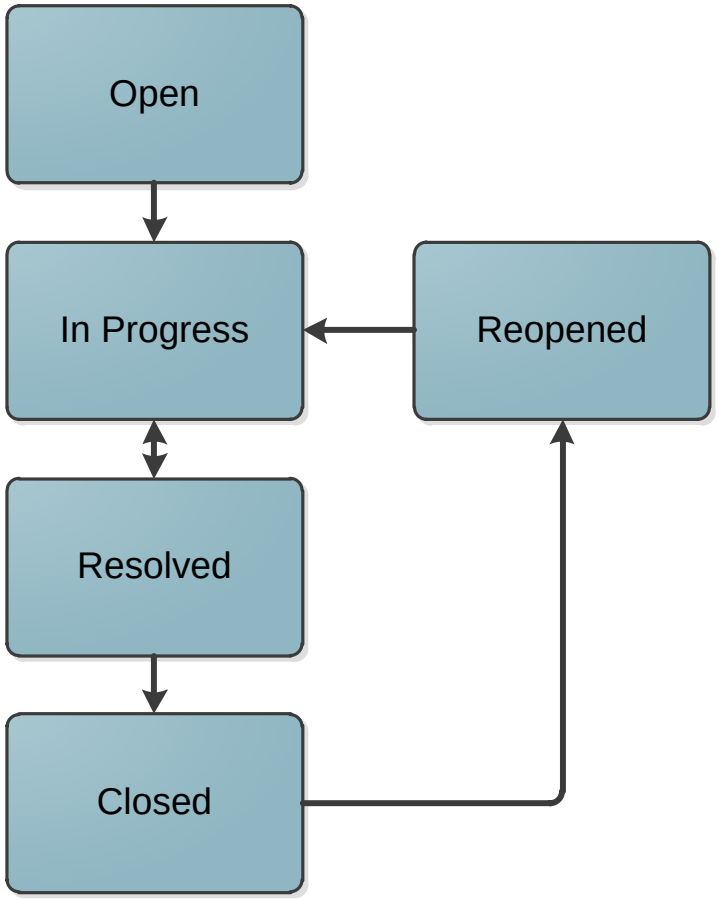
The entries on this page are much like any other bug tracker.

Take time to fill in the information fully and completely

This form is not a placeholder it is the source of information for the change being made.

Describe What, Why and How. Update regularly, provide links to tests etc.

Jira Workflow



Gerrit Code Review

- Gerrit is a Code Review system based on JGit
  http://code.google.com/p/gerrit/

  Also serves as a git server
  adding access control and workflow

  Used by
  - Whamcloud https://review.whamcloud.com/
  - Android https://review.source.android.com/
  - JGit, EGit http://egit.eclipse.org/r/
  - Google, SAP, …

- http://wiki.whamcloud.com/display/PUB/Using+Gerrit

Gerrit

- When one developer writes code, another developer is asked to review that code
- A careful line-by-line critique
- Happens in a non-threatening context
- Goal is cooperation, not fault-finding
- An integral part of the Lustre coding process

Code Review – Tips

- Small changes are much easier to review
- A change should logically do one thing
  - Not many
- No change shall break build or tests
- Split big changes into series of digestible changes
  - These changes depend on each other
  - Last change should switch the new feature on
- Commit message should explain why
  - The What should be obvious from the code change

Jenkins

Jenkins

Jenkins » lustre-master

Back to Dashboard
Status
Changes
Git Polling Log

Build History (trend)

#115 11-May-2011 11:45:42
#114 10-May-2011 09:47:16 3GB
#113 10-May-2011 08:15:43 3GB
#112 09-May-2011 17:15:42 3GB
#111 09-May-2011 14:00:43 3GB
#110 08-May-2011 20:00:42 3GB
#109 08-May-2011 18:15:42 3GB
#108 06-May-2011 09:15:43 3GB
#107 05-May-2011 19:15:43 3GB
#106 05-May-2011 18:00:43 3GB
#105 05-May-2011 09:00:42 3GB
#104 04-May-2011 19:30:41 3GB
#103 03-May-2011 14:45:51 3GB
#102 03-May-2011 08:28:49 3GB
#101 03-May-2011 07:15:51 3GB
#100 02-May-2011 10:32:57 3GB
#12 02-May-2011 10:25:03 6KB
#11 02-May-2011 08:04:16 3GB
#10 02-May-2011 07:31:44 257KB
#4 02-May-2011 06:51:42 260KB
#2 29-Apr-2011 09:31:44 2GB
#1 29-Apr-2011 05:49:37 3GB

Project lustre-master

Build a full matrix for master. See if this one suffers the same problem a

Configuration Matrix

			server	client
i686	el5	inkernel		
		ofa		
	el6	inkernel		
		ofa		
	ubuntu1004	inkernel		
		ofa		
x86_64	el5	inkernel		
		ofa		
	el6	inkernel		
		ofa		
	ubuntu1004	inkernel		
		ofa		

Downstream Projects

coeus

- Left hand side show historical builds
- Every build creates a matrix of binaries
- Click on any orb to go to the output
- No direct control, you have today is to push reviews

Autotest

- No Gui or interface
  - Autotest is a silent agent quitely testing our code
- Autotest takes builds from Jenkins and tests on Whamcloud test hardware
  - Soon the hardware will be expanded to include other community sites
- The results of Autotest can be seen using Maloo

Maloo

- Maloo is the authoritative test results database
  - Autotest **and** Developer results are stored in Maloo
- Testing results from development
  - Results from development provide landing collateral
  - Failures are as important as passes
    - Good to see the transition from failure to pass
- Landing requires passing results in Maloo
  - Maloo / Jenkins / Gerrit work in unison to ensure Reviews, Build and Test have all occurred.

Maloo Screen Shots

whamcloud

latest results | latest sessions | search results | test statistics | upload results | Chris Gearing [settings | logout] whamcloud

Test sessions

Sessions for user: All users Go

1 2 3 4 5 6 7 8 9 ... 28 29 Next →

This is a link to the test suite detail

Host	Group	User	Run at	Imported at	Sets passed	Links
client-23-ib	review	Whamcloud Autotest	2011-05-20 09:58:44 UTC	2011-05-20 13:52:22 UTC	15/16	gerrit:12f8dcc47
client-20-ib	review	Whamcloud Autotest	2011-05-20 03:22:37 UTC	2011-05-20 09:17:00 UTC	15/16	gerrit:c45e7eacc
client-23-ib	review	Whamcloud Autotest	2011-05-20 03:02:25 UTC	2011-05-20 09:32:47 UTC	16/17	gerrit:33340bf6c
client-23-ib	review	Whamcloud Autotest	2011-05-20 00:43:45 UTC	2011-05-20 02:40:59 UTC	1/2	gerrit:4f1aa57ed
client-20-ib	review	Whamcloud Autotest	2011-05-19 22:40:48 UTC	2011-05-20 05:03:31 UTC	16/17	gerrit:2c57a6a60
zwicky1	acc-sm-zwicky1	Prakash Surya	2011-05-19 20:25:08 UTC	2011-05-19 20:45:41 UTC	1/1	
client-8-ib	development	Chris Gearing	2011-05-19 18:03:08 UTC	2011-05-19 21:45:23 UTC	8/10	
client-23-ib	regression	Whamcloud Autotest	2011-05-19 17:05:32 UTC	2011-05-20 00:20:51 UTC	21/21	
client-20-ib	review	Whamcloud Autotest	2011-05-19 16:38:28 UTC	2011-05-19 22:21:54 UTC	17/17	gerrit:1b4f9f99f
client-23-ib	review	Whamcloud Autotest	2011-05-19 09:51:18 UTC	2011-05-19 15:25:47 UTC	17/17	gerrit:f4cded4b3
client-20-ib	regression	Whamcloud Autotest	2011-05-19 06:28:56 UTC	2011-05-19 14:57:04 UTC	20/21	

- Johann Lombardi