LUSTRE
Overview and Features Review

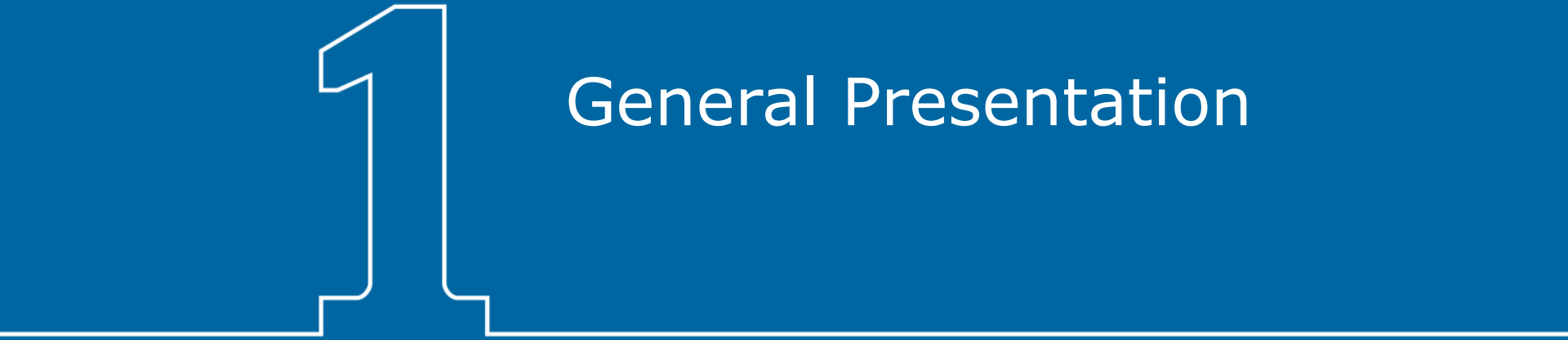
Grégoire Pichon
BDS R&D Data Management
gregoire.pichon@atos.net

10-2016

Bull
atos technologies

Content

- Lustre General Presentation
  - ID, History
- Lustre Architecture
  - Components, Typical Installation
- Lustre Main Features
  - File Striping
  - Directory Striping
  - Networking
  - Distributed Locking
  - High Availability
- Lustre Releases
  - Lustre 2.1, 2.2, 2.3
  - Lustre 2.4
  - Lustre 2.5
  - Lustre 2.6
  - Lustre 2.7
  - Lustre 2.8
  - Lustre 2.9
- What's next



General Presentation

ID	
Type	Distributed file system
Operating system	Linux
Written in	C
Developers	community
License	GNU GPL v2
Website	lustre.org - opensfs.org
Initial release	1.0.0 (December 2003)
Last release	2.8.0 (February 2016)

Lustre in the field

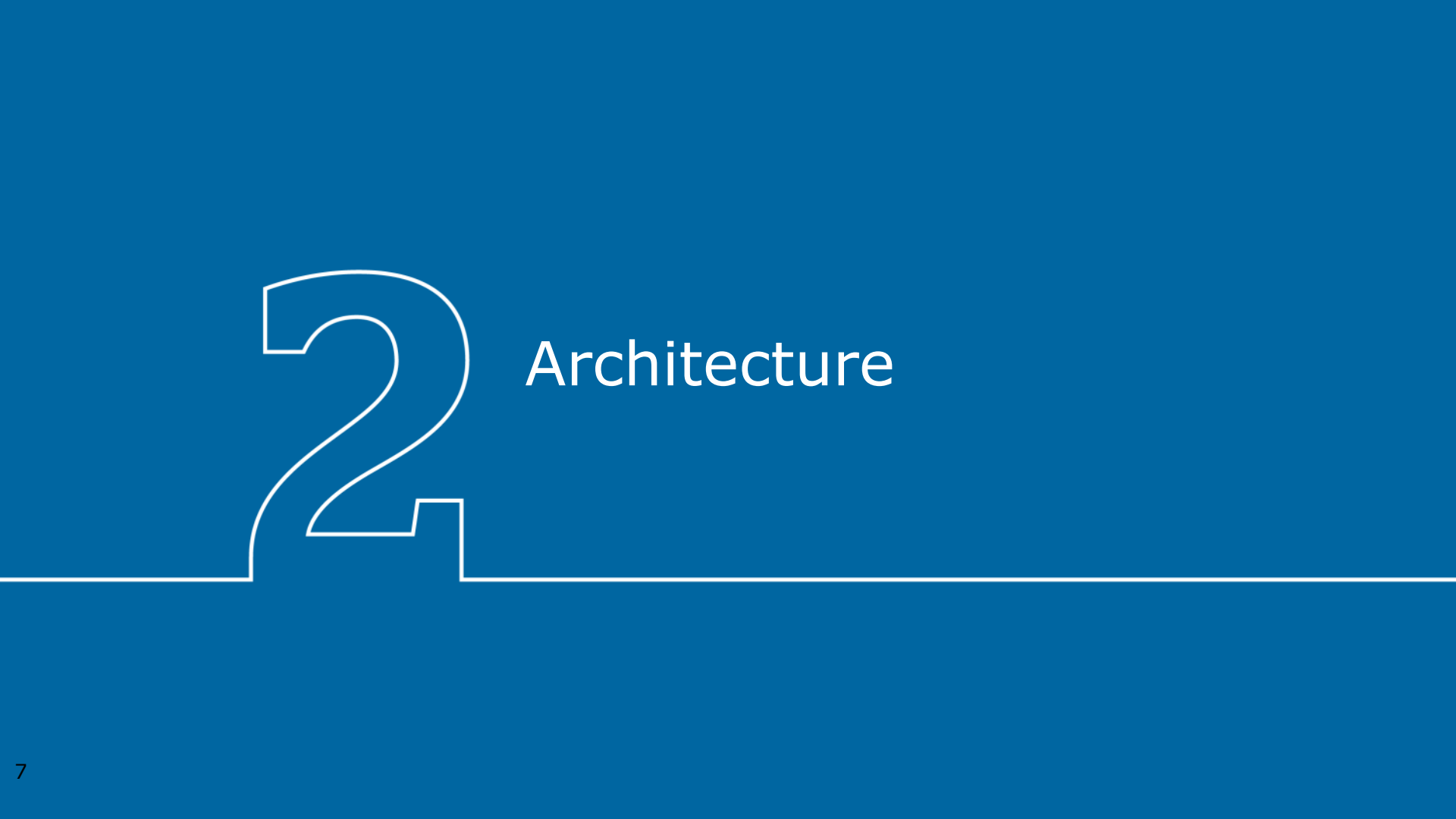
- High performance file system used for large-scale cluster computing
- Scalable to ~10 000 client nodes, ~10 petabytes, ~1 terabytes/s IO throughput

- From Top500 fastest supercomputers in the world, since June 2005, used by
  - at least half of the top10
  - more than 60% of top100

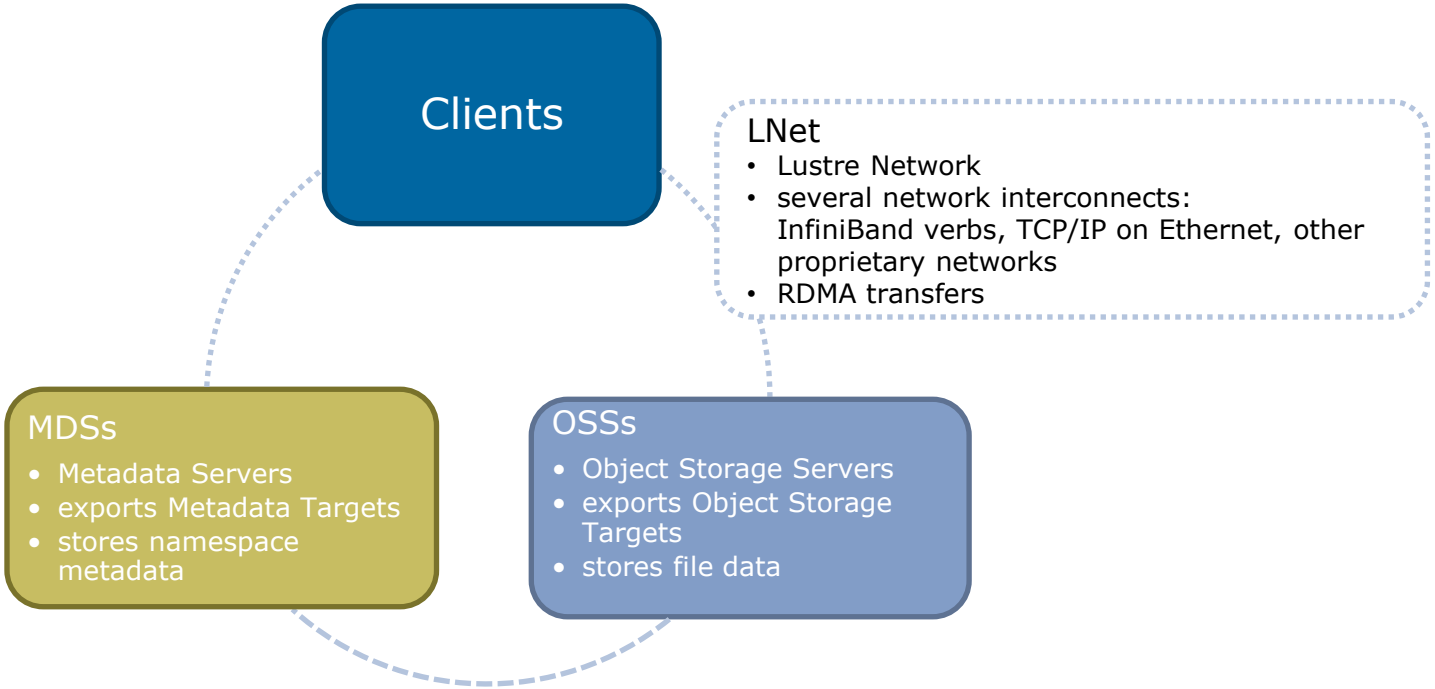
- Top sites with Lustre (Top500 November 2014)
  1. Tianhe-2, National Supercomputing Center
  2. Titan, Oak Ridge National Laboratory
  3. Sequoia, Lawrence Livermore National Laboratory
  4. K computer, RIKEN Advanced Institute for Computational Science
  6. Piz Daint, Swiss National Supercomputing Center
  26. Occigen, GENCI-CINES, delivered by Bull/Atos
  33. Curie, CEA/TGCC-GENCI, delivered by Bull/Atos

Lustre History

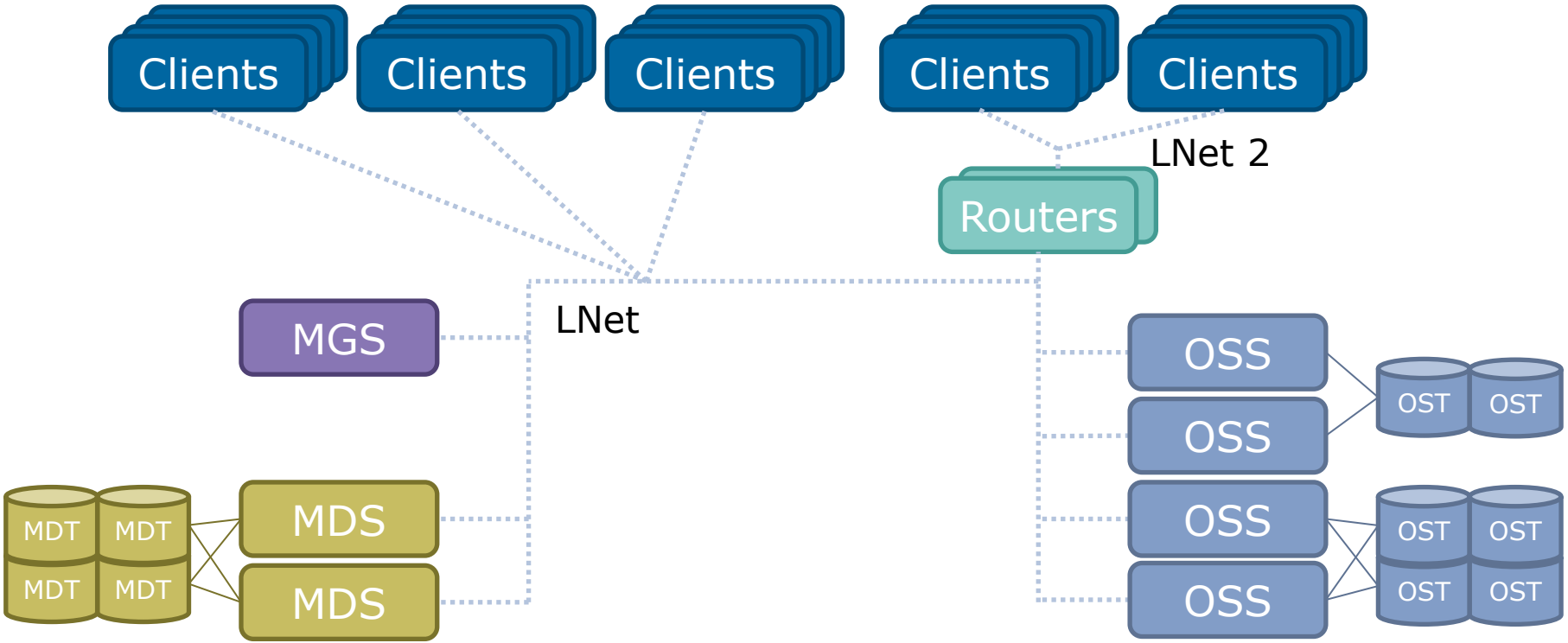
- 1999 - research project by Peter Braam at Carnegie Mellon University
- 2001 - Cluster File Systems company if founded
- 2002... - development of Lustre under the Accelerated Strategic Computing Initiative Path Forward project funded by US Department of Energy
- 2007 - Sun Microsystems acquires CFS. They intent to bring Lustre to ZFS file system and the Solaris operating system
- 2008 - Braam left Sun Microsystems. Eric Barton and Andreas Dilger lead the project
- 2010 - Oracle acquires Sun, but after a few months announce they would cease Lustre 2.x development and place Lustre 1.8 into maintenance-only support
- 2010 - creation of several organizations to provide support and development of Lustre in an open community development model: Whamcloud, Open Scalable File Systems (OpenSFS), European Open File Systems (EOFS).
- 2011 - Whamcloud gets a contract for Lustre feature development and a contract for Lustre 2.x source code maintenance
- 2012 - Whamcloud is acquired by Intel
- 2012 - FastForward project to extend Lustre for exascale computing systems
- 2013 - Xyratex acquires original Lustre trademark, logo and intellectual property


2
Architecture

Lustre Architecture



Lustre Typical Installation



Lustre Key Components

- Clients
  - sees a unified namespace
  - standard POSIX semantics
  - concurrent and coherent read and write access

- Management Server (MGS)
  - stores configuration information of the file systems
  - contacted by Lustre targets when they start
  - contacted by Lustre clients when they mount a file system
  - involved in recovery mechanism
  - not a critical component

Lustre Key Components

- Object Storage server (OSS)
  - exports Object Storage targets (OSTs)
  - provides an interface to byte ranges of objects for read/write operations
  - stores file data

- Metadata server (MDS)
  - exports Metadata targets (MDTs)
  - stores namespace metadata:
    filenames, directories, access permission, file layout
  - controls file access
  - tells clients the layout of objects that make up each file

Lustre Key Components

- OSTs and MDTs
  - uses a local disk file system: ldiskfs (enhanced ext4), ZFS
  - based on block device storage
  - usually hardware RAID devices, but works with commodity hardware

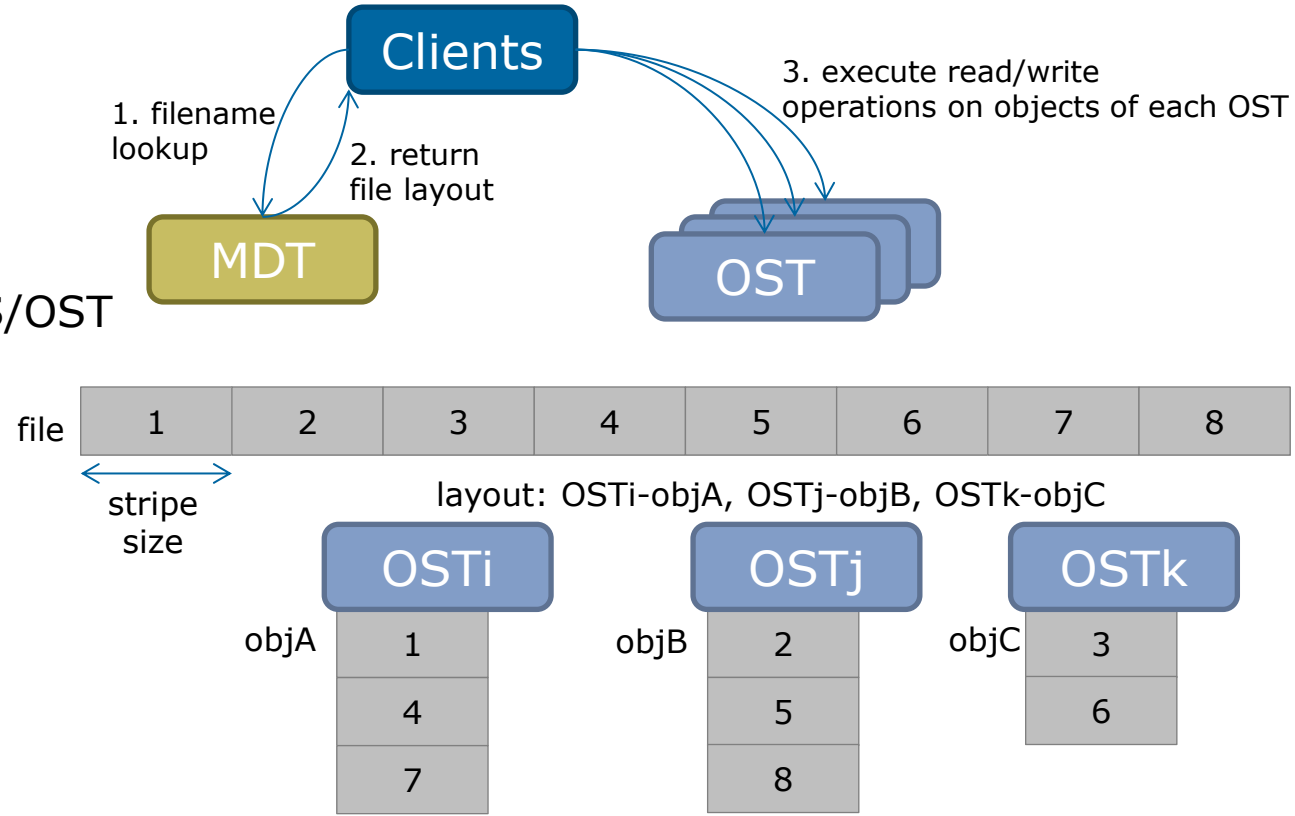
- Routers
  - gateway between two LNets

Main Features

Lustre File Striping

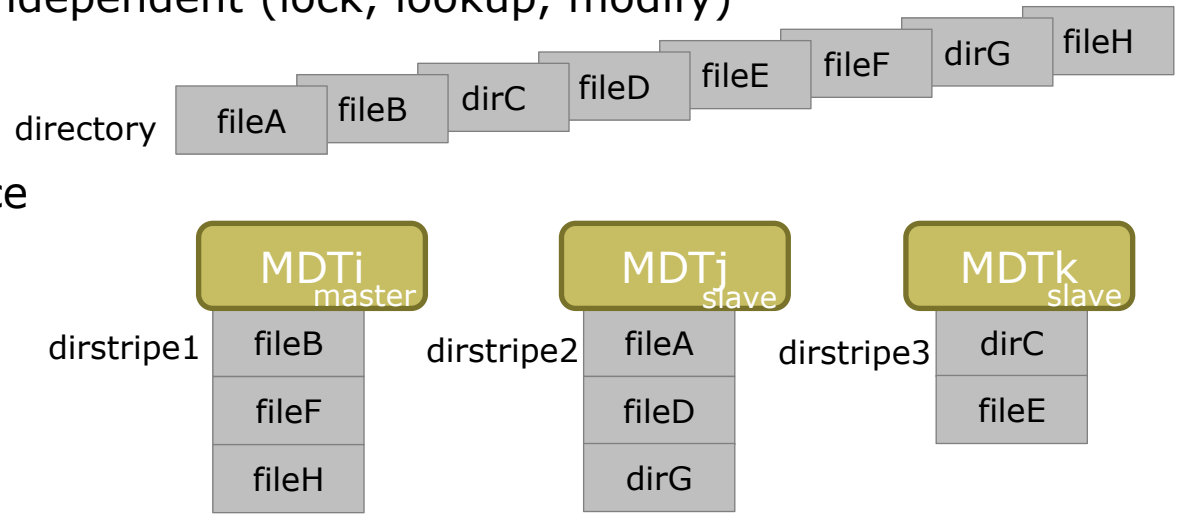
- File access flow
- File I/O bandwidth
  - aggregated OSS/OST bandwidth
- File layout
  - stripe size
  - stripe count

striping is similar to RAID0



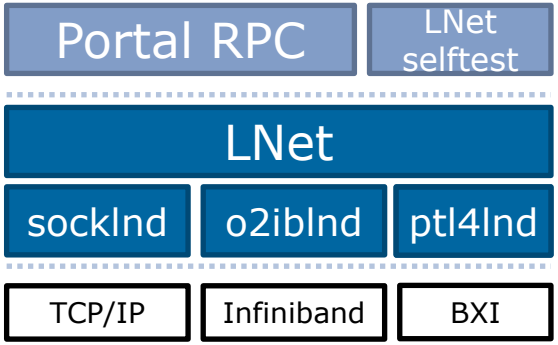
Lustre Directory Striping

- Spread a single directory across MDTs
  - improve performance for large directories (reduce contention)
  - client compute the appropriate MDT for a directory entry using: directory layout + name hash
  - directory stripes are independent (lock, lookup, modify)
- Directory performance
  - aggregated MDS/MDT performance
- Directory layout
  - stripe count
  - master index
  - hash type

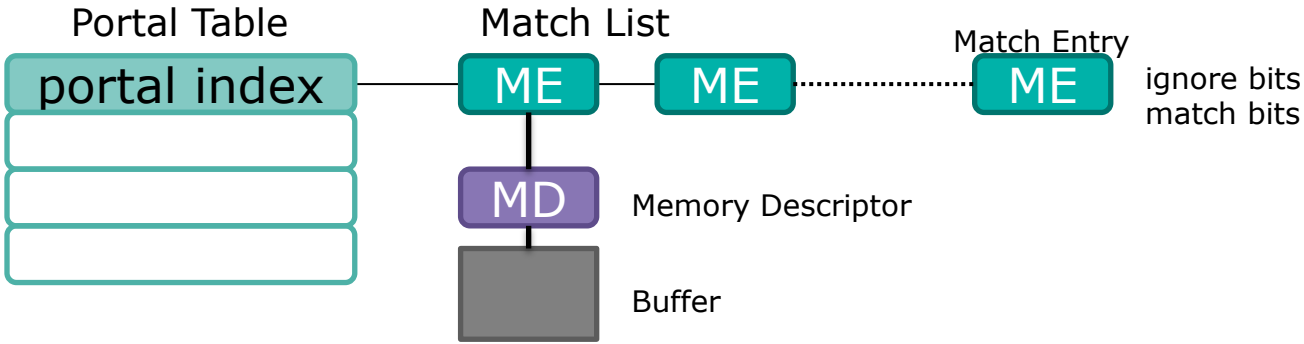


Lustre Networking (LNet)

- Message passing API
  - originated from Sandia Portals
  - supports several network interconnects with Lustre Network Driver (LND)
  - supports Direct Memory Access (DMA)



- LNet addressing scheme



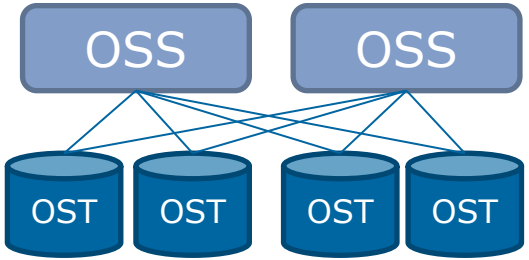
Lustre Locking

- Distributed Lock Manager (LDLM)
  - ensures consistency of concurrent file access by several clients
  - ensures coherency of data cached by the clients

- Data Lock
  - one LDLM instance per OST
  - file content: byte-range extent
  - allow
    - overlapping read extent locks for a file
    - non-overlapping write extent locks for regions of a file

- Metadata Lock
  - one LDLM instance per MDT
  - file lookup: owner and group, permission, mode, ACL
  - state of the inode: directory size, directory contents, link count, timestamps
  - layout: file striping

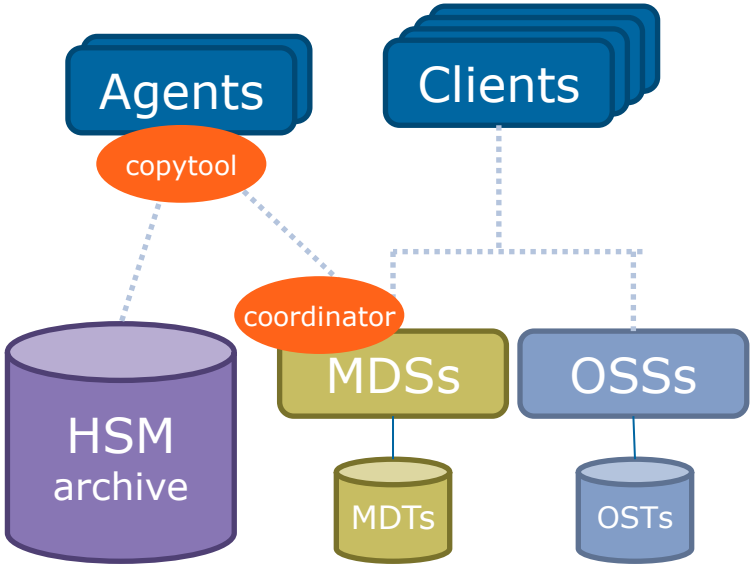
Lustre High Availability



- Server failures and reboots are transparent
- Target failover
  - based on shared storage
  - several OSSs (or MDSs) can alternatively mount each OST (or MDT)
- Recovery mechanism
  - based on the replay of client operations that had not been committed to disk
  - version based recovery: avoid dependency between replay of RPCs on different files

Lustre Hierarchical Storage Management

- Goal
  - extends Lustre targets storage with archive systems
  - transparent to fs users
- Single filesystem namespace
  - file metadata always remains in MDTs
  - file data can be either located on OSTs, on archive, on both
  - actions: archive, release, restore
- External Policy Engine is needed to automatically trigger actions



Lustre Administration Tools

- File system commands
  - mkfs.lustre format a block device for use as a Lustre target
  - tunefs.lustre modify configuration information on Lustre target
  - mount.lustre helper program that starts a Lustre target or mounts the client filesystem

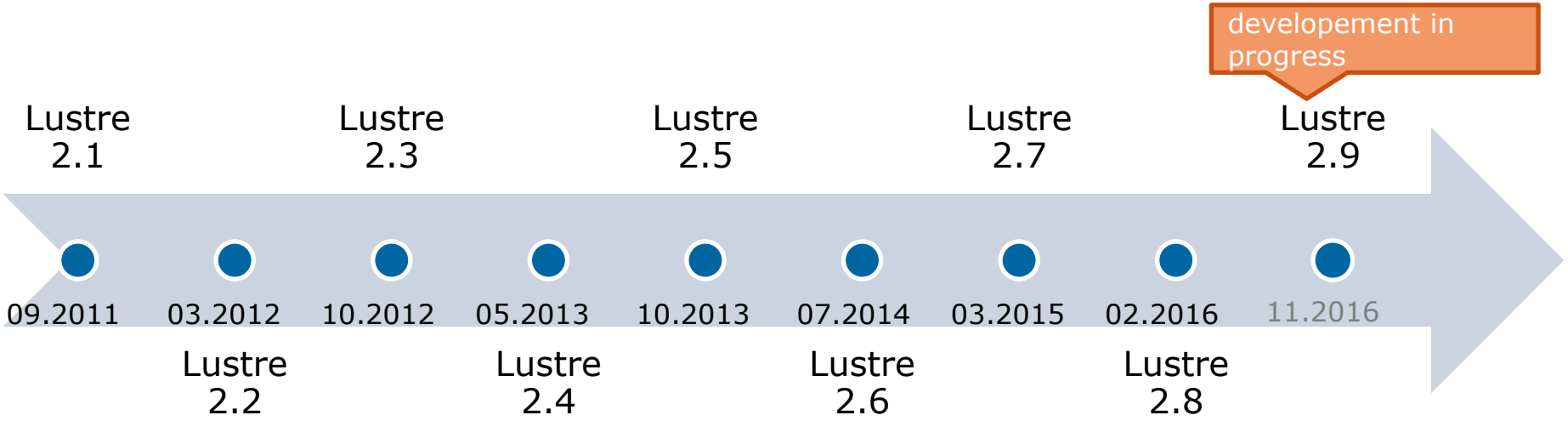
- Low level administration command lctl
  - network configuration (net up/down, list_nids, ping, peer_list, route_list, …)
  - device configuration (device_list, activate, deactivate, abort_recovery, …)
  - parameter configuration (list_param, get_param, set_param)
  - changelog configuration (changelog_register, changelog_deregister)
  - on-line Lustre consistency check and repair (lfsck_start, lfsck_stop)
  - debug configuration (debug_kernel, debug_list, debug_daemon, …)
  - pool configuration (pool_new, pool_add, pool_list, pool_remove, …)

Lustre User Tools

- User command lfs
  - get/set file striping (getstripe, setstripe)
  - get/set directory striping (getdirstripe, setdirstripe)
  - report disk space usage and inode usage (df)
  - show metadata changes (changelog)
  - convert Lustre file id to pathname (path2fid, fid2path)
  - get/set quota usage and limits (quota, setquota)
  - search in the directory tree the files that match some parameters (find)
  - get/set hsm information (hsm_state, hsm_set, …) and trigger actions (hsm_archive,…)
  - swap the data of two files (swap_layout)

Releases

Lustre Releases



Lustre 2.1, 2.2, 2.3

- Lustre 2.1 released in September 2011
  - support Lustre server on Red Hat Linux 6
  - performance and stability improvement
- Lustre 2.2 released in March 2012
  - parallel directory operations: allow multiple clients to traverse/modify a single shared directory concurrently
  - imperative recovery: faster recovery from server failures
  - increased stripe count: up to 2000 OSTs
  - improved single-client directory traversal performance (stat_ahead)
- Lustre 2.3 released in October 2012
  - metadata improvements for fat SMP servers (reduced contention, affinity awareness)
  - LFSCK: verify and repair the MDS Object Index (OI scrub) at runtime
  - allow per-job IO statistics on servers

Lustre 2.4

released in May 2013

- Distributed Namespace (DNE)
  - metadata capacity and performance scaling
- ZFS backing filesystem
- LFSCK (Lustre file system check)
  - scan and verify the consistency of MDT FID and LinkEA attributes
- Network Request Scheduler (NRS)
  - framework to define policies to optimize client request processing
  - example: disk IO ordering, fairness, Qos
- Large bulk IO
  - increase the OST bulk IO maximum size to 4MB
  - more efficient disk IO submission
- Support Lustre client on Linux kernels up to version 3.6

Lustre 2.5

released in October 2013

- Hierarchical Storage Management (HSM)
  - implements clusters with tiered storage solutions
  - file content can be archived on large-capacity long-term storage while still being present in file system namespace
  - needs a PolicyEngine to automatically trigger archive and release policies

Lustre 2.6

released in July 2014

- LFSCK functionality
  - local consistency checks on the OST
  - consistency checks between MDT and OST objects
- Client single thread performance improvement
- Distributed Namespace (DNE) (preview)
  - striped directories
  - allows single large directories to be stored on multiple MDTs
  - improve metadata performance and scalability

Lustre 2.7

released in March 2015

- LFSCK
  - MDT-MDT consistency verification
  - remote directories and striped directories between multiple MDTs
- Dynamic LNet configuration
  - configure LNet at runtime
  - update of network interfaces, routes, routers
- DNE
  - improvement to striped directory functionality
- UID/GID mapping (preview)
  - map UID/GID for remote client nodes to local UID/GID on the MDS and OSS
  - allows a single Lustre filesystem to be shared across clients with different administrative domains
- File striping with list of OSTs

Lustre 2.8

release in February 2016

- LFSCK
  - performance improvements
- DNE
  - asynchronous commit of cross-MDT (improved performance)
  - remote rename and remote hard link functionality
- Single client metadata RPC scaling developed by Bull/Atos
  - allow several modifying metadata requests in parallel, while ensuring consistent recovery
- Security developed by Bull/Atos
  - SELinux support on clients
  - Kerberos revival: authentication and encryption support
- Network Request Scheduler (NRS): Delay policy
  - simulate server load by using NRS for fault injection

Lustre 2.9

in progress

- Shared Key Crypto
  - allow node authentication and RPC encryption using symmetric shared key crypto with GSSAPI
  - avoids complexity in configuring Kerberos across multiple domains
- UID/GID mapping
- Subdirectory Mounts
- Server Side Advise and Hinting
- Large Bulk IO
  - increase the OST bulk IO maximum size to 16MB or larger

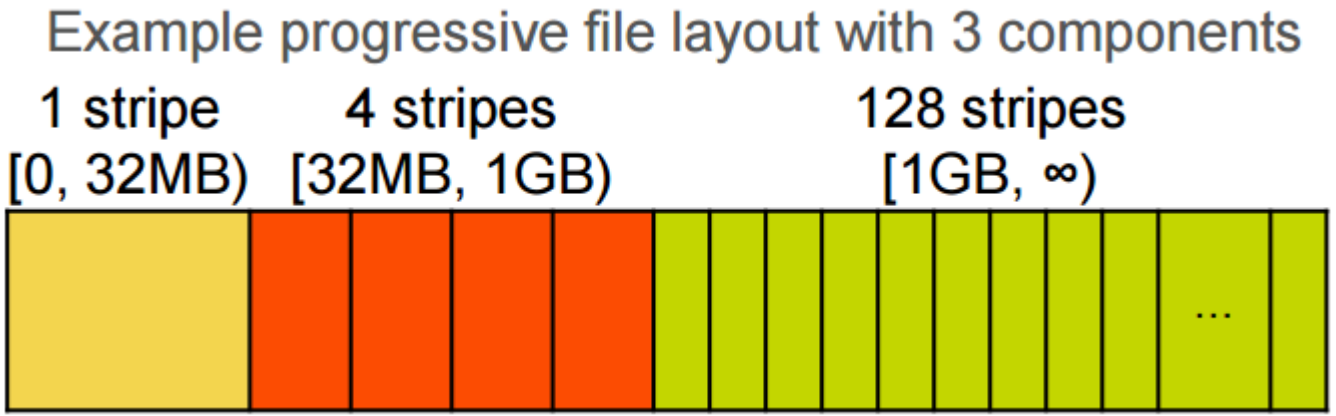
What’s next

What's next ?

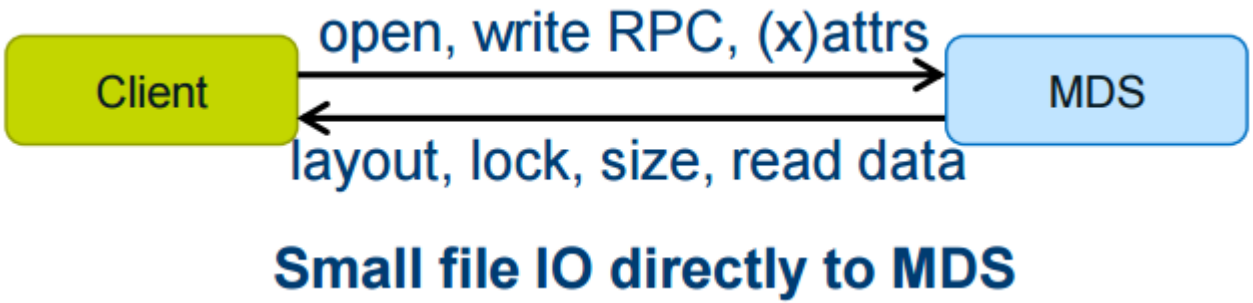
- Patchless server
  - remove Lustre kernel patches
  - allow Lustre servers to be more easily ported to new kernels and to be built against vendor kernels

- Layout Enhancement
  - composite file layout: support for multiple layouts on a single file
  - File Level Replication
    - RAID1 Layout, immediate asynchronous write from client
  - Data on MDT
    - allow small files to be stored directly on MDT for reduced RPC traffic
  - extent-based layouts

Composite File Layouts



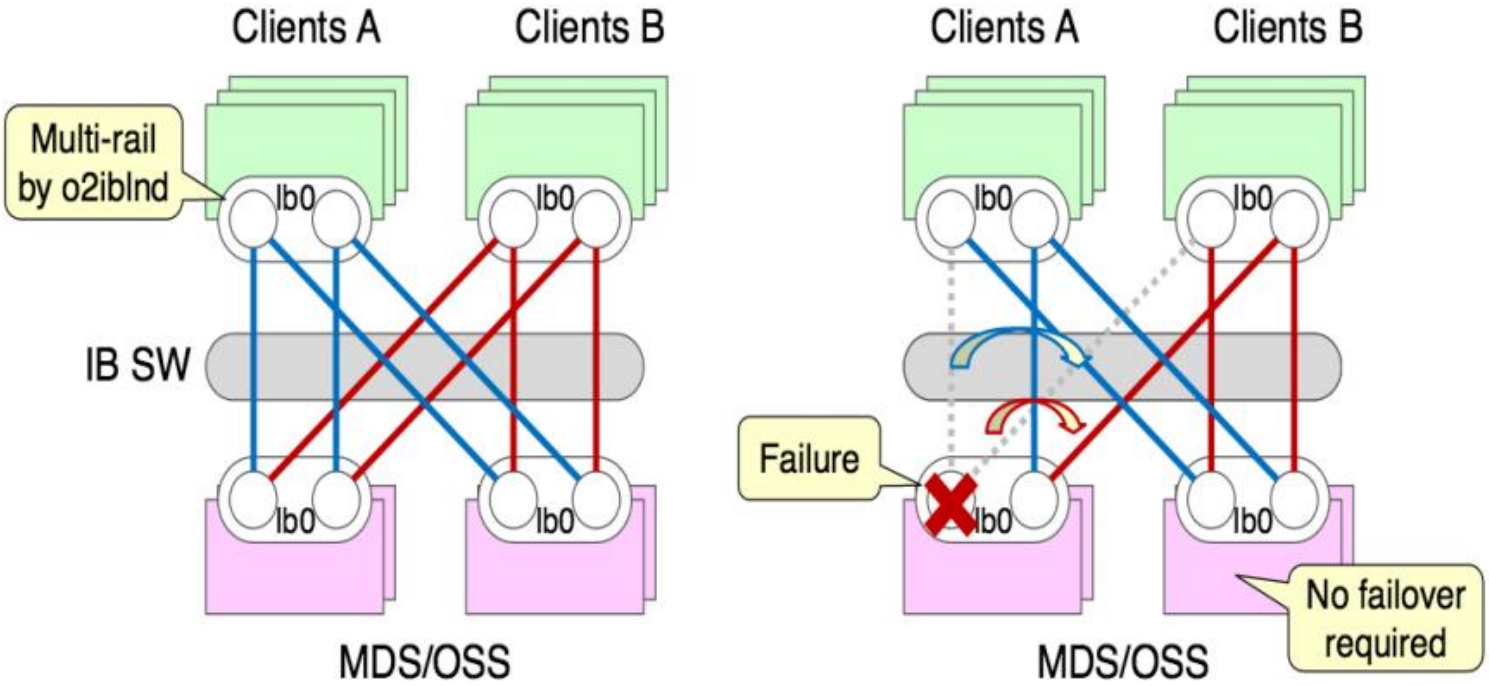
Data on MDT


Client
open, write RPC, (x)attrs
MDS
layout, lock, size, read data
Small file IO directly to MDS

What's next ?

- Multi-rail Lnet
  - allow LNet across multiple network interfaces
- Lustre Client stack improvement
  - CLIO cleanup and speedup
- ZFS Intent Log Support in Lustre osd-zfs
- Quota for project
  - allow specifying a "project" or a "subtree" identifier for files
  - quota accounting to a project, separate from UID/GID
- Support of btrfs as OST backend (osd-btrfs)
- Lock ahead
  - allow user space to request LDLM extent locks in advance of need
  - improve shared file IO performance

Multi-Rail LNet



Bull
atos technologies