Data On MDT For Lustre* Small File IO

Mikhail Pershin, Intel HPDD

Paris LAD, Sep, 2016

Lustre small file performance problem

- Read/write performance is currently optimized for large files
- Large files are striped across many OSTs
  - Data accessed in parallel
- For small files only a single OST is used
  - No parallel data access
  - Just like network filesystem
- Small file access needs more RPCs per MB than large files
  - Increased latency when competing with large file IO from many clients

Large files vs small files IO

Large files

- Data is striped across many OSTs
- Parallel access to all stripes
- Lots of data at one time
- Writes less latency sensitive
- Streaming IO pattern
- Works well with RAID-6

Small files

- Data is placed on single OST
- No parallel access
- More seeking to access data
- More latency sensitive
- Can't do data read-ahead
- Works well with RAID-1+0
- NFS-like but with more RPCs

Why small files are important

Big Data is about Big Files, isn't it?

- Most files (70%) are small, most data (~90%) is in big files*
- Number of small files is big though used space is not
- Small file consumes resources like a big one
- Small files can produce big slowdown
- Latency of access to small files is important

* "2014 NERSC Workload Analysis", page 41+

http://portal.nersc.gov/project/mpccc/baustin/NERSC_2014_Workload_Analysis_v1.1.pdf

Data-on-MDT as possible solution

Data-On-MDT (DoM) was started to improve small file IO performance

- Place small files data on the MDT
- Helps to avoid extra RPCs – less RPC pressure on OSTs
- Less I/O overhead on OSTs
- Avoid blocking small IOs behind large streaming IO workloads
- New benefits from data placed on MDT are possible
- OpenSFS funded the Data-on-MDT design

  http://wiki.opensfs.org/images/b/be/DataonMDSDesign_HighLevelDesign.pdf

Lustre changes required

Changes to client IO path and file layout for DoM

- A new layout for DoM files, set by `lfs setstripe`
  - Example: `lfs setstripe –P mdt –S <stripe size> <path>`
- Client is able to send IO requests to an MDT
  - Extend CLIO stack with MDC
  - Use the existing OSC code
- MDT is able to serve incoming IO requests
  - New IO service threads
  - IO methods at MDT
- Migration from MDT to OST by `lfs migrate`

DoM and PFL

When small file on MDT grows it should be extended to OSTs

Use Progressive File Layout (PFL) for that

- Change layout to use new stripes on OSTs
- The first stripe is kept on MDT
- No migration is needed immediately when file grows
- MDT stripe can be moved by administration tools at any time
- MDT stripe could be moved in background by a policy engine like RobinHood

Implementation overview

- Initial prototype developed on Lustre 2.7
  - will be pushed to master shortly for broader testing and reviews
- Use IO IBITS lock to protect the whole MDT stripe
- DoM layout can be set by `lfs` on file or directory
- Lock at open, glimpse-ahead and read on open optimizations on MDT
- Migration between servers will be possible with `lfs migrate`
- PFL to be used to grow files beyond the maximum MDT stripe size

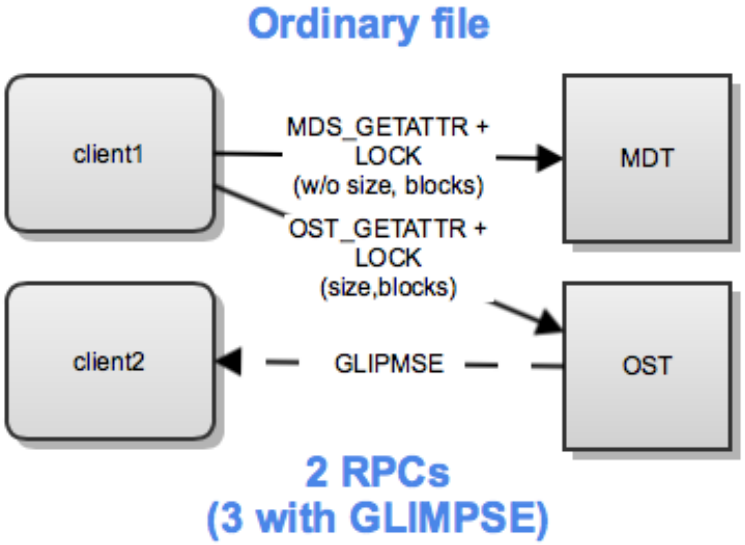
Data-on-MDT optimizations

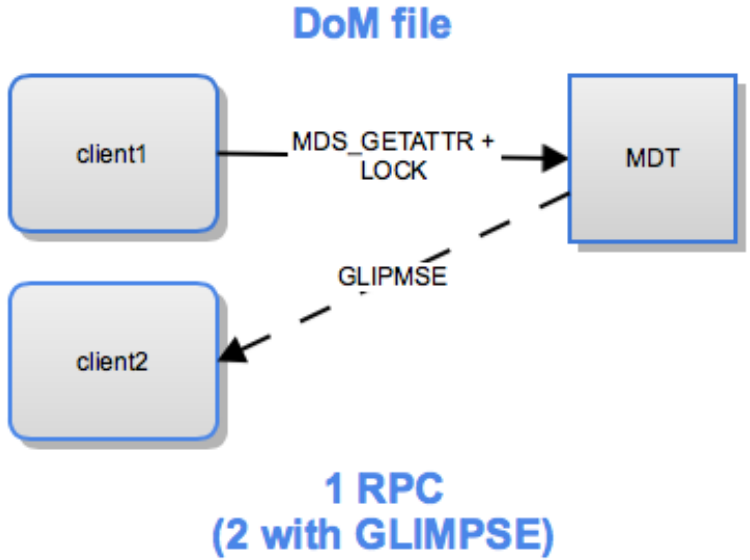
There are a number of optimizations that are possible with DoM

- Glimpse-ahead: since MDT is controlling data locks it is possible to call glimpse callback from MDT when size attribute is returned to the client
- Lock on OPEN: when file is opened for read or write the IO lock can be returned immediately saving us extra ENQUEUE RPC
- Read on OPEN: for small file it is possible to return data in OPEN RPC reply buffer avoiding possible READ RPC, like read-ahead for small files
- Read on STAT: the same as above but during STAT request

Optimization: glimpse-ahead

MDT returns size always with GETATTR and OPEN requests, it does extra GLIMPSE call to the client if it is needed.



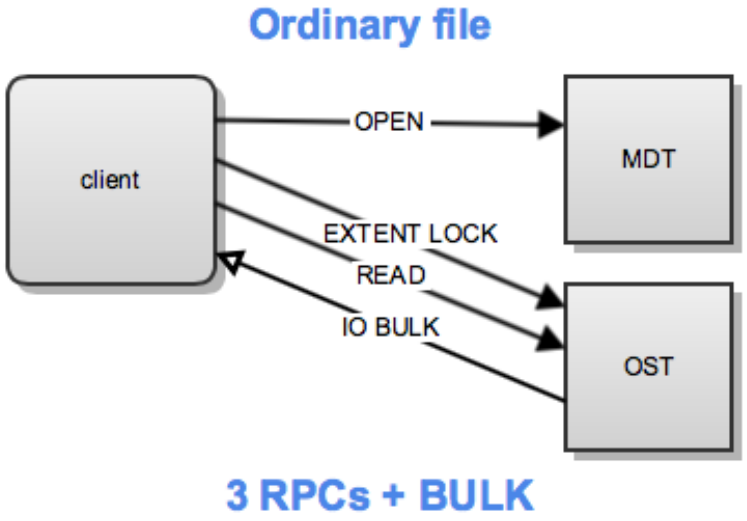


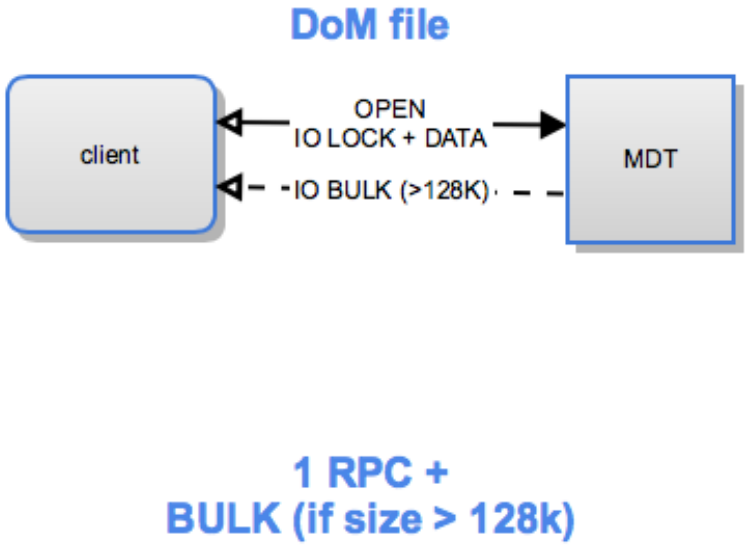
Optimization: Lock on OPEN

MDT takes IO lock along with OPEN and returns it to the client saving extra ENQUEUE RPC.

Optimization: Read on OPEN

MDT reads file data and return in OPEN reply buffer. With buffer growing it is possible to return up to 128K data or use BULK otherwise.





Data-on-MDT Benchmarks

Test setup

1 client, 1 MDS, 2 OST

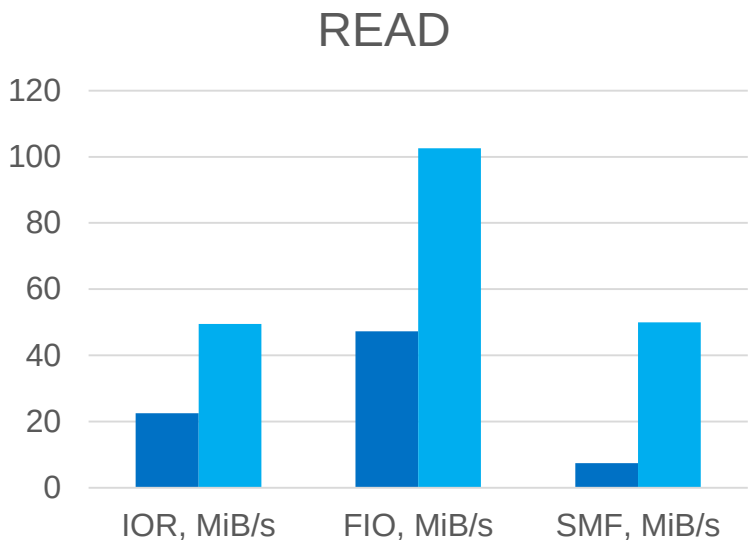
- Intel® Xeon E5-2699 2.3GHz
- 64Gb DDR4
- 1Tb SATA HDD 7200rpm
- Intel InfiniBand*
- All nodes are the same HW

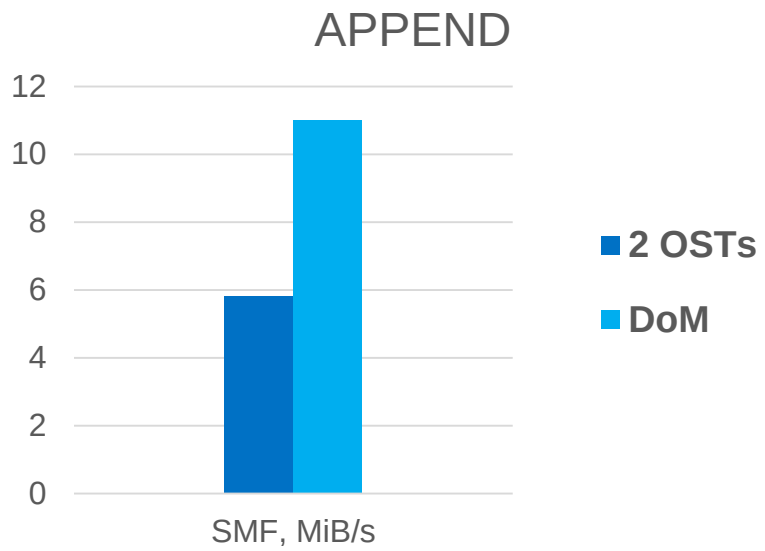
Benchmarks setup

- IOR, FIO, SMF (smallfileio)
- File per process
- Random read/write
- 32KiB filesize
- 4KiB IO size
- 65K files
- 16 threads

READ and APPEND

16 threads, 64k files, 4KiB IO size





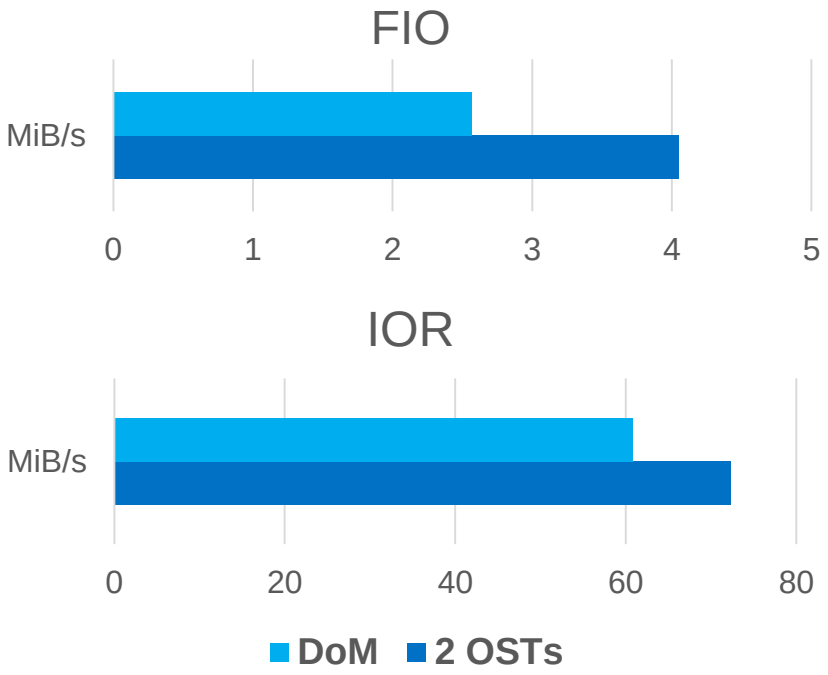
See slide 13 for configurations

WRITE operation

16 threads, 64k files, 4KiB IO size

DoM can’t help with WRITE on the same hardware

- WRITE to many files is balanced between many OSTs
- single MDT serves them all
- DNE helps
- MDS requires better HW



See slide 13 for configurations

Concurrent write workload

Small IO: 64KiB filesize, 65k files, 16 threads
Large IO: 1GiB filesize, 4 files, 4 threads
Both jobs run in loops for 10 minutes (FIO with jobs file)

	All on OSTs		DoM + Large IO on OSTs	
	Bandwidth, MiB/s	Latency, ms	Bandwidth, MiB/s	Latency, ms
Small IO	5.3	*191*	3.5	*377*
Large IO	**2.3**	***7098***	**111**	***147***

See slide 13 for configurations

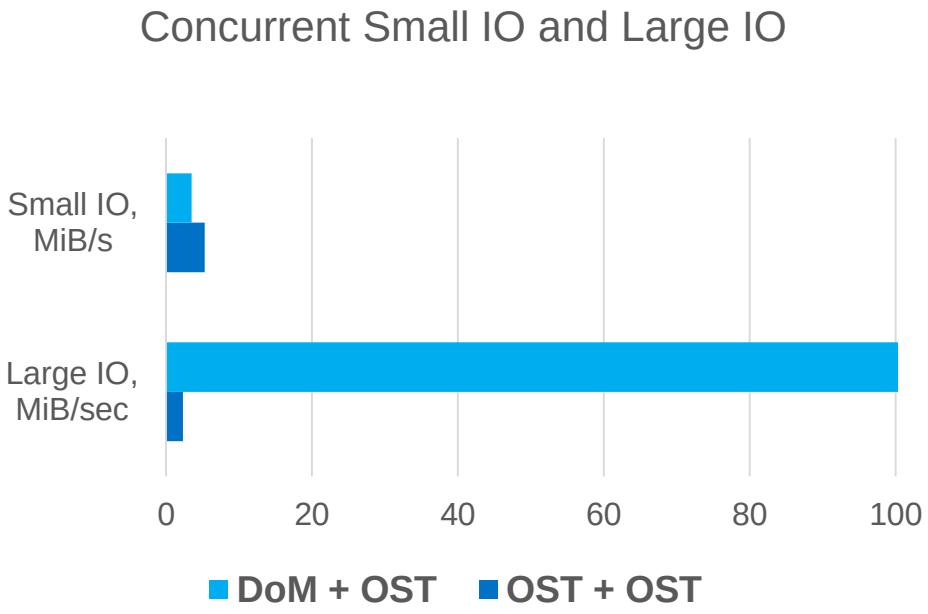
Concurrent write workload cont.

Small and Large IO on OST

- Small random IO affects large streaming IO badly
- Streaming bandwidth is dropped to the level of Small IO bandwidth
- Latency is terrible

Small IO on MDT, Large IO on OST

- Streaming IO is not affected by small random IO and shows full potential



See slide 13 for configurations

New MDS requirements

DoM changes MDS role in general Lustre setups

- MDT needs more space to store small files
- MDS should handle bigger RPC pressure
- That makes sense to use faster storage on MDS
- SSD-based MDS can be perfect for DoM
- DNE setup allows scaling a metadata performance
- Can use larger capacity MDTs for additional DNE MDTs

Availability

- The code will be available in master branch for anyone interested
- Targeted for Lustre 2.11, depends on PFL feature landing first

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.

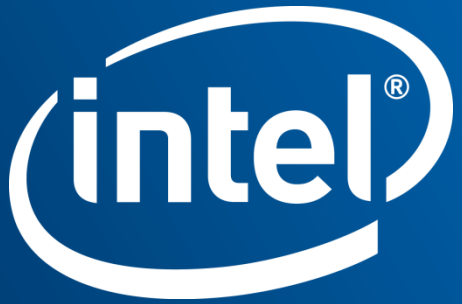
This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

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.


intel®