Burst Buffer and File Level Redundancy

Gabriele Paciucci - RSDD

Jinshan Xiong - HPDD

File Level Redundancy (FLR) goals

Software, network, hardware all contribute to Lustre data unavailability

- FLR should provide redundancy to Lustre file system
  - HA is no longer necessary for OSTs
  - Possible to be deployed on lower cost commodity hardware
  - Configurable redundancy level on per-directory or per-file basis
- Based on Progressive File Layout (PFL)
  - A file replica is composed of one or more PFL components
- Superb I/O performance
  - Write is as fast as for non-FLR files
  - Aggregate read performance is scalable to the number of replicas
- POSIX compliance

FLR Development and Release Plan

Phase 0: Composite Layouts from PFL project (2.10)

Phase 1: Delayed writing mirroring - depends on Phase 0 (2.11)

Phase 2: Immediate write mirroring - depends on Phase 1

Phase 3: Integration with policy engine/copytool - needs Phase 1

Phase 4: Erasure coding for striped files - depends on Phase 2

FLR Development Status

Targeting to land Phase 1 to 2.11

Redundant read

- Data can be read from any available replica
- Client and layout policies to determine which replica to read

Write is partially redundant

- Lustre needs to pick one available replica to write
  - what if the just picked replica becomes unavailable?
- Data will be resynchronized among replicas asynchronously
- Not necessary to resync entire replica, FLR will synchronize modified components only

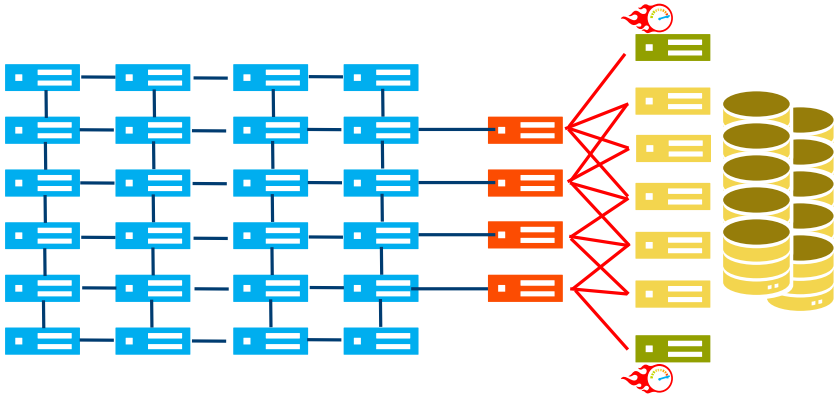
FLR DEMO

FLR use cases

FLR is a fundamental technology that multiple use cases can build upon:

- Multi-tiered storage with preferred read replica
  - SSD OSTs as the first tier and HDD OSTs as the second
  - HDD OSTs will be accessed only when the SSD OSTs are out of reach
  - Pre-stage data before job start for reads
- Burst buffer with preferred write replica
  - NVMe OSTs for fast checkpoint write for HPC
- Increase parallel read performance for many-client reads
- Play it for fun – what a nerd!

Burst buffer market landscape and requirements



- Checkpoint and restart
- File alignment and block alignment
- Staging and demand loading
- Protect from latency bound applications
- Out-of-Core I/O
- Data analysis support

- Several commercial offerings
- Start to be included as requirement in tenders

FLR Based Burst Buffer Solution

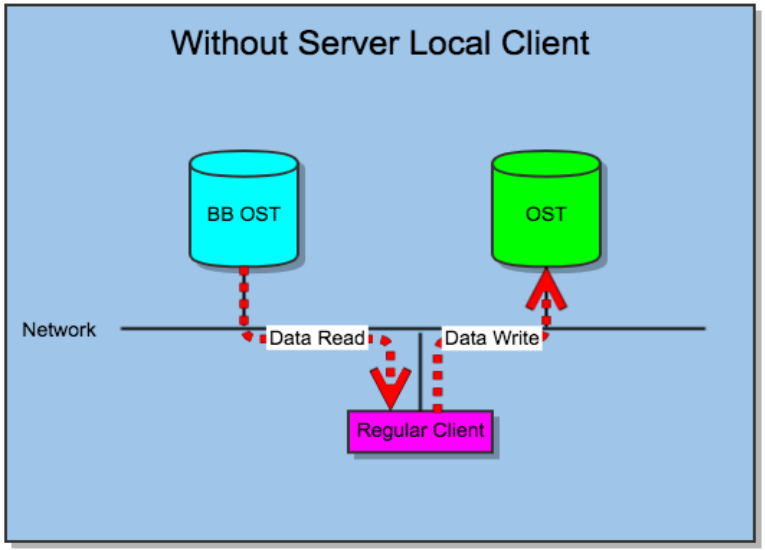
- Benefits of FLR based burst buffer
  - Based on existing mature and proven technology
  - Burst buffer as OSTs in Lustre file system
  - Unified name space, easier for data management
- Data first go to burst buffer OSTs, then migrates to secondary OSTs in background
  - Workload on computing nodes can proceed while data is migrated
- Request burst buffer space on demand, reclaimed by deleting replicas from BB OSTs
  - Highly efficient prestage if checkpoint data are already on burst buffer OSTs
- Burst buffer capacity should be at least three times checkpoint size
  - One checkpoint writing, one checkpoint "safe", one checkpoint migrating

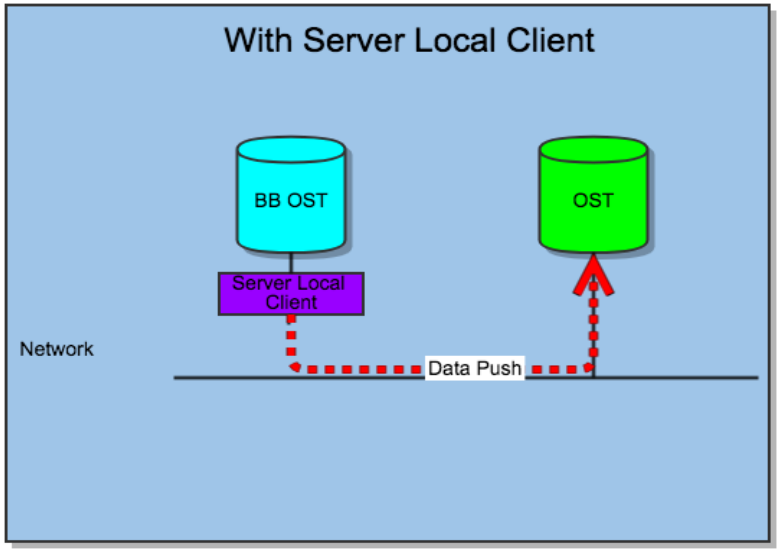
MDT Object Inventory (MOI) File

- Lustre depends on external databases for data management
  - Slow and painful access and update
  - Consistency between Lustre and database is always concerned
  - Limited by Lustre Changelog scalability
- Propose to maintain MOI files on MDTs that list all objects for each OST
  - MOI iteration to list all file FIDs for OSTs under recovery or evacuation
    - Should be as fast as reading a regular file on the MDT
  - No database needed for data management if iteration is fast and compact
    - Policy engine can still maintain in memory cache for optimization
  - Slight overhead on file creation and deletion

Server Local Client

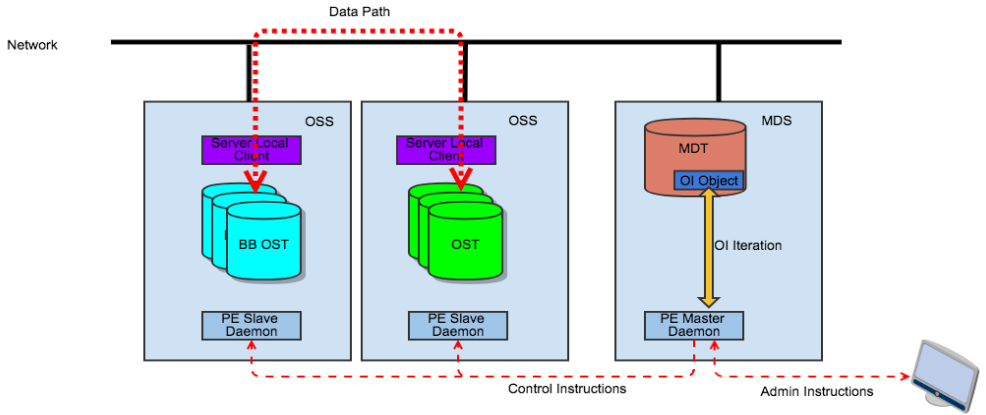
- High efficient data move engine between burst buffer and secondary OSTs
  - Use half of network bandwidth for data transfer, with minimum Lustre protocol overhead





Burst Buffer Policy Engine (PE)

- PE daemon for burst buffer running on each server node
  - No dedicated client node for PE
- Master daemon iterates MOI file
  - Get a list of files created/deleted on burst buffer OSTs
  - Instruct slave daemons for data move
- No database for PE
- Burst buffer command via PE master
  - Imperative data move
  - Burst buffer space monitoring
  - Prestage, etc



Conclusion and call to action

- The FLR technology will modernize Lustre
- Lustre will manage different storage tiers transparently for legacy applications
- Data protection/erasure code at filesystem level simplifies cloud adoption
- Data availability without HA software or multi-port storage
- Intel is supporting these innovations
- Feedback from FLR Phase 1 is welcome
- Intel is looking forward to FLR community adoption and contributions

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.

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, the Intel logo and Intel® Omni-Path 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®
experience
what's inside™