



Progressive File Layout (PFL)

Niu Yawei

Oct. 2016

Legal Information

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.

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.

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.

© 2016 Intel Corporation

Agenda

- Current File Layout
- PFL Goals
- Composite File Layout
- Tools for Using PFL File
- Future Work
- Q/A

Current File Layout

- Only simple RAID0 (striping) layouts are supported
- File layout is decided on creation, can't be changed.
- Several ways to specify layout of a file:
 - Create file by 'lfs setstripe'
 - Inherit default layout from parent directory
 - Inherit FS default layout

PFL Goals

- Minimize stripe count for small files
 - Avoid overhead of creating multiple OST objects
 - Avoid stat/size overhead for many stripes
- Maximum stripe count for large files
 - Use all available OST bandwidth
 - Distribute OST space usage to avoid space imbalance
- Flexibility of layouts for different uses
 - One layout is suitable for a majority of uses and files
 - Allow for variability and flexibility of users, applications, sites

Composite File Layout

- Layout is composite of multiple components
- Each component is a plain V1/V3 layout and describes extent of file
- Component extent can't be disjoint or overlapping for PFL layout



Tools for Using PFL File

- 'lfs setstripe', 'lfs getstripe', 'lfs migrate' and 'lfs find' are modified to support PFL, bunch of new component options introduced:
 - **--component-end** | **-E**: component extent end
 - **--component-start**: component extent start
 - **--component-id** | **-I**: component ID
 - **--component-count**: component count
 - **--component-flags** | **-F**: component flags
 - **--component-add**: Add component(s)
 - **--component-del**: Delete component(s)

Tools for Using PFL File (cont)

- Create a 2 components PFL file
 - `lfs setstripe -E 4M -c 1 -E 256M -c 4 /mnt/lustre/pfl`
- Add/Delete a component to/from an existing PFL file
 - `lfs setstripe --component-add -E -1 -c 16 /mnt/lustre/pfl`
 - `lfs setstripe --component-del -I 3 /mnt/lustre/pfl`
- Migrate a plain file to a PFL file
 - `lfs migrate -E 1M -c 1 -E 64M -c 4 -E -1 -c 8 /mnt/lustre/plain`
- Set default PFL to a directory
 - `lfs setstripe -d -E 1M -c 1 -p pool_fast -E EOF -c 4 -p pool_large /mnt/lustre/dir`

Future Work

- Create OST objects for PFL file on demand during write
 - Currently create all objects at open
- Other layout support: file level replication, etc.

Q&A

