
# Progressive File Layouts Prototype

**John L. Hammond, Intel High Performance Data Division**

**LUG 2015**

**April 14, 2015**

# Agenda

- Project Overview
- About Progressive File Layouts
- Goals & Use Cases
- Design
- Future Work

# Project Overview

- Joint development effort between Intel High Performance Data Division (HPDD) and Oak Ridge National Laboratory (ORNL)
- Funded by ORNL
- Collaborators
  - Michael J. Brim (ORNL)
  - Andreas Dilger (HPDD)
  - Jason J. Hill (ORNL)
  - Neena Imam (ORNL)
  - Josh Lothian (ORNL)
  - Sarp Oral (ORNL)
  - Joel W. Reed (ORNL)
  - Jinshan Xiong (HPDD)

# About Progressive File Layouts (PFL)

- *Layout* is striping (count, width, objects,…)
  - File data backed by one or more OST objects.
  - Fixed attribute of file.
  - Usually determined on creation (default, lfs setstripe, …)
- Progressive Layouts allow increasing the stripe count as file size increases beyond specific thresholds. For example:
  - Use a single stripe for the first 2MB.
  - Use four stripes from 2MB to 256MB (if needed).
  - Use 32 stripes from 256MB to infinity (if needed).

# Goals

The goal of PFL is to resolve the tension between the use cases for small and large stripe counts:

- Small stripe counts offer better performance for operations like create, stat, unlink.
- Files with small stripe counts have better availability than those with large stripe counts.
- Large stripe counts offer better IO performance.
- Large stripe counts are less likely to create out of space conditions on OSTs.

# Use Cases

Progressive layouts allow us to:

- “defer” choosing a stripe count until we know if the file will be small or large (assuming this is not known in advance).
- Place different regions of files on different (types of) targets. For example:

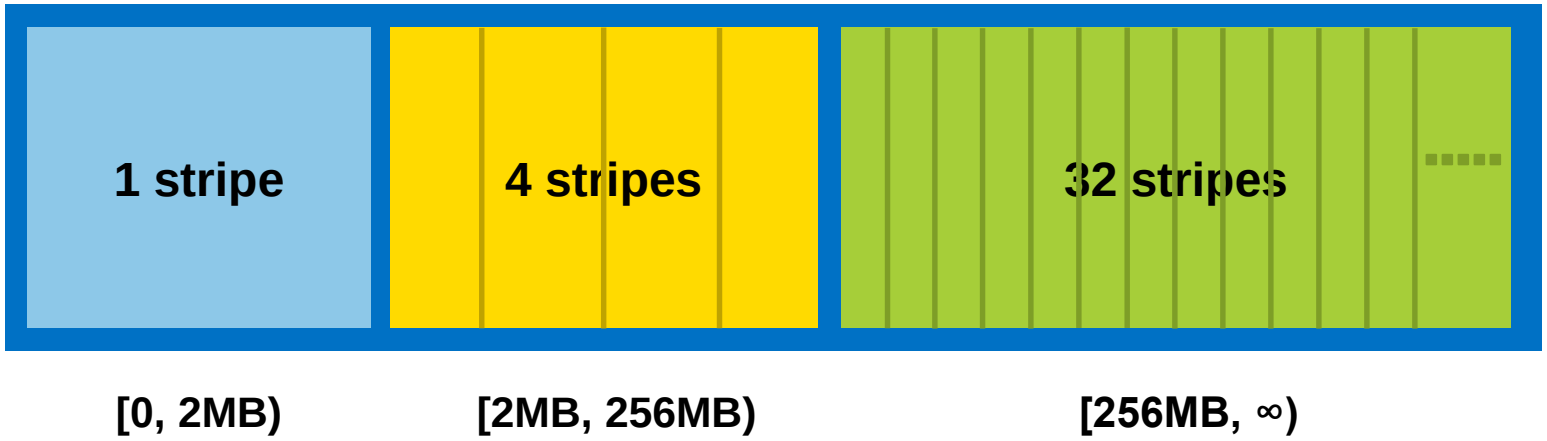
  First 16MB on SSD backed OSTs, remainder on SATA.

- Defer placement of data to avoid out-of-space OSTs.

# Design 1

PFL uses *composite extent-mapped layouts* to allow different RAID-0 layouts to describe different extents of a file.

**File with 3 components:**



# Design 2

Composite layouts are expressed as a list of simple (LMM v1 or v3) layouts with extents:

- Header contains number of components
- Each component has an id, extent, …
- Component layouts stored as blobs

# Design 3

- New ioctl to add new components with specified extent and striping
  - No overlap among component extents.
  - No holes in mapped region, except perhaps at tail.
  - Some restrictions on alignment of extent boundaries.
- Extended lfs setstripe and llapi wrappers for ioctls
- LOV layer interprets composite layouts and dispatches I/Os to appropriate components
  - I/Os to unmapped regions return –ENODATA.
  - Truncate to an unmapped file size returns –ENODATA.

# Prototype Use Pattern

Create FILE with a single stripe in [0, 2MB) using one of

```
$ lfs setstripe [--begin=0] --end=2MB --stripe-count=1 FILE
fd = llpfl_create(FILE, {stripe_count=2}, 0, 2 << 20);
```

Applications may now write (and read) upto [0, 2MB).

Add a four stripe component with extent [2MB, 256MB) using one of

```
$ lfs setstripe [--begin=2MB] --end=256MB --stripe-count=4 FILE
fd = llpfl_setstripe(FILE, {stripe_count=4}, 256 << 20);
```

Applications may now write (and read) upto [0, 256MB).

# Prototype Use

```
$ lfs getstripe FILE
entry_id: 1
  extent_begin: 0
  extent_end: 2097152
  stripe_count: 1
  stripe_size: 1048576
  ...
entry_id: 2
  extent_begin: 2097152
  extent_end: 268435456
  ...
```

# Future Work

- Automagic addition of new components as needed.
- Layout hints (parent directory xattrs, …)
- Template layouts (uninstantiated components with specified layout characteristics).


# Questions?

# Legal Information

- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- 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.
- The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
- Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.
- 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 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
Software