

# PLFS and HDFS: Enabling Parallel Filesystem Semantics In The Cloud

Milo Polte<sup>1</sup>, Esteban Molina-Estolan<sup>2</sup>, John Bent<sup>3</sup>, Garth Gibson<sup>1</sup>, Carlos Maltzahn<sup>2</sup>, Maya B. Gokhale<sup>4</sup>, Scott Brandt<sup>2</sup>

1: Carnegie Mellon University

3: Los Alamos National Laboratory

2: University of California, Santa Cruz

4: Lawrence Livermore National Laboratory

## Hadoop Distributed File System

- Increasingly wide deployment due to prevalence of Hadoop
  - Facebook, Yahoo, Hulu ....
- Good support for resilience
- But lacks some common file system features
  - No support for concurrent writers
  - No support to re-open files for rewrite

## Parallel Log-Structured File System

- Decouples concurrent file access
  - Each writer gets exclusive log file
  - Each node gets exclusive index file
- Designed for checkpointing
- Used for HPC applications
- But we can use this functionality to enrich the semantics of HDFS

## Design

- Augment PLFS to speak to HDFS
- PLFS writes to HDFS files as log files
- Two minor variations from normal PLFS behavior:
  - New log file on each open
  - New index file on every session
- HDFS sees a set of individual writers accessing exclusive files
- Going through PLFS will allow HPC applications to use HDFS as a store

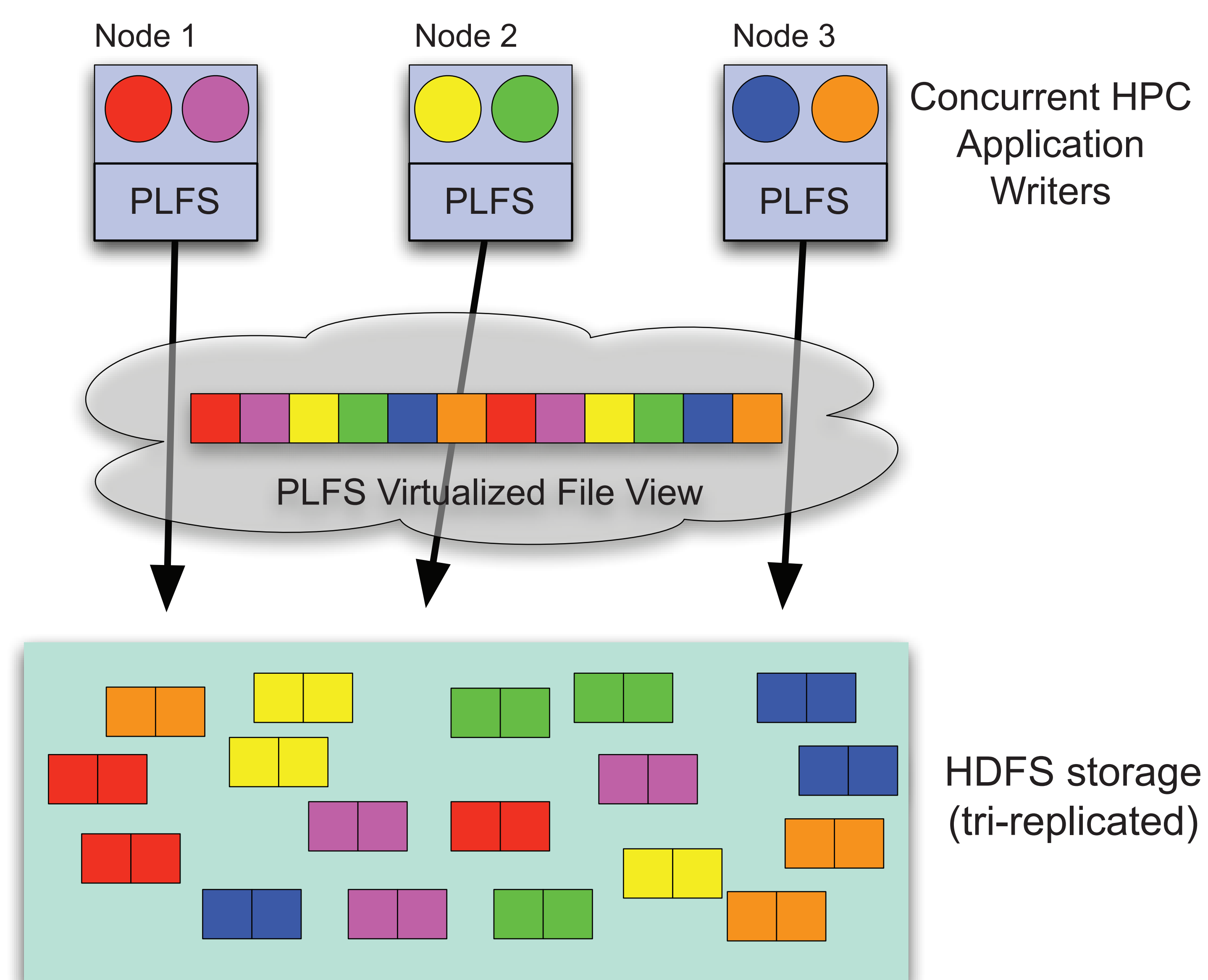
## Challenges

- Teaching PLFS to speak HDFS
  - PLFS is designed for POSIX. HDFS is a new API, semantics
- Output from an HPC application should be available to MapReduce
- PLFS could expose a file map to Hadoop applications
  - But with strided writes, sequential data will be small
  - May create too many map jobs or map jobs that mostly read remotely
- Possible solution: A MapReduce PLFS flattener

## Collaboration and Test Sites

- Collaboration between CMU, UCSU, LANL, LLNL
- Many possible test systems
  - OpenCloud (CMU)
  - 140 node data cluster, 1000 node compute cluster (LLNL)
  - Test clusters (LANL)

### HPC APP ON PLFS-HDFS



### PLFS CONTAINER FOR PLFS-HDFS

