# Parity Logging with Reserved Space: Towards Efficient Updates and Recovery in Erasure-coded Clustered Storage

**Jeremy C. W. Chan, Qian Ding, Patrick P. C. Lee, and Helen H. W. Chan**
**The Chinese University of Hong Kong**

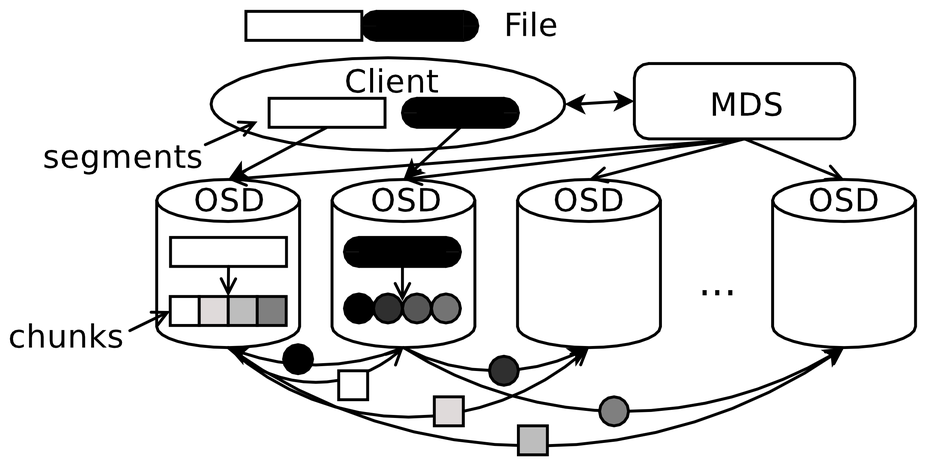
**Source code available at http://ansrlab.cse.cuhk.edu.hk/software/codfs**

## Goals

- Design an erasure-coded clustered storage system for update-dominant workloads
- Mitigate disk seeks for efficient updates and recovery

## Design

- Build **CodFS**, which performs erasure coding on the write path and offloads encoding computations to the storage cluster
- Compute and stripe parity deltas for data updates among storage nodes (OSDs)
- Propose an efficient parity update scheme: **Parity logging with reserved space**




## Parity Logging with Reserved Space (PLR)

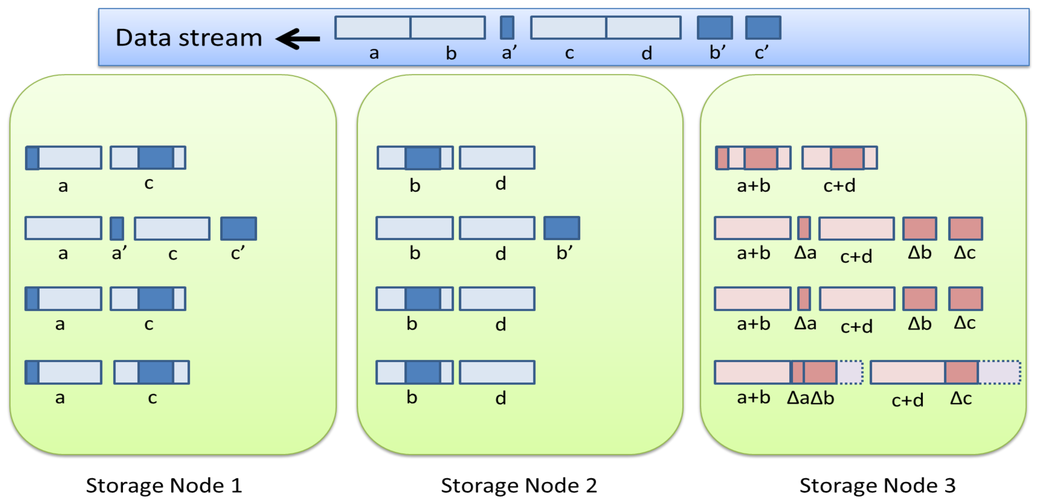
### Idea

- Combine in-place data updates and log-based parity updates
- Reserve storage space next to each parity chunk for keeping parity deltas

| | FO | FL | PL | PLR |
|---|---|---|---|---|
| Data | O | L | O | O |
| Parity | O | L | L | L |

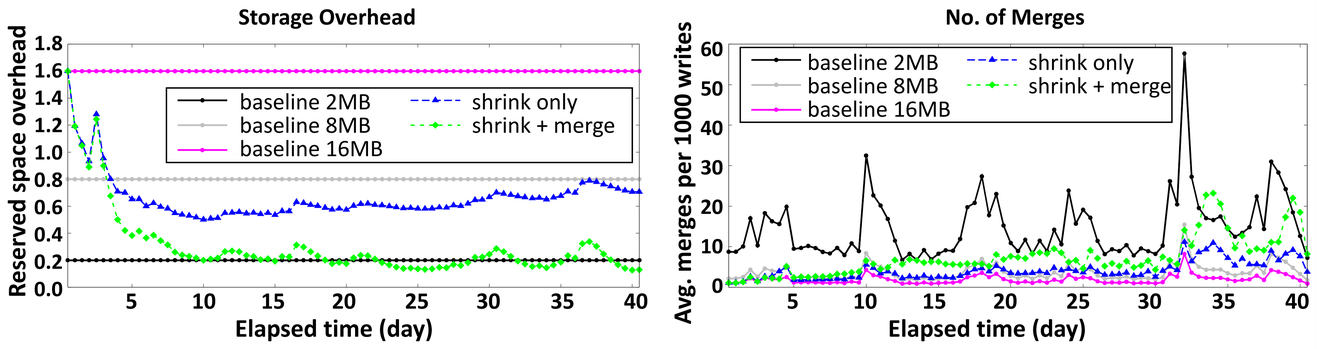
O: Overwrite L: Logging

**FO** Full overwrite
**FL** Full logging
**PL** Parity logging
**PLR** Parity logging with Reserved Space




### Determining reserved space size

- Baseline approach – fixed reserved space for each parity chunks
- Workload-aware approach:
  - predicts the reserved space for each parity chunk using previous workload pattern
  - shrinks the reserved space and releases unused space back to the system
  - merges parity deltas in the reserved space




Storage overhead and no. of merges of different reserved space management strategies under Harvard NFS traces

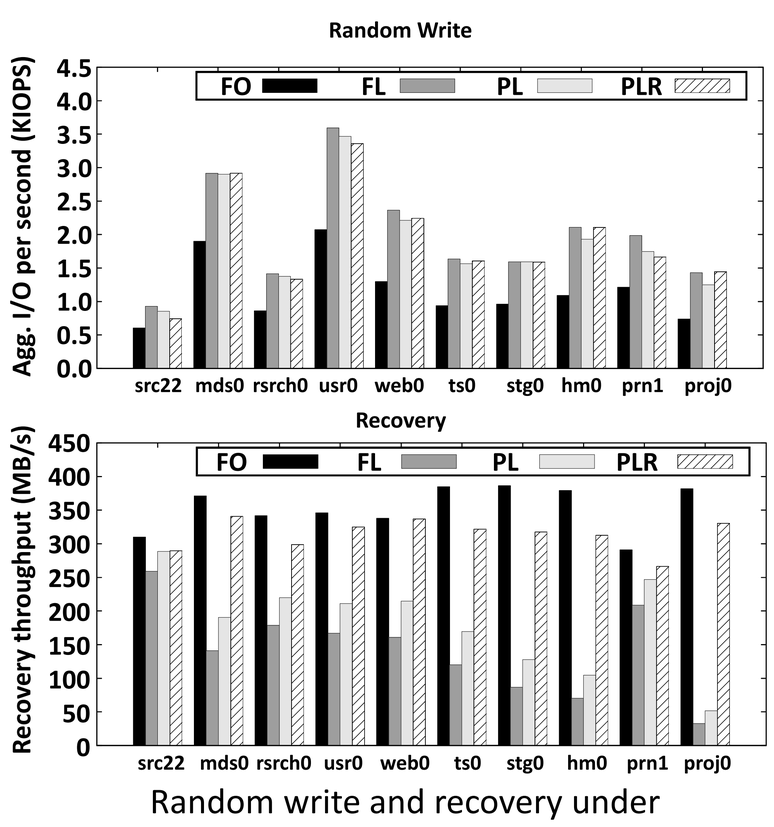
## Testbed Experiments

### Experiments on Real-world Traces

- Random write
  - Parity logging schemes (FL, PL, PLR) are much faster than FO
  - PLR is 63.1% faster than FO
- Recovery
  - PLR saves disk seeks to parity deltas and outperforms both FL and PL in recovery
  - PLR is up to 10x faster than FL

### Experiments on Synthetic Traces

- Show CodFS achieves theoretical throughput
- Show trade-off between reserved space and storage efficiency
- Details in the paper




Random write and recovery under MSR Cambridge traces with RDP (6,4)