OAK RIDGE National Laboratory

File Level Erasure Coding in Lustre

Adam Disney

disneyaw@ornl.gov

LUG2021

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Outline

- Current Progress
- Introduction to Erasure Coding
- Erasure Coding in Lustre's Progressive File Layout (PFL)
- Using File Level Erasure Coding in Lustre
- Limitations and Possible Pitfalls in Initial Version

Current Progress

- Original patches are ~1-2 years old and written by Bobi Jam
  - LU-10911, LU-12186, LU-12187, LU-12188, LU-12189, LU-12668, LU-12669
- I have
  - Inspected these patches
  - Ported to master (a few months ago)
    - f55fdfff5dede69e6674999fb02c1add513704f0
  - Fixed a few bugs
    - EC code errors, logic errors, NULL deref, etc.
  - Tested regression (sanity.sh)
- Currently developing tests and fixing bugs as they're discovered
- Targeting 2.15 release (~80% complete)

File Level Replication in Lustre

- Problem
  - I'm putting my file into Lustre and I don't want it to be lost.
- Solution
  - I'll put multiple copies of my file in Lustre.
- Downside
  - Multiple copies are expensive.
    - 100% overhead per copy

File Level Erasure Coding in Lustre

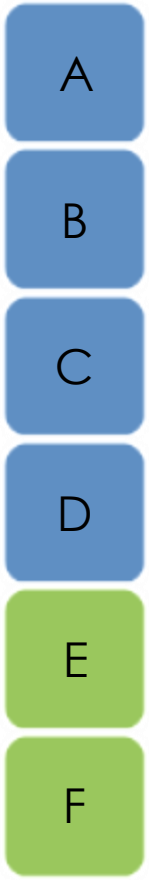
- Problem
  - I'm putting my file into Lustre and I don't want it to be lost.
- Solution
  - I'll store some coding data for my file in Lustre.
- Coding calculations take time but generally requires far less space.

Reed-Solomon Codes

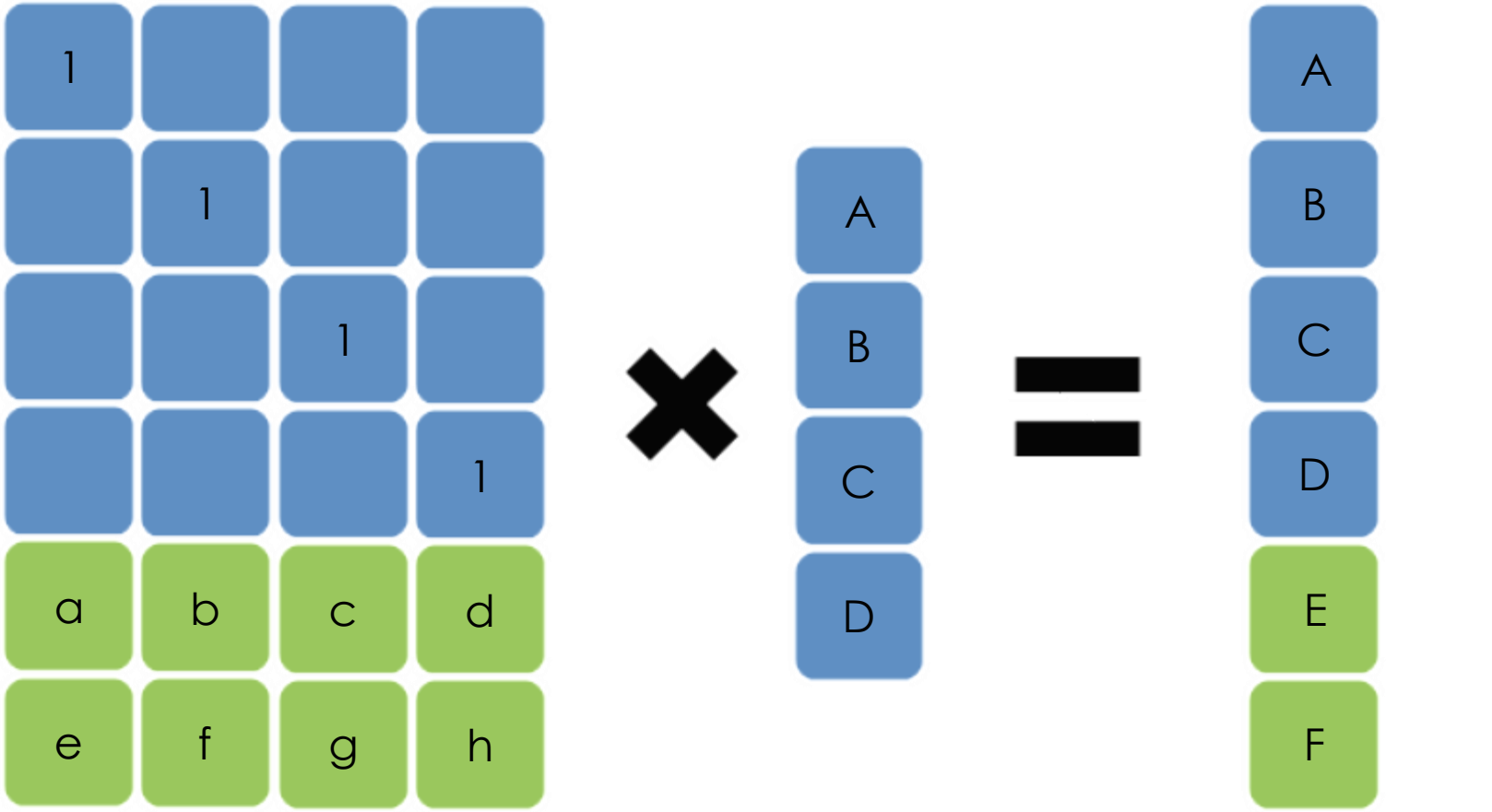
- K = Number of data blocks per stripe
- M = Number of code blocks per stripe
- Data blocks and code blocks are the same size
- Basically, one can pick any K + M
- For any K + M stripe, one can recover all K data blocks from any K blocks

Reed-Solomon Codes (Example 4 + 2)

K = 4
M = 2

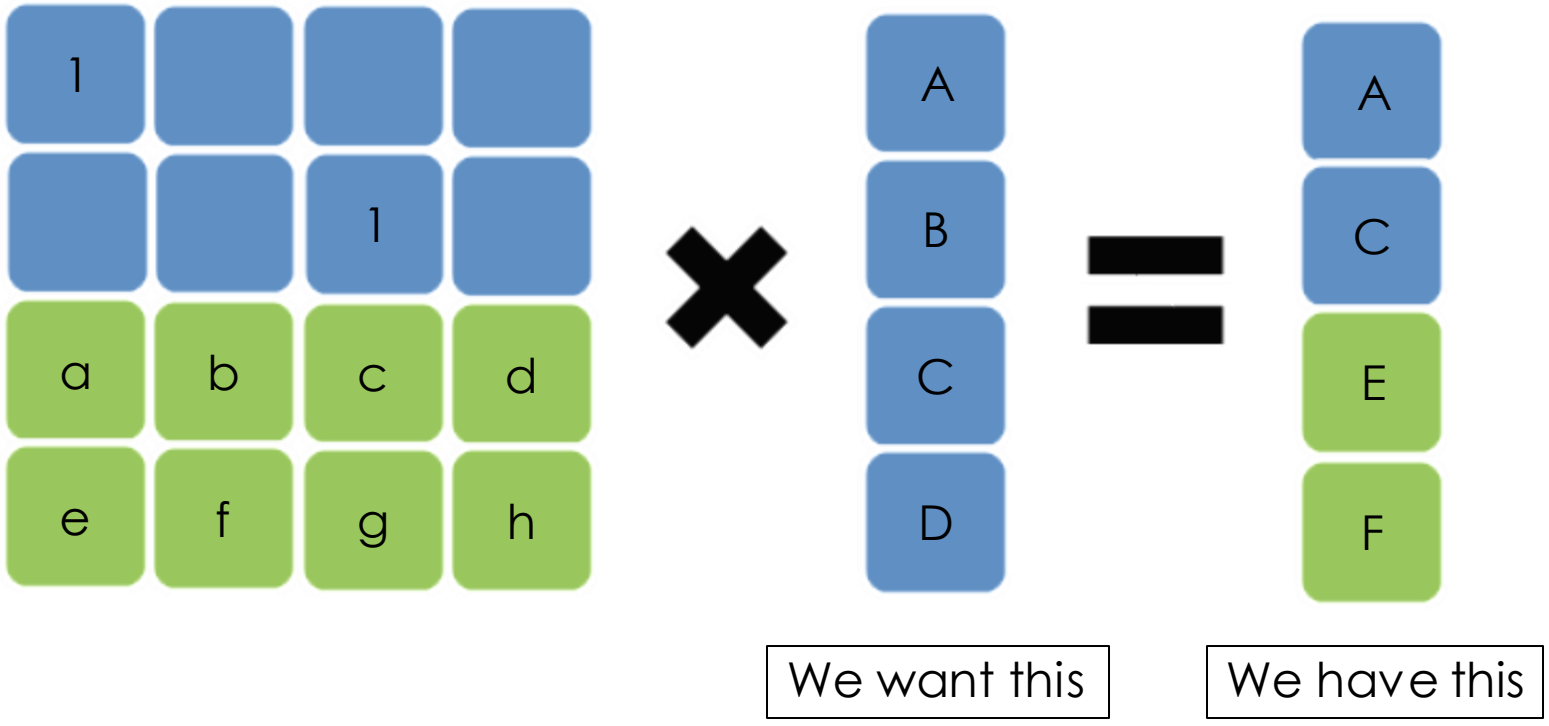


Reed-Solomon Codes (Example 4 + 2)



Reed-Solomon Codes (Example 4 + 2)

Suppose we lost data blocks B and D

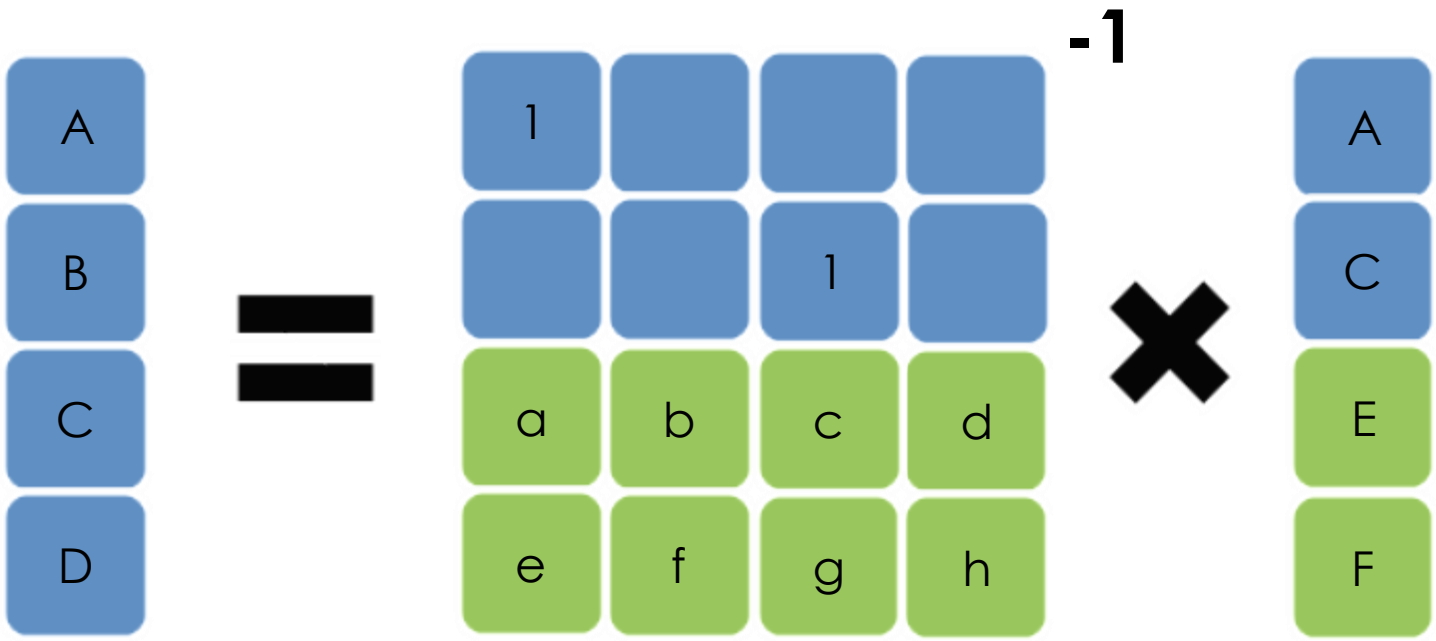


We want this

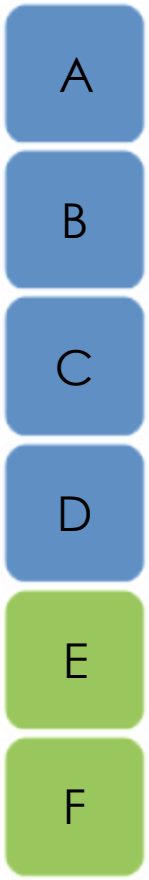
We have this

Reed-Solomon Codes (Example 4 + 2)

Left multiply both sides by the inverse encoding matrix

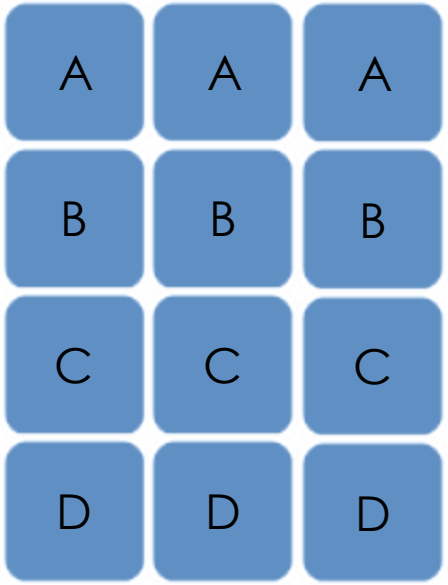


Space Savings



RS 4+2

VS



Triple Replication

Applying Erasure Coding to Lustre PFL


```
lfs setstripe -E 4M -c 1 -E eof -c 4 -E eof -L erasure_code -ec_data_count 4 -ec_parity_count 2
```

Component 0

Component 1

Component 2

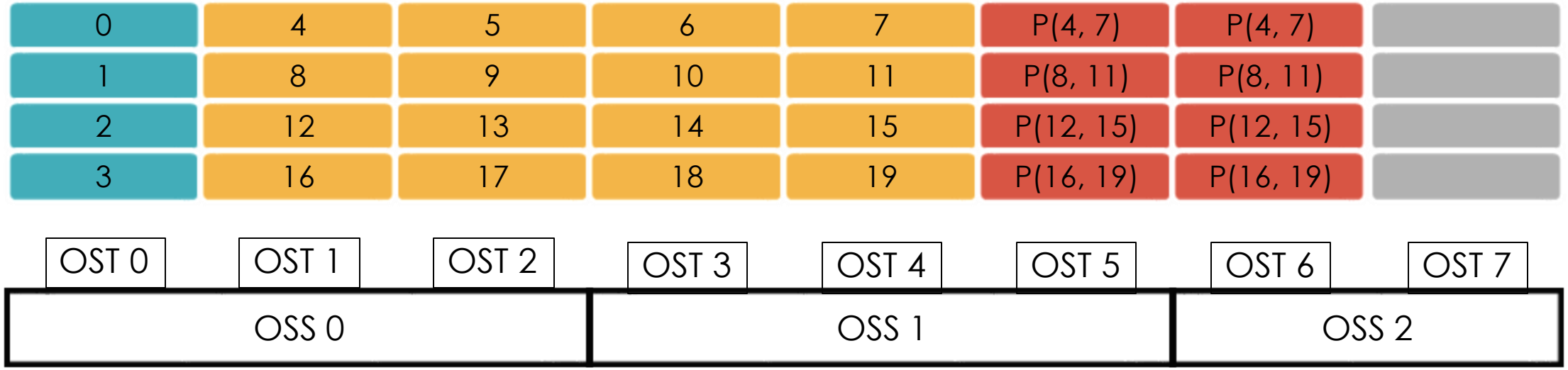
Using File Level Erasure Coding in Lustre

- `lfs setstripe -E eof -c 4 -E eof -L erasure_code -ec_data_count 4 -ec_parity_count 2 file`
- Write to the file
- `lfs mirror resync file`
- When reading the file, if there is a failure that is recoverable, reads succeed as normal
- If file is modified, must resync

Limitations/Concerns in Initial Version

- No automatic data replacement
- No automatic updating of parity
- Slow encoding/decoding
- May need to be careful with data placement

Data Placement




```
lfs setstripe -E 4M -c 1 -E eof -c 4 -E eof -L erasure_code -ec_data_count 4 -ec_parity_count 2
```

Component 0

Component 1

Component 2

Acknowledgments

- Thanks to James Simmons and Rick Mohr on the Lustre lessons!
- This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

OAK RIDGE National Laboratory

File Level Erasure Coding in Lustre

Adam Disney

disneyaw@ornl.gov

LUG2021

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY