



Michal Simon

XRootD Erasure Coding plug-in



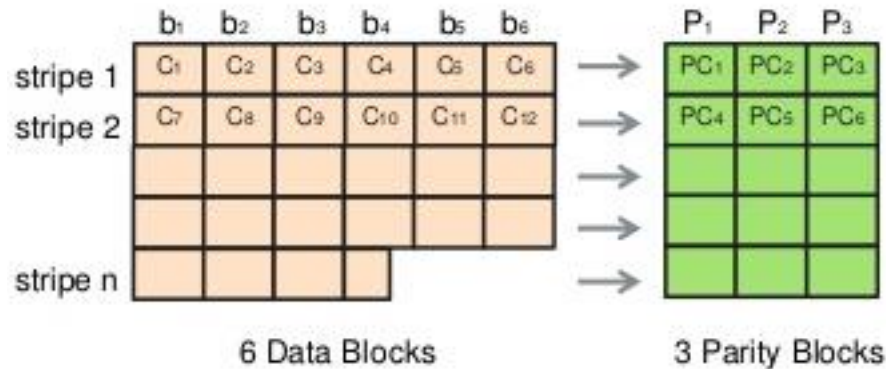
Outline

- Erasure Coding
- Intel ISA-L
- Architecture
- Placement policy
- Status



Erasure Coding

- Transform a block of data of k symbols into a longer block with n symbols such that the original message can be recovered from any k symbols of the new block



Intel: ISA-L

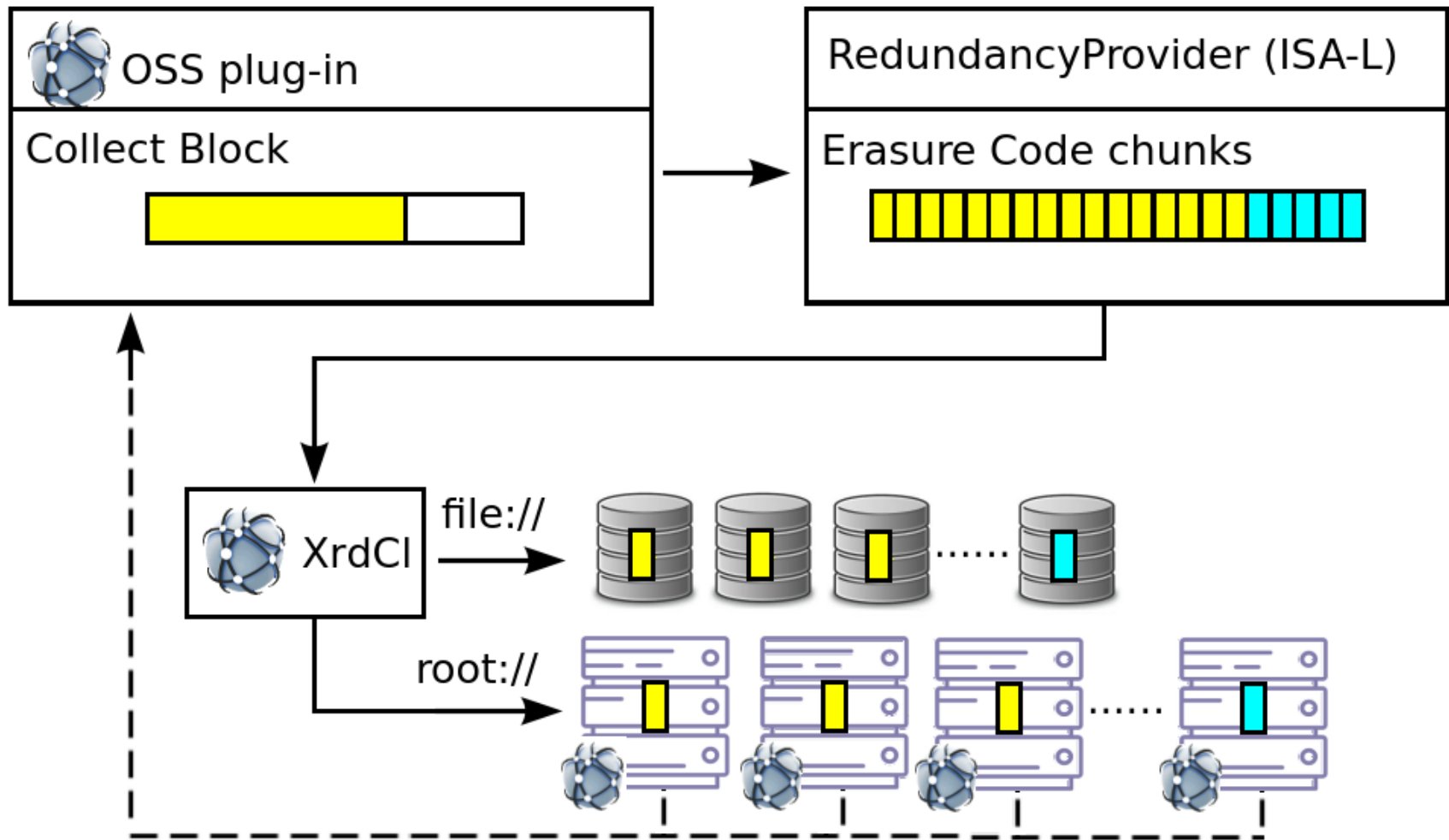
- Intelligent Storage Acceleration Library
 - Written in ASM
 - Bindings for C/C++
 - Distributed under a BSD license
 - Highly optimised Reed-Solomon
 - Automatically chooses an appropriate binary implementation for the detected processor architecture



Architecture

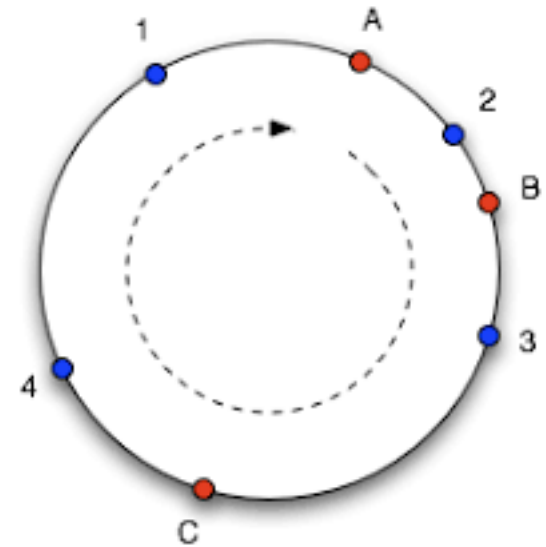
- Goal: easy to manage, minimize need for admin tools
- XRootD OSS plug-in
- Data chunked (e.g. 4MB) and erasure coded (e.g. 18+6)
- Metadata replicated

Data flow (2D EC)



Placement policy

- Consistent hashing for both data and metadata
 - Hash Ring
 - Jump (google)
- Consistent hashing for metadata (higher replication factor) and probabilistic distribution for data
 - metadata contains placement info for data



Status

- Done:
 - RdWrtStream (agresive async IO)
 - RedundancyProvider (Paul Lensing)
 - Placement policy (still room for improvement)
- To Do:
 - Metadata handling (needs optimization)
 - Repairs (not implemented)
- URL:
 - <https://github.com/simonmichal/xrootd/tree/XrdEc>

Questions?

