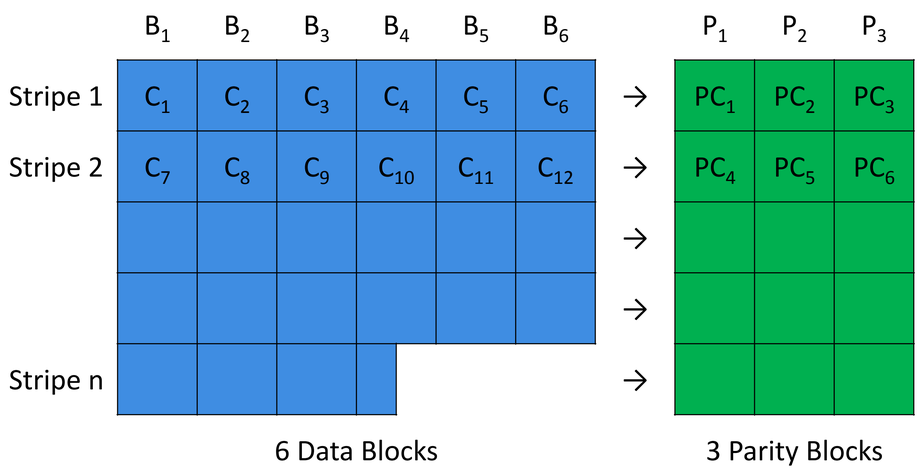
XRootD Erasure Coding Plugin

Michał Simon (CERN IT-ST)
Andreas Peters (CERN IT-ST)

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**
- Widely used e.g. in CDs, DVDs, DSL, RAID6 and satellite communication
- Reed-Solomon codes are optimal erasure codes (maximum distance separable codes)
  - can correct half as many errors as there are redundant symbols added to the block
  - can correct as many erasures (errors whose locations are known) as there are parity blocks

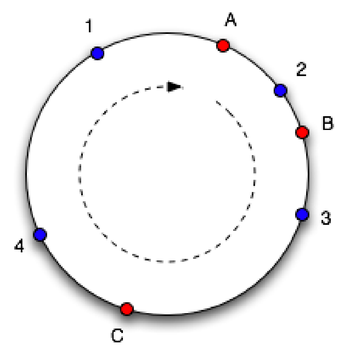


- Intelligent Storage Acceleration Library
  - Written in ASM with **bindings for C/C++**
  - Distributed under a BSD license
  - **Highly optimised Reed-Solomon**
  - Automatically chooses an appropriate binary implementation for the detected processor architecture

Data placement

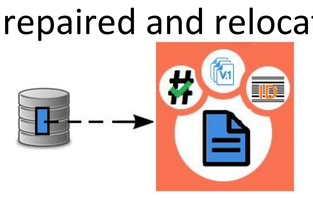
Metadata

- **Stored separately** in order to speed up metadata operations (e.g. stat, ls)
- Replicated for High Availability
- Placement policy – **consistent hashing**, 2 algs considered:
  - Hash Ring
  - Jump (Google)
- Contains **information about data placement**
- Never rehashed to new location



Data

- Erasure coded for High Availability
- **Uniformly distributed between disks or data servers**
- On media failure data blocks are repaired and relocated to new disk
- Blocks contain data + checksum, block index and version



Architecture

Goals:

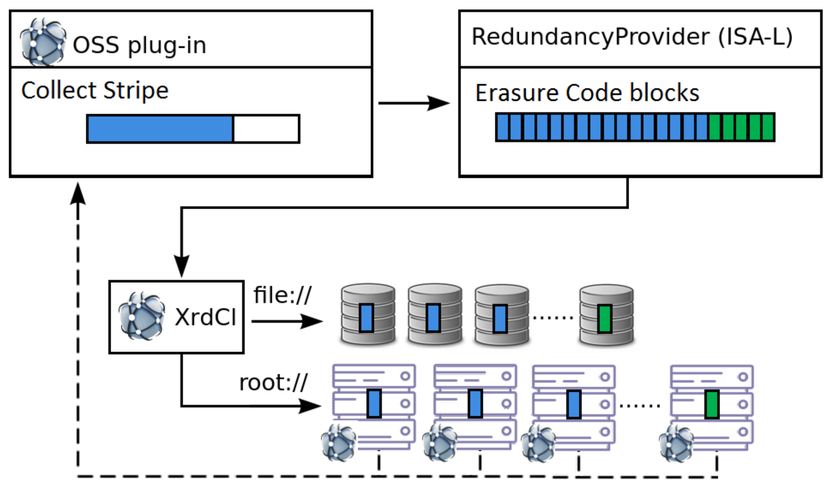
- High Availability
- Aggregate disks and data servers into a single entity
- Fast stream writing and reading (random access can be penalized by repairs)
- 2D Erasure Coding

Use cases

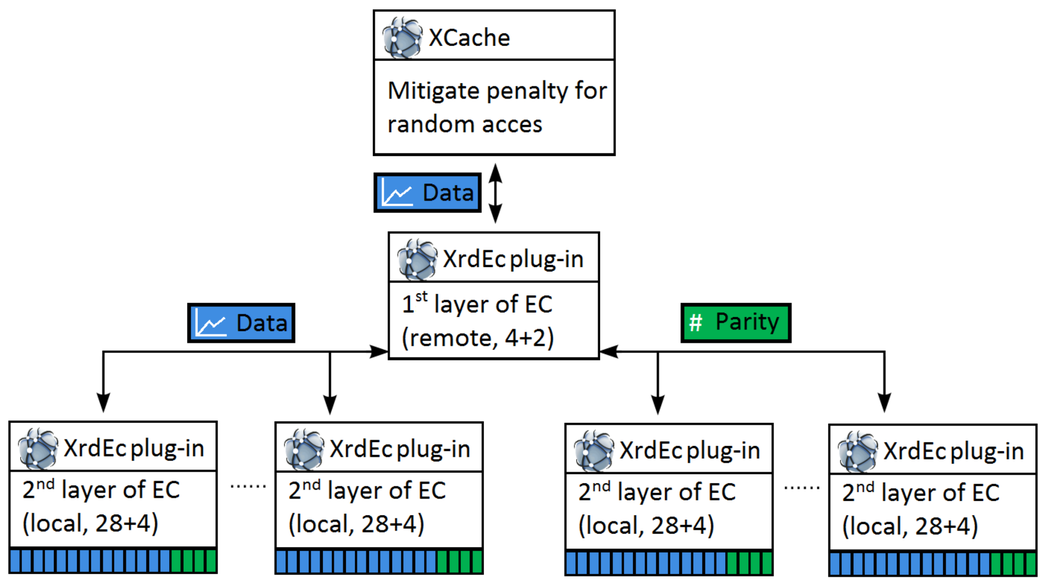
- Large files
- Write-once / read-many

Data flow

- Data are collected until a full stripe can be erasure coded
- Erasure coded data are chunked and written into disks / remote data servers (could be next layer of EC)

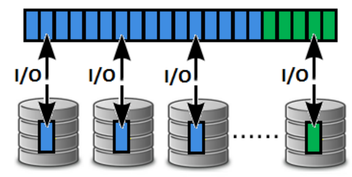


Deployment Model



Status

- An alpha version has been implemented
- Preliminary tests showed promising results
  - **Small overhead** of the XrdEc abstraction layer and erasure coding itself.
  - **Good performance of concurrent I/O**



Work To Do

- Implement missing operations (mkdir, rm, etc.)
- Embrace the new upcoming features of XRootD client
  - **Extended attributes** (xattr)
  - **Bundled requests**