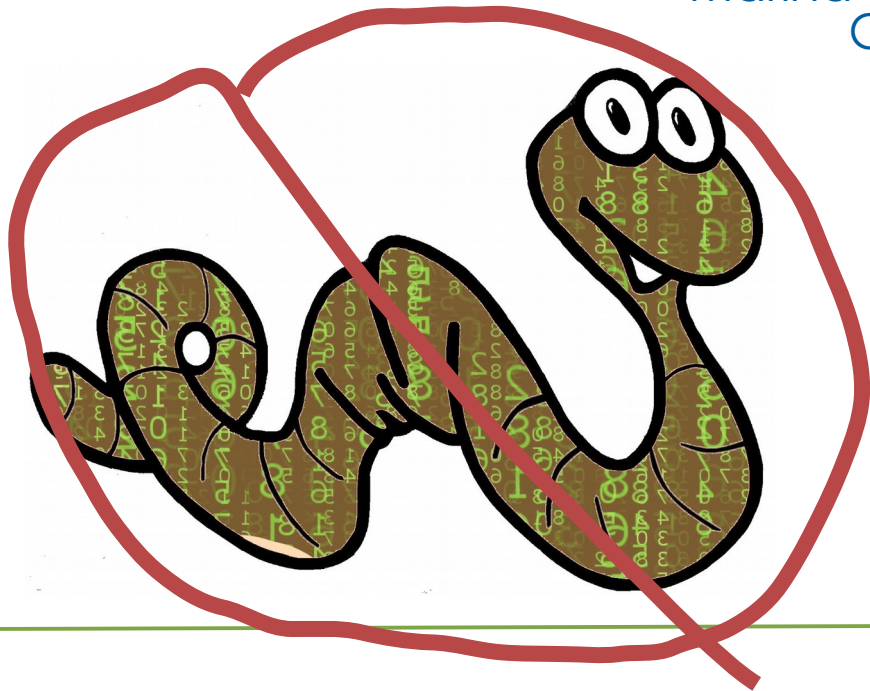


noWORM

Marina Sahakyan for dCache
CHEP 2018, Sofia

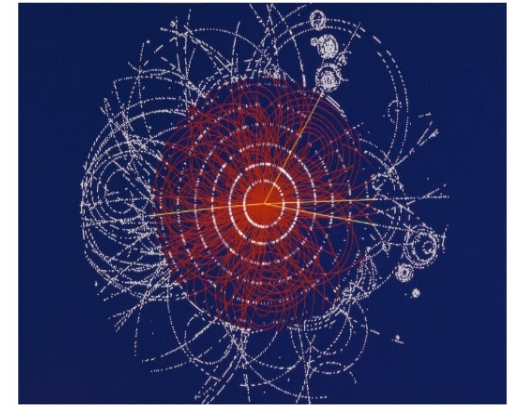


Scientific data challenges

- **Immutability**

Scientific data challenges

- Volume
- Fast ingest
- Chaotic Access
- Sharing
- Access Control
- Persistence
- **Immutability**

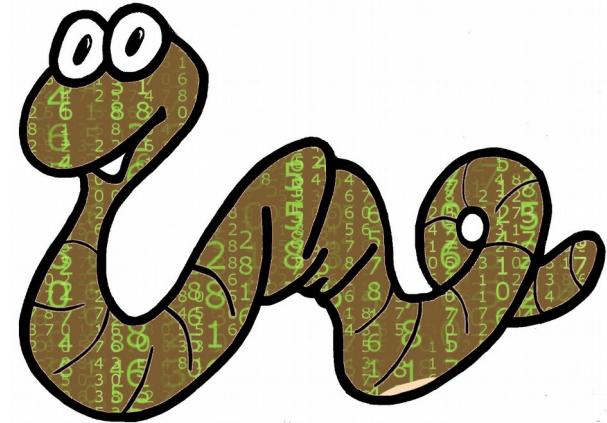


Write Once, Read Many

- Legal organizations
 - SEC Rule 17a-4
- Video hosting providers
 - Youtube
- Streaming Audio
 - Spotify
 - iTunes
- Photo hosting services
 - Flickr
 - Google Photo

Storage Providers

- Provided by High-End storages
 - NetApp
 - EMC
 - IBM
- dCache operation mode



dCache as WORM storage

- Immutable data
 - File ID always points to the same data
 - File ID used by dCache to address data
 - Multiple copies (replicas) are possible

So, why WORM is not enough?

- Too many random-IO NFS workloads
 - xCloud
 - Samba
 - Scratch space
- Show stopper in some workloads
 - HDF5 files (random update)

HDF5 container usecase



Add new content to the container

HDF5 container usecase



Freeze the container

HDF5 container usecase



Ready for Shipping Archival



But wait.....

- File ID points to different data!
 - What about HSM?
 - Replicas?
 - Checksums?!

Worse – POSIX compliance!

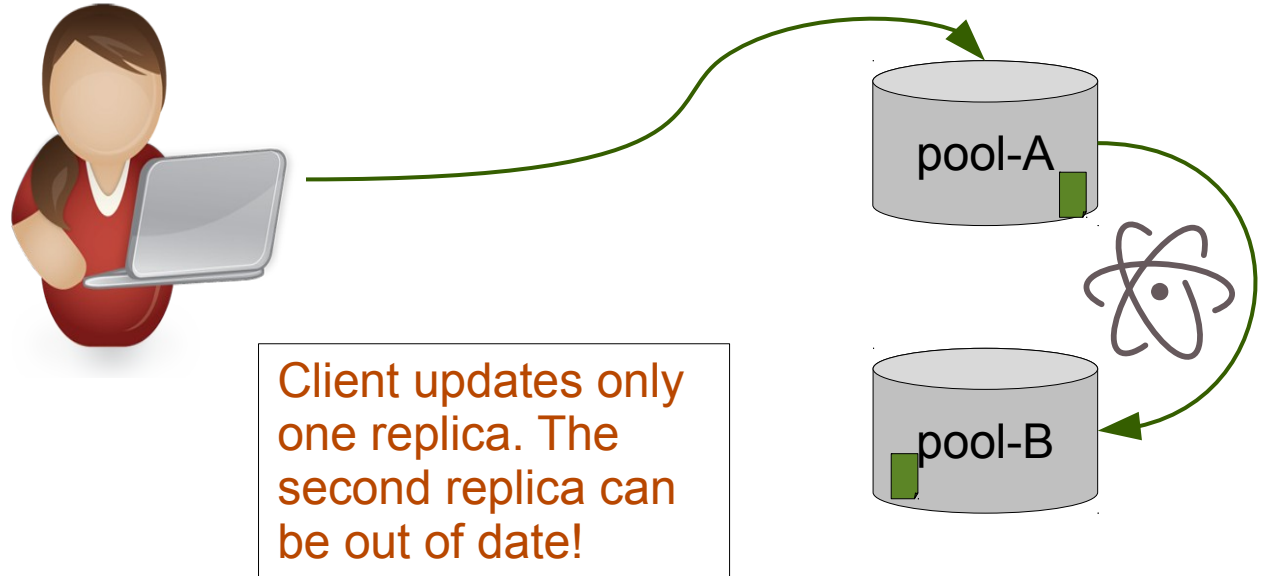
- How much compliance is enough?
- Multi-writer support?
- Multi-protocol support?

WORM as QoS

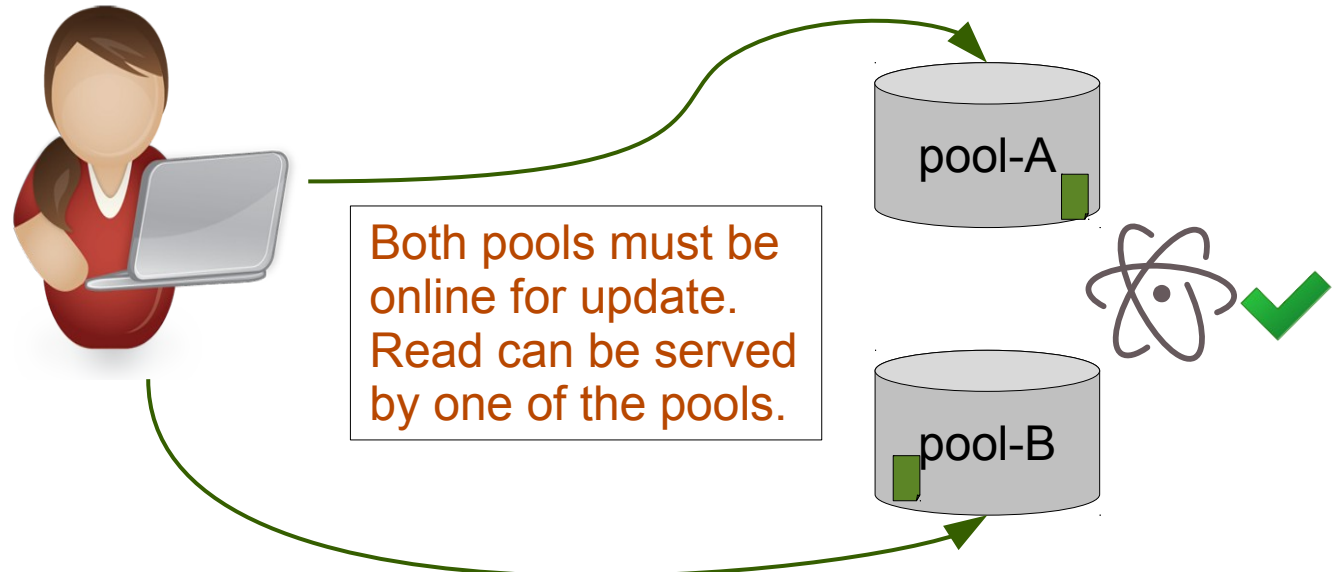
- DISK-ONLY + noWORM
 - No HSM
- Default policy to create WORM files
- Can be changed with QoS transition
 - Directory based
 - File based

Cool, But what about replicas?!

Standard (async) Replication



Client-side Replication



pNFS layout:

- Describes how file's data spread over the data servers
- Which protocol data servers supports (layout types)
 - Block-
 - File-
 - Object-
 - FlexFile-

Client side replication (mirroring)

- Linux client supports multiple layout types
 - RHEL 7.5 (kernel-3.10.0-862.el7) officially supports flexfile
- dCache is the first open-source server
 - Starting from 3.0 (2016)

Current status & WIP

- Working prototype
 - incremental merge into upstream
- Change WORM/noWORM as QoS
- Multi-writer support
 - POSIX locking support
- Pass over 50% of xfstests
- Mirroring

Implementation details

- No HSM, p2p on-load for noWORM files
 - On first upload locations are 'locked'
- non-NFS client may upload initial file
 - Following updates with non-NFS client - rejected
- Client reports IO errors to the door

Summary

- Some workflows require mutable files
- noWORM is a QoS
 - Can be changes for each file
- It's more complicated than it sounds
 - We should not compromise existing stability
- POSIX compliance is a challenge
- But we are working on it!

Thank You!