EOS – Hot Storage for Cool Heads

Presentation by Florian Akos Szabo

Prepared for Clusters, Grids and Clouds seminar during 4-8 of June, 2018

Introduction

- The EOS project was started in April 2010 in the **CERN IT data storage group**.
- Distributed Storage System (=DSS)
- A software solution that aims to provide fast and reliable multi-PB disk-only storage technology for both LHC and non-LHC use-cases at CERN
- Organic storage system – more on it later
- EOS is available under GPL v3 license, so it is FOSS.

EOS - Goals

- The physics analysis **use-cases at CERN** needed performant, reliable and scalable storage
- There was a need to disentangle disk and tap storage (Hot and Cold).
- The system became operational in 2011 for the ATLAS experiment with 2 PB in size and has been growing in four years to 140 PB storage provided by 44k hard disks (see figure 2). The total storage space today is segmented into six independent failure domains (instances) for the four LHC experiments Alice, Atlas, CMS and Lhcb, a shared experiment instance for smaller experiments PUBLIC and a generic user instance USER for all CERN users

Organic Filesystem

- Similar to how living organisms are made of cells that constantly and automatically replenish themselves, even in dynamically changing environments, organic storage allows for continuous **growth in the face of changing requirements**.
- Features:
  - Resilience
  - Self-healing
  - Adaptive accommodation
  - Constant renewal
- It allows servers to be added while applications remain fully functional and provides automatic failover strategies when servers become unavailable, without any manual intervention.

DSS & SDS

- **Storing data on multitude of standard servers, which behave as one storage system although data is distributed between these servers.**
- an advanced form of the “Software-Defined Storage” concept
- typically includes a form of storage virtualization to separate the storage hardware from the software that manages it
- Server hypervisor vs storage hypervisor

Why is DSS becoming so important?

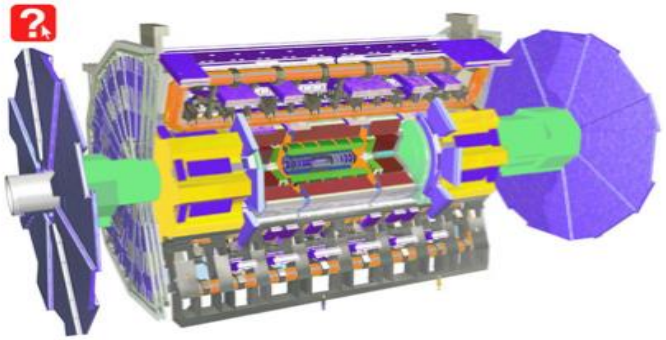
- The current approach to storage does not work anymore: it is not flexible enough, fast enough or the cost is prohibitively high
- By design a distributed storage system solves all of these issues at once.

CERN mainstream use case

tape archive CASTOR

CERN Advanced STORage manager

LHC Detector



O(GB/s)

5-10 GB/s

EOS Open Source Storage

5-10 GB/s

peak 100 GB/s

local batch cluster

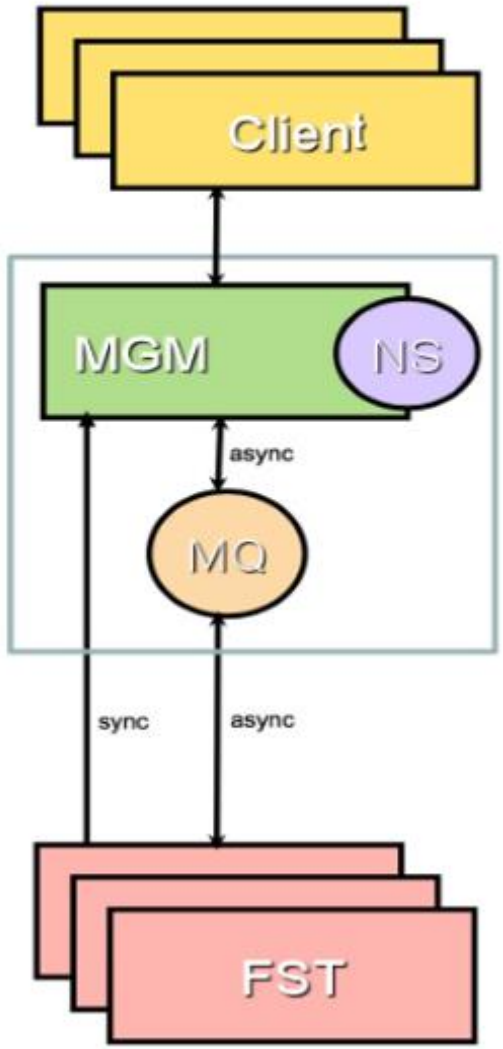
O(10^5 cores)

openstack™

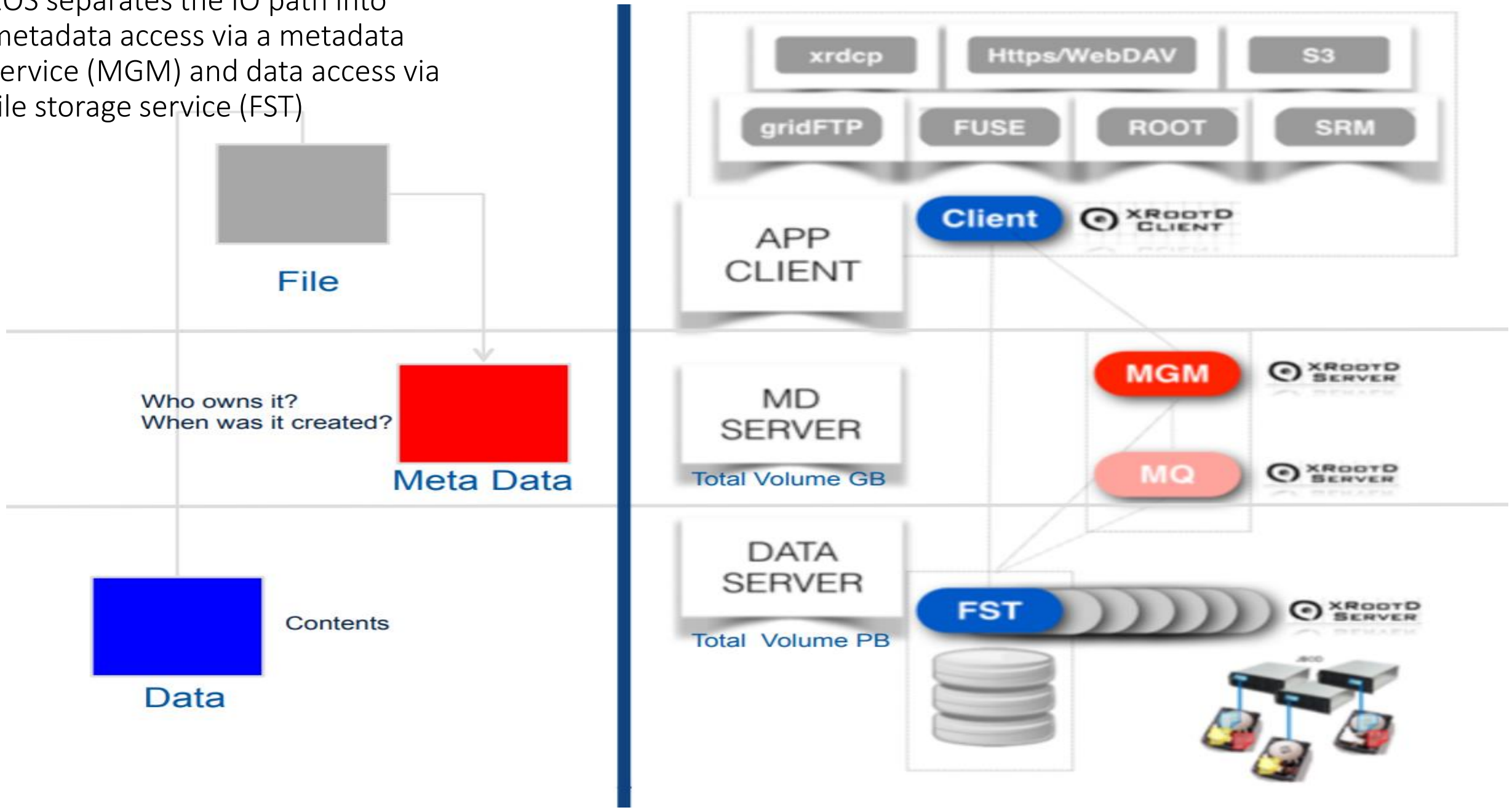
Data Export to Worldwide Computing Grid

EOS - Simple Architecture

- **MGM** - metadata server
- **FST** - storage server
- **MQ** - message broker for asynchronous messaging



EOS separates the IO path into metadata access via a metadata service (MGM) and data access via file storage service (FST)



EOS - Layouts

Files are stored with a so called **layout**.
The following layouts are supported

- **plain** - a file is stored as a plain file in one filesystem
- **replica** - a file is stored with a variable number of replicas in n filesystems
- **rain** - reed-solomon encoded files with data and parity blocks (redundant array of independent nodes)

EOS - Details

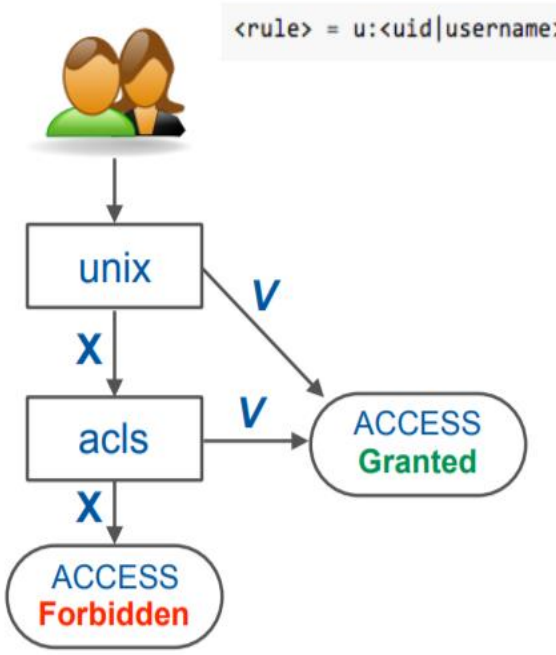
- Strong Authentication via Kerberos & X509 certificates for remote and mounted file access
- Directory-based (non-standard) additive ACLs
- **File-level** checksumming with software hash functions Adler32, SHA1, MD5, CRC32 **block-level** checksumming (algorithm as before) for 4k, 64k, 128k, 256k, 512k & 1M blocksize
- EOS shell as user & administrator interface

EOS - More details

- At the core is the XRootD protocol/framework
- in-memory namespace implementation for low-latency access for meta-data
- master-slave high-availability model
- Quota system for user/group/project
- optinal in-memory namespace cache and KeyValue(KV) store for persistency to improve scalability
- makes use of the XrootD framework which is a fully generic suite for fast, low latency and scalable data access (also acts as a protocol)
- The server runs on Linux, the client runs on Linux and OSX, plus Windows users can also access via SAMBA or WebDav protocols

EOS - Permissions

- Supports Kerberos for local and X.509 for grid access
- File permissions are verified in a 2 step process:
  - UNIX bits (traditional)
  - Access Control Lists (new)




```
<rule> = u:<uid|username>|g:<gid|groupname>|egroup:<name>:{rwxomqc(!d)(+d)(!u)(+u)}
```

tag	definition
r	grant read permission
w	grant write permissions
x	grant browsing permission
m	grant change mode permission
!m	forbid change mode operations
!d	forbid deletion of files and directories
+d	overwrite a '!d' rule and allow deletion of files and directories
!u	forbid update of files
+u	overwrite a '!u' rule and allow updates for files
q	grant 'set quota' permissions on a quota node
c	grant 'change owner' permission on directory children

EOS
-
Recycle Bin

- To prevent accidental data loss
- Delayed deletiotion of files
- Possible to set its **size** and **lifetime** parameters:

```
EOS Console [root://localhost] |/eos/> recycle config --lifetime 86400
```

```
EOS Console [root://localhost] |/eos/> recycle config --size 100G
```

Listing recycle entries

```
EOS Console [root://localhost] |/eos/dev/2rep/subnode/> recycle ls
# Deletion Time           UID      GID      TYPE          RESTORE-KEY      RESTORE-PATH
# ------------------------------------------------------------------------------------------------
Thu Mar 21 23:02:22 2013   apeters  z2       recursive-dir 00000000000007cf /eos/dev/2rep/subnode/tree
```

```
EOS Console [root://localhost] |/eos/> recycle

# ____________________________________________________________________________________
# used 0.00 B out of 100.00 GB (0.00% volume / 0.00% inodes used) Object-Lifetime 86400 [s]
# ____________________________________________________________________________________
```

Example: CernBox

- a Dropbox-like service in production since 2015
- based on OwnCloud software with EOS as storage backend
- Integration meant EOS was enhanced with sync and share functionality
- OwnCloud is using the WebDAV protocol with a few custom extensions
- supports the basic subset of the WebDAV protocol and offers multi-platform file access such as: Android, iOS, Linux, OSX or Windows.

Pros and Cons

- Cheap
- Open Source / FOSS
- Reliable and Fast
- Very Low latency
- Low risk of data loss (logging)
- High consistency
- Code ownership with Cern > can change the way they need

- Slow startup time (takes too long to load the in-memory namespace)
- Very memory intensive due to in-memory namespace
- Code ownership with CERN > No commercial support + responsibilities

Glossary

- https://storpool.com/blog/what-is-distributed-storage-system
- https://cernbox.cern.ch/index.php/s/Kl0hxpeA5bFQ4Ho?path=%2Fpresentations
- https://cernbox.cern.ch/cernbox/desktop/index.php/s/Nbpi3hGZYYqHN93
- https://github.com/cern-eos/eos