EOS as the present and future solution for data storage at CERN

EOS as the present and future solution for data storage at CERN

AJ Peters, EA Sindrilaru, G Adde

CERN IT, Geneva, Switzerland

E-mail: andreas.joachim.peters@cern.ch, elvin.sindrilaru@cern.ch, geoffray.adde@cern.ch

Abstract. EOS is an open source distributed disk storage system in production since 2011 at CERN. Development focus has been on low-latency analysis use cases for LHC[1] and non-LHC experiments and life-cycle management using JBOD[2] hardware for multi PB storage installations. The EOS design implies a split of hot and cold storage and introduced a change of the traditional HSM[3] functionality based workflows at CERN.

The 2015 deployment brings storage at CERN to a new scale and foresees to breach 100 PB of disk storage in a distributed environment using tens of thousands of (heterogeneous) hard drives. EOS has brought to CERN major improvements compared to past storage solutions by allowing quick changes in the quality of service of the storage pools. This allows the data centre to quickly meet the changing performance and reliability requirements of the LHC experiments with minimal data movements and dynamic reconfiguration. For example, the software stack has met the specific needs of the dual computing centre set-up required by CERN and allowed the fast design of new workflows accommodating the separation of long-term tape archive and disk storage required for the LHC Run II.

This paper will give a high-level state of the art overview of EOS with respect to Run II, introduce new tools and use cases and set the roadmap for the next storage solutions to come.

1. Introduction

EOS[1] is a multi-protocol disk-only storage system developed at CERN since 2010 to store physics analysis data physics experiments (including the LHC experiments). 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 44.000 hard disks (see figure 1). 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. Main difference is the average file size with *USER* storing the smallest files.

[1] "**L**arge **H**adron **C**ollider"

[2] "**J**ust a **B**unch **O**f **D**isks"

[3] "**H**ierarchical **S**torage **M**anager"

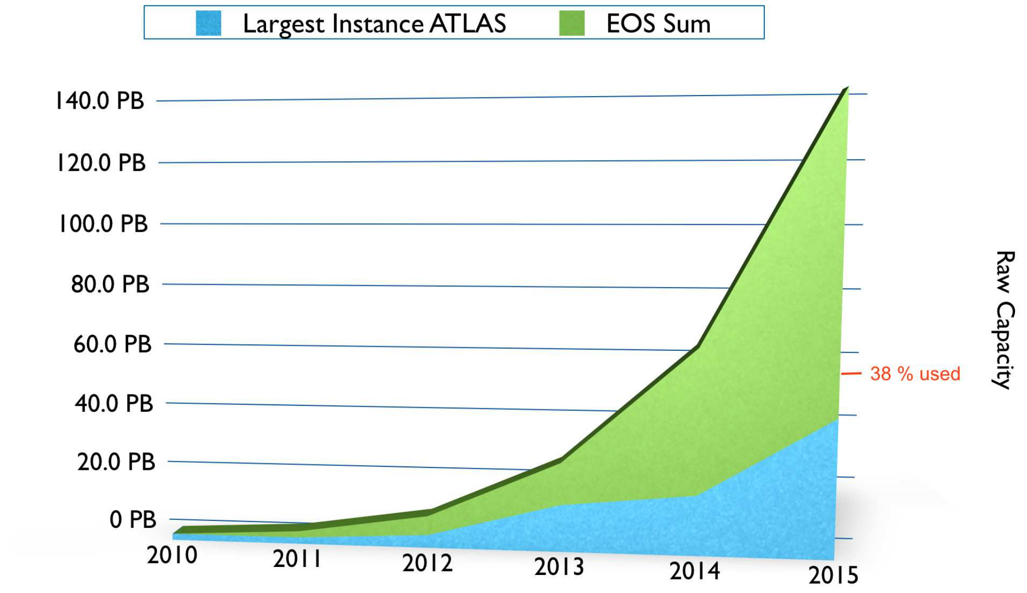

Figure 1. EOS Deployment

2. Architecture

EOS separates the IO path into metadata access via a metadata service *MGM* and data access via file IO services *FST* (see figure 2). To guarantee minimal file access latencies all metadata

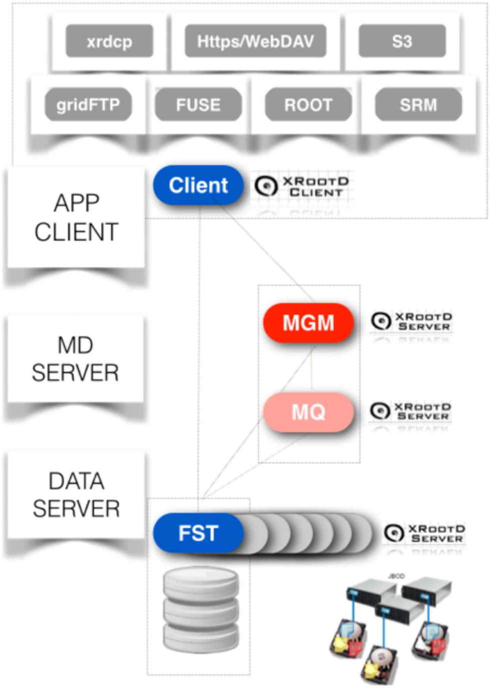

Figure 2. EOS Architecture

Parameter	April 2015
Capacity	140 PB
Server	1.400
Hard Disks	44k
Files	270 M
Directories	26 M
Replicas	0.6 B
theor. Connectivity	13 Tbit
random IOPS	2.2 M
theor. Disk BW	3.3 TB/s
Internal Messaging	150 kHz
State Machine	3 M KV-pairs
Users storing data	3 k
Quota rules	9.600

Table 1. Current numbers for the EOS deployment at CERN.

is kept in-memory on metadata server nodes and persisted using WAL technology[4].

Files are in most cases replicated on two JBOD disks. EOS supports additionally erasure-encoding of files with two or three redundancy stripes based on the *Jerasure*[2] library. The current cluster state and configuration is represented by a shared hash. The state and configuration modifications are exchanged between storage and metadata servers using a message queue service *MQ*. All three services *MGM*, *FST* and *MQ* have been implemented using the XRootD[3] client-server framework.

3. Deployment

EOS services are distributed over two computer centres: the CERN center in Meyrin(CH) and the WIGNER center in Budapest(HU).

Metadata services are deployed as six active-passive pairs with real-time failover capabilities on nodes with 256 GB of memory at the Meyrin center. File storage services are deployed on approx. 1.400 server nodes with attached storage (up to 50 disks per node) distributed in both computer centres.

The EOS service stores currently 270 million files, which account for more than half a billion file replicas. Three thousand CERN users store files in EOS. Access policies are enforced by extended non-POSIX ACLs and more than 9.500 user, group or project quota space rules. The state machines of all six instances are described by three million key-value pairs with a change rate of 150 kHz (see table 1).

4. Releases

EOS releases are named after gemstones. Releases happen frequently (weekly/monthly) and are deployed as needed on production instances. Active development happens on three of the five release branches with individual scope.

4.1. Amber

The amber release has been the first production release until 2012. It is not actively supported or developed anymore.

[4] "**W**rite **A**head **L**og"

ACL 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
!u	forbid update of files
+u	overwrite a '!u' rule and allow updates for files
q	grant 'set quota' permission on a quota node
c	grant 'change owner' permission on directory children
i	make a directory immutable

Table 2. Permission tags in ACLs

4.2. Beryl

Beryl has been used until 2014 and added some new functionality like the active-passive metadata server failover, two site support, a policy engine, erasure encoding and recycle bin functionality.

4.3. Aquamarine

Aquamarine is the current production release. It enhances Beryl with archiving and backup functionality and extended HTTP support.

4.4. Citrine

Citrine is currently a development branch which adds infrastructure aware scheduling and placement for multi-site setups as well as support for the latest XRootD4 framework version.

4.5. Diamond

Diamond is a research branch implementing a namespace and file IO using the Rados[4] object store.

5. Recent Enhancements

5.1. Permission System

Besides common POSIX permissions based on mode bits and user and group identity EOS provides a rich set of ACL permissions. Recently an immutable flag has been added to allow freezing of archived namespace subtrees. ACLs are stored in the system attribute tag *sys.acl*. They are composed by a list of comma separated rules, which are evaluated from left to right. The target of permissions can be *u:uid* to specify a user, *g:gid* to specify a group, *z* to target everyone or *egroup:e-group* to reference a CERN e-group. The available permission tags are shown in table 2.

An examplary EOS ACL looks like:

```
sys.acl=``u:100:rwx!d,g:200:rx,egroup:higgs:rwx!u''
```

Another powerful feature is to enforce sticky ownership using the *sys.owner.auth* extended attribute in a directory. This means that all operations/creations in this directory are executed with the identity of the owner. One can configure the sticky ownership for all users who have write permissions (using *sys.owner.auth=**) or selected users by referencing their authentication principal (e.g. *sys.owner.auth=krb5:nobody,gsi:DN=..*). This feature was very useful to implement directory sharing via CERNBox.

5.2. HTTP protocol extensions
During the last six months CERN has put a file synchronisation and sharing service into production called CERNBox[5]. CERNBox is based on the OwnCloud[7] software with EOS as storage backend. This meant enhancing EOS with sync and share functionality (see [6]). OwnCloud is using the WebDAV protocol with a few custom extensions. EOS HTTP access has been extended to be compatible with the OwnCloud web service. It supports the basic subset of the WebDAV protocol and offers multi-platform file access such as: Android, iOS, Linux, OSX or Windows.

5.3. FUSE access
A frequent incident observed in EOS production instances is the accidental deletion due to use of recursive deletion (rm -rf) on a FUSE mount. The Citrine branch includes a mechanism to identify recursive deletion processes. These are blocked if they are executed too close to the root directory. The hierarchy level from which deletion restrictions apply is fully configurable. The FUSE mount is now able to use a configured kerberos token or proxy certificate to authenticate connections to the metadata server as defined in the environment of a calling process.

5.4. Data Path Proxy
As of today clients outside CERN can read data directly from EOS disk server nodes due to dedicated firewall rules which allow a direct connection. In the future, it is desirable for better security and flow control to channel external IO traffic through a reduced set of IO proxy gateways. This reduces the complexity of the firewall configuration significantly and improves control on the outside connections. The required functionality has been added to the proxy module *PSS* of the XRootD server. The file open function in the metadata server redirects client IO through a set of proxy nodes. The proxy nodes interpret the redirection URL and establish a connection to the relevant file storage node within the private network.

5.5. Infrastructure Aware Scheduling - IAS
EOS storage is deployed in two computer centres separated by 22ms network latency. Batch jobs executing physics analysis jobs are scheduled in either of the two centres and data should be ideally stored in both centres. As a default policy, EOS places one replica of each file in each center. When a client opens a file for reading its location is tagged based on the client's subnet and matched to the closest available file replica. The Beryl branch of EOS was able to provide this functionality only for the special case of two sites. This has now been generalised and allows to define an arbitrary storage tree e.g. with sites, rooms, racks, nodes etc.

When a client creates a new file IAS supports three placement strategies.

- *scattered* - e.g. place replica with a maximum distance between computer centres
- *co-located* e.g. place replica with a minimum distance with respect to the computer center
- *hybrid* e.g. a mixture between scattered and co-located with respect to individual sub-branches

The selection algorithm always places the first replica on resources with minimal distance to the client location in a round-robin fashion. Additional replicas are placed in similiar round-robin fashion according to the placement strategy. Each resource has a precomputed weight influenced by the currently available network bandwidth and storage capacity.

6. Future Work

6.1. Namespace

The in-memory namespace of EOS allows single-threaded internal metadata read access at approx. 160 kHz. Multi-threaded read-only access reaches up to 1 MHz on an average on a Xeon 2.2GHz multi-core CPU. The drawback is the time it takes to boot the namespace in memory. All metadata is available as C++ objects in-memory and the hierarchical relation and speci!dalised views of the metadata are stored in vector, set and map structures (Google Sparse-Hash). Each metadata object (file/directory) requires 0.5-1 kB of memory. During the namesp!dace boot, file and directory metadata change-log files are read sequentially and converted into in-memory objects. The namespace boots in average 100k entries/s e.g. it takes 15 minutes to boot a 100M entry namespace. The current limitation is not the amount of available memory but the time to startup the metadata service.

A new approach is to persist hierarchical and specialised views (quota, filesystem) directly on disk avoiding scanning of gigabyte sized change-log files. There are several useful technologies available:

- Linux filesystems as meta-data store (XFS, EXT4, ZFS, F2FS)
- KV stores (LevelDB, RocksDB, ForestDB)
- Object Storage (RADOS)

The different solutions are now under prototype evaluation and the most promising will be made available as a namespace plug-in replacing the default in-memory change-log file based implementation.

6.2. IO Plugins

The storage market offers new cost-effective and/or operationally effective storage technologies which are currently being integrated into the Citrine development branch (see [9]):

- Kinetic Ethernet Drive Technology [8]
- Rados Object Storage [4]
- External storage systems characterised by access protocol e.g. S3, XRootD, mountable FS storage

These external storage systems require the setup of a new storage service: a pool of gateway machines. Their role is to proxy/bridge from all available external EOS IO protocols to the internally used IO protocols.

7. Summary

EOS provides a very flexible and extendable disk storage platform for a large user community at CERN. The current deployment with 140 PB of storage space and 44.000 hard disks demonstrated an unprecedented scalability for a low-cost storage system in high energy physics. EOS as a service represents the strategic direction as user, group and grid physics data storage. The challenge for the future is to transform the EOS meta-data service into a scale-out solution without losing low-latency file access which is critical for physics analysis. Integration of ethernet drive and object storage technologies will allow to further reduce storage costs and increase reliability and efficiency in the future. A mid-term goal is to merge EOS functionality with widely used open-source storage platforms to extend the software user community and its reusability.

8. Acknowledgements

We thank all members of the CERN IT-DSS group, CERN experiment representatives and our users for participation, help, contributions, operations and fruitful discussions.

References

[1] AJ Peters and L Janyst "Exabyte Scale Storage at CERN" 2011 J. Phys.: Conf. Ser. 331 052015 doi:10.1088/1742-6596/331/5/052015

[2] "Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications", Technical Report CS-08-627, Department of Electrical Engineering and Computer Science University of Tennessee Knoxville, TN 37996, `http://web.eecs.utk.edu/~plank/plank/papers/CS-08-627.html`

[3] "XRootD", `http://xrootd.org`

[4] SA Weil AW Leung SA Brandt C Maltzahn"RADOS: A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters" University of California, Santa Cruz, `http://ceph.com/papers/weil-rados-pdsw07.pdf`

[5] "CERNBox", `http://cernbox.cern.ch`

[6] L Mascetti et al. "CERNBox + EOS: end-user storage for science" CHEP 2015 - theses proceedings

[7] "OwnCloud", `http://owncloud.org`

[8] "Seagate Kinetic - Open Storage Platform" `http://seagate.com`

[9] AJ Peters et al. "Integrating New Storage Technologies into EOS" CHEP 2015 - this proceedings