EPJ Web of Conferences **214**, 04033 (2019) https://doi.org/10.1051/epjconf/201921404033
CHEP 2018

Providing large-scale disk storage at CERN

Hervé Rousseau[1], *Belinda* Chan Kwok Cheong[1], *Cristian* Contescu[1], *Xavier* Espinal Curull[1], *Jan* Iven[1], *Hugo* Gonzalez Labrador[1], *Massimo* Lamanna[1], *Giuseppe* Lo Presti[1], *Luca* Mascetti[1], *Jakub* Moscicki[1], and *Dan* van der Ster[1]

[1]CERN European Laboratory for Particle Physics 1211 Geneva (Switzerland)

Abstract. The CERN IT Storage group operates multiple distributed storage systems and is responsible for the support of the infrastructure to accommodate all CERN storage requirements, from the physics data generated by LHC and non-LHC experiments to the personnel users' files.
EOS is now the key component of the CERN Storage strategy. It allows to operate at high incoming throughput for experiment data-taking while running concurrent complex production work-loads. This high-performance distributed storage provides now more than 250PB of raw disks and it is the key component behind the success of CERNBox, the CERN cloud synchronisation service which allows syncing and sharing files on all major mobile and desktop platforms to provide offline availability to any data stored in the EOS infrastructure.
CERNBox recorded an exponential growth in the last couple of year in terms of files and data stored thanks to its increasing popularity inside CERN users community and thanks to its integration with a multitude of other CERN services (Batch, SWAN, Microsoft Office).
In parallel CASTOR is being simplified and transitioning from an HSM into an archival system, focusing mainly in the long-term data recording of the primary data from the detectors, preparing the road to the next-generation tape archival system, CTA.
The storage services at CERN cover as well the needs of the rest of our community: Ceph as data back-end for the CERN OpenStack infrastructure, NFS services and S3 functionality; AFS for legacy home directory filesystem services and its ongoing phase-out and CVMFS for software distribution.
In this paper we will summarise our experience in supporting all our distributed storage system and the ongoing work in evolving our infrastructure, testing very-dense storage building block (nodes with more than 1PB of raw space) for the challenges waiting ahead.

1 Introduction

The data Storage and services group at CERN provides many services to the physics community. We are going to see some of the operational challenges of running and supporting EOS with an ever increasing traffic and user community. With Castor entering maintenance phase and other storage techniques (mainly object storage) gaining momentum, we will see how the team addresses theses new use cases by offering new services at a minimum cost by leveraging already existing infrastructure and expertise of the group.

2 EOS

EOS[1] is a large-scale organic distributed storage system that supports physics activity at CERN and consists of almost 1300 servers spread over 15 distinct instances for a total raw storage capacity of 250 PB.

In order to provide a consistent experience for users while keeping operational load at a minimum it is mandatory to automate all repetitive tasks where human intervention has no added value, such as Operating System upgrades.

2.1 CentOS 7

The EOS storage fleet was running Scientific Linux 6 for quite some time, but this version is approaching end of life. As such, only security patches are published nowadays. In addition, new features, better hardware support and kernel features are very nice tools to have when troubleshooting a distributed storage system.

It was then decided that all nodes will be updated to CentOS 7 in the beginning of 2018.

Updating the nodes manually would be very error prone and time consuming considering the number of machines. From observation and experimenting we noted that an operator can update around 4 machines per day; clearly this would take way to long, so first we thought about upgrading storage nodes of the instances in parallel, but it turned out that mistakes were made and data was put at risk making this solution not safe although faster.

Table 1. Time needed to upgrade all nodes

Method	Time (days)
Manual (sequential)	325
Manual (parallel, 2)	163
Automatic (sequential)	54
Automatic (parallel, 4)	13

CERN's IT department has extensive knowledge of Rundeck[2] and wrote plugins that allow for tight integration with the Computer Centre's automation tools. We decided to write a job that upgrades storage nodes from SLC 6 to CentOS 7, based on the diagram 1.

Of course, things never go as planned: this process that was supposed to take around 2 weeks to complete ended up taking around 2 months because of various hardware problems that were discovered and required manual intervention at the `BIOS` level on each node of a certain hardware type.

2.2 Namespace

The namespace is the most critical part of the EOS storage system as is contains all metadata about data stored in the cluster such as file name, size, checksum, number and location of replicas. Similar to other distributed storage systems, when a client wants to read or write a file, it has first to talk to an active management node that holds the namespace metadata.

The implementation of the management server used in production is still the original one which loads the entirety of the namespace data from a custom on-disk format to memory before allowing user traffic. The duration of this step is proportional to the number of files and directories stored in the instance as can be seen in chart 2.

There is a fail over mechanism built in the management nodes in order to make planned interventions as transparent as possible. A lot of effort has been put in making this mechanism

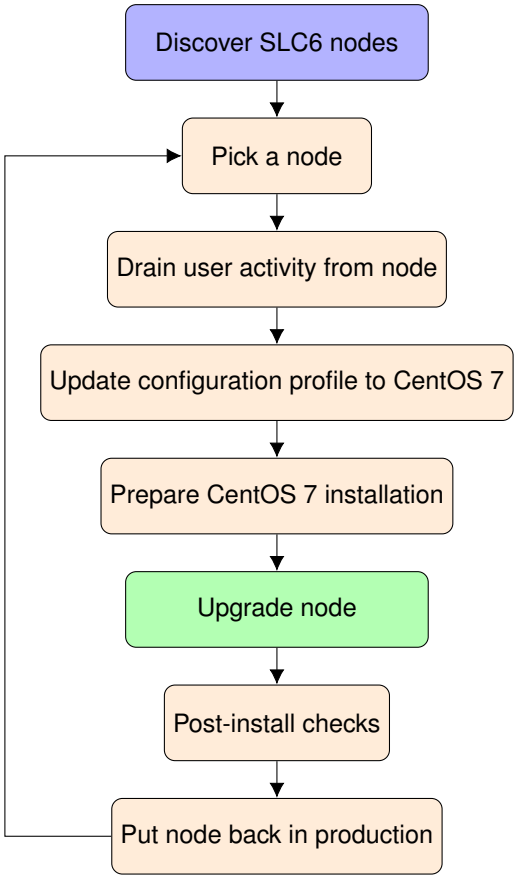

Figure 1. Simplified flowchart of the upgrade process

robust and it is now used with a very high level of confidence. Unfortunately, when hardware problems, bugs or crashes occur, we are forced to do a "cold boot", preventing experiments from writing and reading data for the duration of the process. This is becoming a real issue especially when considering LHC Run 3 and beyond and their huge demands in terms of computing and storage.

Just like with the fail over mechanism, our colleagues have put a lot of work to address this severe limitation and have come up with a distributed key-value store based on the XRootD framework, the REDIS protocol and RocksDB as persistency layer, it uses RAFT as consensus algorithm. This project, called "QuarkDB" has been presented in [3], has been successfully tested with up to 5 billion entries on commodity hardware and will be deployed on EOS physics instances during Long Shutdown 2.

2.3 Optimising resource usage

When designing EOS, one of the goals was to use commodity hardware and benefit from the streamlined hardware procurement process of the IT department to achieve the lowest possible storage cost per Gigabyte. This means that we are using the same servers as our colleagues running the Batch computing farm (see 2), with the addition of a SAS expander.

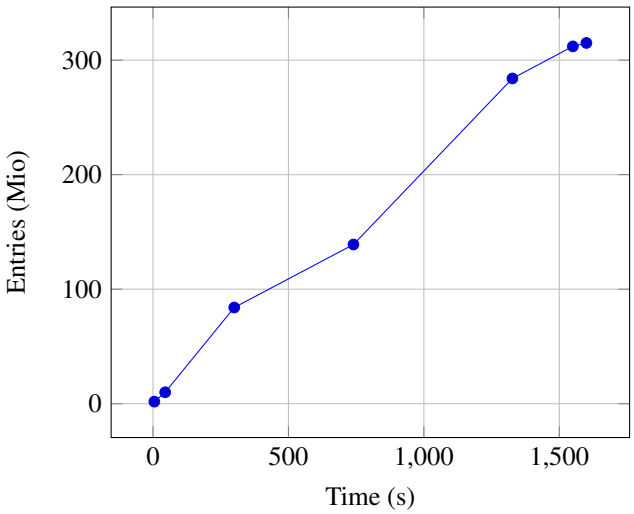

Figure 2. Namespace metadata load time

While this brings a lot of improvements in the day to day operation and management of the Computer Centre, it also means that we're wasting a lot of CPU cycles on the EOS storage machines. A task force called BEER (Batch on EOS Extended Resources) was created and its goal is to allow batch job execution on EOS storage nodes without impact on their performance. The result of this project can be found in [4].

2x Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
128 GB of DDR4 ECC RAM
2x 960 GB SSD Intel(R) DC S3520
HBA LSI SAS3008 ($added for EOS$)

Table 2. Typical EOS server configuration

An alternative to running batch jobs on storage nodes currently being investigated is to attach more than the traditional 2 storage bays with 24 disks each to the SAS expander in order to lower the overall cost overhead of the server in the full storage node. SAS expanders available on the market allow for up to 8 storage arrays to be connected to the machine, bringing the capacity of a single machine to 2.3 PB when using 12 TB disks. There is an experimentation ongoing to understand the implications of running a storage service with such "big" nodes.

Preliminary results tend to show that these configurations are best suited for archival/cold storage use cases mainly because of the bottleneck created by the network interface.

2.4 Network inconsistencies

As already mention in section 2, the size of the EOS fleet exposes it to all possible kind of faults, one of them being "Network inconsistencies". This happens usually because of a faulty router line-card or a misbehaving switch and causes major pain for users and operators,

this is usually reported by users witnessing I/O errors on `read()`, `write()` or `close()` and takes quite some effort to troubleshoot.

In order to improve the availability of the service and find the source of the issue in a more efficient way, we have deployed Consul (a distributed key/value store and service catalog) on all nodes running EOS. It makes use of a fault detection protocol based on SWIM [5] in which nodes monitor each other. This protocol being gossip based, it helps a lot in troubleshooting the kind of problems we sometimes face where nodes can reach the management server but not its fellow storage machines. Consul reports problems in its logs which we have plugged in our logging pipeline and fire alarms when such event occur.

```
2018/07/02 14:41:57 [WARN] memberlist: Was able to connect to lxfsrf16b03.cern.ch but
 ↪  other probes failed, network may be misconfigured
2018/07/02 15:06:32 [WARN] memberlist: Was able to connect to lxfsrf16b03.cern.ch but
 ↪  other probes failed, network may be misconfigured
2018/07/02 15:25:35 [WARN] memberlist: Was able to connect to lxfsrf16b03.cern.ch but
 ↪  other probes failed, network may be misconfigured
2018/07/02 15:43:41 [WARN] memberlist: Was able to connect to lxfsrf16b03.cern.ch but
 ↪  other probes failed, network may be misconfigured
2018/07/02 16:03:21 [WARN] memberlist: Was able to connect to lxfsrf16b03.cern.ch but
 ↪  other probes failed, network may be misconfigured
```

Listing 1: Consul log output

3 Other storages

3.1 Castor

Castor[6] is CERN's Tape Storage system, it is holding today more than 320 PB of data (see 3). It has entered maintenance phase around 2 years ago which means that only bug and security fixes will be implemented. Over the last few years, the disk cache in front of the tape system has shrunk as analysis use cases have moved to EOS.

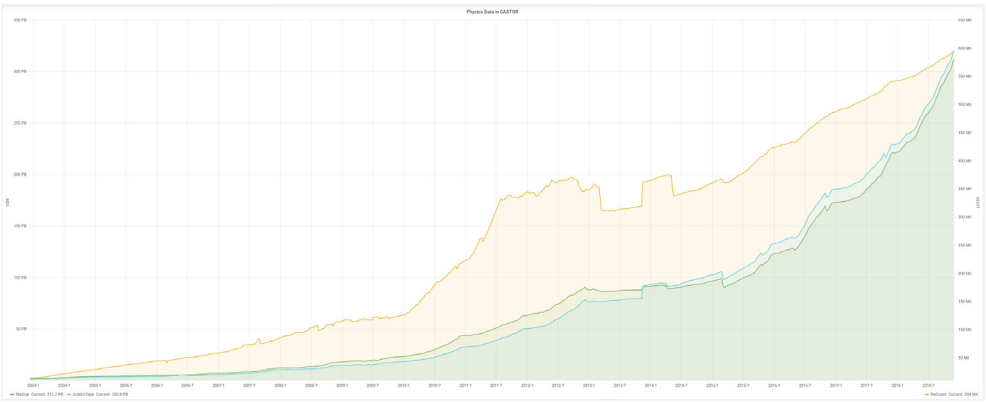

Figure 3. Castor usage

A new tape storage system has been designed and is being validated in order to prepare for the LHC Run-3 and its requirements both in terms of size and performance. This new system is called CTA (CERN Tape Archive) and leverages the disk layer of EOS. A report on its development and deployment is available at [7].

3.2 S3

The ATLAS experiment requested a few years back an object store in order to test Event-level processing[8]. We have provided them with an S3 entry-point to one of our test Ceph clusters. Along ATLAS' experiment with the "Event Service", other users contacted us with similar needs and new use cases such as backups, data processing and storage with Hadoop/Spark. As we have extensive experience with Ceph, it made a lot of sense to use the RadosGW component which provided an HTTP endpoint that is compatible with both SWIFT and S3 protocols.

The service has been designed to be horizontally scalable, just like Ceph itself. We expose RadosGW nodes to user traffic with a DNS Round-Robin entry `s3.cern.ch` and a load-balancing layer based on Traefik[9]. The load balancers can direct traffic to any RadosGW that passes health checks. When a new gateway is started, it registers itself with a Consul cluster that stores the configuration of the nodes. It allows the operators to change the configuration on the fly to all nodes at the same time, via a "watcher" mechanism that notifies the running daemons to reload their configuration.

In addition to that, we rely on the extensive instrumentation of the RadosGW and Traefik in order to have a full view of the status of the cluster and provide us with key metrics such as Time to First Byte, as can be seen in figure 4.

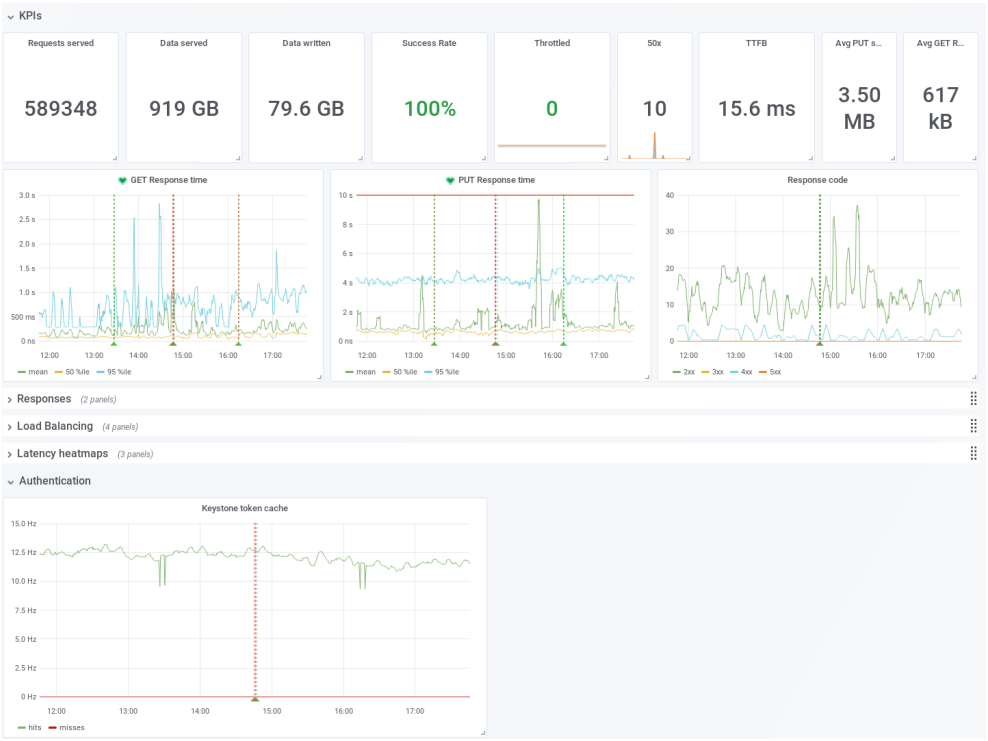

Figure 4. S3 dashboard, combining Traefik and RadosGW metrics

Since 2018, the "S3 service" has become part of CERN's IT Service portfolio. We have integrated it with Openstack to benefit from its extensive project life-cycle management, its

quota and authentication systems. User experience has also improved thanks to a component in Openstack Horizon that allows simple management of a project's buckets and objects.

3.3 NFS

The NFS Filer service is also going through a major overhaul in order to address ever increasing availability constraints. The legacy Filer service was simply made of Openstack Virtual Machines exporting one or more ZFS filesystems hosted on attached block devices. While this is very easy to use, it has severe availability implications as when a VM is down, all of its attached volumes are unreachable for its clients. In addition to that, some users of the service had grown extensively, bringing the Filer service to its limits in terms of concurrent clients and operations that a single VM can support.

Considering all these limitations, and the criticity of some of our users for CERN, we have decided to evolve the service by ditching the NFS layer and protocol and migrating to a CephFS based service.

CephFS is a POSIX compliant distributed filesystem that scales horizontally. Just like with S3 (see 3.2), we leveraged our in-depth knowledge of the Ceph storage system to provide an evolution of the Filer service, first using `ceph-fuse` and now using the kernel module (available from CentOS 7.6). Just like with S3, we have integrated our CephFS cluster with the Manila component of Openstack which effectively provides users with a distributed POSIX compliant self-service filesystem.

References

[1] Peters A and Janyst L, *Exabyte scale storage at CERN* (J. Phys. Conf. Ser. 2011) 331

[2] Rundeck: Platform for Self-Service Operations
https://rundeck.org/ [accessed: 2018-11-28]

[3] Peters A.J., Sindrilaru E.A., Bitzes G. (2017) Scaling the EOS Namespace. In: Kunkel J., Yokota R., Taufer M., Shalf J. (eds) High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science, vol 10524. Springer, Cham

[4] A. Kiryanov et al., Harvesting Cycles on Service Nodes HEPiX Spring 2017 conference (2017)
https://indico.cern.ch/event/595396/contributions/2532584/ [accessed: 2018-11-28]

[5] Scalable Weakly-consistent Infection-style Process Group Membership Protocol
http://www.cs.cornell.edu/info/projects/spinglass/public_pdfs/swim.pdf [accessed: 2018-11-28]

[6] Lo Presti, G. et al. 2007. CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN. In Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies (September 24 - 27, 2007). MSST. IEEE Computer Society

[7] E Cano et al 2015 J. Phys.: Conf. Ser. 664 042007

[8] P Calafiura et al 2015 J. Phys.: Conf. Ser. 664 062065

[9] Traefik: The Cloud Native Edge Router
https://www.traefik.io [accessed: 2018-11-28]