EOS Workshop

Monday 05 February 2018 - Tuesday 06 February 2018

CERN

Book of Abstracts

Contents

Using EOS / **29**

A Microservice Architecture for CERNBOX

Corresponding Author(s): hugo.gonzalez.labrador@cern.ch

Developing EOS & CO / **12**

A new FUSE based file system client for EOS

Andreas Joachim Peters[1]

[1] *CERN*

Corresponding Author(s): andreas.joachim.peters@cern.ch

Since the last workshop, the FUSE client has been rewritten. In this presentation we will discuss in detail the new implementation, its configuration and the new performance metrics.

Using EOS / **1**

Boxed: Docker-based service deployment in private and public clouds

Enrico Bocchi[1]

[1] *CERN*

Corresponding Author(s): enrico.bocchi@cern.ch

Docker containers are rapidly becoming the preferred way to distribute, deploy, and run services by developers and system administrators. Their popularity is rapidly increasing as they constitute an appealing alternative to virtual machines: Containers require a negligible amount of time to set-up, provide performance comparable to the one of the host, and are easy to manage, replicate, and scale-out. Also, Docker containers allow to ship software and deterministically run it by self-containing all the required dependencies and decoupling the execution environment from the host.

In this work, we present Boxed: A container-based version of EOS (the CERN disk/cloud storage for science), CERNBox (Cloud storage & synchronization service), and SWAN (Service for Web-based ANalysis). Boxed is available in two flavors: (i) A one-click setup for personal use where all services run on a single host; and (ii) a production-oriented deployment with the ability to scale out according to the storage and computing needs.

Boxed demonstrates how CERN core services can be deployed in diverse scenarios, ranging from desktop and laptop computers to private and public clouds. In all contexts, Boxed delivers the same fully-fledged services used daily by CERN scientists in demanding scenarios. All in all, Boxed contributes to the adoption of CERN cloud technologies by helping interested partners in deploying CERN services on their cloud infrastructure.

Using EOS / **9**

Building Client-Server APIs using the XRootD Scalable Service Interface

Michael Davis[1]

[1] *CERN*

Corresponding Author(s): michael.davis@cern.ch

This talk will give an overview of the XRootD Scalable Service Interface (SSI), which provides an asynchronous request-response framework with an emphasis on efficient data transfers. This will include a case study explaining how we used SSI and Google Protocol Buffers to develop the API between EOS and the CERN Tape Archive (CTA). The SSI-Protobuf bindings are available as a generic framework which can be used by other projects which need an efficient client-server protocol stack.

Using EOS / **11**

CERN Tape Archive Update

Michael Davis[1] ; German Cancio Melia[1] ; Steven Murray[1] ; Eric Cano[1] ; Julien Leduc[1] ; Anastasia Karachaliou[2] ; Vlado Bahyl[1]

[1] *CERN*

[2] *Ministere des affaires etrangeres et europeennes (FR)*

Corresponding Author(s): eric.cano@cern.ch, michael.davis@cern.ch, anastasia.karachaliou@cern.ch, steven.murray@cern.ch, vladimir.bahyl@cern.ch, julien.leduc@cern.ch, german.cancio.melia@cern.ch

The CERN Tape Archive (CTA) is the tape archival back-end for EOS and the successor to CASTOR. This talk will give an update on CTA developments since last year's EOS workshop.

Using EOS / **6**

CERNBox sharing reloaded: graceful coexistance of POSIX and sync/share in EOS ACLs

Jakub Moscicki[1]

[1] *CERN*

Corresponding Author(s): jakub.moscicki@cern.ch

I'll discuss possible improvements of the EOS permission system to gracefully support ACLS for both sync/share access (CERNBox) and filesystem access (POSIX). This will also include an implementation of automatic synchronization of the shares from EOS.

This is to address functional shortcoming for current CERNBox users and prepare for future massive filesystem access to EOS user instances at CERN.

Using EOS / **18**

CERNBox: the CERN cloud storage driven by EOS

Author(s): Luca Mascetti[1]

Co-author(s): Hugo Gonzalez Labrador [1] ; Jakub Moscicki [1] ; Massimo Lamanna [1]

[1] *CERN*

Corresponding Author(s): massimo.lamanna@cern.ch, jakub.moscicki@cern.ch, hugo.gonzalez.labrador@cern.ch, luca.mascetti@cern.ch

CERNBox is a the CERN cloud storage service. It allows synchronising and sharing files on all major desktop and mobile platforms (Linux, Windows, MacOSX, Android, iOS) aiming to provide universal access and offline availability to any data stored in the CERN EOS infrastructure.
With more than 12k users registered in the system, CERNBox has responded to the high demand in our diverse community to an easily and accessible cloud storage solution that also provides integration with other CERN services for big science: visualization tools, interactive data analysis and real-time collaborative editing.
We report on our experience managing the service and on some insight on the operations of all the underlying technologies that allow us to grow exponentially the service.

Using EOS / **16**

Demo: Setting up QuarkDB

Georgios Bitzes[1]

[1] *CERN*

Corresponding Author(s): georgios.bitzes@cern.ch

During this presentation, we will demo the setup and operation of a highly-available QuarkDB cluster, ready to be used as backend for the new EOS namespace.

Using EOS / **2**

Down the Rabbit Hole: Adventures in Bug-Hunting

Crystal Chua[1]

[1] *AARNet*

Corresponding Author(s): crystal.chua@aarnet.edu.au

This talk covers a journey through fuzz-testing CERN's EOS file system with AFL, from compiling EOS with afl-gcc/afl-g++, to learning to use AFL, and finally, making sense of the results obtained.

Fuzzing is a software testing process that aims to find bugs, and subsequently potential security vulnerabilities, by attempting to trigger unexpected behaviour with random inputs. It is particularly effective on programs or libraries that handle file or input parsing as these areas are often susceptible to buffer overflow or other vulnerabilities, for example libxml2, ImageMagick and even the Bash shell.

This approach to automated bug discovery dates back to the early 1950s, and has been steadily gaining popularity in recent years as fuzzing tools become more sophisticated - and more importantly, easier to use. Of particular note is american fuzzy lop (AFL), a genetic fuzzer written by Michał

Zalewski (lcamtuf@google), which has seen massive success - to date, it has been used in the discovery of over three hundred CVEs and many other non-exploitable bugs, in programs such as firefox, nginx, clang/llvm, and irssi.

Initial experimental fuzzing attempts against EOS with AFL have been promising, and it is hoped that further efforts to establish a process around this will be greatly beneficial in the long run.

Developing EOS & CO / 32

EOS Code Structure & Testing

Elvin Alin Sindrilaru[1]

[1] *CERN*

Corresponding Author(s): elvin.alin.sindrilaru@cern.ch

This presentation will give an overview of the code structure, resources, simple docker-based testing and more.

Using EOS / 8

EOS Ops at CERN:

Herve Rousseau[1]

[1] *CERN*

Corresponding Author(s): herve.rousseau@cern.ch

The EOS operations team at CERN operates multiple instances of EOS for the physics experiments and other activities from the laboratory.

In this presentation we will focus on infrastructure changes, best practices and evolution. A second part will mention the upgrade process we're going through to run Citrine, as well as tools we wrote and use to manage our EOS instances. We will end the talk with a glance at what's coming for 2018 and the implications for the team.

Using EOS / 13

EOS as a Data Lake technology

Simone Campana[1] ; Xavier Espinal Curull[1] ; Maria Girone[1]

[1] *CERN*

Corresponding Author(s): simone.campana@cern.ch, maria.girone@cern.ch, xavier.espinal@cern.ch

The computing strategy document for HL-LHC identifies storage as one of the main WLCG challenges in one decade from now. In the naive assumption of applying today¹s computing model, the ATLAS and CMS experiments will need one order of magnitude more storage resources than what could be realistically provided by the funding agencies at the same cost of today. The evolution of the computing facilities and the way storage will be organized and consolidated will play a key

role in how this possible shortage of resources will be addressed. In this contribution we will describe the architecture of a WLCG data lake, intended as a storage service geographically distributed across large data centers connected by fast network with low latency, and how a prototype of such architecture can be implemented using the EOS technology.

Using EOS / **15**

EOS as storage back-end for geospatial data analysis

Author(s): Armin Burger[1] ; Veselin Vasilev[2]

Co-author(s): Pierre Soille [3]

[1] *European Commission - Joint Research Centre*

[2] *European Commission, Joint Research Centre (JRC)*

[3] *European Commission*

Corresponding Author(s): armin.burger@ec.europa.eu, veselin.vasilev@ec.europa.eu, pierre.soille@jrc.ec.europa.eu

The Joint Research Centre (JRC) of the European Commission has set up the JRC Earth Observation Data and Processing Platform (JEODPP) as a pilot infrastructure to enable the knowledge production Units to process and analyze big geospatial data in support to EU policy needs. This platform is built upon commodity hardware and first operational services were made available mid 2016. It currently consists of processing and service nodes with a total of 1,200 cores, and the EOS system as storage back-end with a total gross capacity of 1.9 petabyte. EOS was deployed on the JEODPP with strong support by the CERN EOS team thanks to the CERN-JRC collaboration agreement. The JEODPP EOS instance relies on the EOS FUSE client given that currently there is no XrootD driver for the Geospatial Data Abstraction Library (GDAL) mainly used for reading and writing geospatial data files.

Multiple data processing levels have been implemented in the JEODPP. The batch processing system based on HTCondor is used for running large-scale data processing tasks based on HTCondor Docker or parallel universes and with all application dependent processes running in Docker containers. The web-based remote desktop level provides access to tools and software libraries for fast prototyping in a standard desktop environment. The interactive data processing in Jupyter notebooks allows for on-the-fly advanced data analysis and visualization.

The JEODPP platform is actively used by more than 15 JRC projects for data storage and various types of data processing and analysis. This required an additional monitoring system based on Grafana to better monitor the platform status. In order to better deal with user needs for data transfers and sharing, JRC will test the usage of CERNBox since it provides better integration with EOS than the currently deployed solution based on NextCloud.

The intensified usage of the platform and new data sources made it necessary to head for a major system extension which is currently underway. This will increase the EOS storage to a total gross capacity of 13 petabytes and the processing and service nodes to a total of 1,600 cores. The EOS service is planned to be migrated to the new Citrine release, and the usage of the new metadata management environment is envisaged once available and stable. The RAIN layout will be tested more extensively in 2018 as an alternative to the replica layout. The storage and processing platform in 2018 is going to be opened to JRC projects with new data domains and shall see a more extensive usage of machine learning technology. This way the platform is becoming the main scientific data hub at JRC.

Using EOS / **4**

EOS at the Fermilab LPC Physics Center

Dan Szkola[1] ; Bo Jayatilaka[1] ; David Alexander Mason[1] ; Marguerite Belt Tonjes[2] ; Lisa Ann Giacchetti[1]

[1] *Fermi National Accelerator Lab. (US)*

[2] *University of Illinois, Chicago*

Corresponding Author(s): lisa@fnal.gov, marguerite.belt.tonjes@cern.ch, bo.jayatilaka@cern.ch, dszkola@fnal.gov, dmason@fnal.gov

We report on operational experiences and future plans with the Fermilab LHC Physics Center (LPC) computing cluster. The LPC cluster is a 4500-core user analysis cluster with 5 PB of storage running EOS. The LPC cluster supports several hundred users annually, from CMS university groups across the US. We anticipate the total EOS storage pool to grow by 50% by the start of Run 3 of the LHC.

Using EOS / 27

EOS status of IHEP site

Haibo li[1]

[1] *Institute of High Energy Physics Chinese Academy of Science*

Corresponding Author(s): lihaibo@ihep.ac.cn

This report will talk about the current status and recent updates of EOS at IHEP Site since the first EOS workshop in 2017, covering storage expansion, issues encountered and other related work.

Using EOS / 10

EOS storage at ALICE using docker

Author(s): Martin Vala[1]

Co-author(s): Miloslav Straka [2] ; Ingrid Kulkova [2]

[1] *Technical University of Kosice (SK)*

[2] *Slovak Academy of Sciences (SK)*

Corresponding Author(s): ingrid.kulkova@cern.ch, miloslav.straka@cern.ch, martin.vala@cern.ch

Talk will present new automatic tool to configure/update EOS storage using docker (eos-docker-utils). Currently plain EOS and ALICE EOS storage configurations are supported. First production storage is running for ALICE experiment (ALICE:Kosice::EOS).

Using EOS / 30

Experience of deployment of CERNbox_SWAN at SPbSU

Corresponding Author(s): andrey.zarochencev@cern.ch

Using EOS / **28**

Extensions for policy driven data management

Corresponding Author(s): andreas.joachim.peters@cern.ch

In this presentation we will briefly explain the foreseen developments to implement the XDC and data lake concepts.

Developing EOS & CO / **33**

Introdution: from workshop to workshop

This presentation will be a short introduction to the workshop agenda and provide some basic context to understand the current status and the future roadmap.

Using EOS / **19**

New CI platform for EOS and XrootD

Jozsef Makai[1]

[1] *CERN*

Corresponding Author(s): jozsef.makai@cern.ch

In the past year, we have migrated the continuous integration platform of EOS, XrootD and all related projects from Jenkins to Gitlab CI in order to provide a more agile, satisfying and all-automated build environment.

Numerous achievements have been reached during the year.

We have introduced builds and packages for new platforms. For EOS, we have created an all-inclusive dmg package for Mac OS Sierra. Both for EOS and XrootD, Debian packaging has been made available with the support of Ubuntu Artful packages for EOS and XrootD, and with the support of Ubuntu Xenial for XrootD. A new, fully-functional apt repository has been established for making widely available the built Debian packages.

For non-release builds, compiler caching has been made available for all platofrms to reduce compilation time as much as possible.

A lot of efforts have been made towards the verification of the EOS software in hope to constantly improve the quality.
We have introduced unit testing based on Google tests framework.
We started to use multiple static analysis tools, Coverity (once a day) and cppcheck with Sonar on a regular basis to detect problems as early as possible.
We introduced a containerized environment based on Docker images (which are built and published for each code changes) to be able to conduct complex tests (FUSE, FUSEX, EOS CLI, stress tests) requiring a fully functional (including authentication) running instance of EOS for each code changes. A similar effort has been made for testing XrootD, as well.

Packages are now automatically signed for released RPMs and all Debian packages.
Our continuous integration environment also has been integrated with Koji to automatically publish release SRPMs which will be rebuilt and client packages will be available in the EPEL repositories.

Developing EOS & CO / 21

New approach of FST metadata storage

Jozsef Makai[1]

[1] *CERN*

Corresponding Author(s): jozsef.makai@cern.ch

EOS FST has been storing file metadata in different relational databases, so far. In order to simplify handling them, the way of storing file metadata is going to be changed to store Base64 encoded, serialized Protobuf metadata objects as extended attributes.

This approach also gave us the advantage to easily compress the metadata, allowing an average compression ratio of 0.5 and saving 50% of space consumed by metadata. This can mean saving up to hundreds of GB storage space per FST machine which could be used as effective storage space instead.
The compression is based on the ZStandrad algorithm using a pre-trained compression dictionary for better compression ratios. We extended it with a wrapper to eliminate bottlenecks and making it thread-safe in order to be able to use it in a massively concurrent environment without losing much time with synchronization.

FST has been extended with an automatic conversion detection from the old way to the new approach in order to be able to perform the conversion in case of necessity on its own.

Using EOS / 26

O2 Disk Buffer - WP15 EOS Performance Testing Framework

Author(s): Pete Eby[1] ; Michael Dean Galloway[1]

Co-author(s): Jeff Porter [2] ; Latchezar Betev [3]

[1] *Oak Ridge National Laboratory - (US)*

[2] *Lawrence Berkeley National Lab. (US)*

[3] *CERN*

Corresponding Author(s): rjporter@lbl.gov, latchezar.betev@cern.ch, gallowaymd@ornl.gov, ebypi@ornl.gov

The ALICE Online/Offline (O2) Disk Buffer project will deploy a 60PB EOS filesystem at CERN to accommodate the Pb-Pb data taking period planned for 2020. An initial ~6PB evaluation system is planned for deployment in May 2018.

Members from CERN, Oak Ridge National Lab (ORNL), and Lawrence Berkeley National Lab (LBNL) are collaborating on Work Package 15 (WP15) in the development of a performance testing and evaluation framework.

One objective of the framework is to validate the O2 disk buffer storage environment through the development of an EOS testing framework which uses synthetic (fio, etc.) and simulated O2 workloads under expected levels of concurrency for standardized, reproducible results and SE performance analysis.

It is envisioned this framework may be of value to the EOS community for storage design and performance evaluation decisions and benchmarking.

This talk presents a design overview of the planned testing framework modules, their implementation, and how to contribute to the development effort.

Using EOS / **25**

T2 Experiences Scaling EOS: Capacity and Performance Observations

Pete Eby[1] ; Michael Dean Galloway[1]

[1] *Oak Ridge National Laboratory - (US)*

Corresponding Author(s): gallowaymd@ornl.gov, ebypi@ornl.gov

During the last two years Oak Ridge National Laboratory (ORNL) has administered the ORNL::EOS T2 site which has seen two storage capacity expansions with installed capacity increasing from 1PB to 2.5PB. As utilization and capacity have grown observations on the performance impact of underlaying storage architecture, RAID size, filesystem design decisions, and performance tunings have been evaluated. While deploying the latest 1PB expansion performance tests iterated through different storage layouts to identify performance effects and help identify a more optimized storage configuration to meet EOS demands. Will will share observations and the evolution of their effects on deploying new capacity.

Using EOS / **3**

The Adventures of AARNet Across the EOS Dimension

David Jericho[1]

[1] *AARNet*

Corresponding Author(s): david.jericho@aarnet.edu.au

AARNet's use of EOS for both our production CDN and our CloudStor platform over the last two years has been an adventure in collaboration, experiencing bugs, and extracting esoteric knowledge from both people and the code base.

EOS exists in a space that isn't met by any existing open source scale out storage solutions. Neither Ceph, or any of the less common scale out systems provide the capabilities that EOS can deliver at tens of petabytes per cluster. That is even assuming they can scale to such a size.

AARNet is investigating how to scale up to the tens of petabytes on their continent spanning EOS storage environment, while maintaining high availability of data. The major concern is not the technical development of EOS, but rather the surrounding issues of governance, technical debt, maintenance and documentation.

This presentation discusses in brief some of the issues that have been experienced, how they were resolved (or not), and proposes some possible solutions to taking EOS from the targeted in-house open source project by CERN, to a possible contender in the increasingly common massive storage scale clusters.

Developing EOS & CO / **5**

The EOS Citrine Scheduler and new Centralized Drain

Andrea Manzi[1]

[1] *CERN*

Corresponding Author(s): andrea.manzi@cern.ch

This presentation will show the status and plans for the EOS Citrine Scheduler component focusing in particular on the configuration aspects. The talk will also introduce the new implementation of the Drain subsystem which now uses the GeoTreeEngine component for the drain placement selection.

Developing EOS & CO / 31

The EOS Citrine Version

Elvin Alin Sindrilaru[1]

[1] *CERN*

Corresponding Author(s): elvin.alin.sindrilaru@cern.ch

This presentation will cover the development and current status of the EOS Citrine release.

Using EOS / 7

The eXtreme Data Cloud (XDC) Project

Oliver Keeble[1]

[1] *CERN*

Corresponding Author(s): oliver.keeble@cern.ch

EOS is participating in the EU-funded eXtreme Data Cloud (XDC) Project which will support work on distributed deployment, caching and federation. This contribution gives an overview of the project and EOS's role within it.

Developing EOS & CO / 20

The new EOS website

Maria Arsuaga Rios[1]

[1] *CERN*

Corresponding Author(s): maria.arsuaga.rios@cern.ch

The aim of this presentation is the introduction of the new EOS website, where users and developers can find all the information that they need in one place with an easy interaction and accessibility from all type of devices.

Developing EOS & CO / 14

The new namespace and QuarkDB

Georgios Bitzes[1] ; Elvin Alin Sindrilaru[1] ; Andreas Joachim Peters[1]

[1] *CERN*

Corresponding Author(s): andreas.joachim.peters@cern.ch, georgios.bitzes@cern.ch, elvin.alin.sindrilaru@cern.ch

EOS has outgrown the limits of its legacy in-memory namespace implementation, presenting the need for a more scalable solution. In response to this need we developed QuarkDB, a highly-available datastore capable of serving as the metadata backend for EOS.

We will present the overall system design, and several important aspects associated with it, such as our efforts in providing comparable performance to the in-memory namespace through extensive caching and latency-hiding techniques.

Developing EOS & CO / 17

User authentication in eosxd: A tale of /proc/pid/environ and kernel deadlocks

Georgios Bitzes[1] ; Andreas Joachim Peters[1]

[1] *CERN*

Corresponding Author(s): andreas.joachim.peters@cern.ch, georgios.bitzes@cern.ch

Supporting multiple parallel users in eosxd requires some mechanism of distinguishing their identities, and assigning a different set of credentials to each.

In this presentation, we detail our efforts in implementing the eosxd authentication subsystem based on process environment variables.

However, reading the environment variables of a process (/proc/pid/environ) from within a FUSE daemon comes with a major caveat: The possibility of triggering a deadlock in the Linux kernel. We will outline the root cause of this issue, and describe various mitigations and workarounds for preventing it, thus making environment-based authentication in a FUSE daemon feasible.

Developing EOS & CO / 22

WLCG Accounting of EOS and smart files

Jozsef Makai[1]

[1] *CERN*

Corresponding Author(s): jozsef.makai@cern.ch

WLCG Accounting is an important task to monitor the available and used resources of the LHC computation grid. Accountable resources involve EOS storage space for the experiments.

In order to support this task force from the EOS side, EOS has introduced a new accounting interface (see accounting CLI command) to make the necessary information easily available. The accounting information consist of the quota nodes statistics and other custom, user specified data which can be provided as specific extended attributes. The output of the command is JSON text standardized for this purpose. It also supports a wide range of caching possibilities.

EOS has introduced a new feature, called “smart files” to make this new feature easy to access. The purpose of this is to be able create special (empty) files in the EOS namespace which execute specified EOS command instead upon reading them. So the accounting command can be configured as a “smart file” and the report can be easily accessed through the REST interface.

Developing EOS & CO / 23

XROOT development update - year 2017

Michal Kamil Simon[1]

[1] *CERN*

Corresponding Author(s): michal.simon@cern.ch

XRootD is a distributed, scalable system for low-latency file access. It is the primary data access framework for the high-energy physics community, and the backbone of EOS project.
In this contribution we (briefly) discuss the most important new features introduced in year 2017 including: support for systemd socket inheritance, XrdSsi, Caching Proxy v2, support for local files and redirections and extreme copy. Also, we report on the most important bugfixes and enhancements to the
client. Finally, we give an overview of the plans for the year 2018.

Developing EOS & CO / 24

XrootD Erasure Coding Plugin

Michal Kamil Simon[1]

[1] *CERN*

Corresponding Author(s): michal.simon@cern.ch

In order to bring the potential of Erasure Coding (EC) to the XrootD / EOS ecosystem an effort has been undertaken to implement a native EC XrootD plugin based on the Intel Storage Acceleration Library (ISAL). In this contribution we discuss the architecture of the plugin, carefully engineered in order to enable low latency data streaming and 2D erasure coding. Also, we report on the status, and the future work.