EOS deployment on Ceph RBD/CephFS with K8s

Federico Fornari[1], Alessandro Costantini[1], Alessandro Cavalli[1], Daniele Cesini[1], Doina Cristina Duma[1], Antonio Falabella[1], Enrico Fattibene[1], Luca Mascetti[2], Lucia Morganti[1], Andreas-Joachim Peters[2], Andrea Prosperini[1] and Vladimir Sapunenko[1]

[1] CNAF - Italian Institute for Nuclear Physics, Bologna, Italy
[2] CERN - IT Storage, Genève, Switzerland

E-mail: federico.fornari@cnaf.infn.it

Abstract. The present activity focused on the integration of different storage systems (EOS [1] and Ceph [2]) with the aim to combine the high level scalability and stability of EOS services with the reliability and redundancy features provided by Ceph. The work has been carried out as part of the collaboration between the national center of INFN (Italian Institute for Nuclear Physics) dedicated to Research and Development on Information and Communication Technologies and the Conseil Européen pour la Recherche Nucléaire (CERN), with the goal of evaluating and testing different technologies for next-generation storage challenges. This work leverages the well-known open-source container orchestration system, Kubernetes [3], for managing file system services. The results obtained by measuring the performances of the different combined technologies, comparing for instance block device and file system as backend options provided by a Ceph cluster deployed on physical machines, are shown and discussed in the manuscript.

1. Introduction

Due to the increasing interest on data management services capable to cope with very large data resources allowing the future e-infrastructures to address the needs of the next generation extreme scale scientific experiments, the national center of INFN (Italian Institute for Nuclear Physics) dedicated to Research and Development on Information and Communication Technologies (CNAF) and the Conseil Européen pour la Recherche Nucléaire (CERN) joined their experiences on storage systems to evaluate and test different technologies for next-generation storage challenges.

The present work consisted on investigating the possibility to exploit EOS, an open-source storage software solution for multi-PetaByte storage management at CERN Large Hadron Collider, in order to deploy a distributed file system over a storage backend provided by Ceph, an open-source software platform capable to expose data through interfaces for object, block and posix-compliant storage.

The motivation for this activity resides in the possibility to integrate the two storage solutions to combine the high-level scalability and stability of EOS services with the reliability and redundancy features provided by Ceph. The documentation of both products clearly explains that EOS [1] and Ceph [2] services can be deployed as containers and orchestrated by Kubernetes, the open-source container-orchestration system for automating computer application deployment, scaling and management. In this respect, Kubernetes has been employed to test different cluster-deployment scenarios (both on cloud and bare-metal) and assess their performances, bringing important improvements in terms of system operations, management and scalability, and comparing for instance block device and file system as backend options on Ceph side.

2. Materials and Methods

To carry out the present study, a testbed was created, using both cloud and bare-metal resources. In particular, the activities have been performed by relying on two Kubernetes clusters (cloud and bare-metal), a Ceph cluster (on bare-metal) and EOS services (deployed using bare-metal resources).

Ceph Nautilus (release 14.2.14) has been used to deploy and configure an 8-node cluster with a total amount of 320 CPUs and 1.5TB RAM (see Figure 1 for more details). The network connection between the nodes of the Ceph cluster is provided by a 2×10 Gbit/s bonding channels setup. Each node has 27 disks with 8TB each, connected via Serial Attached SCSI interface from 4 JBODs with 60 disks each. The total capacity of the Ceph cluster reaches 216 disks, with an overall capacity of 1728TB of raw storage space. The storage deployment relies on 2 different replication strategies: replica 3 for Ceph RADOS Block Device (RBD) and erasure code 6+2 for CephFS.

On the Kubernetes side, one cluster has been deployed on 3 Openstack-provided Virtual Machines based on CentOS 7 operating system and with network connectivity at 1 Gbit/s (see Figure 1). These nodes, 1 master and 2 workers, with an overall capacity of 12 virtual CPUs and 12GB of RAM, have been provided with Kubernetes 1.20.1 components on top of Docker 20.10.1 as container engine. The second, bare-metal, Kubernetes cluster has been deployed on the same servers hosting the Ceph cluster (see Figure 2 for more details) with one additional bare-metal server acting as master.

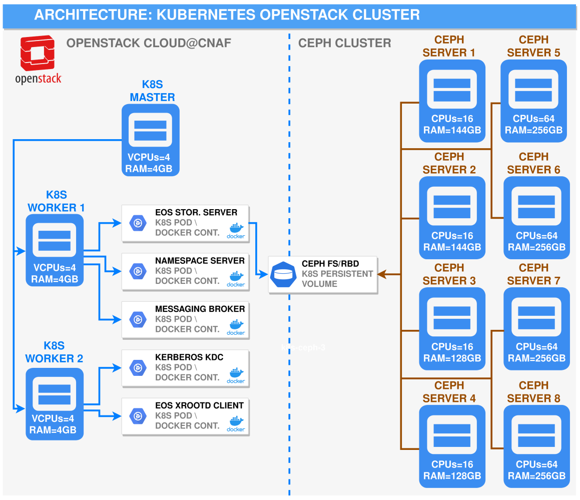
ARCHITECTURE: KUBERNETES OPENSTACK CLUSTER
OPENSTACK CLOUD@CNAF
openstack
CEPH CLUSTER
K8S MASTER
VCPUs=4 RAM=4GB
K8S WORKER 1
VCPUs=4 RAM=4GB
K8S WORKER 2
VCPUs=4 RAM=4GB
EOS STOR. SERVER K8S POD \ DOCKER CONT.
NAMESPACE SERVER K8S POD \ DOCKER CONT.
MESSAGING BROKER K8S POD \ DOCKER CONT.
KERBEROS KDC K8S POD \ DOCKER CONT.
EOS XROOTD CLIENT K8S POD \ DOCKER CONT.
docker
CEPH FS/RBD K8S PERSISTENT VOLUME
CEPH SERVER 1 CPUs=16 RAM=144GB
CEPH SERVER 2 CPUs=16 RAM=144GB
CEPH SERVER 3 CPUs=16 RAM=128GB
CEPH SERVER 4 CPUs=16 RAM=128GB
CEPH SERVER 5 CPUs=64 RAM=256GB
CEPH SERVER 6 CPUs=64 RAM=256GB
CEPH SERVER 7 CPUs=64 RAM=256GB
CEPH SERVER 8 CPUs=64 RAM=256GB

Figure 1. High level architecture of Kubernetes cluster deployed on cloud resources.

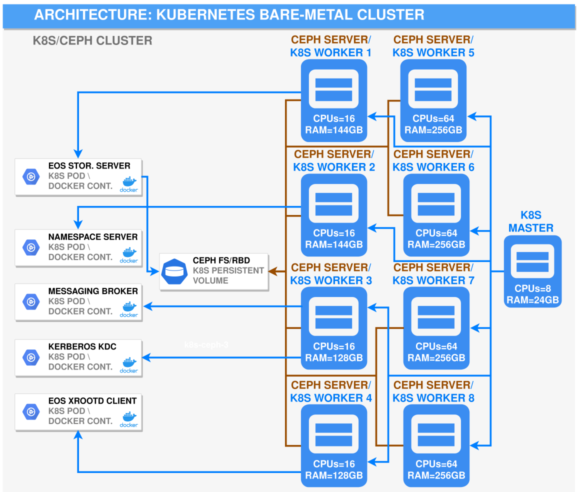
ARCHITECTURE: KUBERNETES BARE-METAL CLUSTER
K8S/CEPH CLUSTER
CEPH SERVER/ K8S WORKER 1 CPUs=16 RAM=144GB
CEPH SERVER/ K8S WORKER 2 CPUs=16 RAM=144GB
CEPH SERVER/ K8S WORKER 3 CPUs=16 RAM=128GB
CEPH SERVER/ K8S WORKER 4 CPUs=16 RAM=128GB
CEPH SERVER/ K8S WORKER 5 CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 6 CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 7 CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 8 CPUs=64 RAM=256GB
K8S MASTER CPUs=8 RAM=24GB
EOS STOR. SERVER K8S POD \ DOCKER CONT.
NAMESPACE SERVER K8S POD \ DOCKER CONT.
MESSAGING BROKER K8S POD \ DOCKER CONT.
KERBEROS KDC K8S POD \ DOCKER CONT.
EOS XROOTD CLIENT K8S POD \ DOCKER CONT.
docker
CEPH FS/RBD K8S PERSISTENT VOLUME

Figure 2. High level architecture of Kubernetes cluster deployed on bare-metal resources.

Regarding EOS (version 4.8.57), it has been previously mentioned that the related services can be deployed as containers. In this respect, integration activities began by using solutions developed by CERN that allow to automatically deploy an EOS cluster [4] comprising also plugins [5] such as Storage Server Plugin (FST), Meta Data Namespace and Scheduling

Redirector Plugin (MGM), Message Queue Server Plugin (MQ), clients and Kerberos Key Distribution Center (KDC) [6], available from a public GitHub repository [7].

Due to some limitations, mainly related to the Kubernetes deployment within the project, an important work has been carried out to allow the initialization of the EOS file system on volumes attached to the storage servers by using Kubernetes Persistent Volume Claims based on Ceph RBD or CephFS backends. Improvements consisted also in adding the possibility to specify the storage space to be reserved for the Persistent Volumes. Moreover, storage servers configuration has been also modified in order to make the associated Kubernetes Pods part of a StatefulSet [8] that allows the migrated Pods to keep their Persistent Volumes with originally stored data in case of failures or re-deployment.

To test the integration among Ceph and EOS, a proof-of-concept EOS cluster has been deployed using cloud resources. Functionality tests involving EOS with Ceph RBD and CephFS have been performed on K8s Openstack instances, deploying a EOS cluster with 1 storage server and 1 client. The EOS file system has been configured with a replica 1 layout and a 4MB block size. Different volume sizes have been tested by sequentially transferring 1GB and 10GB files until reaching total occupancy; the related results in terms of average file transfer rate are shown in Figure 3.

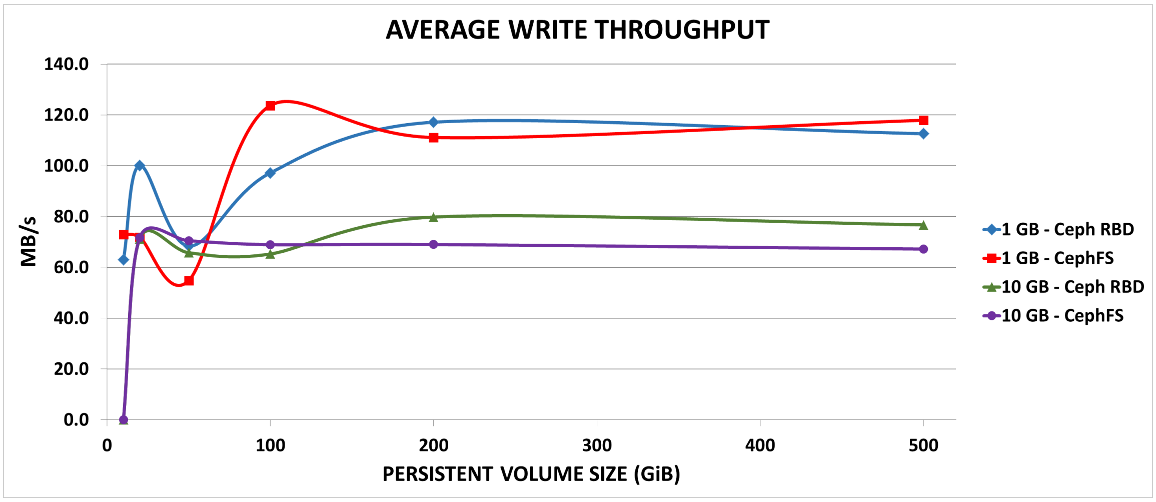

Figure 3. Average write throughput rates on Kubernetes cluster, deployed on cloud, using different persistent volume size.

These preliminary results show a bandwidth saturation at 1 Gbit/s reached with 1 client, proving good stability of the setup. The throughput observed with 10GB files is slightly below saturation. This can be due to the limited amount of RAM chosen during the resource provisioning for the Openstack virtual machines. Data replication can be easily verified on Ceph side by monitoring the storage allocation of the replica 3 RBD pool used for the test. Writing 1 file on EOS, in fact, the occupied space seen with `ceph df` is 3 times the file size copied on EOS. Fail-over has been also tested by shutting down a worker node hosting a storage server and verifying that the previously written data continue to be accessible after re-deploying the Kubernetes Pod on a different worker node.

This preliminary work has been fundamental to enhance the `eos-on-k8s` project features and to start a massive study aimed at testing the performances on a EOS/Ceph/K8s bare-metal deployed cluster, as extensively described in the next section.

Setup	Disks	Replica Strategy	Server Pod(s)	Client Pod(s)	W/R protocol
CephFS	216	Erasure Coding 6+2	1	from 1 to 8	XRootD
Ceph RBD	216	Replica 3	1	from 1 to 8	XRootD
EOS-CephFS	216	Erasure Coding 6+2	1	from 1 to 8	XRootD
EOS-Ceph RBD	216	Replica 3	1	from 1 to 8	XRootD

Table 1. Kubernetes setups adopted for the performance tests.

3. Performance Tests

Starting from the results described in the previous section, extended tests have been carried out by using bare-metal resources, where a Kubernetes cluster has been deployed. As already mentioned, the K8s workers share the same resources of the Ceph cluster (see Figure 2). The purpose of this choice is to assess performances and improvements given by the adoption of highly coupled resources when using Ceph as EOS storage backend. For the tests, the XRootD [9] server Pods have been used to expose a storage area deployed on a Ceph Persistent Volume, in order to compare the behavior of stand-alone Ceph and EOS integrated with Ceph when transferring data via XRootD protocol. As a matter of fact, EOS exposes data via XRootD by default. Furthermore, the setup related to clusters and services adopted in the present study is given in Table 1 and explained hereafter:

(i) 1 XRootD server Pod with Persistent Volume on CephFS and a maximum of 8 XRootD client Pods (CephFS in Table 1).

(ii) 1 XRootD server Pod with Persistent Volume on Ceph RBD and a maximum of 8 XRootD client Pods (Ceph RBD in Table 1).

(iii) 1 EOS storage server Pod with Persistent Volume on CephFS and a maximum of 8 XRootD client Pods (EOS-CephFS in Table 1).

(iv) 1 EOS storage server Pod with Persistent Volume on Ceph RBD and a maximum of 8 XRootD client Pods (EOS-Ceph RBD in Table 1).

For each setup mentioned above, the tests consisted in sequentially writing and reading 100 1GB files via XRootD by using from 1 to 8 parallel clients. As the available workers are 8, the client Pods have been deployed maximizing the usage of the available resources. All the client Pods are connected to the same storage server Pod which provides the Persistent Volume Claims for the tests.

The results have been collected and reported in Figure 4 and Figure 5 where the average disk servers read and write throughput for each deployed setup, respectively, have been shown.

As can be seen from the figures, CephFS, in general, shows better read/write scores if compared with Ceph RBD. Such behavior can be explained and it is actually due to the different access patterns to the disks. CephFS, in fact, acts as a file system shared over the network, where different machines can access it all at the same time. On the other side, RBD uses images shared over the network.

Another important result shown by the tests is that the EOS-Ceph integration has a better throughput, if compared to Ceph alone, when the number of clients increases. This can be due to a cache effect among EOS and Ceph that becomes evident by distributing the clients on the Kubernetes worker nodes. The same cache effects can also explain the performances shown on Figure 5, where the average write throughputs are reported.

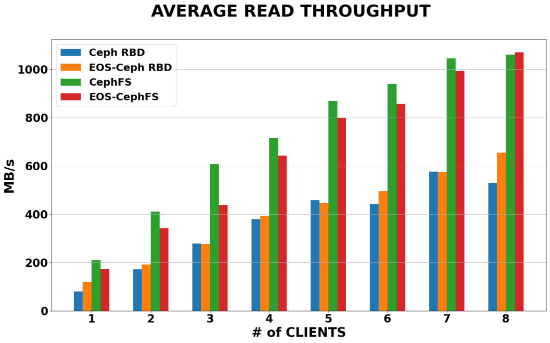

Figure 4. Average read throughputs using different Pod clients.

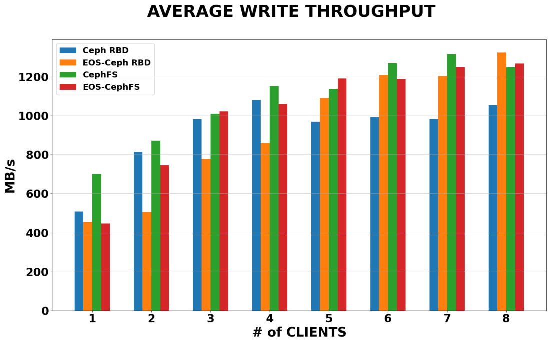

Figure 5. Average write throughputs using different Pod clients.

4. Conclusions

The present study, performed within a collaboration between the national center of INFN (Italian Institute for Nuclear Physics) dedicated to Research and Development on Information and Communication Technologies and CERN, has been able to evaluate and test different technologies for next-generation storage challenges at CNAF.

In particular, the integration between EOS and Ceph, using Kubernetes as a natural framework to test different cluster-deployment scenarios, gave good results in terms of scalability and stability (given mainly by EOS services), reliability and redundancy (provided by Ceph), integration and management (provided by Kubernetes) and overall performances.

Testing different scenarios, in fact, allows to deal with different problems for which proper solution have been developed, bringing also important improvements in the integration of such services.

Stimulated by the important results obtained for the present study, new advancements are planned for future work. In particular, further analyses implying setups with higher number of servers and parallel clients are foreseen.

References

[1] EOS webpage: `https://eos-docs.web.cern.ch/`
Last seen: Feb 2022

[2] Ceph webpage: `https://ceph.io/`
Last seen: Feb 2022

[3] K8S webpage: `https://kubernetes.io/`
Last seen: Feb 2022

[4] EOS GitHub project: `https://github.com/cern-eos/eos`
Last seen: Feb 2022

[5] EOS GitHub project: `https://github.com/cern-eos/eos#project-directory-structure`
Last seen: Feb 2022

[6] Kerberos protocol: `https://en.wikipedia.org/wiki/Kerberos_(protocol)`
Last seen: Feb 2022

[7] GitLab@CERN webpage: `https://gitlab.cern.ch/eos/eos-on-k8s`
Last seen: Feb 2022

[8] K8S Statefulset webpage: `https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/`
Last seen: Feb 2022

[9] XRootD webpage: `https://xrootd.slac.stanford.edu/`
Last seen: Feb 2022