Integration of XRootD into the cloud infrastructure for ALICE data analysis

2015 J. Phys.: Conf. Ser. 664 022036

(http://iopscience.iop.org/1742-6596/664/2/022036)

Journal of Physics: Conference Series **664** (2015) 022036 doi:10.1088/1742-6596/664/2/022036

Integration of XRootD into the cloud infrastructure for ALICE data analysis

Mikhail Kompaniets[1], Oksana Shadura[2,3,4], Pavlo Svirin[2,3], Volodymyr Yurchenko[3,4] and Andrey Zarochentsev[1]

[1] Saint Petersburg State University, Russia
[2] CERN, Switzerland
[3] National Technical Univ. of Ukraine "Kyiv Polytechnic Institute", Ukraine
[4] Bogolyubov Institute for Theoretical Physics, Ukraine

Abstract.

Cloud technologies allow easy load balancing between different tasks and projects. From the viewpoint of the data analysis in the ALICE experiment, cloud allows to deploy software using Cern Virtual Machine (CernVM) and CernVM File System (CVMFS), to run different (including outdated) versions of software for long term data preservation and to dynamically allocate resources for different computing activities, e.g. grid site, ALICE Analysis Facility (AAF) and possible usage for local projects or other LHC experiments.

We present a cloud solution for Tier-3 sites based on OpenStack and Ceph distributed storage with an integrated XRootD based storage element (SE). One of the key features of the solution is based on idea that Ceph has been used as a backend for Cinder Block Storage service for OpenStack, and in the same time as a storage backend for XRootD, with redundancy and availability of data preserved by Ceph settings.

For faster and easier OpenStack deployment was applied the Packstack solution, which is based on the Puppet configuration management system. Ceph installation and configuration operations are structured and converted to Puppet manifests describing node configurations and integrated into Packstack. This solution can be easily deployed, maintained and used even in small groups with limited computing resources and small organizations, which usually have lack of IT support. The proposed infrastructure has been tested on two different clouds (SPbSU & BITP) and integrates successfully with the ALICE data analysis model.

1. Introduction

Distributed storage provides access and location transparency, high scalability, migration options and failure recovery. In the same time cloud computing is focused on maximizing the effectiveness of the shared resources, so cloud model is usually based on resources not only shared by multiple users but also dynamically reallocated per demand.

- We tried to use storage solution suitable for ALICE experiment computing purposes - cloud-based XRootD service. Our goal was to create a possibility for small organizations with limited resources to participate in ALICE data analysis process.
- Main task was to prototype fast deployment schema for cloud by using one unified answer file in Packstack, easily modified by administrators.
- We attempted to provide lightweight solution that can manage clusters either with a small quantity of computing nodes or large computing farms.

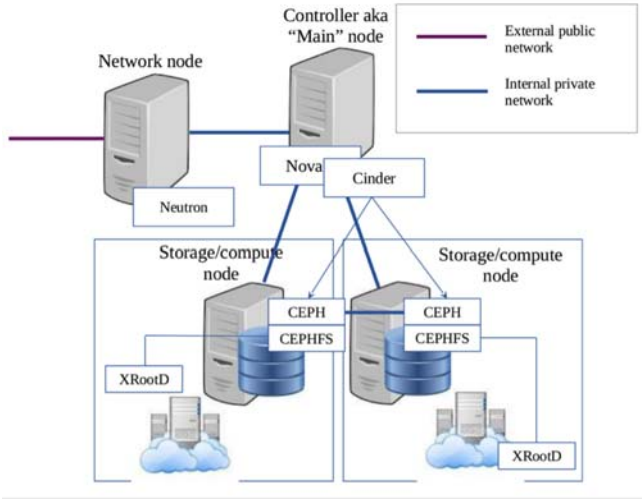


Figure 1. Principle schema: Controller, Network and Compute Nodes

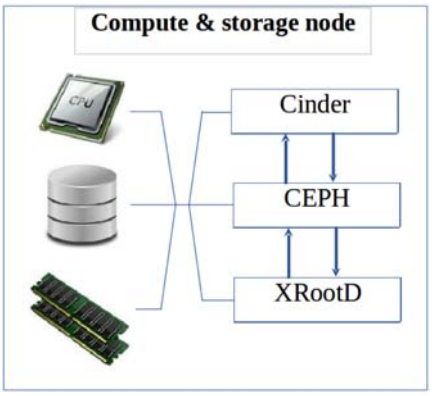


Figure 2. Prototype resources sharing schema

2. Software solutions used in prototype

2.1. Packstack RDO

Packstack RDO [1] is a command line utility that uses Puppet modules (easy and scalable configuration management tool) to support rapid deployment of OpenStack on existing servers over an SSH connection. Packstack is suitable for deploying both single node proof of concept installations and more complex multi-node installations. It has modular structure that allowed us to add new features for installation and configuration of new software modules into OpenStack.

2.2. Ceph

Cinder, OpenStack storage component, is responsible for exposing block devices to virtual machines. It has 27 storage drivers, among which are: Ceph RBD [2] [3], GlusterFS [4], NFS, LVM.

While comparing Ceph and LVM, possibility to achieve high availability remains difficult in case of LVM. If LVM goes down, all Cinder blocks will not respond anymore and will stay unavailable for a certain amount of time. At the same time Ceph provides more reliability, due to idea of providing a distributed storage, where all data is replicated over the cluster of Object Storage Devices.

However, Ceph is not a panacea for all problems. It has some issues that still need to be solved as a clock drift sensitivity and redundant data replication caused by Ceph policy.

2.3. XRootD

XRootD [5] is a system for scalable cluster data access, challenged by thousands of parallel batch jobs and file sessions widely used in LHC experiments.

2.4. CVMFS & mCernVM

CernVM [6] tools already become a de-facto standard of organization and management access to main HEP software for data analyses and scientific researches in experimental physics.

3. Deployment schema

Describing our system let's point out main components:

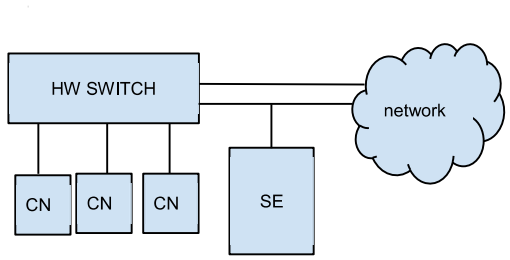


Figure 3. Network structure

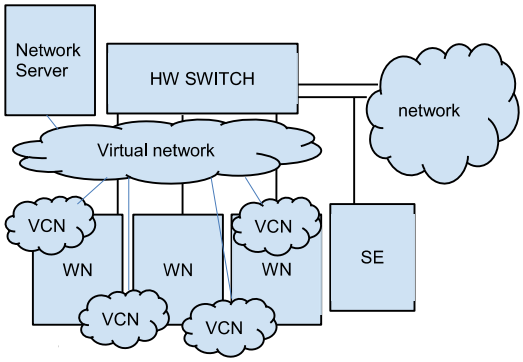


Figure 4. Resources sharing schema

- On Figure 1 shown Controller node as a dedicated server for management of all services on machines in the cluster. Cinder service handles storage requests and interacts with Ceph service located on this node. Due to distributed nature of Ceph, Cinder has quick access to all the data on all storage nodes.
- Network node is responsible for managing all connections between machines in the cluster. Neutron service usually handles all external and internal connections.
- Other machines are considered as unified nodes, which act as both storage and compute nodes in the same time. Virtual machines, running on nodes, can use XRootD service. It connects to Ceph storage, situated on the same node and mounted with CephFS with the help of set of developed scripts. XRootD writes data to the storage on this node, and Ceph replicates data over the cluster.

All resources (CPU, RAM, storage) are shared between OpenStack with Ceph, RadosFS/CephFS & XRootD services (Figure 2). It allows to use them more efficiently, while storage has single logical structure and can be accessed by all services through Ceph. Since these services are located on the same machine, they access shared memory and CPU and use as much resources as they need and when they need.

4. Solution for a network infrastructure

The original OpenStack network implementation, also known as Nova Network, assumed a basic model of performing all isolation through Linux VLANs and Iptables. These are typically sufficient for small and simple networks, but larger customers are likely to have more complex network requirements.

Neutron introduces the concept of a plugins, which is a backend implementation of the OpenStack Networking API. A plug-in can use a variety of technologies to implement the logical API requests and offer a rich set of network topologies.

5. Network structure specifics for a distributed data storage

A standard ("pure") computation cluster consists of the following elements (Figure 3):

- hardware computing element, which is a set of computing nodes,
- separate storage element, which has one or many common network interfaces,
- hardware switch, which provides connectivity from compute nodes to data storage and also serves as a gateway to another network.

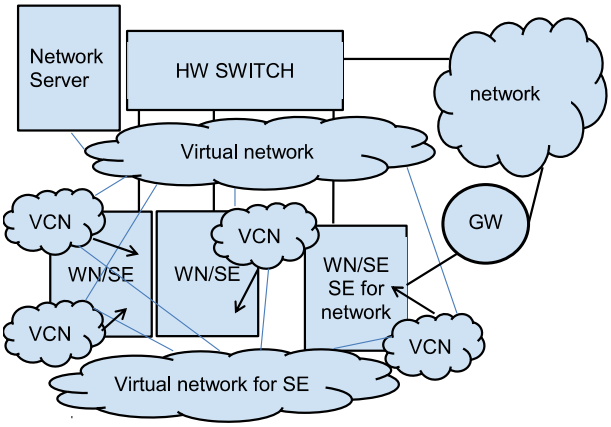


Figure 5. Resources sharing schema with dedicated network

In Figure 3 every node has direct access to data storage through the hardware router.

The difference from a standard schema is that nodes are substituted by a virtual compute nodes (VCNs) running on hardware worker nodes (WNs), considered as a basic entity of our logical schema and a network server with a virtual router and a virtual internal network.

Hardware WNs access SE only through network server, which acts as a gateway for the virtual network and may be a reason of the bottlenecks. This problem could be solved by adding a virtual network interface on SE and applying Iptables rules to it. The virtual interface on Figure 4 is displayed with a dotted line.

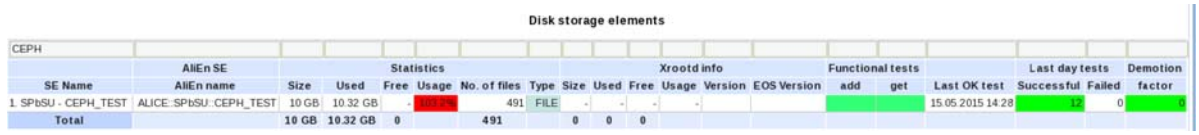
Let's discuss possibility when SE works as a distributed device and not as separated. For an outbound access there is one node that used for read and write requests from other sites and could cause a bottleneck. We can avoid it by adding a virtual network interface on the same node or any other node which hosts monitor service. Preliminary results of testing are shown in "Tests and benchmarks" section.

Distributed storages give one more access point through different storage nodes, which can be used for performance improvement. Virtual compute nodes connected to a storage talk not to random hardware WN, but to the node where the virtual compute node runs. However main specific of cloud environments is that a virtual machine does not depend on a WN and does not know the node where it is running on. We solved it by running a script, which has an access to cloud control node and responds to a request from a virtual machine with its hardware WN IP address.

Considering this it is recommended to create a dedicated network for SE operations (a network architecture is presented on Figure 5) (arrows show the connections from virtual compute node to carrier hardware WN while copying data).

We managed to construct the architecture described above through creating additional virtual network using Neutron tools, applying network rules for network and implementing of next scripts:

- Work node script (WNS) creates a virtual interface with an access to dedicated storage network and assigns an IP to interface with further monitoring of interface status.
- Cloud controller script (CCS) listens to a TCP port and responds to requests with IP of a WN host where the requesting virtual machine is running on.
- Cron-based Virtual machine script (VMS) is running CCS for quering address of parent WN and comparing received virtual interface IP against resolved IP of SE.


Disk storage elements

CEPH

	AliEn SE		Statistics						Xrootd info						Functional tests			Last day tests		Demotion
SE Name	AliEn name	Size	Used	Free	Usage	No. of files	Type	Size	Used	Free	Usage	Version	EOS Version	add	get	Last OK test	Successful	Failed	factor
1. SPbSU - CEPH_TEST	ALICE::SPbSU::CEPH_TEST	10 GB	10.32 GB	-	103.2%	491	FILE	-	-	-	-					15.05.2015 14:28	12	0	0
Total		10 GB	10.32 GB	0		491		0	0	0									

Figure 6. Ceph-based ALICE storage element; http://alimonitor.cern.ch/stats?page=SE/table

On every WN we had setuped an XRootD server connected to CERN network. While contacting server or redirector from a VM the final request will be done to XRootD running on the same WN. It was observed a significant decrease in speed or timeouts while copying big amounts of data to external SEs or to internal SE through the gateway. The problem was solved by turning off TCP Segmentation Offload (TSO) on virtual machines.

6. XRootD and Rados with CephFS/RadosFS backend

One of the Ceph's features is to use it as a storage for block devices and as data storage in the same time. Ceph could be used as a storage for disk images for virtual machines (Ceph-based Cinder). The second approach is to use it for integration of a distributed storage, but still no implementation for ALICE had been managed. In the first scenario suggested by A.Peters [7] RadosFS is used for a direct access to Ceph-based Rados [8]. Second scenario suggests a simpler way to mount Ceph pools in UNIX-like way by using CephFS.

While testing RadosFS, it was discovered that AliEn copy operations fail because of specific checks, based on possibility of changing the pemissions for files being copied. Thus, in spite of the successful attempts with RadosFS, we decided to move to CephFS for final prototype, however we still continue RadosFS integration, but its current status does not allow to compare RadosFS performance against CephFS correctly.

7. Tests and benchmarks

According to described requirements we developed an automatic cluster configuration toolkit which was tested on SPbSU and BITP sites. We used ALICE storage based only on CephFS solution, caused by lack of compatibility of RadosFS with our concept. On Figure 4 we proved that Ceph-based storage could be easily used in ALICE environment. In the same time the main goal was to evaluate the performance of proposed architecture with more complicated test suits checking I/O and other comparison benchmarks.

7.1. LVM vs Ceph

As a next step for testing of Ceph as a object distributed storage for VM it was suggested to use 3 different settings:

- dedicated storage on LVM - Cinder is configured with LVM backend (Conf 1 on Figure7);
- dedicated distributed storage - Cinder is configured with Ceph backend running on additional 3 nodes (Conf 2 on Figure 7);
- distributed storage on compute nodes - Cinder is configured with Ceph backend running on the compute nodes (Conf 3 on Figure 7).

On Figures 7 and 8 there are presented the results of testing of 3 virtual machines on two different sites. These results were obtained using Bonnie test suit. These figures demonstrate only approximate comparison results from different clusters and not represent full statistical picture of our research.

From tests we can easily notice that configuration of SPbSU cluster is not properly managed according to write/read operations comparing with BITP ones and need to be investigated, tuned

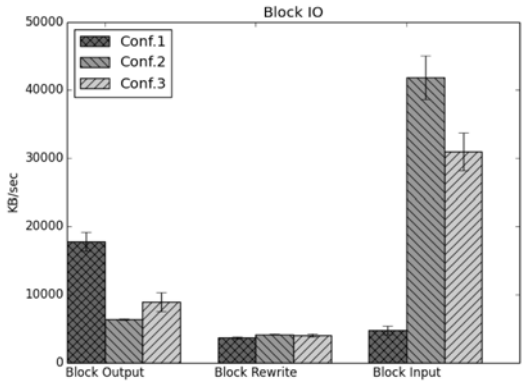


Figure 7. Bonnie tests on SPBSU cluster

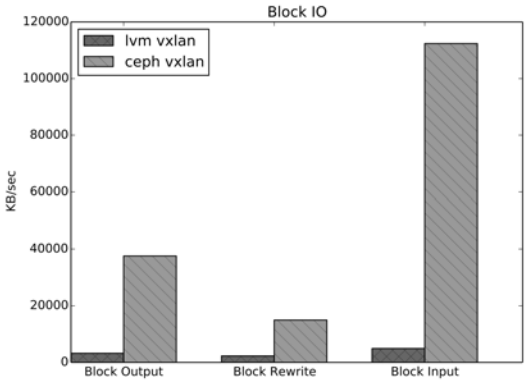


Figure 8. Bonnie tests on BITP cluster

and re-checked again. In the same time it is still showing benefits of reading from CEPH storage. Testing of BITP Ceph configuration showed significant benefits of using CEPH comparing to LVM backend.

7.2. PROOFBenchmarks

In order to evaluate the CPU load and I/O performance we used a PROOFBenchmark tool. On Figure 9, MakeDataset test results are presented where datasets with 30000 events were created on 8 virtual machines and 3 worker nodes writing data to an external SE. The tests have been run on 6 VMs each with 4 cores on 3 nodes on SPbSU cluster, VM images stored in Cinder (Ceph). See below a list of used dataset targets:

- dataset, created on VM disks - "VM";
- dataset, created on XRootD SE, mounted on WNs using NFS - "WN-nfs";
- the same as previous case, but write operations are managed through gateway "GW-WN-nfs";
- dataset, created on XRootD mounted on WNs using CEPHFS SE with write operation on parent nodes "WN-ceph";
- dataset, created on XRootD mounted on WNs using CEPHFS with write operation through gateway server "GW-WN-ceph";
- dataset, created on XRootD SE without NFS and with write operation through gateway server "GW-WN-xfs";
- dataset, created on external XRootD on XFS SE "GW-SE".

Figure 9 represents an example MakeDataSet output, while Figure 10 demonstrates speed of MakeDataSet for different targets.

The tests show small slowdown while writing to Ceph on parent nodes, however, at the same time there is also slowdown noticed while writing to SE mounted with NFS comparing to write operation to local storage. Possible explanation of these misbehavior could be reasoned by further review of correct behavior of PROOFBench testing algorithm in case of usage on different backends.

7.3. Tests based on copy check strategy

We reproduced configuration of PROOFBenchmark tests using simple copy tests. These tests don't show load rate of computing elements, but allow to reinterpret results from other point of

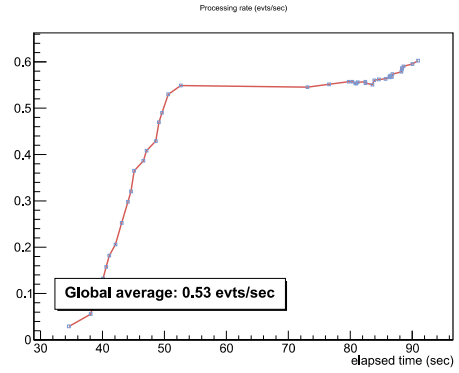


Figure 9. PROOFBenchmark (MakeDataSet())

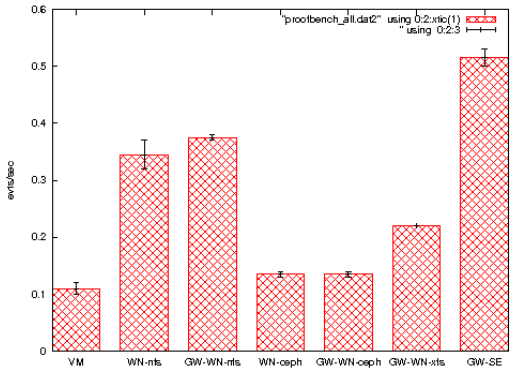


Figure 10. PROOFBenchmark tests (speed/SE type)

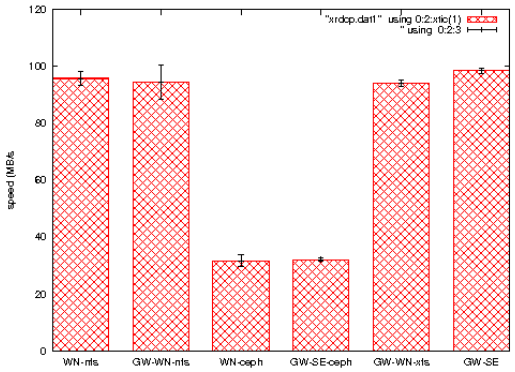


Figure 11. Speedup of Xrdcp() depending of SE backend

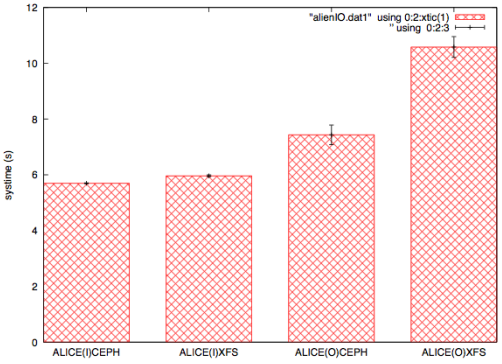


Figure 12. AliEn I/O benchmarking (Ceph/XFS backends)

view.

From results shown on Figure 11 we can see, that Ceph concedes in comparison with XFS (local file system), but not significantly enough, roughly around factor of 2.5 on proposed workload.

7.4. Read/write testing using AliEn

Last test was provided to check status of read-write operations through AliEn on Ceph and XFS.

Figure 12 demonstrates system time used for copy procedure and according this case Ceph shows some visible benefits comparing to XFS, where input is managed through copy procedure from Vobox to SE and output is read operation from SE to Vobox.

In general we can say, that in the case of overload of virtual machines on nodes, read-write speed of Ceph on those nodes decreases, as expected, but not critically. The effectiveness of suggested system of traffic readdressing on parent node is not proven, we still hope to find more consistent test.

8. Problems, possible issues and proposed solutions

During first tests we used RadosFS solution but all attempts were unsuccessful because of permission problems (chmod) between XRootD layout and RadosFS. Our final procedure of

testing was managed using CephFS backend.

Concerning network - issues were solved creating special network solution described in Section 5 "Solution for a network infrastructure". Some additional settings also were made in Neutron service to ensure correct Modular Layer 2 option support.

While testing CernVM images that are considered to be a baseline for CERN LHC experiments, it appeared that CernVM solutions are configured to use old style EC2 APIs without Neutron API support, while there is no possibility in OpenStack with Neutron network service to use EC2 services without additional network support (EC2 VPC network should be visible as Neutron network entity with support of public/private networks).

9. Conclusions

9.1. Benefits of using Ceph

We gained speedup using Ceph as a main storage backend comparing to known storage solutions, but in the same time there are still some misunderstandings and listed problems to be solved in the next iteration of our research.

9.2. Packstack: use or not?

Packstack usage shows significant advantages comparing with alternative Openstack installations, while it has a fast development cycle and development of our modules always should be compatible to all possible OS and changes in other services of RDO development. The main problem is that our provided network solution is not so flexible, thus possible solution could be to manage our own Neutron plugin for providing redundant network.

10. Other considerations

As a possible step ahead we should mentioned the fact that Docker container usage leads to some echoing opinions that cloud computing environments should abandon virtual machines and replace them with containers due to their lower overhead and potentially better performance. Currently intensive development of Ceph inside Docker containers has started [9] and while using RBD as backend for the Docker containers, Ceph could be installed as part of the system and mount outside of the container before launching. That could lead for additional researches to be done in a future.

11. Acknowledgment

The present work is supported for MK and AZ in part by Saint-Petersburg State University research grants 11.38.242.2015

References

[1] Packstack RDO `https://openstack.redhat.com/`

[2] CEPH `http://ceph.com/`

[3] Ceph and openstack integration `http://ceph.com/docs/next/rbd/rbd-openstack/`

[4] Glusterfs and related distributed data projects `http://www.gluster.org/`

[5] Xrootd webpage `http://xrootd.org/`

[6] Cernvm portal `http://cernvm.cern.ch/portal/`

[7] Peters A J 2014 *US-ALICE Grid operations review*

[8] Radosfs - a filesystem library based in rados `https://github.com/joaquimrocha/radosfs`

[9] Run Ceph OSDs in Docker `https://registry.hub.docker.com/u/ceph/osd/`