Journal of Physics: Conference Series **513** (2014) 042047 doi:10.1088/1742-6596/513/4/042047

Building an organic block storage service at CERN with Ceph

Daniel van der Ster, Arne Wiebalck
CERN, Geneva, Switzerland
E-mail: daniel.vanderster@cern.ch

Abstract. Emerging storage requirements, such as the need for block storage for both OpenStack VMs and file services like AFS and NFS, have motivated the development of a generic backend storage service for CERN IT. The goals for such a service include (a) vendor neutrality, (b) horizontal scalability with commodity hardware, (c) fault tolerance at the disk, host, and network levels, and (d) support for geo-replication. Ceph is an attractive option due to its native block device layer RBD which is built upon its scalable, reliable, and performant object storage system, RADOS. It can be considered an "organic" storage solution because of its ability to balance and heal itself while living on an ever-changing set of heterogeneous disk servers.

This work will present the outcome of a petabyte-scale test deployment of Ceph by CERN IT. We will first present the architecture and configuration of our cluster, including a summary of best practices learned from the community and discovered internally. Next the results of various functionality and performance tests will be shown: the cluster has been used as a backend block storage system for AFS and NFS servers as well as a large OpenStack cluster at CERN. Finally, we will discuss the next steps and future possibilities for Ceph at CERN.

1. Introduction

The Large Hadron Collider (LHC) at CERN is a very large data producer, generating more than 25PB of particle collision data per year. In the years leading up to the LHC, the storage requirements generated by the detectors were unique in the world. In preparation for LHC computing, CERN developed new systems to store and manage the vast quantities of collision data to both disk and tape. These systems are now successfully managing more than 100PB of data. However, in recent years, innovations from companies such as Google, Yahoo, and Facebook have demonstrated that the Big Data problems seen by other communities are approaching, and often surpassing, those of the LHC. As such, it is crucial for CERN to investigate the potential of these new storage technologies and evaluate if they may be leveraged by our community.

Ceph [1] is one such technology. It is a flexible storage product offering a variety of interfaces. First, it is a reliable object store with basic and advanced GET/PUT semantics made available via a command line interface as well as an API. The object store is composed of a cluster of object storage daemons, which use replication and data integrity checking to achieve a fault-tolerant and self-healing system. Next, it builds upon the object store to offer network block devices, Amazon S3 and OpenStack SWIFT protocol access, and finally a general purpose network filesystem.

During 2013, CERN IT has been evaluating Ceph via a series of test instances growing from 50TB to 3PB in size. This paper reports on these tests and proposes the next steps for exploiting this promising technology in the HEP community.

2. Motivation for a new storage service at CERN

The storage and management of physics and user data at CERN is handled by a suite of services. First, the physics data is stored on two systems, CASTOR and EOS. CASTOR [2] is a mature disk and tape management system which presently holds close to 90PB of data, with more than 70PB on tape [3]. EOS [4] is a relatively young storage system which provides rapid file location lookup and data access via the XROOTD [5] protocol; EOS already holds close to 20PB of data and is growing rapidly as it takes over much of the responsibility for physics data from CASTOR. Next, there is AFS, which is a globally-accessible network filesystem offering users universal access to their home directories, workspaces, and project areas. With close to 2 billion files stored and around 75kHz access rates [6], the service is active and growing. Finally, CERN manages a series of NetApp-based NFS boxes for databases and other applications, as well as a large DFS service for Windows data management.

Recently the CERN IT department has been evolving its infrastructure to leverage the cloud computing paradigm developed in industry. In this Agile Infrastructure project [7], the department is building a large OpenStack cluster (designed to run of order 100000 virtual machines) and using Puppet for configuration management. Further, CERN is deploying a new remote data centre in Budapest, Hungary, and the increasing scale of data storage is demanding a move towards uniform storage hardware across all services. Finally, the department has new storage services on the horizon, including a Dropbox-like desktop synchronisation service [8] and the scientific data publishing service Zenodo [9].

All of these conditions have created an environment which demands a new type of storage service at CERN. First, and fundamentally, the service must offer image and block storage to OpenStack Glance and Cinder; these will enable the core IT services to migrate to the cloud without constraints on data consumption. Next, it must offer the possibility to consolidate the backend storage of AFS/NFS/OwnCloud/Zenodo onto a new block storage layer. And finally, it needs to be performant (throughput and latency), reliable (fault tolerance and bitwise correctness), and operable (scalable and self-healing).

3. A brief introduction to Ceph

Ceph [1] is an open software project which is designed with goals similar to those of the evolving CERN IT infrastructure. The first and foremost component is RADOS, a Reliable and Autonomic Distributed Object Store. RADOS exposes a cluster of physical disks via object storage daemons (OSDs) – each of which is responsible for the integrity of objects it stores. Data is arranged logically into pools, each of which being composed by a configurable number of placement groups. Placement groups are implemented physically as a folder on an OSD. The mapping of placement groups to OSDs is implemented via the CRUSH (Controlled Replication under Scalable Hashing) algorithm [10] – this algorithmic approach to data placement eliminates the need for (and problems associated with) a data location catalog in Ceph. CRUSH is infrastructure-aware, meaning that it allows OSDs to be weighted and organised logically by room, row, rack, host, etc.; CRUSH's programmable placement rules instruct the OSDs how to replicate data across failure domains to address correlated failures. In a healthy cluster, data is written and read via the master copy of an object, but in a degraded state clients may communicate with a replica, which logs the operations so that the master replica may replay the changes once it recovers. Clients and OSDs get a consistent view of the cluster by consulting one of N Ceph Monitor daemons – the monitors are redundant and can tolerate up to $floor(N/2)$ failures.

In addition to a simple GET/PUT/RM command line interface, RADOS offers a rich synchronous and asynchronous API with bindings in most popular languages. This service is sufficient for applications which require object storage and whose backend can be modified to communicate directly with RADOS. For applications which require Amazon S3 or OpenStack SWIFT compatibility, the RADOS Gateway (RGW) service can be deployed. RGW is implemented as a cluster of front end http daemons which handle the S3 and SWIFT API requests; buckets, user accounts and quotas are all implemented by the gateways. During a PUT operation, S3 and SWIFT objects are broken into n megabyte chunks for performance.

Beyond the object storage use-cases, Ceph implements a block storage service, RBD, and a general file system, CephFS. RBD is an iSCSI-like interface which allows physical and virtual machines to use a block device (i.e. `/dev/rbd1` on Linux) which can be formatted with a filesystem and used like a local physical disk. Clients for RBD are present in qemu-kvm and the Linux Kernel. In both cases, RBD devices are implemented by storing many n megabyte chunks as objects in RADOS – the object names are generated by concatenating the device's name with the chunk address. The qemu-kvm client supports RBD images which stripe writes across many OSDs for performance. CephFS is a general purpose network filesystem which exposes POSIX-like file access to clients. Like in RGW and RBD, files are chunked and may be striped, however CephFS requires an additional daemon, the MDS, to store file metadata. CephFS is still in development and has not yet been deemed to be production quality.

4. First steps with a 50 terabyte prototype

In early 2013 the AFS/NFS team in CERN IT proposed the evaluation of Ceph in order to handle the OpenStack block storage use-case. The initial test deployment was performed on ten former CASTOR disk servers, each with 24 disks. Three Ceph monitors and one RGW were deployed as Microsoft HyperV virtual machines.

Installing the cluster was relatively simple; the software was installed via the RPM packages made available by the community. While the `ceph-deploy` tool did not at that time support RedHat-based distributions, configuration of the disks and daemons was not overly complicated and a 250TB cluster was running within two days.

The initial basic tests, including writing/reading objects and testing the response of the system to OSD failures, were successful. However, while evaluating the data rebalancing features (i.e. reweighting the OSDs) the cluster encountered a large problem from which it never recovered. During object recovery and rebalancing, Ceph OSDs require around 2GB of memory each in order to track the states of the many objects being moved between disks. Our 250TB was severely disproportionate with respect to memory, with less than 500MB available to each OSD.

After reconfiguring the cluster to match the memory constraints (decreasing from 24 OSDs per server down to 4 or 5, making 50TB usable storage) the tests were repeated and this time succeeded. RADOS, RBD, RGW, and CephFS were all confirmed to function as advertised. Some basic tests of the OpenStack volumes on RBD were performed and it was found that the qemu-kvm version shipped by RedHat didn't support RBD. After compiling in the necessary RBD driver (a task which was later performed by Inktank Inc., the commercial enterprise employing the core Ceph developers), the OpenStack volumes functioned correctly.

The eventual success of this 50TB cluster convinced the department on the merits of Ceph and justified further evaluation in a petabyte-scale test deployment.

5. Deploying a 3 petabyte Ceph Cluster

In mid-2013 the department provisioned 48 new storage servers and 5 monitor servers in order to build a 3PB Ceph cluster. The hardware configuration is detailed in tables 1 and 2.

Table 1. Hardware configuration of the Ceph OSD Servers

CPU	Dual Xeon E5-2650	32 threads including hyperthreading
Memory	64GB RAM	
Network	Dual 10Gig-E NICs	Only one connected
System Disks	3x 2TB Hitachi disks	3-way mirror
Storage Disks	24x 3TB Hitachi disks	Eco drives, 5900RPM

Table 2. Hardware configuration of the Ceph Monitors

CPU	Dual Xeon L5640	24 threads including hyperthreading
Memory	48GB RAM	
Network	Dual 1Gig-E NICs	Only one connected
System Disks	3x 2TB Hitachi disks	3-way mirror

5.1. Deploying Ceph with Puppet

Using Puppet manifests derived from those written by the community [11], the cluster deployment and operations have been fully automated. When new servers are installed, they are simply added to the Puppet host group for our OSDs; the `puppet-ceph` code then detects the OSD disks, formats them with XFS, prepares the OSD filesystem, retrieves an authorisation key for the OSD daemon, and generates the required `ceph.conf`. The last step, starting the OSD daemon, is left to the system administrator in order to permit him or her to schedule the resulting data rebalancing. `Mcollective` is used for subsequent management of the cluster, including Ceph version upgrades and other bulk operations.

In order to better integrate into the CERN environment, some changes to `puppet-ceph` were necessary. First, since the Puppet environment is shared within CERN, the mechanism used to distribute the Ceph admin keyring needed to be modified from being a exported resource (which any Puppet managed machine at CERN would have been able to import) to using a kerberos-authenticated secure copy. Next we have changed the management of physical disks in order to better handle disk replacements and server reboots. These changes [12] will be pushed upstream to the community after some time in preproduction at CERN.

5.2. Placement Group and OSD Configuration

Ceph allows for an arbitrary number of data pools, which are configured to have N_{PG} placement groups having $N_{replica}$ replicas each. An ideal Ceph cluster should have between 50-100 placement groups per OSD, and following the rule

$$50 \leq \frac{N_{PG} * N_{replica}}{N_{OSD}} \leq 100 \tag{1}$$

we find that that N_{PG} for our cluster ($N_{OSDs} = 1152$, $N_{replica} = 3$) should range between 19200 and 38400.

In order to separate placement group replicas as widely as possible, the CRUSH map has been configured to separate OSDs by room, rack, and host using the `osd crush location` option, for example `osd crush location = room=0513-R-0050 rack=RJ35`.

5.3. Cluster Monitoring

In order to monitor the health of the cluster, two probes have been implemented. First, an hourly cron job checks the status of the cluster an alerts the operators if any OSDs have failed

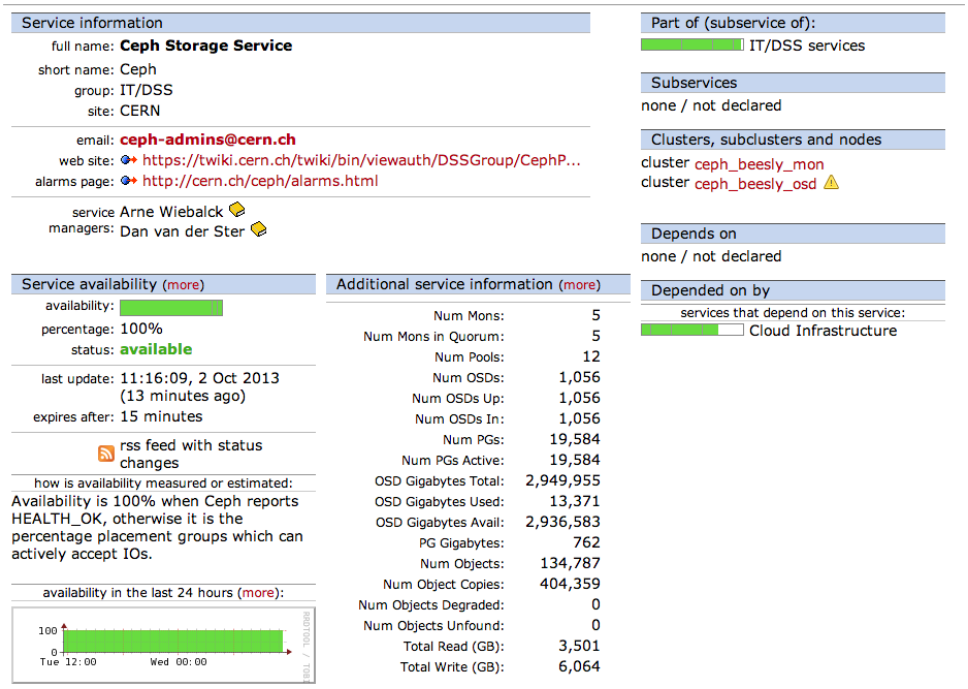

Figure 1. Service Level Status of the 3 petabyte cluster at CERN.

or placement groups have become degraded. Next, Ceph has been integrated into CERN's Service Level Status (SLS) system, where availability and other service metrics are recorded. The SLS probe has been written using the Ceph JSON data dumps via a Python API; it may be useful to other Ceph administrators and is available online [13].

5.4. Benchmarking

Using the built-in RADOS benchmarking tool (`rados bench`), we have measured the performance of the cluster.

5.4.1. Single client/100 threads write and read rados bench When writing to a single replica test pool, the cluster achieves a write performance of 997MB/s using 100 threads from a single client machine and a read performance of 962MB/s also using 100 threads. Increasing the number of replicas to 3 decreases the write performance to roughly 650MB/s while the read performance is not largely affected.

5.4.2. 120 million file test In order to evaluate the scalability of Ceph as the number of files increases, `rados bench` was used to write 120 million tiny objects into the OSDs. These 0-byte objects were created (with 3 replicas) at a rate of 15kHz using 1000 client threads. During this exercise, there was no increased static load or memory consumption noticed on the OSD daemons, so the scalability of Ceph by this dimension was validated. However, it was noted that in a cluster with 120 million tiny objects, the data rebalancing or recovery resulting from adding or removing OSDs will be substantial – we exercised the addition of one new server to the cluster and the resulting rebalancing required 24 hours to complete. This test highlights that even though Ceph makes server and OSD changes operationally trivial and transparent to end users, significant changes to a cluster should be planned with appropriate at-risk periods

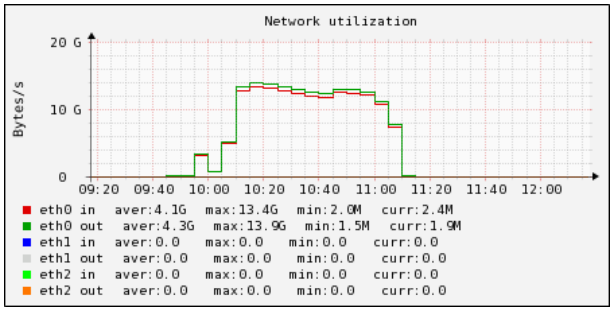

Figure 2. Network usage at 14 gigabytes per second during an all-to-all rados bench test.

while the cluster rebalances its data.

5.4.3. Simultaneous rados bench from all disk servers Finally, in order to saturate the backend network switch, `Mcollective` was used to execute a one-hour `rados bench` on all OSD servers. The result is depicted in figure 2, showing a saturation at 14GB/s. It is not clear if this limitation was generated by the backend network switch or the write limitations of the OSDs themselves.

6. Use-Cases for Ceph at CERN

The availability of a 3PB cluster has attracted various novel use-cases within the IT department. These are described here, and note that all use-cases are still being evaluated.

6.1. OpenStack images and volumes

As mentioned previously, the original motivation for building a Ceph cluster was partly to provide block storage for OpenStack Cinder volumes and Glance images. Using the qemu-kvm package provided by Inktank, this is possible. However, a few problems were encountered. First, it was noted that the performance of RBD volumes is greatly improved when striping writes across many objects; striped objects are not available by default in OpenStack Grizzly, so our team developed a patch to add this functionality.

Next, we observed an problem resulting in failed Glance image uploads. It was found that our large cluster puts a requirement on the clients to create more than 2000 processes. Finally, because the deployment of Grizzly at CERN made use of the experimental *Cells* feature, Cinder volumes were not supported on our installation. Detailed benchmarking and optimisation of the Cinder/RBD driver is planned for the future before this is put into production at CERN.

6.2. A high performance tape buffer for CASTOR

CASTOR requires a high performance tape buffer in order to supply the O(100) tape drives with data at 250MB/s each. Without having the functionality to stripe reads across many disk servers, CASTOR itself cannot supply data at the required rates. By employing a RADOS-based buffer between the disk and tape servers, the striping features of Ceph may be leveraged to deliver the required read performance.

Since it was found that single threaded RADOS GET does not exceed 200-250MB/s, a simple streaming RADOS client was developed. This tool, tentatively named `cephcp`, breaks an object into n pieces and writes those in parallel int RADOS, thereby allowing the client to communicate with many OSDs at once during write and read operations. Using cephcp, PUT and GET operations from a single client using a stripe width of 16 transferred at around 1GB/s, close to the limit of the 10Gig-E network.

6.3. Backend Storage for AFS/NFS/DPM

AFS and NFS services operated using dedicated and specialised hardware; DPM is a HEP filesystem developed and tested at CERN. By consolidating the backend storage provisioning on a standard IP-based service such as Ceph RBD, the operations of the backend storage can be optimised by eliminating development and operational redundancies.

Prototype deployments of AFS and DPM servers have already been tested using the kernel RBD client. This work is still in progress.

6.4. Object Storage for OwnCloud/Zenodo

OwnCloud is a desktop synchronisation service similar to Dropbox; it is being evaluated by CERN IT [8] in order to see if it works well enough to attract CERN users from the popular Dropbox-like services on the market (which have security and confidentiality implications for CERN documents and data). OwnCloud advertises that it can write its data to an S3 storage backend, so the RGW is an attractive offering to build this service upon. Unfortunately, the initial tests of the OwnCloud S3 feature demonstrated that this functionality is not yet mature enough for production.

Zenodo [9] is a scientific data publication service which is being developed at CERN. Currently the service uses AFS as a backend data store, however the developers are currently evaluating the use of RADOS to allow the service to scale up as its popularity increases.

7. Future Plans and Conclusion

In our tests of Ceph at CERN, it is clear that many users expect a native POSIX-like network filesystem similar to AFS and NFS. A gateway-based solution, which exposes underlying Ceph storage via AFS or NFS, could be usable in the short term; however the ideal solution (for performance and reliability) would use Ceph directly from the clients. Whereas CephFS is already available to evaluate, its usage outside of test instances is not currently recommended by the community. However, in the timescale of 6 to 12 months we expect the filesystem to mature to a point where we can begin offering it to the CERN community.

Looking forward, it is clear that Ceph has already presented a novel solution to the new backend storage requirements of CERN IT. We have only just started the first large scale tests of the services with OpenStack, CASTOR, AFS, NFS, DPM, and others, though thus far the service has delivered in terms of performance, scalability, and reliability.

References

[1] S. Weil et al. Ceph: A scalable, high-performance distributed file system. *Proc. of the 7th symp. on Operating systems design and implementation*. 2006.
[2] G. Lo Presti et al. CASTOR: A distributed storage resource facility for high performance data processing at CERN. *Proc. of the 24th IEEE Conf. on Mass Storage Systems and Technologies, 2007*. pp. 275280.
[3] X. Espinal et al. Disk storage at CERN: handling LHC data and beyond. *These proceedings*.
[4] A. Peters et al. Exabyte Scale Storage at CERN. *J. Phys.: Conf. Series 331 052015*. 2011.
[5] A. Dorigo et al. Xrootd - A highly scalable architecture for data access. *WSEAS Trans. Comput.*. April 2005.
[6] D. van der Ster et al. Toward a petabyte-scale AFS service at CERN. *These proceedings*.
[7] B. Moreira et al. Production Large Scale Cloud Infrastructure Experiences at CERN. *These proceedings*.
[8] J. Moscicki. Rethinking how storage services are delivered to end-users at CERN: prototyping a file sharing and synchronisation platform with ownCloud. *These proceedings*.
[9] Zenodo Website. Online at *http://www.zenodo.org*.
[10] S. Weil et al. CRUSH: Controlled, scalable, decentralized placement of replicated data. *Proc. of the 2006 ACM/IEEE conf. on Supercomputing*. 2006.
[11] Git repository for `enovance/puppet-ceph`. Online at *https://github.com/enovance/puppet-ceph*.
[12] Git repository for `cernceph/puppet-ceph`. Online at *https://github.com/cernceph/puppet-ceph*.
[13] Git repository for `cernceph/ceph-scripts`. Online at *https://github.com/cernceph/ceph-scripts*.