Journal of Physics: Conference Series **664** (2015) 042054 doi:10.1088/1742-6596/664/4/042054

Ceph-based storage services for Run2 and beyond

Daniel C. van der Ster, Massimo Lamanna, Luca Mascetti, Andreas J. Peters, Hervé Rousseau
CERN, Geneva, Switzerland

E-mail: daniel.vanderster@cern.ch

Abstract. In 2013, CERN IT evaluated then deployed a petabyte-scale Ceph cluster to support OpenStack use-cases in production. With now more than a year of smooth operations, we will present our experience and tuning best-practices. Beyond the cloud storage use-cases, we have been exploring Ceph-based services to satisfy the growing storage requirements during and after Run2. First, we have developed a Ceph back-end for CASTOR, allowing this service to deploy thin disk server nodes which act as gateways to Ceph; this feature marries the strong data archival and cataloging features of CASTOR with the resilient and high performance Ceph subsystem for disk. Second, we have developed RADOSFS, a lightweight storage API which builds a POSIX-like filesystem on top of the Ceph object layer. When combined with Xrootd, RADOSFS can offer a scalable object interface compatible with our HEP data processing applications. Lastly the same object layer is being used to build a scalable and inexpensive NFS service for several user communities.

1. Introduction

The CERN IT Department has operated a 3 petabyte Ceph [1] cluster in production for roughly 18 months. With a primary use-case of hosting OpenStack [2] Cinder volumes and Glance images, the cluster presently hosts around 176 terabytes of user data, replicated 3 times. This data is exposed to end-user virtual machines via close to 3000 RADOS block devices, specifically as 1200 Cinder volumes and 1800 Glance images. The cluster is currently hosted in Meyrin, Switzerland and a second cluster is planned for CERN's data centre in Budapest.

Over the course of these 18 months, our team has gathered a variety of operations experience, much of which will be summarized in this short article. Section 2 presents the OpenStack use-cases in more detail, and highlights our conclusions regarding tuning Ceph for IOPS capacity. Section 3 summarizes ongoing work to exploit Ceph—RADOS in particular—to store physics data. Finally, we conclude in section 4.

2. Block Storage for OpenStack

The model offered by OpenStack Cinder volumes, whereby users and developers can attach an arbitrarily large block device to a virtual machine, simplifies the operations of core IT and LHC experiment services. Notable examples include ATLAS PanDA, CMS GlideinWMS, and the CMS Webtools, all of which have replaced dedicated hardware with virtual machines and attached storage. In addition to most core IT services, the Data and Storage Services Group is in the process of migrating NFS users out of a proprietary filer appliance and onto a solution composed of ZFS-formatted volumes on virtual NFS servers. Nearly 100 terabytes of these "mini-Filer" servers have already been deployed. The CERN central CVMFS service also employs

attached volumes in the Squid Proxy services, and work is currently underway to migrate the stratum zero data from a filer appliance onto attached volumes. Indeed, with close to 1000 attached volumes, it is clear that the virtual block device model is proving essential to the virtual CERN IT infrastructure.

2.1. IOPS and SSDs

One key feature that makes Cinder volumes operable is QoS types, which allows operators to throttle the IOPS and bandwidth available to each attached volume. Paying attention to available and consumed IOPS is critical to a successful Ceph service. Using a write-ahead journal to guarantee write durability, Ceph acknowledges a write only after it has been persisted on all replicas. Therefore, the latency of the Ceph journal is central to the available IOPS on a given Ceph object storage daemon (ceph-osd).

In our tests, we have concluded that the two core Ceph use-cases (namely, object and block storage), can be treated differently with regard to journal requirements. For object storage, overall throughput is normally more important than latency; co-locating the journal on the ceph-osd disk usually offers adequate performance, even after considering the double-write penalty coming from journalled writes. On the other hand, block storage has been observed to rely on a low small write latency in order to offer a responsive experience; we have concluded that a solid-state journal is essential for ceph-osd being used for block storage.

In particular, we have observed a 5-10x increase in the 4kB write IOPS capacity after moving ceph-osd journals from a co-located spinning disk onto an SSD [3]. Additionally, SSD journals have been measured to decrease the small write latency from 50ms to around 5ms. Given that SSD journal devices are usually partitioned for many ceph-osd's, it is important to select a drive with adequate random write performance, and of course being a write-only journal the endurance offered by the drive should be evaluated carefully. The Ceph user community has concluded that the Intel DC S3700 is a good choice for this use-case; we use the 200GB version and store five ceph-osd journals per SSD.

2.2. Tuning Ceph and Linux

One issue that our cluster suffered during the Dumpling release cycle was that the ceph-mon LevelDBs were ever increasing in size, sometimes by many gigabytes per day. As a workaround we scheduled a manual compaction on each ceph-mon every two days. Thankfully this issue seems to have been resolved after upgrading to the Firefly. Early in the life of the cluster we found that the ceph-mon reliability was highly dependent on the latency of the LevelDB filesystem; we currently store ceph-mon data directories on SSDs and no longer observe monitor elections that can plague a non-responsive cluster.

For the ceph-osd processes, we have paid particular attention to the scrub and deep-scrub processes. By default, each placement group (PG) is scrubbed weekly, and this often happens at the same time each week. After observing that this *thundering herd* of scrub processes was hurting client latencies, we developed two strategies to mitigate the problem. First, we wrote a process to preemptively schedule PG scrubs in order to smear them out over the week and decrease the number of concurrent scrubs. Second, we requested changes to the ceph-osd process to allow the ceph-osd disk thread to be `ionice`'d to the idle priority. With these two mechanisms applied, scrubbing is now generally transparent to our end-users.

Handling many small files on a ceph-osd also exposes a few weaknesses in Linux, or at least in RHEL 6 systems. First, we have found the NUMA zone reclaim feature can lead to whole servers freezing for several minutes while the kernel drops the inode caches as a result of a NUMA zone running out of memory. We now have a general policy of disabling this feature with the sysctl `vm.zone_reclaim_mode = 0`. UpdateDB can be similarly disruptive if it is configured to index the ceph-osd FileStores; we now add `/var/lib/ceph` to the `PRUNEPATHS` in

`/etc/updatedb.conf`. Finally, Ceph is widely known to use an excessive number of threads and sockets; servers are recommended to increase `kernel.pid_max` to its maximum possible value, 4194303, and clients are required to increase the allowed number of open files with `ulimit -n` to a value larger than 4096.

2.3. Significant Incidents

Our Ceph cluster has been exposed to several disk and host failures during 18 months of operations. Disks fail monthly on average, and this type of failure has been found to be completely transparent to our end-users. The SSD journals currently have a perfect reliability record, though the authors acknowledge that they are tempting fate by mentioning this statistic. On two occasions an entire ceph-osd host went offline. In both of these cases the host and data could be recovered within a few hours by an operator. After this experience we now disable automatic backfilling for host failures by setting the `mon osd down out subtree limit` option to `host`.

In October 2014 we suffered a power cut and UPS failure which took down 3 out of 5 ceph-mon daemons. The cluster was down for 18 minutes as a result. In the postmortem, it was found that during this outage block device IOs were paused in flight and resumed once the cluster quorum was restored. No filesystem corruptions were reported, and we have therefore concluded that virtio-blk is superior to virtio-scsi, which would have timed out the IOs.

In March 2015 a central router issue led to significant packet loss on the Ceph networks. This led to OSDs being marked up and down repeated over many hours, and around 20 OSDs reached a heartbeat timeout and "committed suicide" (a data safeguard feature). This outage required manual intervention to restart the down OSDs and backfilling was ongoing for around 12 hours. Again, no filesystem corruptions were reported after this incident.

3. Ceph for Physics Data

In addition to block storage, we are also investigating how to exploit RADOS within our existing physics data stores [4]. CASTOR version 2.1.15 added support for the RadosStriper; with this feature CASTOR disk servers would act as thin gateways to objects stored in RADOS [4]. Additionally, various strategies for using RADOS are being explored for the EOS Diamond release, included use-cases for both metadata and data [5].

For other corner cases, we have been evaluating an S3 service built with the Ceph RADOSGW daemon. Our deployment uses civetweb and haproxy for horizontal scalability and the most visible user is currently BOINC. We presently store more than 40 million objects behind the Ceph S3 service.

3.1. A 30 Petabyte Test

These physics data use-cases demonstrate that larger Ceph clusters may soon be needed. However, we are not aware of production Ceph clusters above the 3-5 PB size.

In order to evaluate Ceph's scalability, we provisioned 150 servers—each with 48x 4TB drives and only 64GB of RAM[1]—for a short term test [6]. After optimizing our Puppet configuration templates, we succeeded to deploy the cluster within roughly 2 days and then performed various performance tests which demonstrated up to 52 GB/s write performance and little performance degradation during backfilling.

During the testing we exposed one scalability limitation in which ceph-osd processes consume an amount of memory proportional to the number of OSDs in the cluster. Ceph uses an *osdmap* to represent the cluster's structure, including the directory and statuses of OSDs, the CRUSH map, etc... Having 7200 OSDs, the osdmap was 4MB in size, and since each OSD caches up

[1] This is acknowledged to be below the recommended 2GB RAM per OSD.

to 500 previous versions of the osdmap, each OSD consumed 3-4GB RAM. With 48 OSDs per server, this implied that at least 200GB of RAM is needed per server just to cache the osdmaps. We feel this is not an efficient usage of expensive RAM, especially given old osdmaps are rarely used in a healthy Ceph cluster. On the positive side, we found that by decrease several options related to the osdmap cache we were able to decrease memory usage to under 500MB per ceph-osd, thereby achieving a stable cluster. However, we are looking forward to any improved design which would remove this limitation.

4. Summary

This short paper presented the operational status of Ceph in the CERN IT Department. The production cluster is currently 3PB in size and is primarily used for OpenStack Cinder and Glance block devices. This block storage service is proving to be quite popular and indispensable, with close to 1000 virtual machines relying on Ceph to provide attached volumes.

Looking forward, we have ongoing developments in CASTOR and EOS to exploit Ceph object storage for better reliability and scalability. These projects motivated a 30 petabyte Ceph test in order to evaluate the solution at a scale relevant to physics data storage. This test demonstrated that operating Ceph at that scale is feasible with workarounds, however we identified limitations which prevent further scaling in terms of more ceph-osd processes.

References

[1] Weil, Sage et al. Ceph: A scalable, high-performance distributed file system. Proceedings of the 7th symposium on Operating systems design and implementation. USENIX Association, 2006.
[2] Bell, Tim et al. Scaling the CERN OpenStack cloud. CHEP 2015. Okinawa, Japan.
[3] Van der Ster, Daniel. Block Storage Service: Status and Performance. CERN-IT-Note-2014-002. June 2014. *https://cds.cern.ch/record/1971198*.
[4] Mascetti, Luca et al. Disk storage at CERN. CHEP 2015. Okinawa, Japan.
[5] Peters, Andreas Joachim et al. Integrating CEPH in EOS. CHEP 2015. Okinawa, Japan.
[6] Van der Ster, Daniel and Rousseau, Hervé. Ceph 30PB Test Report. CERN-IT-Note-2015-002. May 2015. *https://cds.cern.ch/record/2015206*.