

Old Habits Die Hard

just another scalable weblog

Ceph, a free unified distributed storage system

By Javier on 26 February 2016 11:00 PM | [Permalink](#) | [Comments \(0\)](#)



Over the last few months I have been working in [Ceph](#), a free unified distributed storage system, in order to implement some missing features in [RADOS gateway](#), help some customers with Ceph clusters in production and fixing bugs.

This effort is part of my daily work here in [Igalia](#) working in upstream projects. As you could know, Igalia works in the Cloud arena providing services on development, deployment and orchestration around interesting open projects.

Together with Ceph (storage) we are also working upstream in [Qemu](#) (compute) and [Snabb](#) (networking). All these projects are in the core to create private and public clouds with Open Source.

My goal with this first post is introducing Ceph in a simple and easy way to understand this marvelous piece of software. I will cover the design and main innovations in Ceph together with its architecture, major use cases and relationship with [OpenStack](#) (a well-known free and open-source software platform for cloud computing).

Understanding Ceph

Ceph is an [object storage](#) based free software storage platform that stores data on a single distributed computer cluster. I would say this definition catches the essence of Ceph perfectly. It is also the foundation to understand its innovations, the architecture and the performance/scalability factors in Ceph.

Let's start with the object storage. The object storage is a storage architecture that manages data as objects, as opposed to other storage architectures like file systems which manage data as a file hierarchy and block storage which manages data as blocks within sectors and tracks. Each object typically includes the data, a variable amount of metadata, and a globally unique identifier.

On top of this object storage, Ceph provides a block interface ([RBD](#)), an object interface ([RGW](#)) and a filesystem interface ([CephFS](#)).

If we add a smart cluster approach in the previous design we will have a reliable object storage service that can scales to many thousands of devices. This reliable object storage service is known as [RADOS](#) (Reliable Autonomic Distributed Object Storage) in the current Ceph implementation.

But what is a 'smart cluster approach' here? At the petabyte and exabyte scale, systems are necessarily dynamic. They are built incrementally, they grow and contract with the deployment of new storage and decommissioning of old devices, devices fail and recover on a continuous basis, and large amounts of data are created and destroyed. RADOS takes care of a consistent view of the data distribution and consistent read and write access to data objects.

RADOS also provides storage nodes with complete knowledge of the distribution of data in the systems, devices can act semi-autonomously using peer-to-peer like protocols to self-manage data replication, participate in failure detection and respond to device failures and the resulting changes in the distribution of data by replicating or migrating data objects.

If we consider the minimal configuration together with the basic components needed to set up a RADOS system, we will have a set of object storage daemons ([OSDs](#)) and a small group of monitors ([MONs](#)) responsible for managing OSD cluster membership.



Contact Info

Javier M. Mellid
hacking at Igalia

jmunhoz@igalia.com

Other places

Follow Javier on [twitter](#) if you want to know what he is current reading or thinking about.

Syndication

Subscribe to this weblog's
[atom feed](#) or [rss feed](#)

Archives

[All postings](#)

Tags

[Ceph](#) | [Drones](#) | [Kernel](#) | [Security](#) | [Testing](#)

Recent Entries

[KubeCon / CloudNativeCon Europe 2019](#)
[Cephlocon Barcelona 2019](#)
[Ceph Days Galicia 2019](#)
[RGW/S3 Archive Zone goes upstream in Ceph](#)
[On Ceph RGW/S3 Object Versioning](#)
[Open Source UAV, autopilot integration and service layer](#)
[Open Source UAV, USS client, driver and controller](#)
[Open Source UAV and integration into controlled airspace](#)
[Open Source UAV, UTM, DEM and elevation profile](#)
[Open Source UAV, SITL in Docker, Mission Planner and MAV tools](#)
[Attending Panda Security Summit 2018](#)
[Attending AWS Summit Madrid 2018](#)
[Ceph Day in Santiago de Compostela](#)
[Ceph RGW/S3 demo container technical notes](#)
[Open Source UAV and survival analysis with R](#)

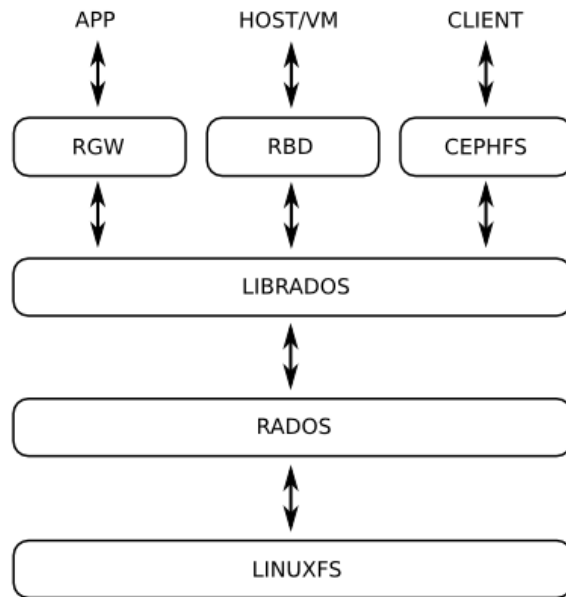
In Ceph this OSD cluster membership requires a [cluster map](#). This cluster map specifies cluster membership, device state and the mapping of data objects to devices. The data distribution is specified first by mapping objects to placement groups (PGs) and then mapping each PG onto a set of devices. The algorithm taking care of these steps is known as [CRUSH](#) (Controlled, Scalable, Decentralized Placement of Replicated Data)

With this information in mind we may consider two major innovations in Ceph RADOS:

- The CRUSH algorithm. The way how Ceph clients and Ceph OSD daemons compute information (hashing function) about object location instead of having to depend on a central lookup table
- Smart daemons. The Ceph's OSD daemons and Ceph clients are cluster aware. This enables OSDs interact directly with other OSDs and MONs. Ceph clients interacts with OSDs directly.

Both items add significant intelligence in the solution to avoid bottlenecks and, at the same time, pursue hyperscale at the petabyte and exabyte scale.

In this point we should have enough information to understand the raw Ceph architecture. Let's have a look in the usual block diagram for Ceph:



- RGW. A web services gateway for object storage, compatible with S3 and Swift
- RBD. A reliable, fully distributed block device with cloud platform integration
- CEPHFS. A distributed file system with POSIX semantics and scale-out metadata management
- LIBRADOS. A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)
- RADOS. A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

Mapping out the major components involved under the hood and their interactions makes it still possible getting a more detailed version of this architecture:

[Open Source UAV and mobile cellular networks](#)

[CVE-2005-3252 - Snort 2.4.0-2 remote code execution](#)

[Attending LibreCon 2017](#)

[Open Source UAV API, DroneKit-Python and Gepppy](#)

[Open Source UAV Autopilot with Ardupilot and Pixhawk](#)

[Building and running RISC-V Linux rev 1.9 on QEMU](#)

[AWS4 browser-based upload goes upstream in Ceph](#)

[CVE-2017-7269 - Binary patch diffing](#)

[CVE-2017-7269 - IIS 6.0 WebDAV remote code execution](#)

[Ceph RGW AWS4 presigned URLs working with the Minio Cloud client](#)

[Multipart Upload \(Copy part\) goes upstream in Ceph](#)

[Attending ApacheCon and Apache Big Data Europe 2016](#)

[AWS4 chunked upload goes upstream in Ceph RGW S3](#)

[Virtual Data and Control Paths in Software-Defined Storage](#)

[Ansible AWS S3 core module now supports Ceph RGW S3](#)

[The Ceph RGW storage driver goes upstream in Libcloud](#)

[Scalable placement of replicated data in Ceph](#)

[The Outscale OSU driver goes upstream in Libcloud](#)

[Requester Pays Bucket goes upstream in Ceph](#)

[AWS Signature Version 4 goes upstream in Ceph](#)

[Ceph, a free unified distributed storage system](#)

[On S3, endpoints, regions, signatures and Boto 3](#)

[Windows 10 Kernel debugging on QEMU](#)

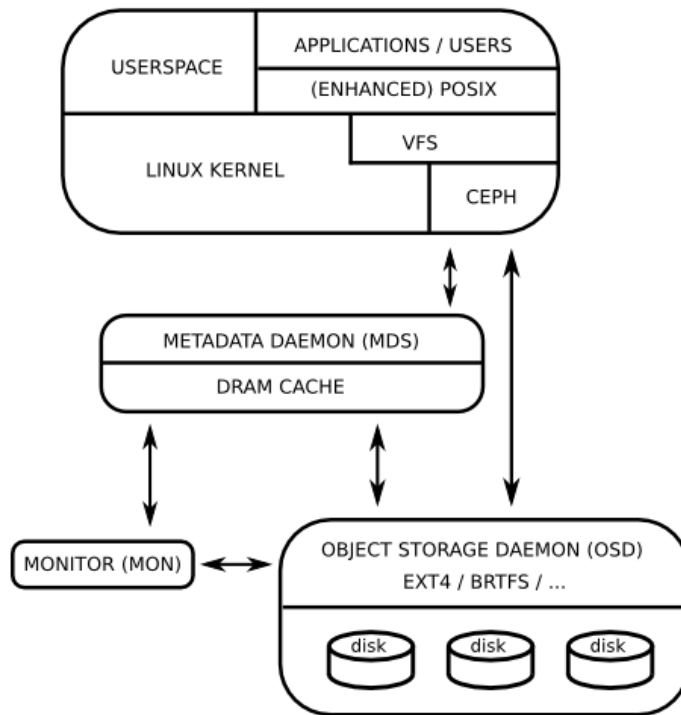
[Pflua and high performance packet filtering](#)

[Visit to INTECO's Cyber-Security Headquarters](#)

[Collaborating with the Carnegie Mellon Software Engineering Institute on browser security](#)

[Detecting and removing computer virus with OCaml](#)

[On Technology of Controls for Accelerators and Detectors](#)



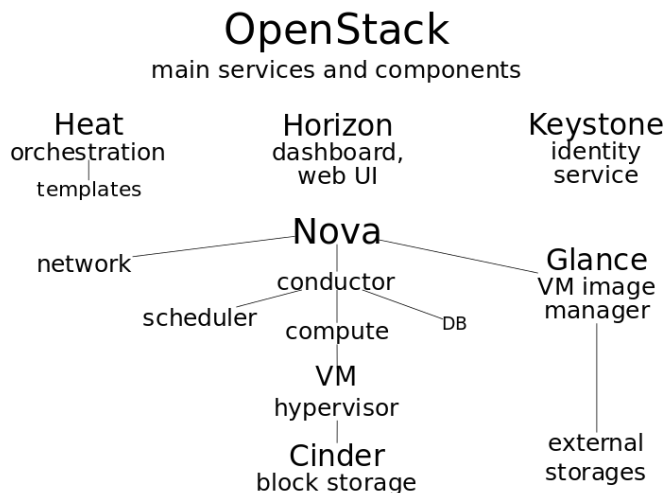
The OpenStack basics

Although this is an introduction post in Ceph I will describe OpenStack and its relationship with Ceph briefly. It will be useful later.

Ceph may be used alone but some of its most interesting use cases take place as part of OpenStack. A quick overview on OpenStack will be useful to understand how the OpenStack and Ceph components work together to provide reliable and scalable storage.

The current stable release for OpenStack is 'Liberty' and it includes 17 components (compute, image services, object store, etc). All those components have well-known code names (Nova, Glance, Swift, etc)

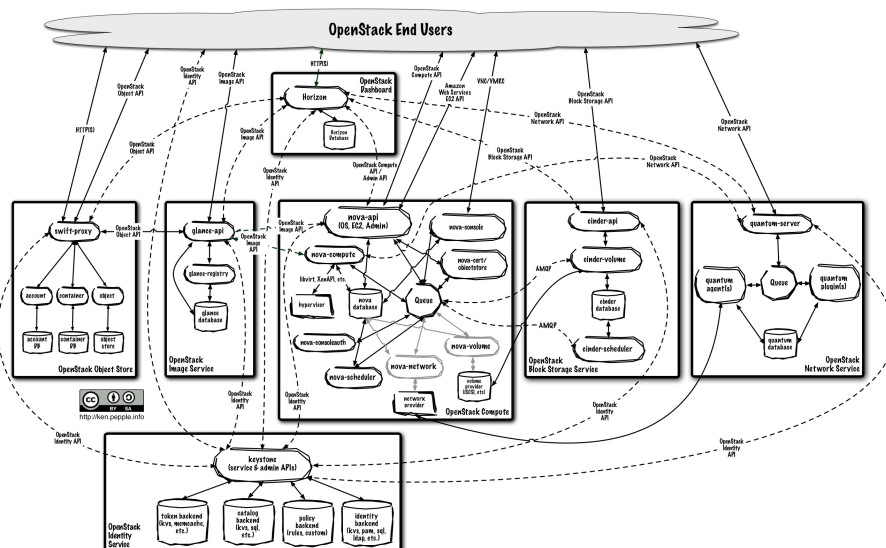
The next picture catches a very high level abstraction for OpenStack:



As you can see, Glance (VM image manager) and Cinder (block storage) are two core services in the solution.

We mentioned the previous picture shows a simple view of OpenStack. A more accurate diagram together with the relationships among the services is available in the next picture for 'Folsom', a previous release (2012)

While OpenStack evolves and include new services, this 'Folsom' picture should be good enough to introduce the services related to storage and the level of complexity of OpenStack.



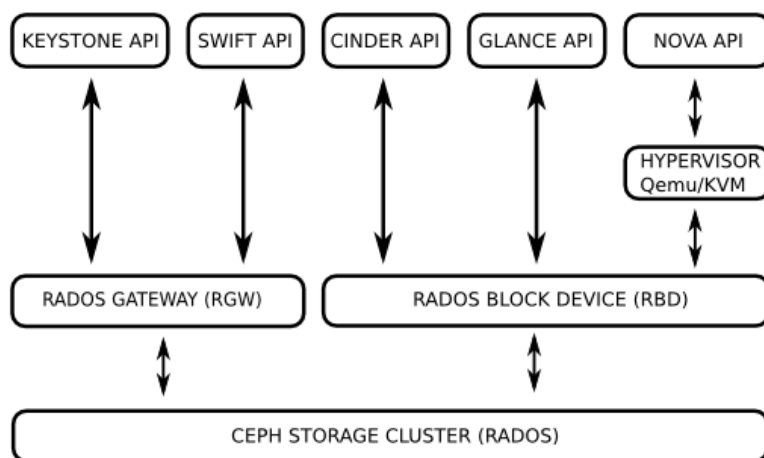
So the storage services in place are Swift (object store service), Glance (image service) and Cinder (block storage service).

Those services work in tandem to cover the general and specific requirements for storage in OpenStack.

Using Ceph in OpenStack

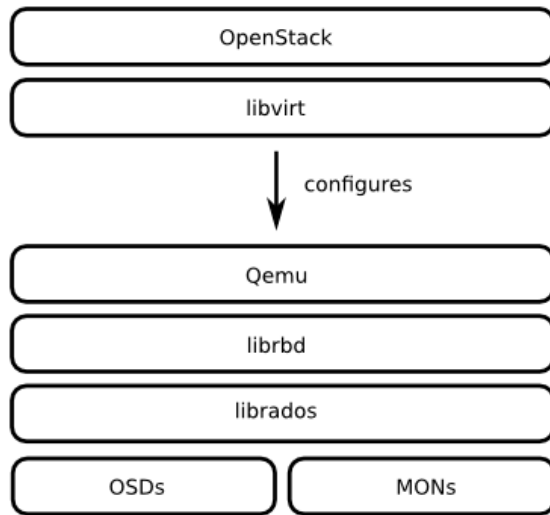
The main integration points between OpenStack and Ceph are the object and block device interfaces.

The RADOS gateway (RGW) and the RADOS block device (RBD) interfaces are used to provide the required storage to 5 services (Keystone, Swift, Cinder, Glance and Nova)



It is worth mentioning the compute service (Nova) interfaces the RBD layer via a hypervisor. An open source hypervisor working like a charm with Ceph is Qemu/KVM. It uses librbd and librados.

Other component to mention in the stack is libvirt. OpenStack uses libvirt to configure Qemu/KVM properly.



Ceph RBD dominates the choice for Cinder drivers currently, as stated in the [sixth public survey of OpenStack users](#) (page 31)

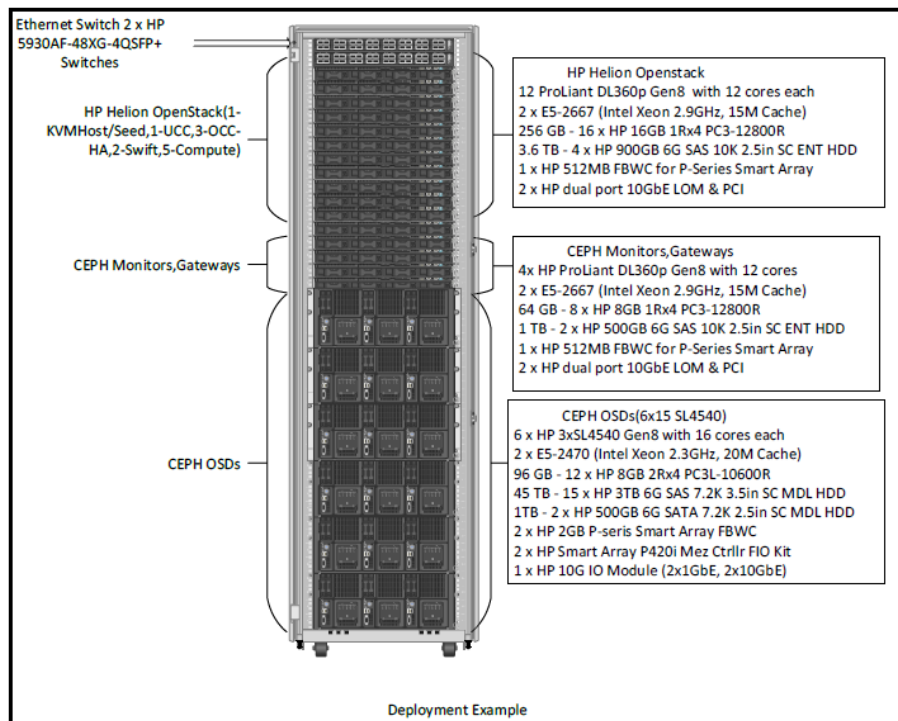
The physical deployment of Ceph and OpenStack

Setting up and operating a reliable and scalable storage cluster is always demanding. It requires a careful planning along many different aspects. Some of these critical decisions are related to the cluster capacity (RAM, disks, number of nodes, use profiles, etc)

Although it is always possible going with your own custom configuration some hardware providers offer several standard configurations.

As a random and arbitrary example, we can have a look in the HPE Helion portfolio. This set of solutions is a mix of open-source software and integrated systems for enterprise cloud computing.

The next picture shows the physical space required and how it compares to the different logical components in the architecture.



The new and old use cases

The production of data is expanding at an astonishing pace. Two major drivers in this rapid growth of global data are the analog-to-digital switch (software is everywhere) and the rapid increase in data generation by individuals and companies.

The new use cases related to storage nowadays are radically different of the previous ones a few years ago. These new use cases are all about storing and retrieving unstructured data like photos, videos and social media in massive scale. All this stuff requires real-time analytics and reporting together with efficient processing.

To get these requirements together, some companies are extending/migrating their current datacenters to support software-defined approaches. As consequence, those new datacenters leverage virtualization concepts such as abstraction, pooling, and automation to all of the data center's resources and services to achieve IT as a service. In this vision all elements of the infrastructure (compute, storage, networking and security) are virtualized and delivered as a service.

In this context, we can identify some new and well-known use cases along the next 5 different categories. The original classification is used by the [RedHat Storage](#) team. Take into consideration I am merging Cloud infrastructure and Virtualization here.

- **Big data analytics.** Storing, integrating, and analyzing data at petabyte scale
- **Cloud infrastructure and Virtualization.** Virtual machine storage and storage for tenant applications (Swift/S3 API)
- **Rich media.** Massive scalability and cost containment (scaling out with commodity hardware)
- **File sync and share.** Secure mobility, collaboration and the need for anytime, anywhere access to files
- **Archival data.** Agile, scalable, cost-effective and flexible unified storage (objects, blocks and file systems)

Ceph is used to support all these use cases in production with great results.

Pushing Ceph to the limit

Some folks in the [CERN IT Department](#) are pushing Ceph to the limit. They use Ceph as part of an OpenStack deployment and I have to say the numbers are great.

The solution is a large distributed OpenStack infrastructure with around 10,000 VMs and 100,000 CPU cores (1000 Cinder volumes and 1500 Glance images). The Cloud is predominantly used for physics data analysis but they also reported on a long tail of conventional IT services and user-managed application VMs.

If you want to know more on this Ceph cluster operated by CERN, I would recommend to watch this [video](#) at Vancouver Summit 2015.

In brief, and beyond of the great insights shared along the talk, the current Ceph version scales out to 10 PB. In that scale it just works. Over that threshold, it requires extra configuration adjustments.

Wrap-up!

I told you! This piece of software is marvelous!

I plan to add new blog entries to cover some of the new features implemented in the previous months. They are upstream code now so you will be able to enjoy them in Jewel!

If you are looking for some kind of support related to development, design, deployment, etc. in Ceph or you would love to see some new feature in the next releases. Feel free to contact me!

Related posts

- [AWS Signature Version 4 goes upstream in Ceph](#)
- [On S3, endpoints, regions, signatures and Boto 3](#)

[« On S3, endpoints, regions, signatures and Boto 3](#) [AWS Signature Version 4 goes upstream in Ceph »](#)

Comments

0 Comments

Old Habits Die Hard

 Login ▾ Recommend Tweet Share

Sort by Best ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

ALSO ON OLD HABITS DIE HARD

Old Habits Die Hard - ClamAV ISO 9660 built-in support

2 comments • 7 years ago



jmunhoz — Hi Sergio, I released this patch for devel-20060419 eleven months ago but it isn't officially supported by the ClamAV

Old Habits Die Hard - security workshop slides

1 comment • 7 years ago



tsao — Thanks to you for your speech. You talked in a not usual way about *privative* and free antivirus. We enjoyed a lot! :-)

Old Habits Die Hard - AWS4 chunked upload goes upstream in Ceph RGW S3

3 comments • 3 years ago





Javier — The default chunk size is client side stuff. You need to configure it via API if possible. It is usually hardcoded in source

Old Habits Die Hard - VAX virtual bare-metal programming

1 comment • 6 years ago



KeepOnLearning — Notice this old DEC logo was usurped by the set designers of "The Newsroom" as the logo for the series'

 Subscribe  Add Disqus to your siteAdd DisqusAddGenerated by [Jekyll](#)