Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

The Study of Mixed Storage Scheme of Private Cloud Platform Based on Ceph

Zuoyang Qu [a)], Chenhui Xie [b)], Chengliang Liu [c)]

China Academy of Railway Sciences Corporation Limited, Beijing 51849150, china.

[a)] quzuoyang@rails.cn, [b)] xiechenhui@rails.cn, [c)] liucl@rails.cn

Abstract. Ceph is a unique unified system that supports object storage, block storage, and file storage with the characteristic of high available, manageable, and free. To build a distributed storage system based on Ceph in a private cloud environment with a high level of data privacy, it can support object storage, block storage, and file storage simultaneously. Ceph FS and MDS in the current community version are friendlier to ecological environment of OpenStack. However, community versions of Ceph have many problems with their actual use. This article discusses a hybrid storage solution that can be used in a real environment. Based on Ceph's original framework, it adds some extensions to make the native Ceph more suitable for the enterprise's production environment. There is a very attractive solution by using Ceph as a back-end storage for a private cloud platform whether consider about cost and business considerations.

Key words: Private Cloud, Backend Storage, OpenStack, Ceph.

ORIGINAL STORAGE DESIGN FOR OPENSTACK

OpenStack, the most popular open-source cloud platform, is an important solution for enterprises to implement private cloud platforms to provide IaaS-style services [1]. There are many interdependent components that are included in OpenStack. For example, each component depends on Keystone. Nova also relies on Glance, Neutron, and Cinder. Among these components, Swift, Glance and Cinder are required for backend storage support.

Unfortunately, there is no support for unified storage in original OpenStack, and it is different from the backend storage of cloud hosting service (Nova), mirroring service (Glance), and cloud hard disk service (Cinder), which is cause the severe internal power consumption. For example, it is usually takes one to three minutes when creating a virtual machine. This kind of design lacks reasonable lateral expansion and various problems will inevitably occurs. when the system pressure increases. Therefore, the storage must be redesigned, when building a cloud platform. All data from cloud platform is concentrated in the Ceph's resource pool, whilst the operations can avoid unnecessary data transmission, such as creating virtual machines, migrating, expanding, shrinking, etc.

DISTRIBUTED STORAGE KEY TECHNOLOGY IN CEPH

Ceph is a unified and distributed storage system which is designed for superior performance, reliability, and scalability. It can provide three functions of object storage, block storage, and file storage system at the same time through a storage system, so as to simplify deployment and operating maintenance on the premise of meeting different application requirements. The object storage is compatible with Amazon's S3 and OpenStack Swift, it can be accessed by call in Ceph library through C, C++, Java, Python, PHP, or store data in the form of objects through the Restful gateway. Block storage can be mounted directly like a hard disk. File system mounted like a network file system and compatible with POSIX interfaces. Distributed storage system for Cech [2] means decentralized structure, and there is no theoretical upper limit in systematic scalability.

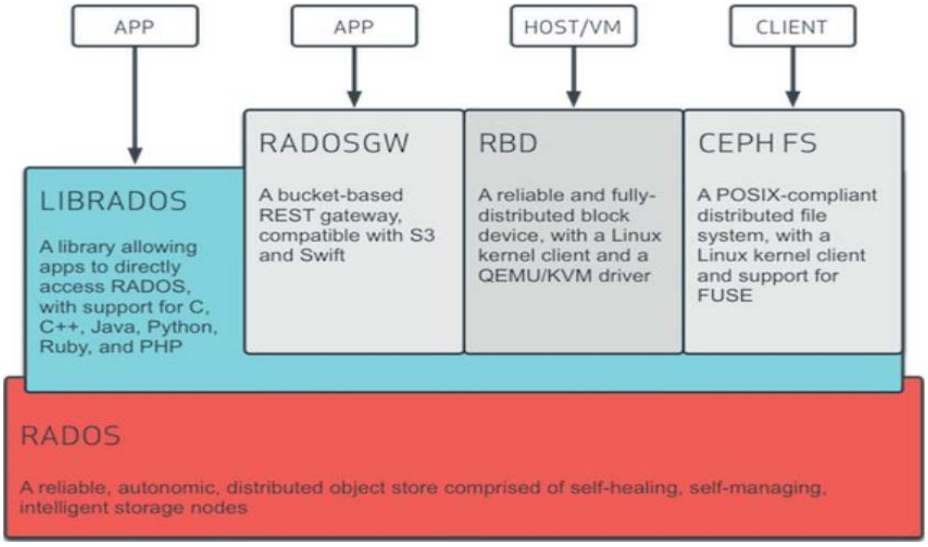

FIG 1. System level of ceph

With the architecture for Ceph, object storage provided by LIBRADOS and RADOSGW, RBD work for block storage, CEPH FS provides a file system, whilst LIBRADOS interface for its call under the three, and it is ultimately stored in RADOS [3] as objects. There are three roles that Ceph cluster node can bear:

Monitor: Monitor the health of the cluster and send the latest CRUSH map to the client (Contains the topology of the current network).

OSD: maintaining objects on the node, respond to client requests and synchronize with other OSD nodes.

MDS, providing metadata for the file and providing high-level application interface Ceph FS [4].

As a distributed storage, files in Ceph have been divided evenly and randomly on each node. It uses the CRUSH algorithm to determine the storage location of the object. Ceph client can directly calculate the file storage location as long as it knows current cluster topology. The client communicates directly with the OSD node to obtain the file location without having to ask the central node, thus avoiding the single point risk. Ceph is a relatively mature storage system currently. It is an ideal storage backend for OpenStack and can also be used as a storage backend for Hadoop.

Ceph and Gluster are flexible storage systems that performed excellent in cloud environments [5]. The reason why we chosen Ceph is not only it's easier to integrate with Linux and friendlier to Windows, but also the different ways of its accesses storage is more likely make it to a more popular technology. Ceph is already a part of the mainline Linux kernel (2.6.34), and it will be gaining more attention as an excellent open source project as its high performance, reliability, and scalability.

PRIVATE CLOUD BACKEND STORAGE TO INTEGRATE CEPH

There are two approachs to apply Ceph to the private cloud (based on OpenStack) backend storage. One is to use Ceph to provide block storage for OpenStack, while still use Swift to provide object storage. Since the private cloud contains the Swif assembly as an object store [6], Ceph is written in C++ and Swift is written in Python, so Ceph have dominant performance. But unlike Ceph, Swift is focused on object storage, which was integrated well with OpenStack and validated by mass production practices as one of the OpenStack components [7]. The other way is that the storage backend of Nova/Glance/Cinder is provided by Ceph, and there is no data transmission between these three modules, thus only need to manage a unified storage when quickly creating a virtual machine. Here we take the second method to further exploration [8].

Experimental Environment

Integrated the experimental resources we can use at the moment to build experimental clusters [9]. The overall cluster deployment distribution is shown in the FIG2, some physical storage nodes in the cluster are equipped with 3.2 TB NVME SSD cards, which is support PCIe 3.0 and above interfaces, NVMe 1.2, stable 4K IOPS (read no less

than 800,000, write no less than 150,000), and 4K time delay (read no more than 90μs, write less than 20μs). Its MTBF ≥ 200W hours, and its annual failure rate ≤ 0.44%. The standard configuration of all storage nodes is two 2.5-inch 600GB 10K SAS hot-swappable hard drives, two 480GB SSDs (intel S3520 series) hot-swappable hard drives, eight 6T 7.2K SATA hot-swappable hard drives, and RAID cards with ≥1GB cache as standard. they support RAID0,1,5,6,10,50,60, it is configured power-failure protection, which can expand ≥ 16 hot-swappable 3.5-inch disk slots.

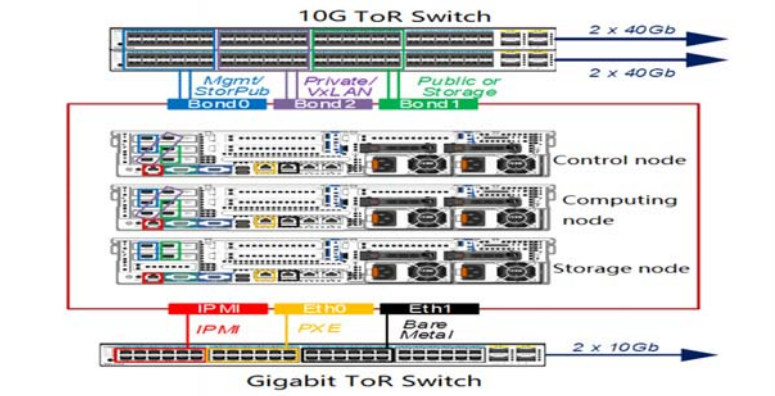

FIG 2. The deployment of experimental cluster

Used CentOS Linux release 7.3.1611 (Core) as operating system, the host cluster deploys PXE network of enterprise private cloud based on the M version of OpenStack's in-depth developed by deploying nodes. The deployed network architecture is shown in FIG 3.

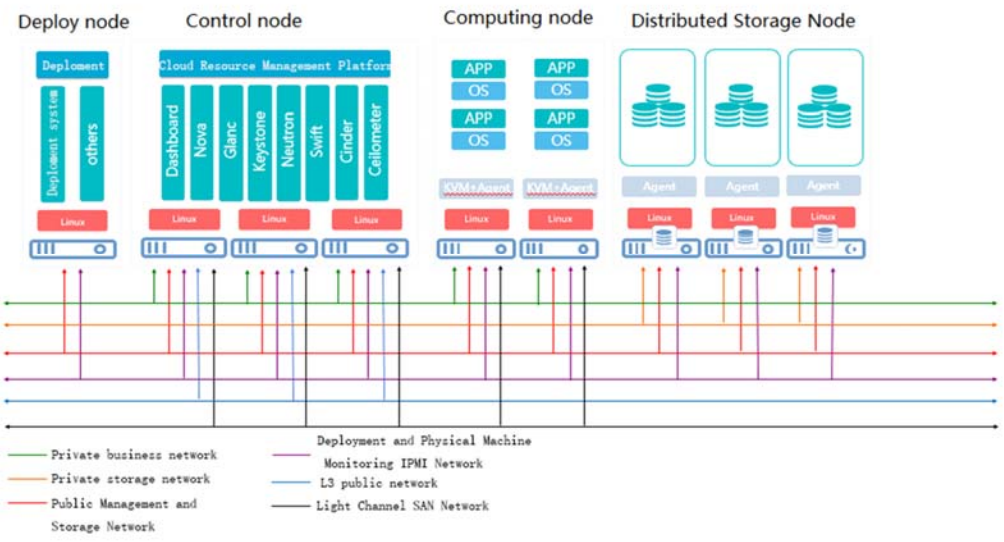

FIG 3. The architecture of cluster deployment network

Control node: storage public-Dual 10G, L3 Public-Dual Gigabit, business Private-Dual Gigabit, deployment and IPMI-Single Gigabit, management-Dual Gigabit (Management network and storage Public network can be merged).

Computing nodes: Storage Public-Dual Gigabit, Business Private-Dual Gigabit, Deployment and IPMI-Single Gigabit, Management-Dual Gigabit (Management network and storage Public network can be merged).

Storage nodes: Storage Public-Dual Gigabit, Storage Private Network-Dual 10 Gigabit (IB available), Deployment and IPMI-Single Gigabit, Management-Dual Gigabit (Management network and storage Public network can be merged).

Community Ceph Supports Private Cloud Backend Storage

Used Ceph FS as the local file system for Nova nodes. As a shared instance storage in OpenStack, Ceph block device mirroring which is treated as a cluster object can be used in OpenStack, and OpenStack Glance can also be used to store image in Ceph block device. Accordingly, there was no data transfer between OpenStack's Nova Glance and Cinder, and high availability clusters only need to manage a unified storage [10]. Here we used Ceph version 10.2.5.

The Ceph cluster in this study have total capacity of 479TB with three copies, which have available capacity of approximately 159TB. We used two SSD disks to record the log in all nodes. Each SSD disk was divided into 4 areas with each partition capacity was 40G, and each SSD partition corresponds to an OSD. There are three pools for Ceph, which is named image, volumes and backups, and each pool with three copies. Owing to there are two different storage mediums for the host node, we divided the SSD storage and mechanical hard disk into many different resource pools for testing to maximize the effectiveness of the hardware. According to the official recommended architecture [11], we did the following fusion to modify the configuration files in the /etc directory of the OpenStack control node. The fusion architecture is shown in FIG.4.

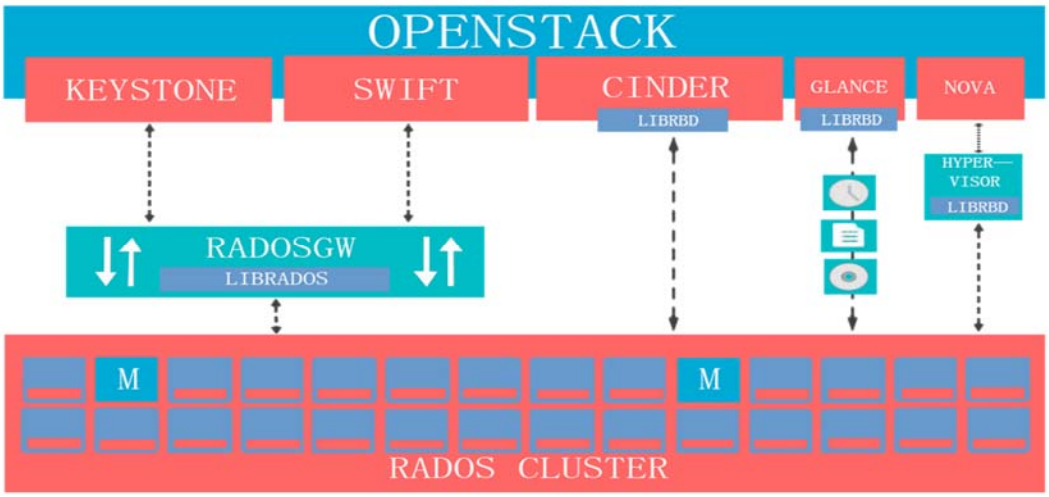

FIG 4. Fusion architecture

First, we instantiated a host on a private cloud, which distribution was Ubuntu14.04 LTS amd_64. After that, two 80G cloud disks are instantiated in Ceph‘s pool on two different storage media. Started the cloud host and mount two cloud drives respectively for the FIO test. After mounted the cloud disk from the SSD pool to the host, we used the command fdisk -l to display the path as /dev/vdb. The fio test with read/write mixed mode was as follows:

fio -filename=/dev/vdb -direct=1 -iodepth 1 -thread -rw=randrw -rwmixread=70 -ioengine=psync -bs=16k -size=80G -numjobs=30 -runtime=100 -group_reporting -name=ssd

Unmounting the cloud disk and mount the cloud disk from the mechanical hard disk pool, repeat the above steps and read and write mixed mode FIO test. Then uninstall the cloud hard disk, log off the instance and re-apply, repeat the above steps 6 times. The results are shown in TAB 1, TAB.2.

TAB 1. The results of community edition ceph-fio-ssd pool

Key indicators		Ceph-SSD POOL					
		1	2	3	4	5	6
Read	bw(KB/s)	330136	365295	332364	335429	329473	302375
	iops	20633	20453	21023	20868	20089	22186
Write	bw(KB/s)	141713	146601	145744	139912	144320	139847
	Iops	8857	9012	8463	8356	8217	8541
Hard disk usage		100%	100%	100%	100%	100%	100%

TAB 2. The results of community edition ceph-fio-sata pool

Key indicators		Ceph-SATA POOL					
		1	2	3	4	5	6
Read	bw(KB/s)	161171	159791	166523	162351	160716	163224
	iops	10073	10000	10524	10740	10221	10092
Write	bw(KB/s)	68983	69300	68172	70778	69862	68051
	Iops	4311	4326	4278	4390	4328	4291
Hard disk usage		100%	100%	100%	100%	100%	100%

Follow the results above, the SSD resource pool reads and writes about twice as fast as the SATA resource pool by using community edition Ceph pooling resources.

Optimize Ceph Support for Private Cloud Backend Storage

From the above results we can see there is some performance loss with the community version of Ceph. According to industry statistics, if the community version of Ceph does not have a good operation and maintenance development team, the number of storage nodes can hardly exceed 20 nodes. There is a lot of room for improvement in network communications, thread scheduling, memory management, etc.

FileStore technology will be used when Ceph reads and writes data. Writing the log first and then writing the data causes double writes while writing the data block, resulting in a great performance penalty. But if we can separate metadata from data, only metadata writes logs, which can improve efficiency. Another improvement is hotspot read-ahead cold-pool sleep. In order to improve overall efficiency, we storing hot data with high frequency of access in high-speed media, it gradually drops in the mechanical hard disk as the heat drops, at the same time, control the cold storage pool hardware, reduce energy consumption and increase disk life. We also optimize network communications by aggregating TCP links. By the way, it is also important to support data interface, like FC and ISCSI, supports link redundancy to ensure the security of service links. We also added compression and disaster recovery strategies. Including 1-6 data copies, erasure codes and other different strategies. The improved architecture is shown in FIG 5. The red label is the optimization section.

Test according to the test procedure described in section 3.2.

Through the above tests, we can see that the efficiency of the optimized Ceph is significantly higher than that of the community version of Ceph. According to our tests, it can increase the original efficiency by 130%.

SUMMARY

In general, Ceph is for large-scale storage applications to solve complex situations of various application types of enterprises and requires professional technical service teams to provide technical support. Unfortunately, Small and medium-sized enterprises do not have so much data to store and there is not enough money for professional technical service teams to operate and maintain. Therefore, an ideal scenario, would be use NFS or ISCSI to store virtual machine images or as an additional volume of virtual machines for use with OpenStack.

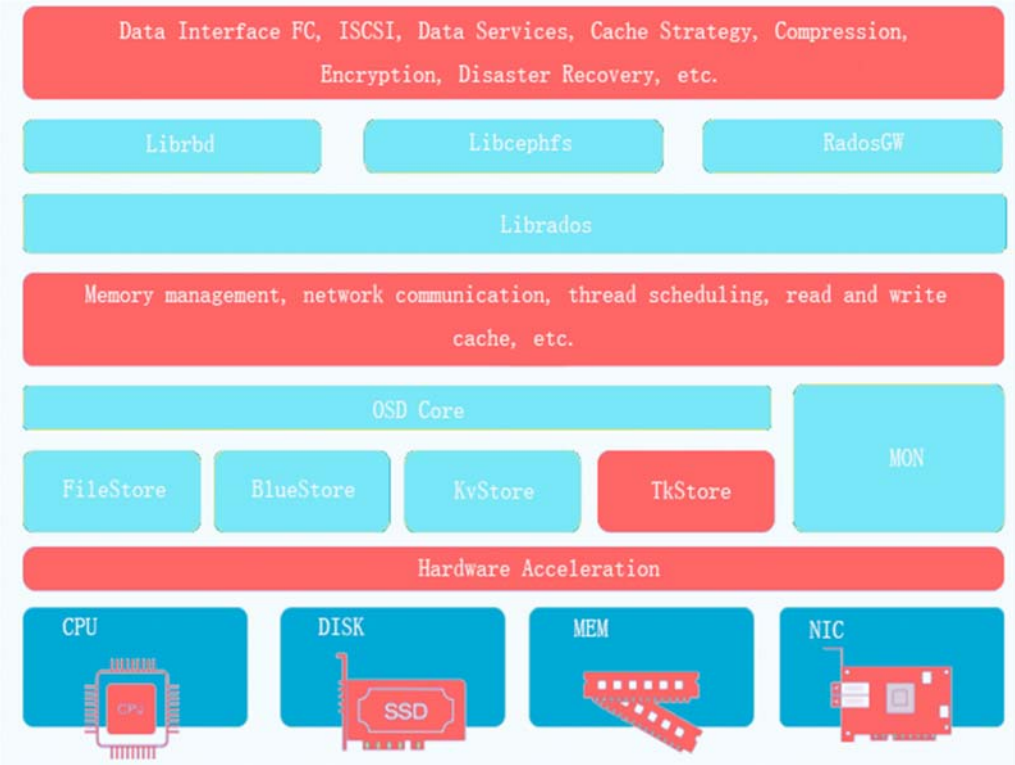

FIG 5. Optimization about ceph

TAB 3. The test results of optimized ceph-fio-ssd pool

Key indicators		Optimized Ceph-SSD POOL					
		1	2	3	4	5	6
Read	bw(KB/s)	436279	436943	439025	435619	439051	434190
	iops	27267	27666	29125	28223	28416	27191
Write	bw(KB/s)	212246	214946	212037	211598	214549	211031
	Iops	13266	13434	13252	13224	13409	13189
Hard disk usage		100%	100%	100%	100%	100%	100%

REFERENCES

1. Zhenhua Xiong. Cloud Storage Technology Base on OpenStack[D]. Jilin University,2014.
2. Weil SA, Brandt SA, Miller EL, et al. CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data. SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing .Tampa, FL, USA.2006:31.
3. Weil SA, Leung AW, Brandt SA, et al. RADOS: a scalable, reliable storage service for petabyte-scale storage clusters[C] International Petascale Data Storage Workshop. DBLP, 2007:35-44.
4. Weil S A, Brandt S A, Miller E L, et al. Ceph: a scalable, high-performance distributed file system[C] Symposium on Operating Systems Design and Implementation. USENIX Association, 2006:307-320.
5. Johanes J, Johari MF, Khalid M, et al. Comparison of Various Virtual Machine Disk Images Performance on GlusterFS and Ceph Rados Block Devices. International Conference on Informatics Applications. 2014.
6. Azagury A, Dreizin V, Factor M, et al. Towards an Object Store[C] MASS Storage Systems and Technologies. IEEE, 2003:165-176.
7. Wei Kong, et al. Multi-level image software assembly technology based on OpenStack and Ceph[A]. IEEE Beijing Section,Global Union Academy of Science and Technology, Chongqing Global Union Academy of Science and Technology. Proceedings of 2016 IEEE Information Technology, Networking, Electronic and

Automation Control Conference (ITNEC 2016) [C]. IEEE Beijing Section, Global Union Academy of Science and Technology, Chongqing Global Union Academy of Science and Technology:2016:4.
8. Gudu D, Hardt M, Streit A, et al. Evaluating the performance and scalability of the Ceph distributed storage system[J]. Big Data (Big Data), 2014 IEEE International Conference on, Washington, DC, USA ,2014:177-182.
9. Tiezhu Zhao. Research on Performance Modeling and Application of Distributed File System[D].South China University of Technology 2011.
10. Xiang Li. Research and Performance Testing of the Ceph Distributed File System[D]. XIDIAN University, 2014.
11. Bin Wang. Application of Ceph Storage on the Railway Station-Train Wi-Fi System [A]. The Eleventh China Intelligent Transportation Conference Proceedings [C]. Intelligent Transportation Association China,2016:8.