Implementation of a Software-Defined Storage Service with Heterogeneous Storage Technologies

Chao-Tung Yang[1]*, Shuo-Tsung Chen[2], Wei-Hsiang Lien[1], and Vinod Kumar Verma[3]
[1]Department of Computer Science, Tunghai University, Taichung, Taiwan R.O.C.
[2]College of Engineering, National Yunlin University of Science and Technology, Taiwan R.O.C.
[3]Department of Computer Science and Engineering Deemed University
under Ministry of Human Resource Development,
Government of India Longowal, Sangrur District, India

Abstract

With the popularity of cloud computing, the demand for cloud infrastructure also increases. Because the requirement of cloud computing increases steeply, the infrastructure of cloud service expands too. Spending the least cost to achieve the best effect on the cloud is a big problem for every researcher, and virtualization is one of the answers to resolve this problem. Virtualization is a technology that divides the real resources into logically manageable resources, which are then provided to the users to control and manage them efficiently and legitimately. Software-Defined Storage (SDS) is a kind of virtualization which integrates the storage resources and different storage devices by software to increase the usability and activity. According to user requirements, SDS can be adjusted to achieve the optimal performance. In recent years, SDS becomes more and more popular, and several companies have announced their product. However, the generic standard still has not appeared; most products are only appropriate for their devices, and SDS can integrate a few storages. In this paper, the OpenStack is adopted to build and manage the cloud service, and software is used to integrate storage resources including Hadoop HDFS, Ceph, and Swift on OpenStack to achieve the concept of SDS. In such software platform, different storage devices are harmonized to provide an integrated storage array and build a virtual storage pool; so that users do not feel restrained by the storage devices. The software platform also provides a web interface for managers to arrange the storage space, administrate users, and configure security settings. For allocation of the storage resources, we make a policy and assign the specific storage array to the machine that acquires the resources according to that policy. Then, the performance tests for file systems are conducted to prove the system runs correctly. From the experimental results, it is shown the performance of Hadoop HDFS is better than the other two storages; i.e., except a few instances, Hadoop HDFS outperforms the other two storages in the same environment.

Keywords: cloud service, virtualization, software-defined storage

1 Introduction

Cloud computing, one of today's hottest topics, is being developed very fast. There are many large companies such as Google, Amazon, and Yahoo offering cloud services to millions of users at the same time. A lot of people including programmers in the enterprise, government, and research centers, or even some common people, want to build cloud computing environments. But the operating cost of a cloud may be very expensive; the cost includes acquirement and maintenance of hardware facilities, and salaries paid to operating engineers and so on. So how to reduce the cost is a big issue.

Journal of Internet Services and Information Security (JISIS), volume: 9, number: 3 (August, 2019), pp. 74-97
*Corresponding author: Department of Computer Science, Tunghai University, No. 181, Sec. 3, Taichung Port Rd., Situn District, Taichung City 40704, Taiwan R.O.C., Email: ctyang@thu.edu.tw, Tel: +886-4-23590121 (ext: 33804)

Cloud computing is a concept, in which computers over networks are able to cooperate with each other to provide far-reaching network services [7, 26, 8, 13, 23]. The basic approach of cloud computing is to computing through terminal operations over the Internet by moving hardware, software, and shared information from users to the server side. Due to this situation, we can say that we are living in a world that is surrounded by the cloud technology, and we almost use the cloud every day. Now, more and more enterprises join the procession of cloud service providers. There are many kinds of cloud services in the market. When a company wants to build a cloud, a difficult problem it must face is how to choose the right cloud product to suit its need. Software-Defined Storage (SDS) is a kind of virtualization which integrates the storage resource and different storage device by software to increase the usability and activity; the users can adjust SDS according to their requirement to achieve the optimal performance. In recent years, SDS is an increasingly popular concept, and several companies have announced related products. But the generic standard still has not formed; hence, most of its products are only appropriate for some devices.

In this paper, we implement a cloud service to integrate several heterogeneous storage technologies. In order to achieve the goal of high availability, we choose the technologies with a redundant mechanism, in which crash of machines is considered a normal situation and mechanisms exist to resolve it. Additionally, the open source technology is adopted to let the system have good compatibility. It can also help the users to freely build their systems with high flexibility. These features are presented in the design concept of Software-Defined Storage.

This paper is organized as follows. Section 2 describes background information and related works including cloud computing, virtualization, and software-defined storage. Section 3 introduces not only experimental environment and methods, but also the overall architecture. Section 4 lists the hardware information measurement method and software version while presenting and analyzing experimental results. Finally, section 5 summarizes this paper by pointing out its major contributions and future work directions.

2 Background Review and Related Work

In this section, the related technologies and related works used in this system is presented.

2.1 Cloud Computing

Cloud computing is Internet-based computing, in which shared hardware, software resources, and data can be provided to users according to their demand of computers and other devices. The user does not need to know the details of the cloud infrastructure, or has appropriate expertise, and is without direct control. Cloud computing allows companies to fast deploy applications and reduce the complexity of management and maintenance costs, and enables rapid changes of IT resources reallocation to respond business needs. Cloud computing, usually involving with the Internet, describes a new Internet-based services to increase IT use and delivery models, and is easy to provide dynamic and virtual extension of the resource. The cloud is a metaphor of networks or the Internet. Cloud computing can be considered to include the following levels of service: infrastructure as a service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)

Cloud computing relies on the sharing of resources in order to achieve economies of scale in similar infrastructure. Service providers integrate a large number of resources for use by multiple users. Users can easily request more resources and adjust usage at any time, without the need to release all resources back to the whole structure. Users do not need to buy a lot of computing resources; they only need to enhance the amount of rent for short-term spikes of resource demands, and release some resources during

low demand periods. Service providers are able to lease unused resources to other users, and even adjust rents in accordance with the demand of the whole.

2.2 Virtualization

With virtualization, the computer's physical resources, such as servers, network, memory, and storage, are abstracted after conversion, so that users can apply those resources in a better way than the original configuration. Virtualization is commonly referred to virtualized resources including computing power and data storage; in this thesis virtualization is specifically referred to virtualization of servers [17, 11, 24, 21, 10]. The server virtualization software technology refers to the use of one or more of the host hardware settings. It has the flexibility to configure the virtual hardware platform and operating system, like real hardware. In this way, a variety of different operating environments (for example, Windows, Linux, etc.) can operate simultaneously on the same physical host, and be independently operating in different physical hosts. Virtualization solutions can be broadly divided into three categories: full virtualization, para-virtualization, and hardware-assisted virtualization.

2.2.1 Storage Virtualization

After the server virtualization, the interactive mode of the server and storage device changed from one to one to the same application with several VMs to one storage device. It causes read/write of data more complicated like Figure 1. In addition to the application which has to request immediately such as DB or ERP, there are more and more new type of cloud and big data applications to be released like Node.js, Phthon and Hadoop and so on. Because the appearance of these new type technology, the ability of storage device to support them is challenged time and again. Additionally, there are several kinds of different architecture and medium of storage. For instance the new architecture like NAS and SAN or the medium like SATA, SAS, SSD and many other storage architecture developed for virtualization; the more and more new technologies let situations of data services become more complicated [19, 20].

2.2.2 Software-Defined Storage

Software-Defined Storage (SDS) is a term for the computer data storage technology which separates storage hardware from the software that manages the storage infrastructure. The software enabling SDS environments can provide policy management for feature options such as deduplication, replication, thin provisioning, snapshots and backup[27, 16, 4].

Characteristics of SDS could include any or all of the following features[15, 6]:

- Abstraction of logical storage services and capabilities from the underlying physical storage systems, and in some cases, pooling across multiple different implementations. Since data movement is relatively expensive and slow compared to compute and services (the ”data gravity” problem in infonomics), pooling approaches sometimes suggest leaving it in place and creating a mapping layer to it that spans arrays. Examples include

- Automation with policy-driven storage provisioning with service-level agreements replacing technology details. This requires management interfaces that span traditional storage array products, as a particular definition of separating the ”control plane” from ”data plane,” in the spirit of OpenFlow. Prior industry standards efforts include the Storage Management Initiative – Specification (SMI-S) which began in 2000.

- Commodity hardware with storage logic abstracted into a software layer. This is also described as a clustered file system for converged storage.

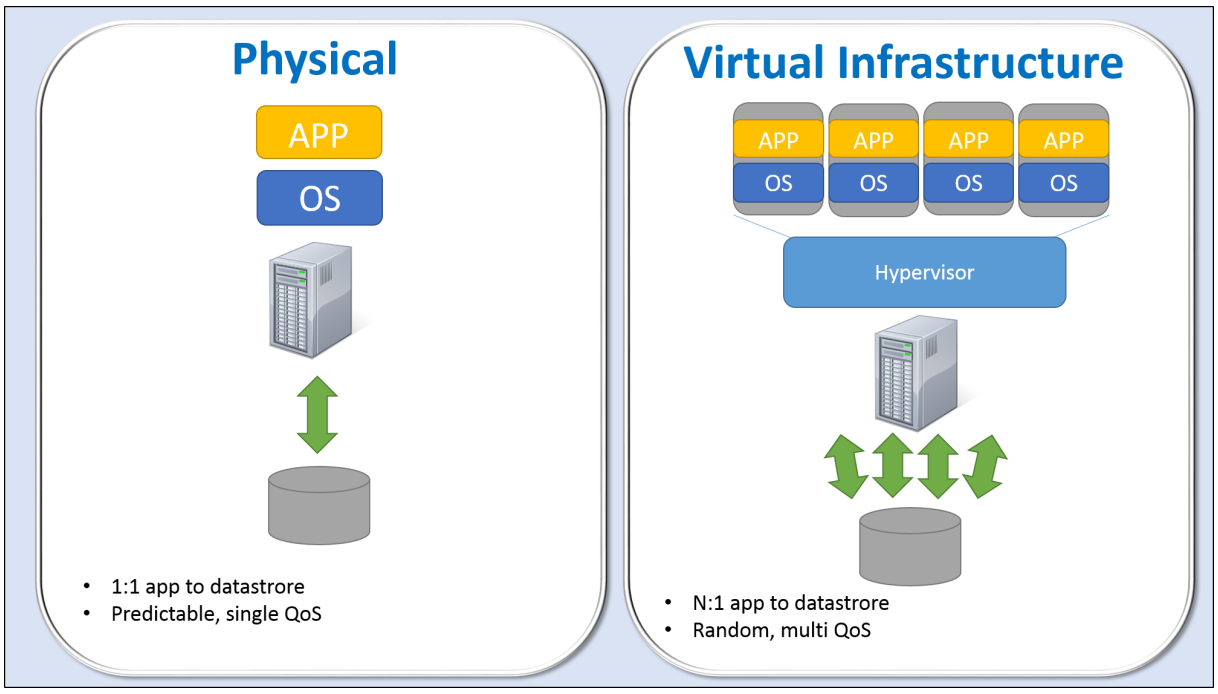


Figure 1: the storage situation compare from the before and now

- Scale-out storage architecture.

2.3 OpenStack

OpenStack[2] is an IaaS cloud computing project for public and private clouds. It is free open source software released under the terms of the Apache License. The project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and features rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution. Founded by Rackspace Hosting and NASA, OpenStack has grown to be a global software community of developers collaborating on a standard and massively scalable open source cloud operating system. Its mission is to enable any organization to create and offer cloud computing services running on standard hardware. The project is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote, protect and empower OpenStack software and its community.

OpenStack offers flexibility and choice through a highly engaged community of over 6,000 individuals and over 190 companies including Rackspace, such as Intel, AMD, Canonical, SUSE Linux, Inktank, Red Hat, Groupe Bull, Cisco, Dell, HP, IBM, NEC, VMware and Yahoo. It is portable software, but is mostly developed and used on operating systems running Linux.

2.4 Swift

OpenStack Object Storage (Swift) is a scalable redundant storage system. Objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster. Storage clusters scale horizontally simply by adding new servers. Should a server or hard drive fail, OpenStack replicates its content from other active nodes to new locations in the cluster. Because OpenStack uses software logic to ensure data

replication and distribution across different devices, inexpensive commodity hard drives and servers can be used.

2.5 HDFS

The Hadoop Distributed File System (HDFS) is an Apache Software Foundation project and a subproject of the Apache Hadoop project. HDFS is a distributed file system designed to hold very large amounts of data (terabytes or even petabytes), and to provide high-throughput access to this information. Files are stored in a redundant fashion across multiple machines to ensure durability to failure and high availability to parallel applications [29]. HDFS has many similarities with other distributed file systems, but is different in several respects. One noticeable difference is HDFS's write-once-read-many model that relaxes concurrency control requirements, simplifies data coherency, and enables high-throughput access [28, 12, 18, 5]. Another unique attribute of HDFS is the viewpoint that it is usually better to locate processing logic near the data rather than moving the data to the application space. HDFS rigorously restricts data write to one write at a time. Bytes are always appended to the end of a stream, and byte streams are guaranteed to be stored in the written order.

2.6 Ceph

Ceph[25, 30, 1] is a software storage platform designed to present object, block, and file storage from a single distributed computer cluster. Ceph is a distributed storage designed to provide excellent performance, reliability and scalability. Ceph was made possible by a global community of enthusiastic storage engineers and researchers. It is open source and freely-available. Ceph software runs on commodity hardware. The system is designed to be both self-healing and self-managing and strives to cut both administrator and budget costs.

2.7 Related Work

Chengzhang Penga and Zejun Jiangb [14] proposed a cloud storage service system, in which a solution is suggested to build a cloud storage service system based on the open-source distributed database. Their system was designed as three-tiered architecture. And they demonstrated their prototype of CS3 built based on the framework.

A file system architecture for organization of data and metadata that will efficiently organize and enable sharing besides exploiting the power of storage virtualization and maintaining simplicity in such a highly complex and virtualized environment proposed by Ankur Agrrawal et al [3].

Tahani Hussain et al.[9] assessed the performance of an existing enterprise network before and after deploying distributed storage systems. And the simulation of an enterprise network with 680 clients and 54 servers followed by redesigning shows improvements in the storage system throughput by 13.9% with 24.4% decreased average response time and 38.3% reduced packet loss rate.

Dejun Wang[22] proposed an efficient cloud storage mode for heterogeneous cloud infrastructures. And he verified by extensive tests to the model with numerical examples. He thought cloud storage system with traditional storage systems has some differences; such as the demand from the performance point of view, data security, reliability, efficiency and other indicators need to be considered for cloud storage services, which are services in a wide range of complex network environment for the demands of large-scale users.

3 System Design and Implementation

3.1 System Architecture

The main goal of this paper is to build a cloud service platform integrating with several heterogeneous storage technologies. The cloud storage combines different storage technologies that are the most popular and perform great in the same type of technology. These storages that we chose are open source; hence, it will be helpful for further development and maintenance in the future. Additionally, we will offer a simple GUI for users to manage the cloud service. In this paper, we will overview the whole system.

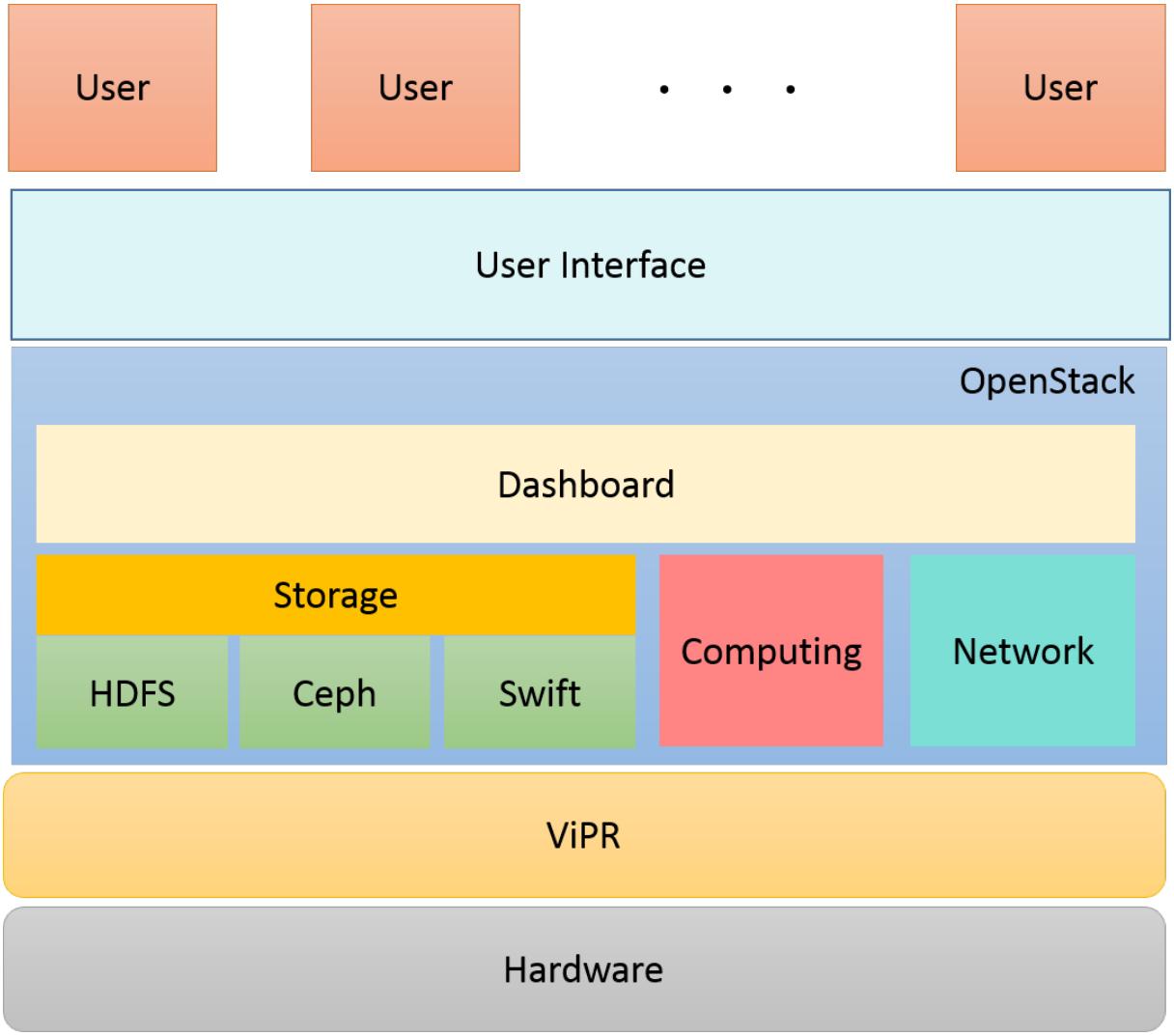


Figure 2: System Architecture

The system architecture consists of some components. The first part is a server node built by OpenStack. And we also used OpenStack to build some VMs to be the storage nodes. Second, we used three kinds of different storage technology to build the storage of OpenStack. And we used software ViPR to control the different storage devices and manage storage spaces for the storage of OpenStack. ViPR will display the storage with a simple appearance and it can be used to integrate different storage technologies. The system architecture is shown in Figure 2.

In our experiment, we focus on the inside of system. The basic of system was built by OpenStack, which was combined with several components, including the computing side, network side and storage side. We integrated some different storage to be the storage side for the OpenStack cloud. And the storage controller integrated them by the APIs of these storage.

3.2 System Implementation

- Architecture of Ceph: there are three daemons in the Ceph side, including OSD, MDS and MON. The OSD is responsible for saving data as an object; it used an independent partition, and it will partition to a xfs type. It at least has one node, and the capacity of Ceph is defined by the Equation(3.1), where i means the number of osd from 0 to n, and n means the last number of OSD. The function shows the total capacity of OSD as the sum of all OSDs.

$$CephSize = \sum_{i=0}^{n} partitionsize(OSD)_i, n \geq 1 \tag{1}$$

The second is MDS, responsible for storing the data structure of the data index, and writing it into the OSD. It at least has one node, and it uses a redundant mechanism. The last is MON which is responsible to monitor the running status of OSD and MDS, and according to the situation to adjust them. It at least has one node, and it uses a redundant mechanism too. The Ceph architecture is shown in Figure 3.

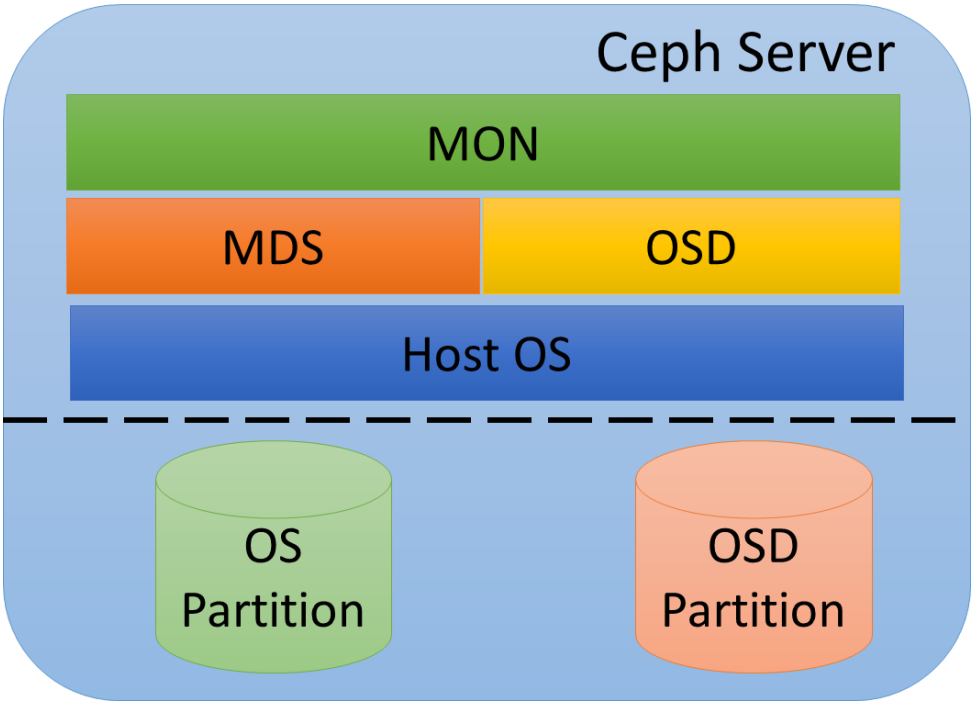


Figure 3: Ceph Architecture

Every OSD has an independent partition; and the three daemons will not conflict with one another. They can run on a same node.

- Architecture of HDFS: the HDFS is combined with the master node and the slave node which is called NameNode and DataNode. And we built a HDFS architecture consisting of one NameNode and two DataNodes shown in Figure 4. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from clients of the file system. The DataNodes also performs block creation, deletion, and replication upon instruction from the NameNode shown in Figure 5.

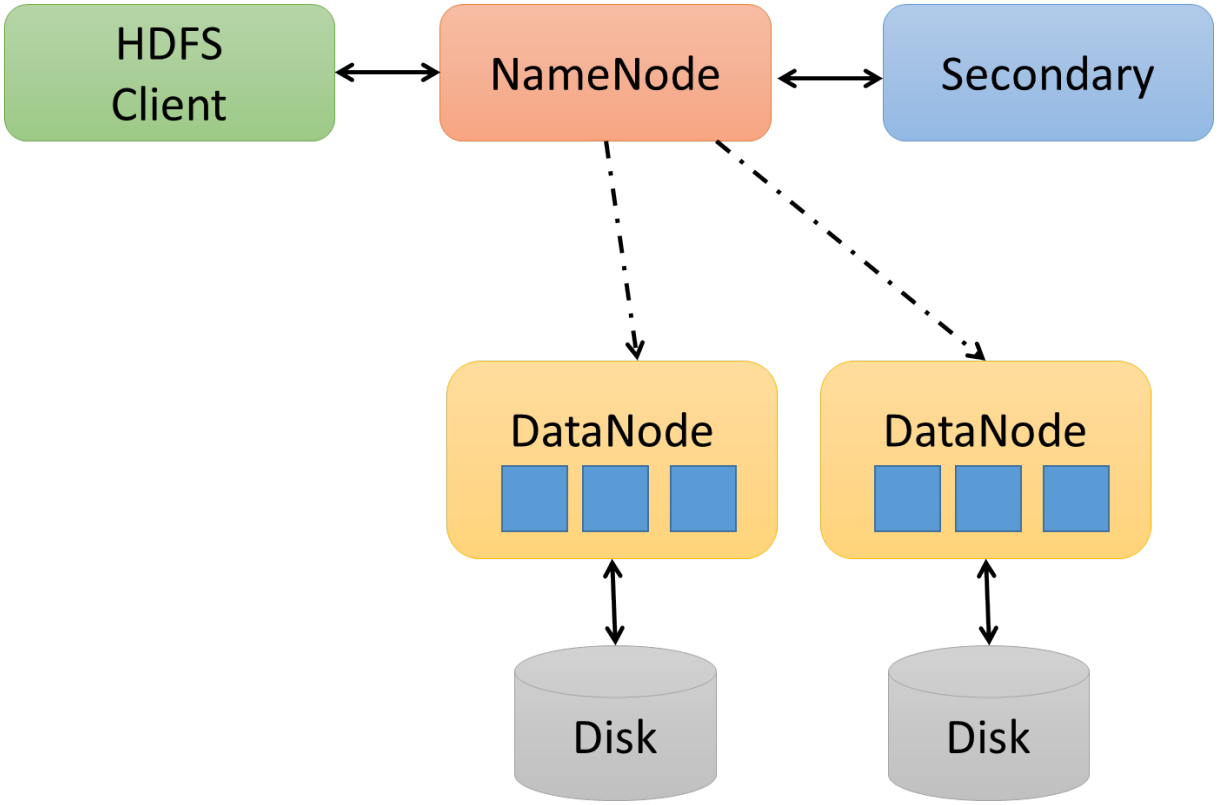


Figure 4: HDFS Service Architecture

- OpenStack: OpenStack was used to build our cloud; it is open source software to manager cloud. It was built on a VM with Ubuntu OS, and then some VMs were created to form a storage system cluster. The overview of the system is shown in Figure 6 and Figure 7.

- EMC ViPR: traditional data center is just a set of some servers, storage, network and security and etc. In the traditional data center, to build a new application usually takes several weeks. But the process can be simplified by the data center driven by completely dynamic software, making it like building a new VM. If we can design a series of policies that can be implemented on a variety of new applications to automatically support infrastructure services included in a container, i.e., a virtual data center, then, in a few minutes or a few seconds, we can complete the deployment of a new application.

  EMC Virtualization Platform Reinvented (ViPR) is a logical storage system, not a physical storage. It can integrate EMC storage and third-party storage in a storage pool, and manage it as a single system, but still save the worth of original storage. ViPR can replicate data across several different place and data center with different store product, and it provides a unified block store, object store

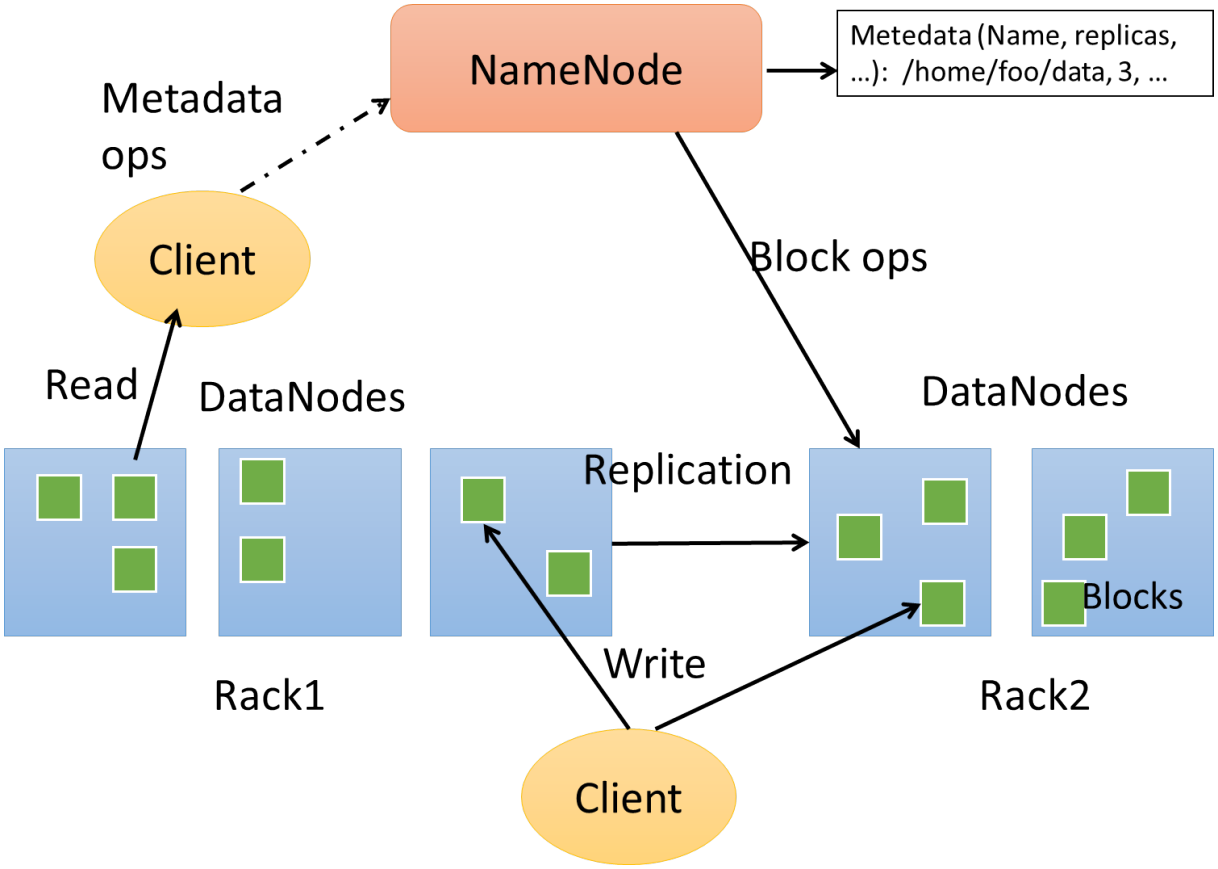


Figure 5: HDFS Architecture

and file system and other services. At the same time, ViPR provides a unified metadata service and self-service deployment, measurement and monitoring services. In addition to this, ViPR also applies to multi-tenant environments. EMC proposed a model about software-defined storage, and described the relation with OpenStack. The architecture is shown in Figure 8.

ViPR run on three to five virtual server machines. It consists of the control plane and data plane; the first part realizes automatic store, and the second part provides data service by building on the first part. ViPR provides some APIs to let users to use services and manage storage, as shown in Figure 9.

One part of ViPR is the control plane. It can manage all applications of storage array, such as data mining, storage resource searching, and capacity counting and reporting. The controller controls the application of virtual storage array, and manages the storage resource. In order to achieve the goal, ViPR virtualizes the control path of physical storage, and runs the function with it. By the virtual plane, people can manage the storage pool and split it to different virtual storage arrays with policies, just like virtualization of the server.

In addition to the controller and data service, ViPR provides the open RESTful API. The developers can develop new services without the limit of the hardware. Through these APIs, people can access data, and add, delete, modify, monitor and measure the logical storage resources.

ViPR is built by the scale-out architecture; if it is built with at least three clusters, the architecture will provide the ability of high availability, load balance and system upgrade and so on. It

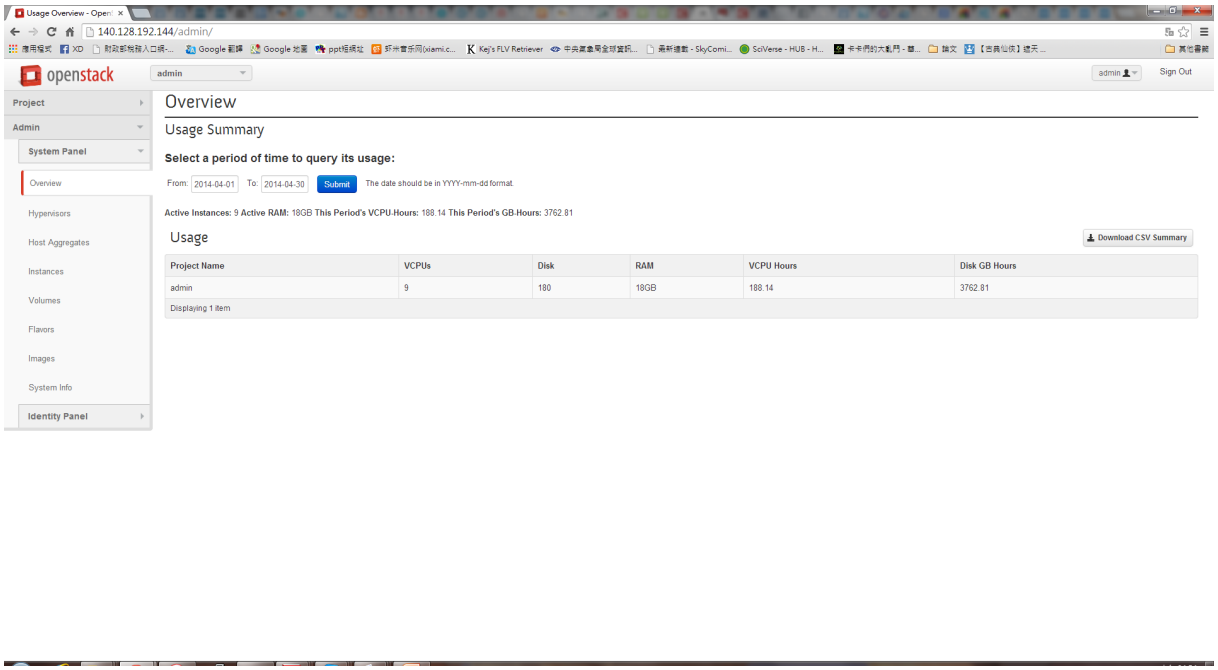


Figure 6: OpenStack Overview

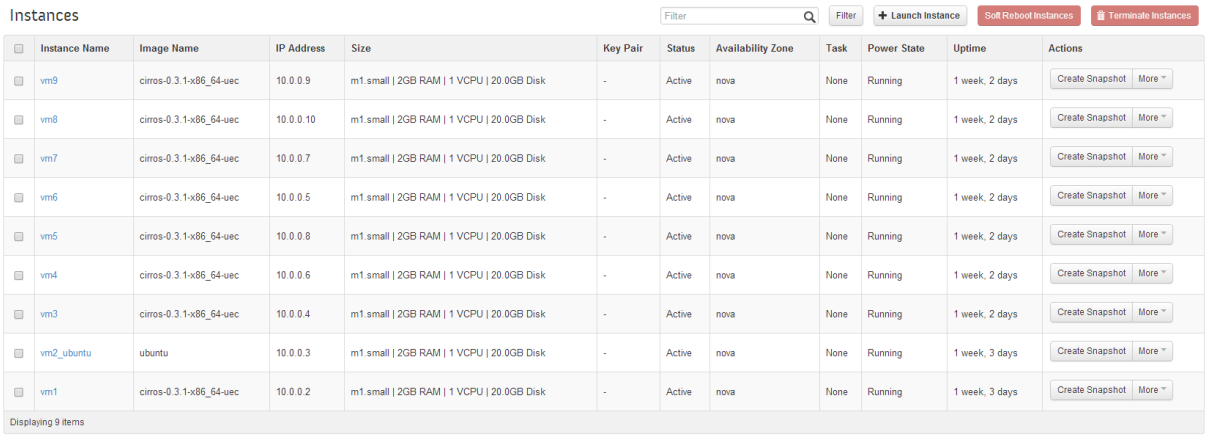


Figure 7: VM Instances

provides RESTfulAPI, GUI(console), CLI and SDK to let user control it with high flexibility as shown in Figure 10. And ViPR can provide the automatic of disaster recovery. Through the continuous remote replication technology, it can provide the ability of data replication continuously to successfully achieve the goal of disaster recovery.

4 Experimental Results

4.1 Experimental Environment

We used OpenStack to build our cloud platform, which then was used to create and manage the distributed storage system. In the system we integrated three heterogeneous storage technologies. And we

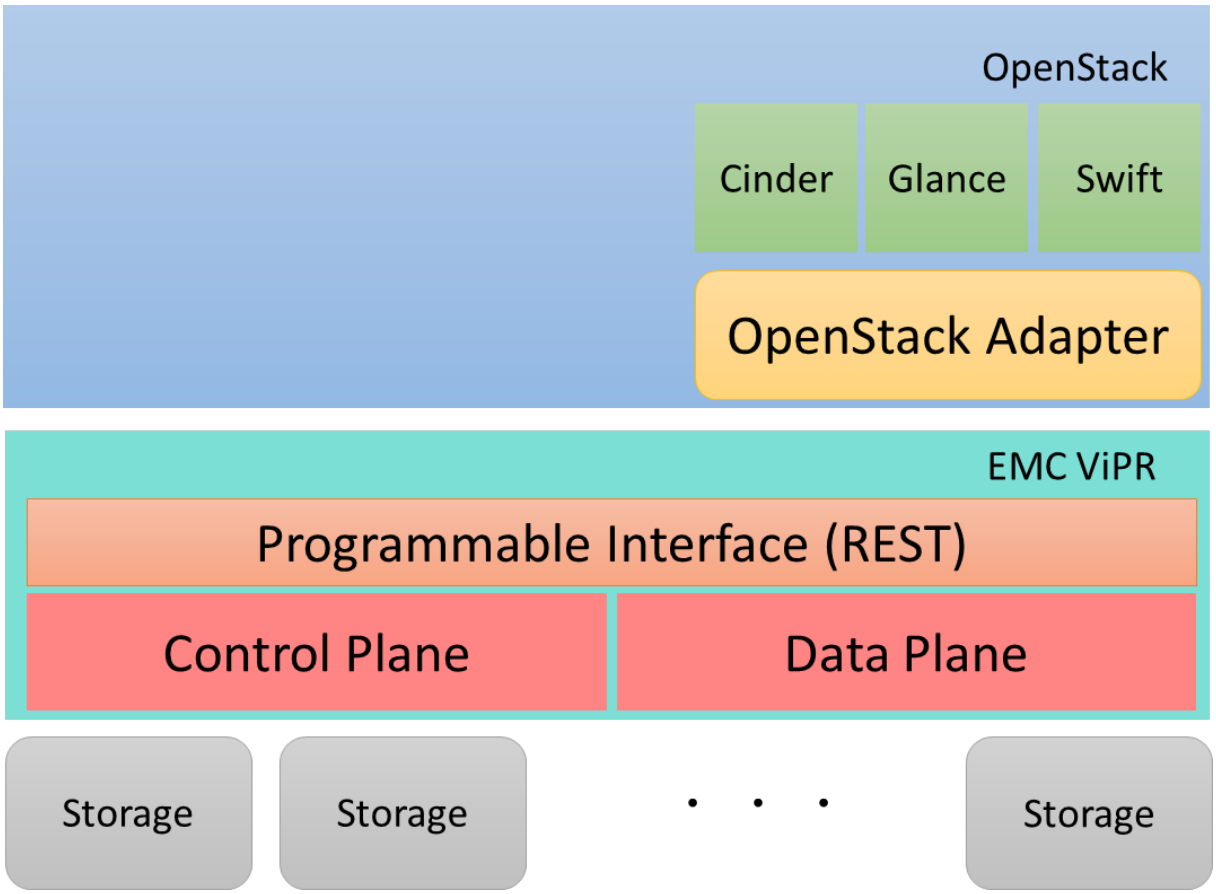


Figure 8: ViPR with OpenStack Architecture

built the storage system by some VMs, in which Ceph was constructed by three VMs with specifications of 1 core CPU, 2GB memory, and a total of 60GB storage space. The first node was MON and OSD, the second was MDS and OSD, the third was OSD. These were components in the Ceph cluster. And the HDFS part was constructed by three virtual machines including one NameNode and two DataNodes with specifications of 1 core CPU, 2GB memory, and total of 40GB storage space.

4.2 Performance of Data Transport

In the first part of our experiment, we compared the three different kinds of storage technology, including Ceph, Hadoop HDFS and GlusterFS. The reason to choose them is that they are famous storage technology in the popular distributed storage systems today.

In this experiment, we built three storage systems: Hadoop, Ceph and GlusterFS and recorded the data of the performance of uploading and downloading files on the three systems. We used three cases to upload and download a single file. The first case was for files with the size of 1MB to 16GB. The second is tests of small sized files of 1MB to 100MB, with 16GB total file size. And the last one is files100MB to 1000MB, with 16GB total file size.

The result for the experiment to upload a single file from 1MB to 16GB from the client node to the server node is shown in Figure 11. We observe that when the file size is smaller than 512MB, the performance of the three storage systems are similar. But when the file size is larger than 512MB, the performance of Hadoop is better than GlusterFS and Ceph. According to this result, we can say the

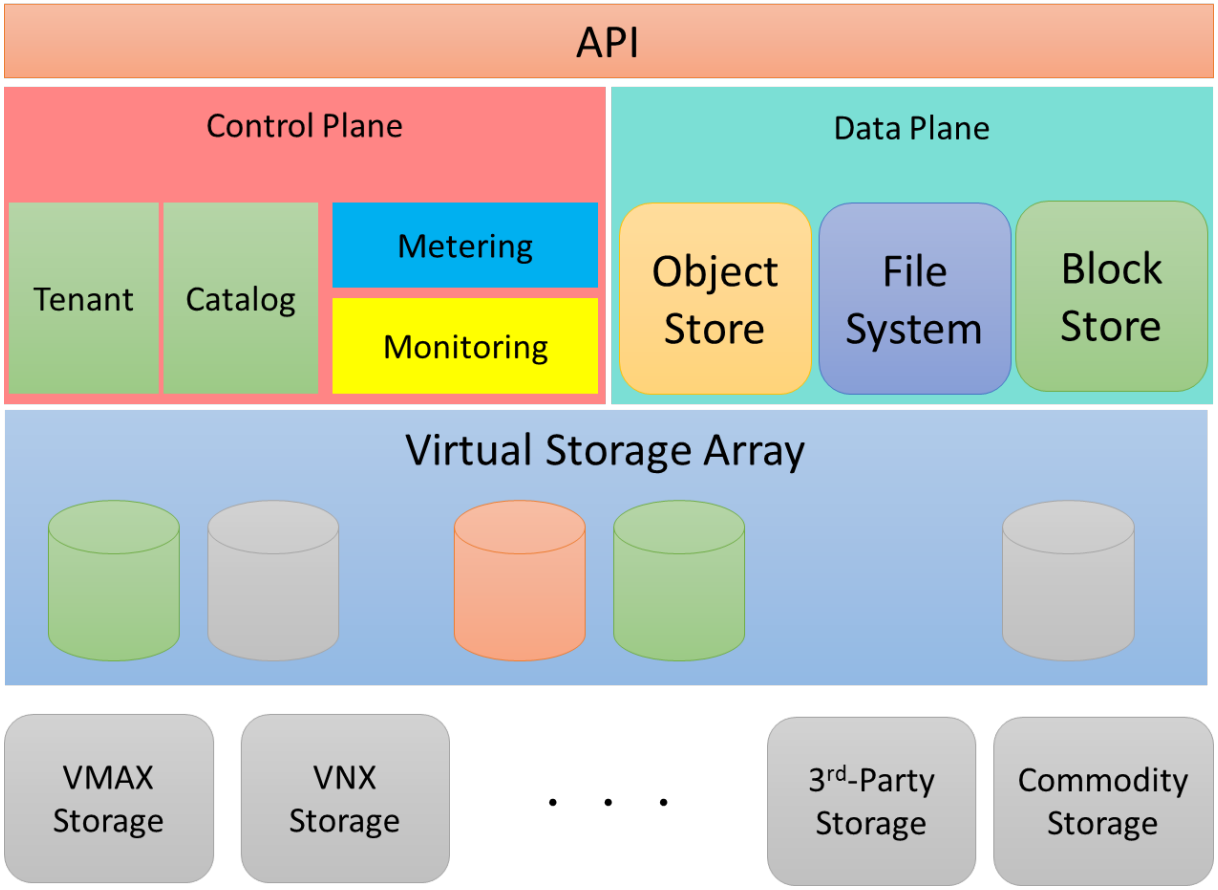


Figure 9: ViPR System Architecture

ability of Hadoop is better than GlusterFS and Ceph when uploading large files.

The result for the experiment to download a single file from 1MB to 16GB from the client node to the server node is shown in Figure 12. From the result, we observe that when the file is smaller than 512MB, the performance of the three storage systems are similar. But when the file size is larger than 512MB, the performance of GlusterFS is better than the other two. However, when the file size is larger than 4GB, the performance of Hadoop is better than Ceph.

In this part, we uploaded 16GB file in total, and split the file into smaller files of 10MB to 100MB. We will use it to test the effect of performance when transporting files of different sizes. In Figure 13, we can see the curve is almost stable; however when the file size is just larger than 60MB, the run time of Hadoop increase fast. And when the file size is between 30MB to 60MB, its performance is the best. The performance of GlusterFS and Ceph are similar to each other. And Hadoop performs better than they.

In this part, we downloaded 16GB file in total, and split the file to be 10MB to 100MB. We will use it to test the effect of performance of downloading different file sizes, as shown in Figure 14. In this experiment, we find the results have not consistent trends; they are close. The performance of Hadoop is better than Ceph most of the time, but when the file size between 50MB and 80MB, the performance of Ceph and Hadoop is close.

As shown in Fig. 15, we uploaded 16GB file in total, and split the file from 100MB to 1000MB; we wanted to test the effect of performance of data transport between larger file and small file. From the results, we can observe that the performance of Hadoop is better than the other two. Moreover, when the

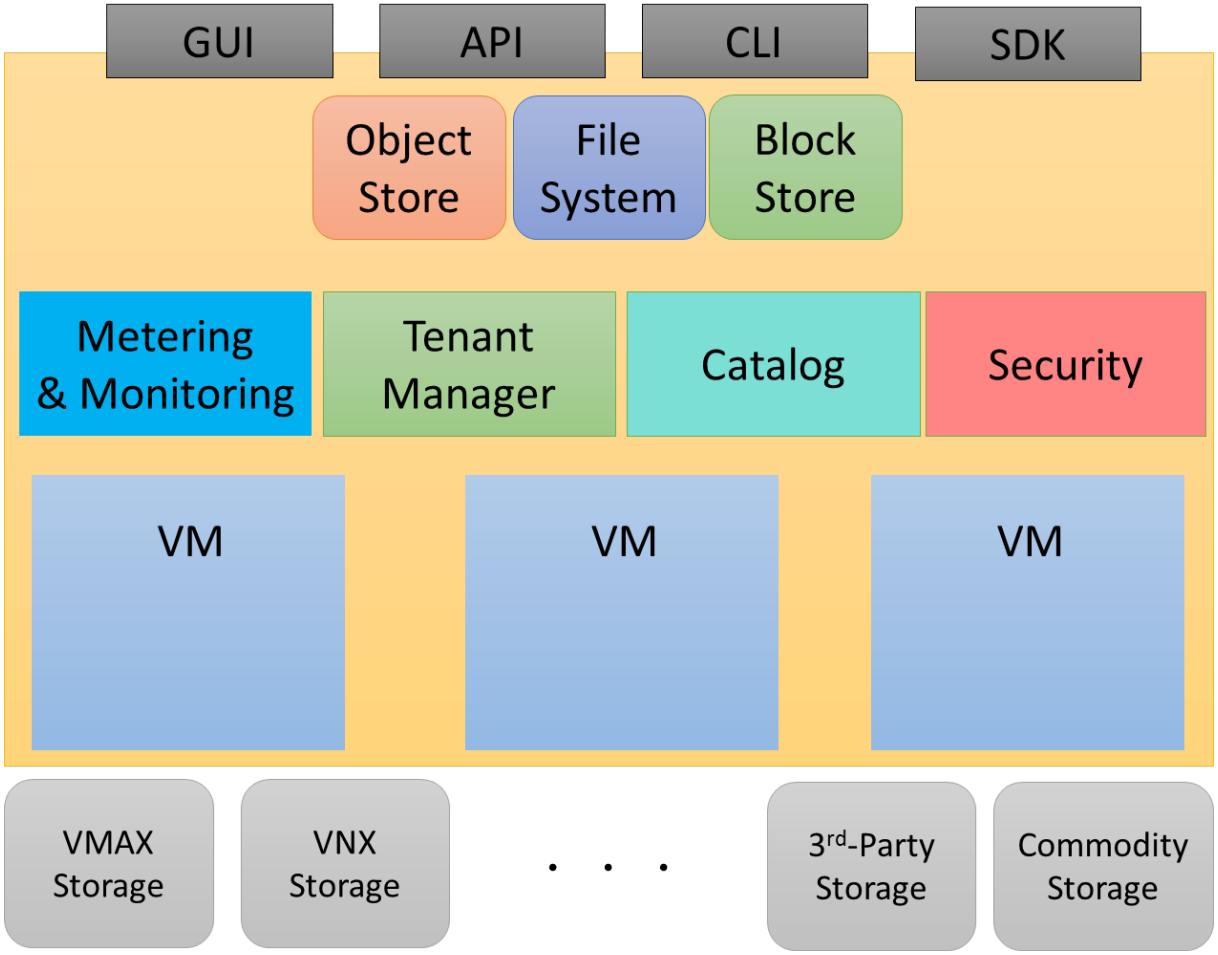


Figure 10: ViPR Cluster

file size is smaller than 500MB, Ceph is better than GlusterFS, but if the file size is larger than 500MB, Ceph is either similar to GlusterFS or even worse.

As shown in Fig. 16, we download 16GB file in total, and split the file from 100MB to 1000MB. From the result, we found that the performance of them is close, but Ceph rises suddenly when the file size is between 500MB and 600MB. And Hadoop is the best in them.

4.3 User Interface of ViPR

ViPR can integrate some different original storage architecture through a data service, and provide some automatic mechanism for the storage systems to simplify the management work and decrease the work complexity. It can transform the physical storage from different company or system to be a single, scalable and open virtual storage platform. It can manage all storage resource by a single interface, as shown in Figure 17.

As shown in Figure 18, people can register their storage system to the physical assets part. And people can add some computers in the physical storage array to be a host, as shown in Figure 19.

ViPR transforms the physical storage to be a virtual storage plane, and it can create or manage the virtual storage pools automatically by the policy from the manager. When people create the policy, he can set how to create the virtual storage pool according to the storage performance which the service

Table 1: Hardware Specification

Host name	CPU	Memory	Disk	OS
Management node	vCPU 9 cores	20GB	200GB	Ubuntu 12.04
Ceph node1	1 cores vCPU	2GB	20GB	Ubuntu 12.04
Ceph node2	1 cores vCPU	2GB	20GB	Ubuntu 12.04
Ceph node3	1 cores vCPU	2GB	20GB	Ubuntu 12.04
HDFS Namenode1	1 cores vCPU	2GB	20GB	Ubuntu 12.04
HDFS Datanode1	1 cores vCPU	2GB	20GB	Ubuntu 12.04
HDFS Datanode2	1 cores vCPU	2GB	20GB	Ubuntu 12.04
GlusterFS server1	1 cores vCPU	2GB	20GB	Ubuntu 12.04
GlusterFS server2	1 cores vCPU	2GB	20GB	Ubuntu 12.04
GlusterFS client	1 cores vCPU	2GB	20GB	Ubuntu 12.04

Table 2: Software Specification

Software	Version
OpenStack	Havana
Ceph	V0.72 EMPEROR
HDFS	2.2.0
Swift	
GlusterFS	3.5

needs, as shown in Figure 20 and Figure 21.

There are some data service will run on the plane, and the virtual storage pool and virtual storage array can provide the application with the data service, as shown in Figure 22. People can split the virtual storage pool to be some virtual storage arrays, and manage them by the policy. There are three types in the data service:

- Object: the file systems can store, take a file and modify the unstructured data by the data service, and the origin data application system can read or write the data easily.

- HDFS: collocation the service, people can transform the origin storage device to be a big data storage place. And they can use it to build a big data environment quickly without building an analysis cluster.

- Block: in the block data service, people can take a data across several storage array, and it can support many kinds of applications, file systems, database and virtual platforms.

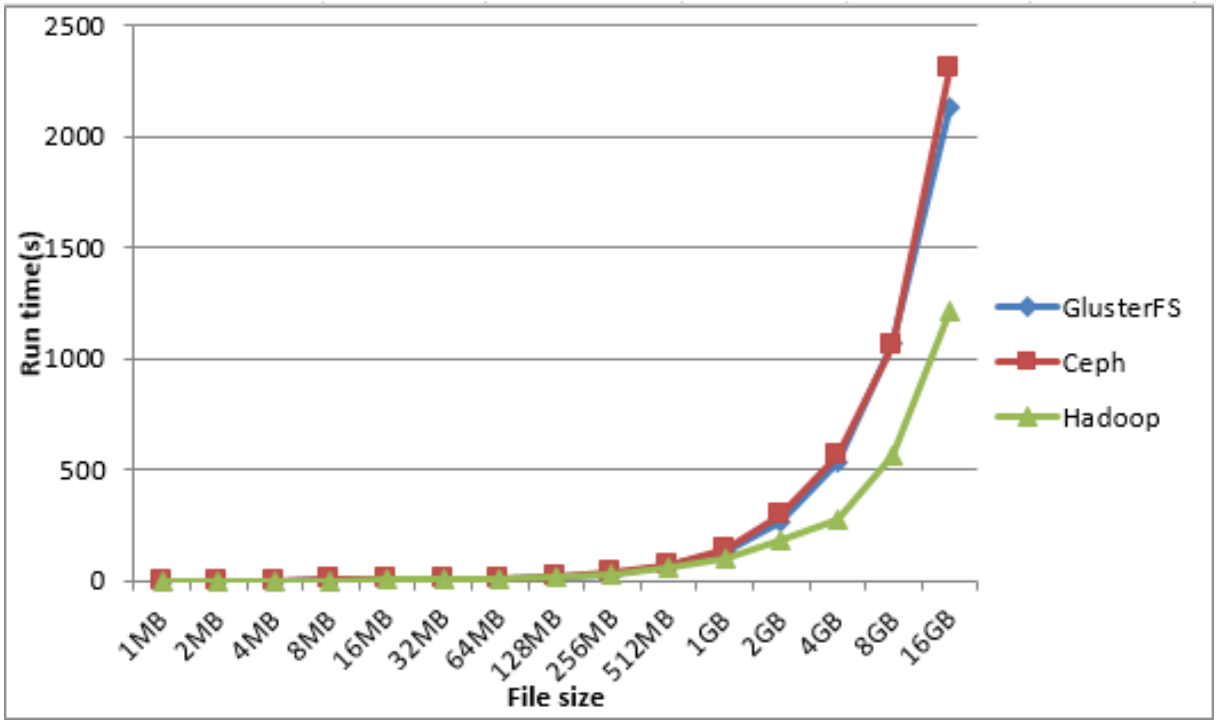


Figure 11: Write single file with different file size

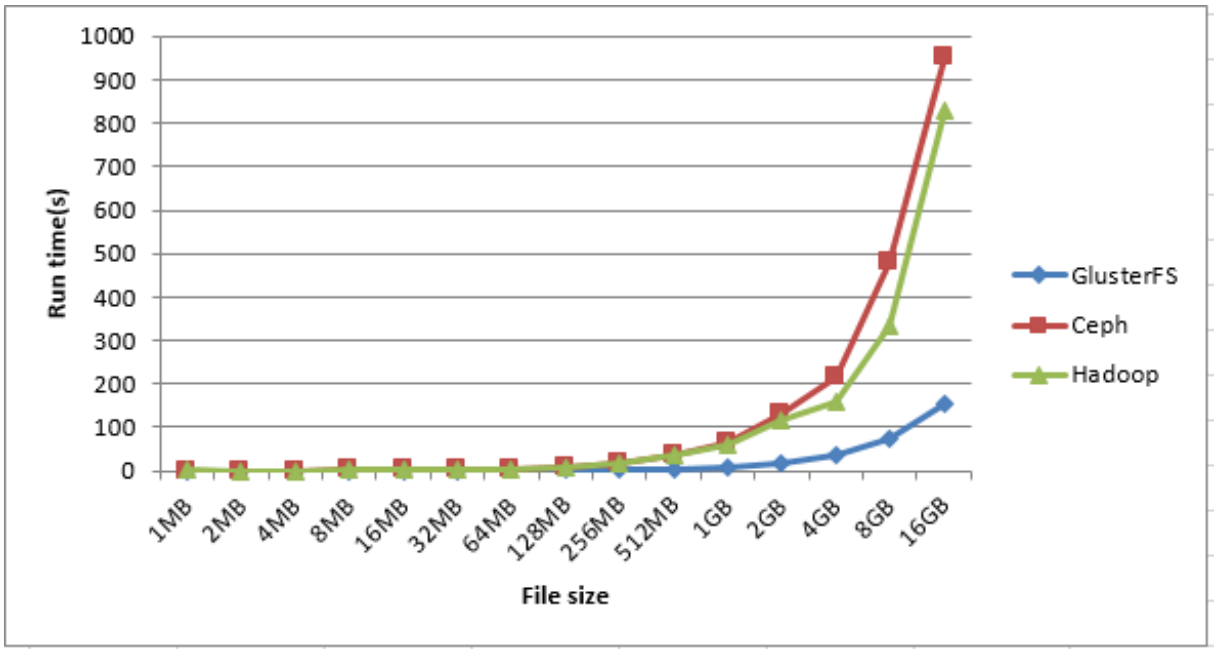


Figure 12: Read single file with different file size

4.4 Discussion

We used ViPR to provide the storage management to system; it can manage storage array simply through the data plane and control plane. The data plane is an advantage data service which runs on storage arrays,

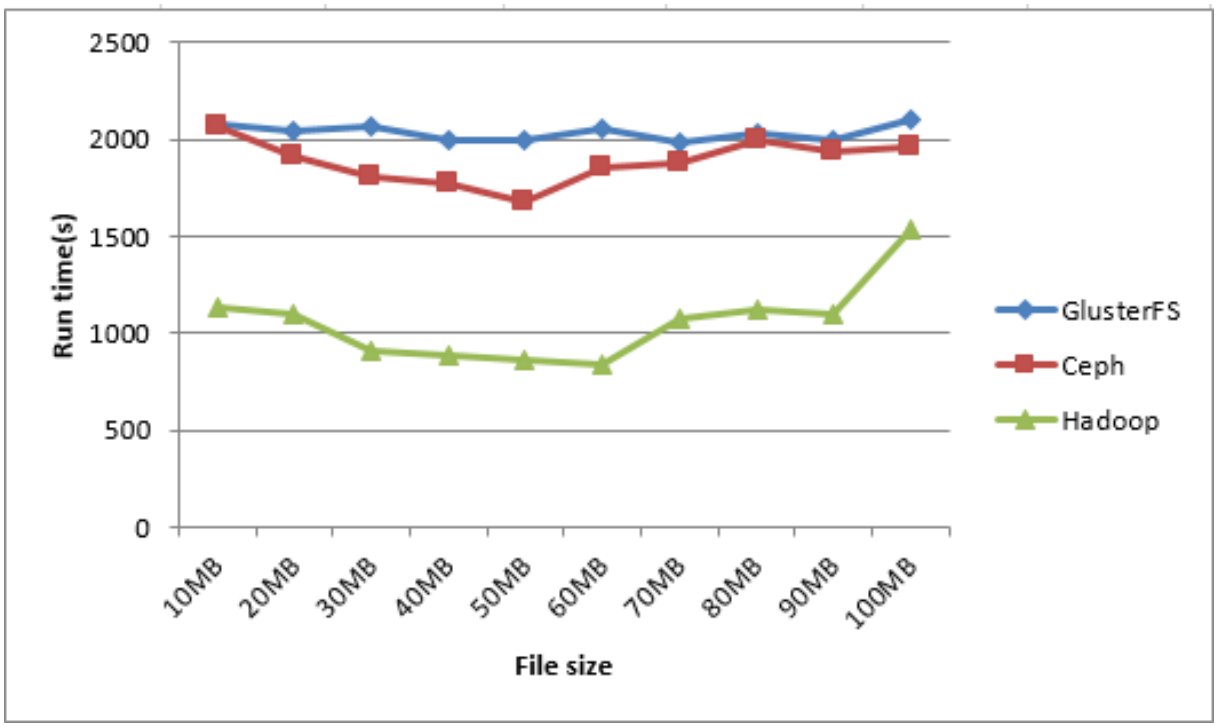


Figure 13: Write small file

Figure 14: Read small file

for example it can offer ability of object storage on file system storage platforms or offer ability of block storage on normal storage platforms. The data service which can virtualize the storage system combines

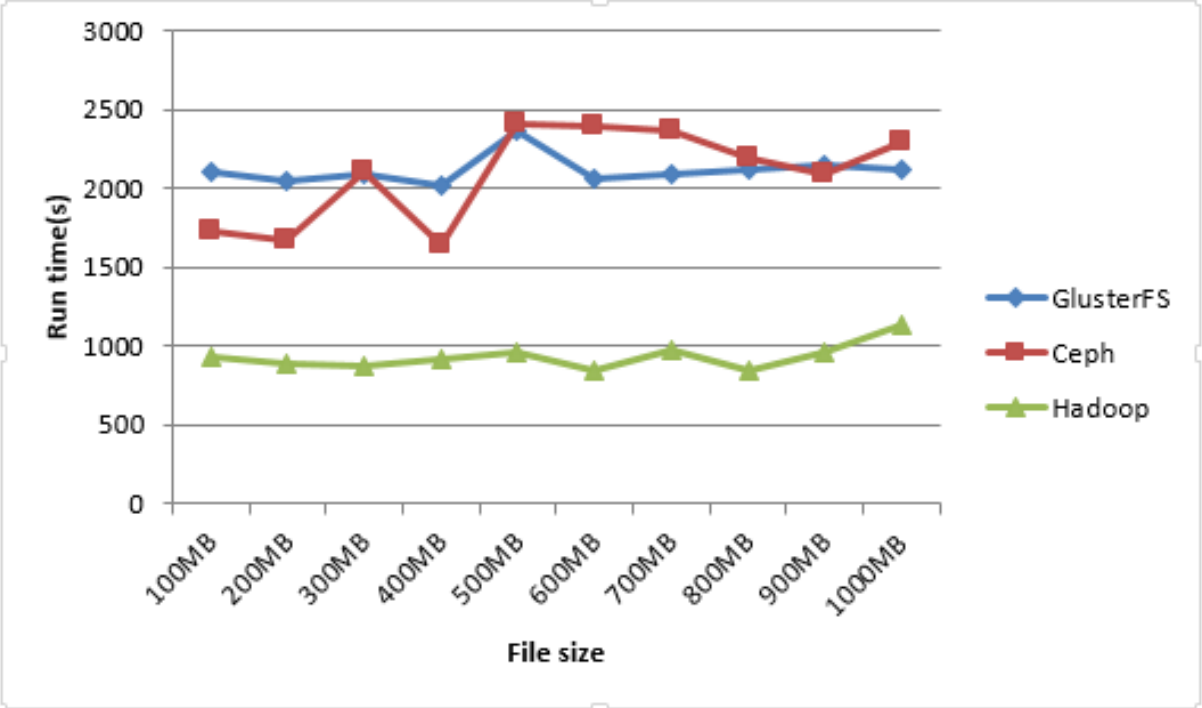


Figure 15: Write small file

Figure 16: Read small file

different data types (file, object, and block), protocol (iSCSI, NFS, and REST), security, usability. We used ViPR to offer a storage pool to our system, and we prove that the system will run normality by

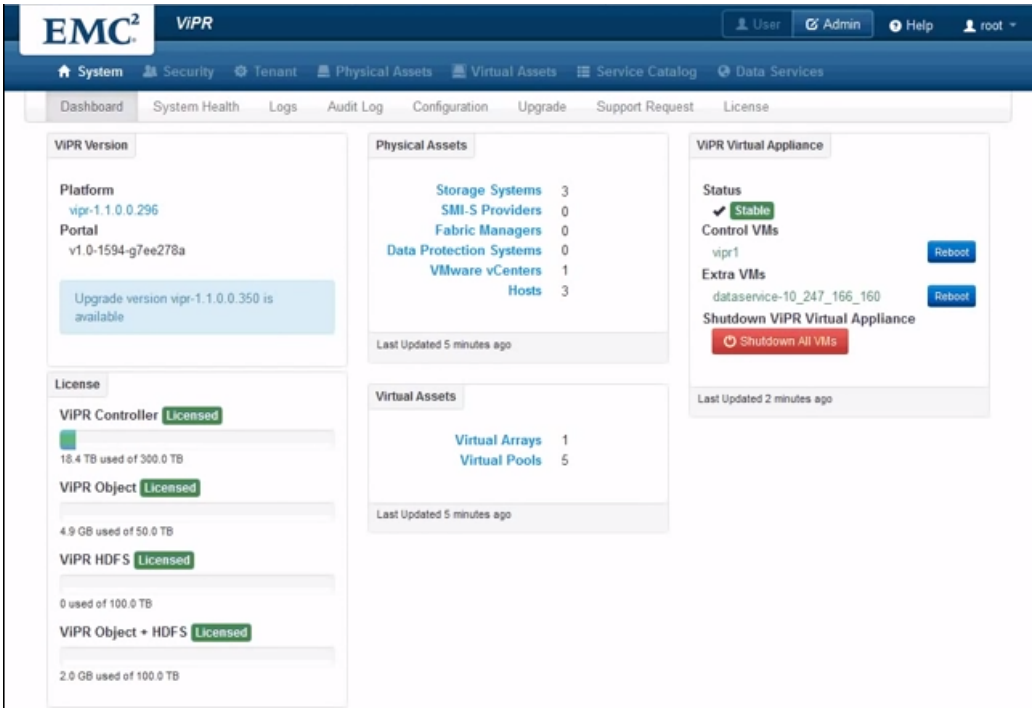


Figure 17: Dashboard

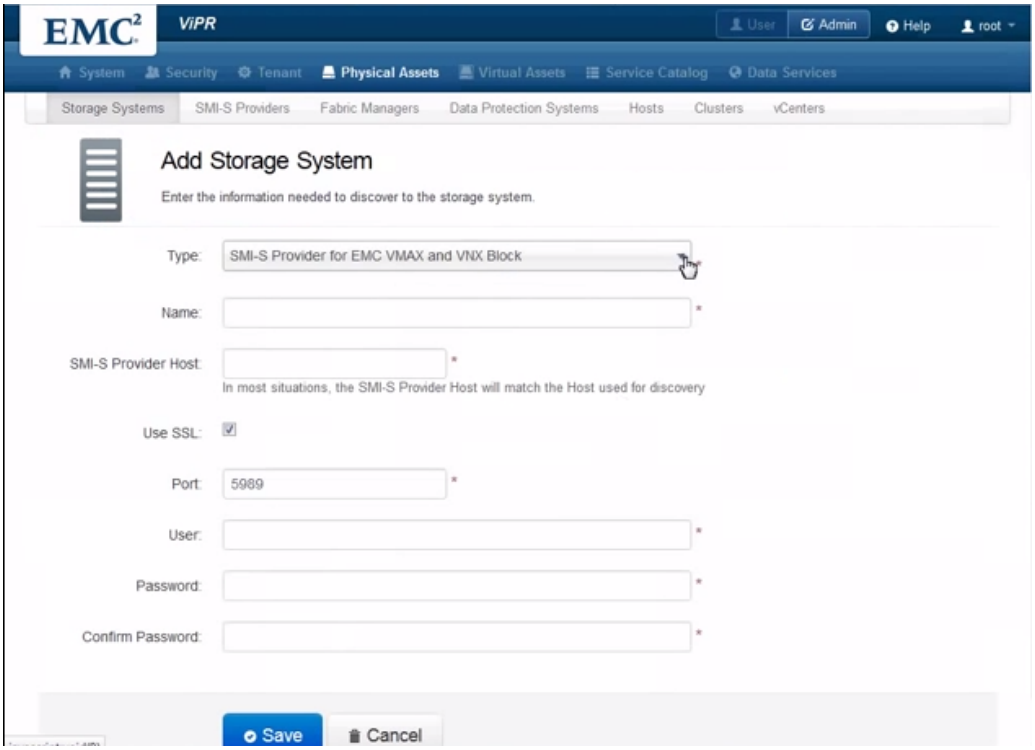


Figure 18: Add PhysicalAsset

the experiments in this paper. According to the experimental results, we can find that the performance

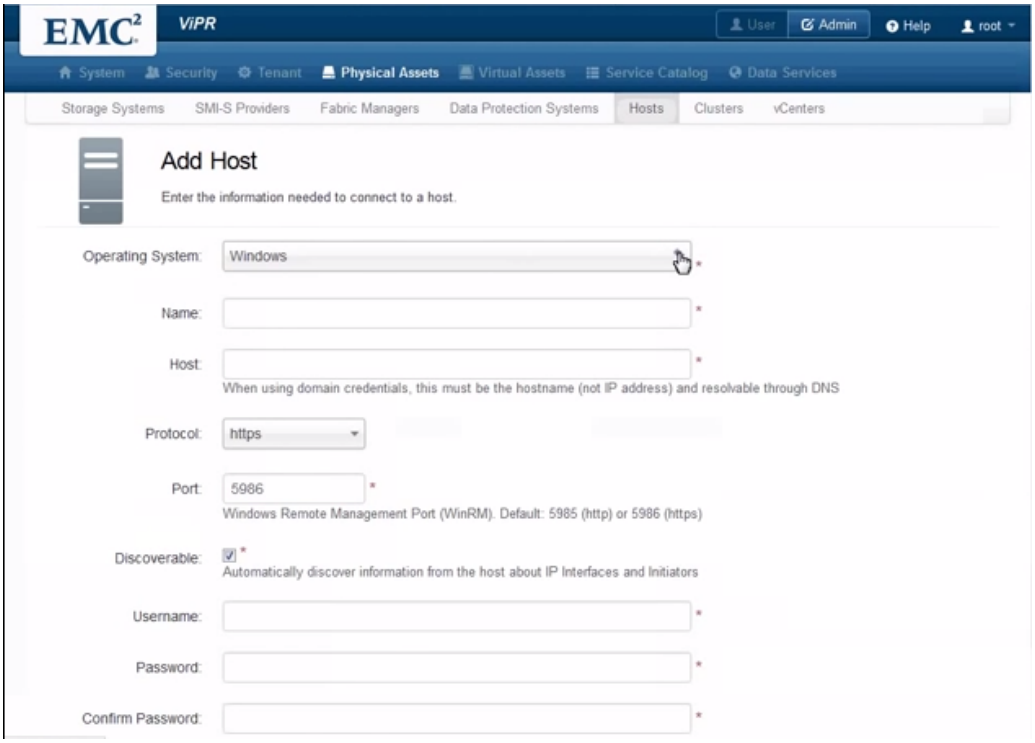


Figure 19: Add Host

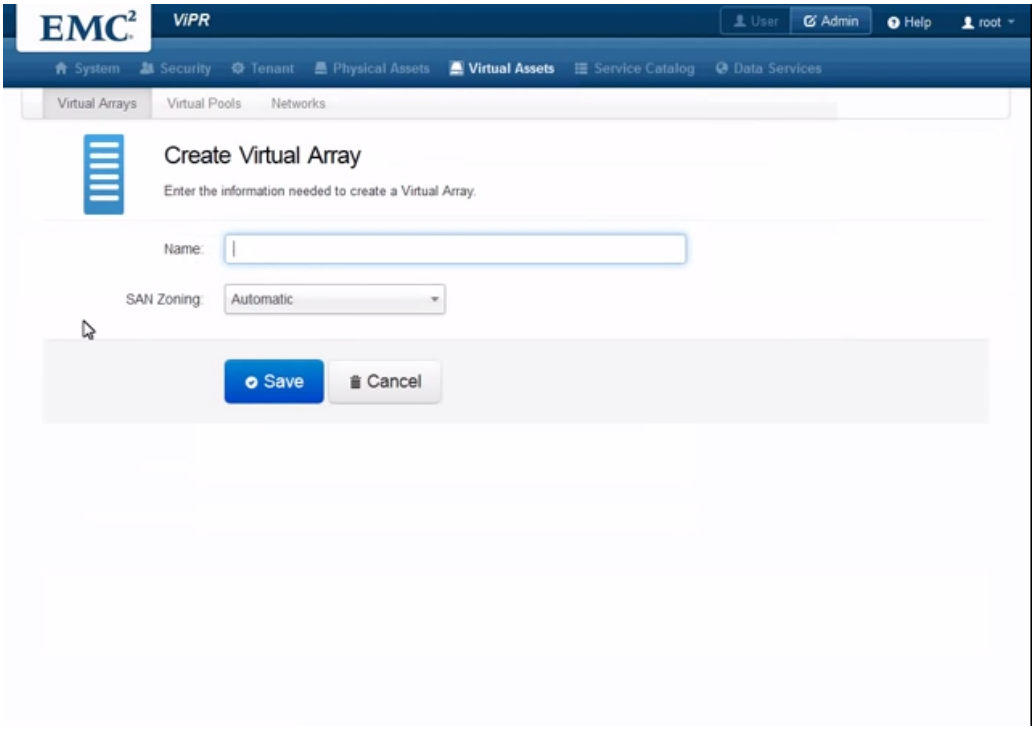


Figure 20: Add VirtualAsset

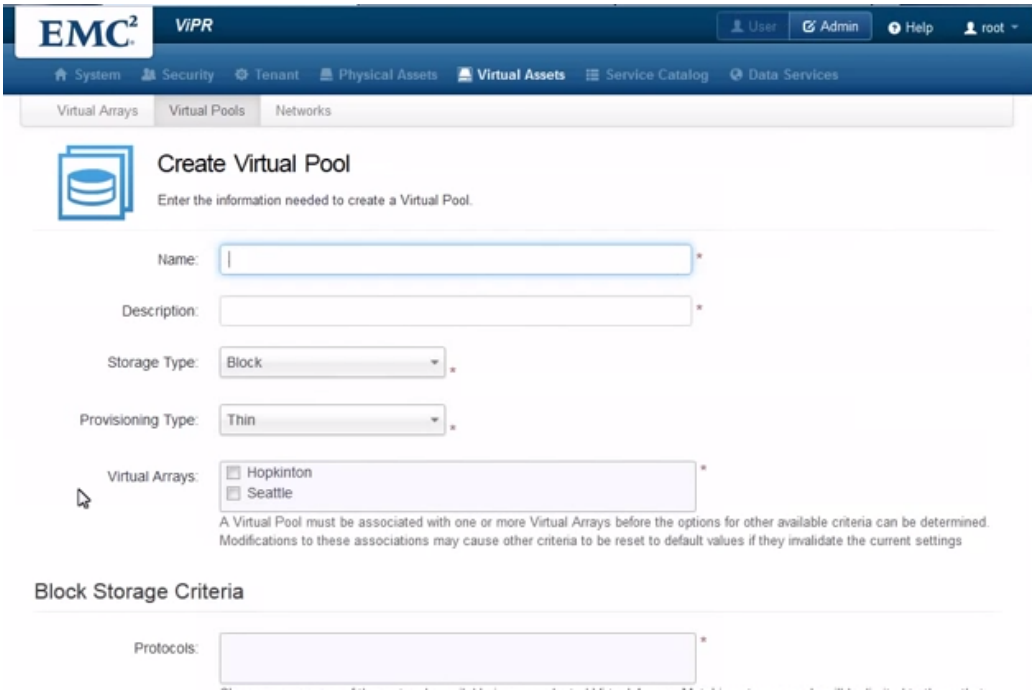


Figure 21: VirtualPool Create

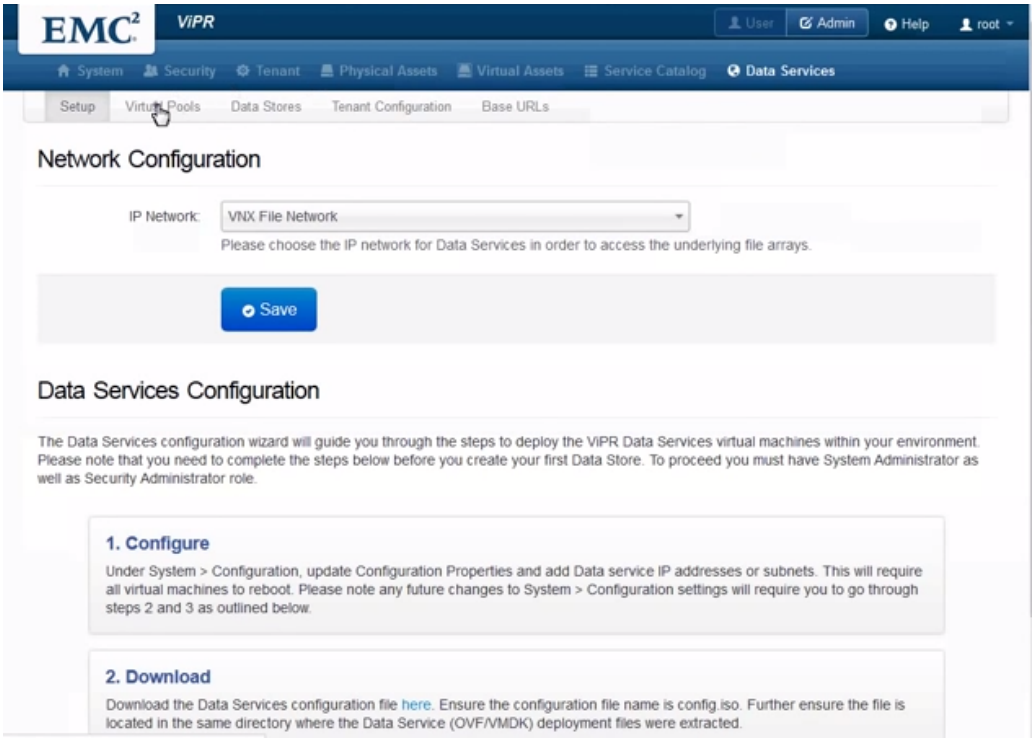


Figure 22: DataService Setup

of Hadoop HDFS is better than the others, and thus, Hadoop HDFS is more mature and has higher performance than the others. And ViPR can create virtual storage pool and assign the storage space to

different kinds of storage.

5 Conclusion and Future Work

5.1 Conclusion Remarks

Cloud computing is a concept, in which computers over networks can cooperate to provide far-reaching network services. The basic approach to computing through the Internet terminal operations of cloud computing is to move hardware, software, and shared information from users to the server side. Due to the situation, it can be said that our life is surrounded by cloud technology, and we almost use the cloud every day. In recent years, the request for cloud service increases quickly every day. How to manage the cloud more humanely and make cloud deployment simpler are very important. In this paper, a cloud platform was developed to achieve the concept of Storage-Defined Storage by software through distributed cloud architecture to build a high scalable and reliable cloud service, which integrates various storage technologies. Finally, the system also provides a user-friendly interface to allow users to use this system with ease.

5.2 Future Work

The proposed system still has many places to improve. The first problem is the storages which the proposed system can support are limited. It just can support some storage technologies. The other problem is that the environment of the system is small. Therefore, it is necessary to develop and deploy software that can support more storage technologies. It is also required to build a larger environment with the system architecture, and improve the performance of the data transport from the management node to the storages.

References

[1] USENIX Association, 2006.

[2] Openstack. `https://www.openstack.org/` [Online; Accessed on August 2, 2019], 2014.

[3] A. Agrrawal, R. Shankar, S. Akarsh, and P. Madan. File system aware storage virtualization management. In *Proc. of the 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM'12), Bangalore, India*, pages 1–11. IEEE, October 2012.

[4] M. Carlson, A. Yoder, and L. Schoeb. Software defined storage. `http://snia.org/sites/default/files/SNIA%20Software%20Defined%20Storage%20White%20Paper-%20v1.0k-DRAFT.pdf/` [Online; Accessed on August 2, 2019], April 2014. SNIA Working DRAFT.

[5] H.-C. Chao, T.-J. Liu, K.-H. Chen, and C.-R. Dow. A seamless and reliable distributed network file system utilizing webspace. In *Proc. of the 10th International Symposium on Web Site Evolution (WISE'08), Beijing, China*, pages 65–68, October 2008.

[6] Coraid, Inc. The fundamentals of software-defined storage. `https://www.alyseo.com/wp-content/uploads/2014/12/SB-Coraid_SoftwareDefinedStorage.pdf`, [Online; Accessed on August 2, 2019], 2013. SOLUTION BRIEF.

[7] P. T. Endo, G. E. Gonçalves, J. Kelner, and D. Sadok. A survey on open-source cloud computing solutions. In *Proc. of the 28th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC'10), Gramado, Brazil*, pages 3–16, May 2010.

[8] S. Figuerola, M. Lemay, V. Reijs, M. Savoie, and B. S. Arnaud. Converged optical network infrastructures in support of future internet and grid services using iaas to reduce ghg emissions. *Journal of Lightwave Technology*, 27(12):1941–1946, June 2009.

[9] T. Hussain, P. N. Marimuthu, and S. J. Habib. Managing distributed storage system through network redesign. In *Proc. of the 15th Asia-Pacific Network Operations and Management Symposium (APNOMS'13), Hiroshima, Japan*, pages 1–6. IEEE, September 2013.

[10] M. Kim, H. Lee, H. Yoon, J.-I. Kim, and H. Kim. IMAV: an intelligent multi-agent model based on cloud computing for resource virtualization. In *Proc. of the 2011 International Conference on Information and Electronics Engineering (ICIEE'11), Bangkok, Thailand*, pages 199–203. IJIEE, May 2011.

[11] Q. Ma, H. Wang, Y. Li, G. Xie, and F. Liu. A semantic qos-aware discovery framework for web services. In *Proc. of the IEEE 2018 International Conference on Web Services (ICWS'08), Beijing, China*, pages 129–136. IEEE, September 2008.

[12] G. Mackey, S. Sehrish, and J. Wang. Improving metadata management for small files in hdfs. In *Proc. of the 2009 IEEE International Conference on Cluster Computing and Workshops (CLUSTER'09), New Orleans, Los Angeles, USA*, pages 1–4. IEEE, August-Month 2009.

[13] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *Proc. of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09), Shanghai, China*, pages 124–131. IEEE, May 2009.

[14] C. Penga and Z. Jiang. Building a cloud storage service system. *Procedia Environmental Sciences*, 10:691–696, 2011.

[15] S. Robinson. Software-defined storage: The reality beneath the hype. `http://www.computerweekly.com/opinion/Software-defined-storage-The-reality-beneath-the-hype`, [Online; Accessed on August 2, 2019], March 2013. Computer Weekly.

[16] M. Rouse. software-defined storage, 2013. `http://searchsdn.techtarget.com/definition/software-defined-storage`, [Online; Accessed on August 2, 2019].

[17] J. Sahoo, S. Mohapatra, and R. Lath. Virtualization: A survey on concepts, taxonomy and associated security issues. In *Proc. of the 2nd International Conference on Computer and Network Technology (ICCNT'10) , Bangkok, Thailand*, pages 222–226. IEEE, April 2010.

[18] J. Shafer, S. Rixner, and A. L. Cox. The hadoop distributed filesystem: Balancing portability and performance. In *Proc. of the 2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS'10), White Plains, New York, USA*, pages 122–133, March 2010.

[19] J. Spillner, J. Müller, and A. Schill. Creating optimal cloud storage systems. *Future Generation Computer Systems*, 29(4):1062–1072, June 2013.

[20] B. Tang and G. Fedak. Analysis of data reliability tradeoffs in hybrid distributed storage systems. In *Proc. of the IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPS'12 Workshops), Shanghai, China*, pages 1546–1555. IEEE, May 2012.

[21] P. Vichare, A. Nassehi, S. Kumar, and S. T. Newman. A unified manufacturing resource model for representing cnc machining systems. 25(6):999–1007, December 2009.

[22] D. Wang. An efficient cloud storage model for heterogeneous cloud infrastructures. *Procedia Engineering*, 23:510–515, 2011.

[23] L. Wang, D. Chen, Y. Hu, Y. Ma, and J. Wang. Towards enabling cyberinfrastructure as a service in clouds. *Computers & Electrical Engineering*, 39(1):3–14, January 2013.

[24] P. Wang. Qos-aware web services selection with intuitionistic fuzzy set under consumer's vague perception. *Expert Systems with Applications*, 36(3):4460–4466, April 2009.

[25] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proc. of the 7th symposium on Operating systems design and implementation (OSDI'06), Seattle, Washington, USA*, pages 307–320. USENIX Association, November 2006.

[26] Wikipedia. Cloud computing. `http://en.wikipedia.org/wiki/Cloud_computing`, [Online; Accessed on August 2, 2019], 2013.

[27] Wikipedia. Software-defined storage, 2014. `http://en.wikipedia.org/wiki/Software-defined_storage`, [Online; Accessed on August 2, 2019].

[28] C.-T. Yang, W.-C. Shih, G.-H. Chen, and S.-C. Yu. Implementation of a cloud computing environment for hiding huge amounts of data. In *Proc. of the 2010 International Symposium on Parallel and Distributed Processing with Applications (ISPA'10), Taipei, Taiwan*, pages 1–7. IEEE, September 2010.

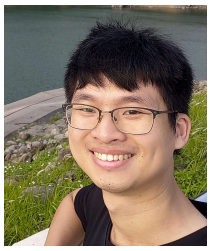
[29] C.-T. Yang, W.-C. Shih, and C.-L. Huang. Implementation of a distributed data storage system with resource monitoring on cloud computing. In *Proc. of the 7th International Conference on Grid and Pervasive Computing (GPC'12), Hong Kong, China*, volume 7296 of *Lecture Notes in Computer Science*, pages 64–73. Springer, Berlin, Heidelberg, May 2012.

[30] J. Yun, Y. H. Park, D. M. Seo, S. J. Lee, and J. S. Yoo. Design and implementation of a metadata management scheme for large distributed file systems. *IEICE Transactions on Information and Systems*, E92-D(7):1475–1478, 7 2009.

Author Biography

Chao-Tung Yang is a Distinguished Professor of Computer Science at Tunghai University in Taiwan. He received his Ph.D. in computer science from National Chiao Tung University in July 1996. In August 2001, he joined the Faculty of the Department of Computer Science at Tunghai University. He is serving in a number of journal editorial boards, including International Journal of Communication Systems, KSII Transactions on Inter- net and Information Systems, Journal of Applied Mathematics, Journal of Cloud Computing. He has published more than 280 papers in journals, book chapters and conference proceedings. His present research interests are in cloud computing and service, big data, parallel computing, and multi-core programming. He is a member of the IEEE Computer Society and ACM.

Shuo-Tsung Chen received the B.S. degree in Mathematics from National Cheng Kung University, Tainan in 1996 and M.S. degree in Applied Mathematics from Tunghai University, Taichung in 2003, Taiwan. In 2010, he received the Ph.D. degree in Electrical Engineering from National Chinan University, Nantou, Taiwan. Now he is an Adjunct Assistant Professor in Department of Electronic Engineering, National Formosa University and Department of Applied Mathematics, Tunghai University.



Wei-Hsiang Lien received the B.S. degree in Computer Science from Tunghai University, Taichung in 2014 and M.S. degree in the same department of Tunghai University in 2016.

Vinod Kumar Verma born in Kalka (Haryana), India. Dr. Verma is an Assistant Professor in the Department of Computer Science and Engineering at Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India. He earned his B.Tech. in Computer Engineering from Kurukshetra University and his MS in Software Systems from BITS, Pilani, in 2005 and 2008 respectively. Dr Verma earned his Ph.D. in Computer Science and Engineering from SLIET, Longowal. Dr Verma has published many research papers in the International Journals of IEEE, Springer, Elsevier, Taylor and Francis. Dr. Verma is serving as Lead Guest Editor for a Special issue in International Journal of Distributed Sensor Networks (JCR@Thomson Reuters Journal), SAGE Journals Publishing, United Kingdom (UK).