

Public Cloud Storage for the Seismic Big Data Based on Amazon EC2 Cluster and Hadoop

Jie Xiong¹ and Song Zhang^{2*}

¹*School of Electronics and Information, Yangtze University, Jingzhou, China*

^{1,2}*Department of Computer Science and Engineering, Mississippi State University, Starkville, MS, USA*

¹*xiongjie@yangtzeu.edu.cn*, ^{2*}*szhang@cse.msstate.edu*

Abstract

The seismic data expanded rapidly in recent years, whose size could be up to hundreds TBs, as modern seismic acquisition technologies were employed. How to store and access the seismic big data efficiently is an emergency problem for the oil industry and scientific research. A public cloud storage scheme for the seismic big data is proposed based on the Amazon EC2 and Hadoop. The IO performance evaluation results show that the proposed public cloud storage scheme has advantages of high IO performance and good scalability. It is suitable for the seismic big data storage and access.

Keywords: *Public cloud storage, Big data, Seismic data, Hadoop, Amazon EC2*

1. Introduction

Seismic exploration is the most important method for the oil and gas resources exploration [1, 2]. The standard file format of seismic data is SEG-Y defined by the Society of Exploration Geophysicists [3]. In recent years, the size of seismic data expanded rapidly, as the new seismic acquisition technologies were employed, and much more wide area were explored [4]. The size of typical seismic data varies from tens of MBs to hundreds of GBs, or even up to hundreds of TBs. How to store and access these seismic big data efficiently is very important for the oil industry and scientific research.

Many researchers have been dedicated in this field recently. Liu [5] designed a distributed seismic data file system (DSFS) based on a computer cluster and gain very high I/O efficiency. Li [6] constructed a high performance cloud storage system on the private cloud cluster. Jin [7] proposed a storage framework for the seismic big data based on the storage cluster and parallel file system. Tayir [4] analyzed the storage effect to the seismic big data on the cluster platform and proposed a scheme to improve the storage efficiency. However, all these solutions mentioned above are based on whether the private cluster or private cloud cluster, which lead to the difficult to share the seismic big data on the Internet and to extend the cluster.

Cloud is a cheap alternative to supercomputers and clusters, a more reliable platform than grids, and a more scalable platform than the largest of commodity clusters. The cloud computing provides a distributed, shared infrastructure for data storage and processing [8]. As one of the biggest cloud computing vendors, Amazon provides a web service named Elastic Compute Cloud (EC2) [9] which could be used to build a virtual cloud cluster [10]. There are many successful commercial big data application based on Amazon EC2, such as FINRA, Yelp, and AdRoll. Apache Hadoop [11] is an open-source distributed computing framework to provide with Hadoop Distributed File System (HDFS) as its distributed file system and MapReduce as the programming model [12]. Hadoop can

* Corresponding Author

combine with the Amazon EC2 easily to build a virtual cluster, which can scale the computation capacity, storage capacity and IO bandwidth by simply adding virtual servers.

In this paper, a public cloud storage scheme for the seismic big data based on Amazon EC2 and Hadoop is proposed firstly; a cloud storage system is built up on the Amazon cloud secondly; its IO performance is evaluated lastly.

2. Design of Cloud Storage for the Seismic Big Data

2.1. Seismic Data

The standard seismic data file (SEG-Y) format was defined by the Society of Exploration Geophysicists [3], which is illustrated in Figure 1.

SEG-Y Tape Label (Optional)	3200 byte Textual File Header	400 byte Textual File Header	1~N th 3200 byte Extended Textual File Header (Optional)	1 st 240 byte Trace Header	1 st Data Trace	...	1 st 240 byte Trace Header	M th Data Trace
--------------------------------------	---	--	--	--	----------------------------	-----	--	----------------------------

Figure 1. Standard Seismic Data File (SEG-Y) Format [3]

The SEG-Y file consists of two main parts, as shown in Figure 1. (1) File header (gray part in the Figure1), which consists of a 3200 byte text file header, a 400 byte binary file header, and several optional information; (2) Seismic data (yellow part in Figure 1), which consists of 1~Mth trace data, each trace data consists of a 240 byte binary trace header and an array (named Data Trace in the figure) containing actual data of that trace.

The size of SEG-Y file varies from MBs to TBs, depending on how large the exploration area is, how many traces per km² there are, and how many sample points per trace have.

2.2. Amazon EC2

Amazon EC2 is a web services which provides resizable compute capacity in the Amazon cloud. We can obtain and configure a virtual cloud cluster by renting the Amazon EC2 service, and be charged according to the computation ability, storage ability, network performance, and the online time.

In this paper, we apply 16 virtual computers (named instances) to build up a virtual cluster, using the free tier Amazon account. The configuration of these instances are listed in Table 1.

Table 1. Configuration of virtual cluster

	CPU	Memory(GB)	Disk(GB)	OS
Master (1 node)	Intel Xeon E5-2676, 2.4GHz	1	20	Ubuntu Server 14.04 LTS
Slaves (15 nodes)	Intel Xeon E5-2676, 2.4GHz	1	30	Ubuntu Server 14.04 LTS

2.3. Hadoop

Hadoop is a framework for distributed processing of large data set across computer cluster. The core modules included in Hadoop [11] are listed in Table 2.

An important characteristic of Hadoop is the partitioning of data and computation across many (could be up to thousands) of hosts, and executing application computations in parallel close to their data [13].

Table 2. The Core Modules of Hadoop

Module Name	Description
Hadoop Common	The common utilities that support the other Hadoop modules.
Hadoop Distributed File System (HDFS)	A distribute file system that provides high-throughput access to application data.
Hadoop YARN	A framework for job scheduling and cluster resource management.
Hadoop MapReduce	A YARN-based system for parallel processing of large data set.

2.4. Cloud Storage Scheme for the Seismic Big Data

We design cloud storage scheme for the seismic big data based on Amazon EC2 and Hadoop are illustrated in Figure 2.

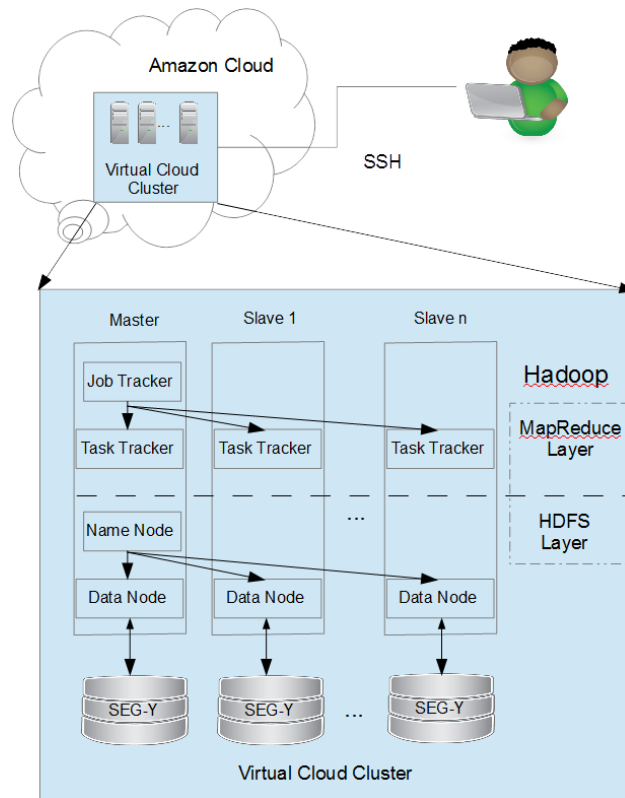


Figure 2. Cloud Storage Scheme for the Seismic Big Data Based On Amazon EC2 Virtual Cluster and Hadoop

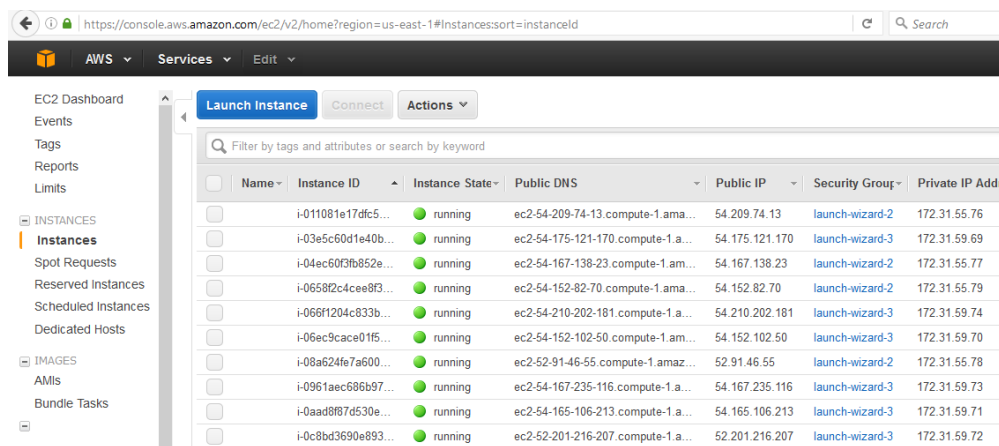
The virtual cloud cluster are located in the Amazon Cloud, consisting of a master node and n slave nodes. The Hadoop is deployed on this virtual cloud cluster, which consists a name node and several data nodes. The name node located at master node is the head of the HDFS, while the data nodes located at master and slave nodes are where the data stored in. The seismic big data (SEG-Y files) are stored in this HDFS. The distributed file IO is under the control of the MapReduce layer. Since the seismic big data are stored distributed, they can be read and write in parallel to speed up the IO performance efficiently. In addition, the computation and storage capacity of the virtual cluster can be enhanced easily by renting more virtual servers from Amazon EC2.

3. Implementation the Cloud Storage for the Seismic Data

3.1. Build a Virtual Cluster On Amazon EC2

There are three steps to build a virtual cluster on Amazon EC2.

- (1) Create a new Amazon Web Service account at <https://aws.amazon.com>.
- (2) Open the Amazon EC2 console at <https://console.aws.amazon/ec2>, from the dashboard. Launch “Ubuntu Server 14.04 LTS AS an Amazon Machine Image” instance.
- (3) Choose the number of instances, and download the key pair and launch the instances. The running instances in the EC2 dashboard are shown in Figure 3.



The screenshot shows the AWS Management Console interface for the EC2 Dashboard. The left sidebar contains navigation options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Images, AMIs, and Bundle Tasks. The main area displays a table of running instances with columns for Name, Instance ID, Instance State, Public DNS, Public IP, Security Group, and Private IP Address. All instances are in a 'running' state.

Name	Instance ID	Instance State	Public DNS	Public IP	Security Group	Private IP Address
	i-011081e17dfc5...	running	ec2-54-209-74-13.compute-1.ama...	54.209.74.13	launch-wizard-2	172.31.55.76
	i-03e5c60d1e40b...	running	ec2-54-175-121-170.compute-1.a...	54.175.121.170	launch-wizard-3	172.31.59.69
	i-04ec60f3fb852e...	running	ec2-54-167-138-23.compute-1.am...	54.167.138.23	launch-wizard-2	172.31.55.77
	i-06582c4cee8f3...	running	ec2-54-152-82-70.compute-1.ama...	54.152.82.70	launch-wizard-2	172.31.55.79
	i-066f1204c833b...	running	ec2-54-210-202-181.compute-1.a...	54.210.202.181	launch-wizard-3	172.31.59.74
	i-06ec9cace01f5...	running	ec2-54-152-102-50.compute-1.am...	54.152.102.50	launch-wizard-3	172.31.59.70
	i-08a624fe7a600...	running	ec2-52-91-46-55.compute-1.amaz...	52.91.46.55	launch-wizard-2	172.31.55.78
	i-0961aec686b97...	running	ec2-54-167-235-116.compute-1.a...	54.167.235.116	launch-wizard-3	172.31.59.73
	i-0aad8f87d530e...	running	ec2-54-165-106-213.compute-1.a...	54.165.106.213	launch-wizard-3	172.31.59.71
	i-0c8bd3690e893...	running	ec2-52-201-216-207.compute-1.a...	52.201.216.207	launch-wizard-3	172.31.59.72

Figure 3. Running Instances in the EC2 Dashboard

3.2. Build the Hadoop on Amazon Virtual Cluster

There are four steps to build the Hadoop on the Amazon virtual cluster.

(1) Preparation

Update the packages, install Java in Ubuntu, and download Hadoop.

(2) Setup environment variables

Add the following script at the end of “\$HOME/.bashrc” file of each node:

```
export HADOOP_CONF=/home/ubuntu/hadoop/conf
export HADOOP_PREFIX=/home/ubuntu/hadoop
export JAVA_HOME=/usr/lib/jvm/java-7-oracle
export PATH=$PATH:$HADOOP_PREFIX/bin
```

(3) Remote SSH authorization

Execute the following command on master node to configure the remote SSH authorization.

```
$ eval `ssh-agent -s`
$ ssh-add MyFirstKey.pem
```

(4) Setup Hadoop cluster

Add “export JAVA_HOME=/usr/lib/jvm/java-7-oracle” into “hadoop-env.sh” file of master node. Add the following script into the “core-site.xml” file of master node.

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://ec2-54-209-221-112.compute-1.amazonaws.com:8020</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/ubuntu/hdfstmp</value>
</property>
</configuration>
```

ec2-54-227-103-131.compute-1.amazonaws.com:50070/dfshealth.jsp

NameNode 'ec2-54-227-103-131.compute'

Started: Tue Aug 16 15:37:06 UTC 2016
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

7 files and directories, 1 blocks = 8 total. Heap Size is 32.08 MB / 966.69 MB (3%)

Configured Capacity : 235.16 GB
DFS Used : 260 KB
Non DFS Used : 22.68 GB
DFS Remaining : 212.48 GB
DFS Used% : 0 %
DFS Remaining% : 90.35 %
Live Nodes : 8
Dead Nodes : 0
Decommissioning Nodes : 0
Number of Under-Replicated Blocks : 0

(a) Cluster status

Live Datanodes : 8

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
ec2-107-22-148-155	2	In Service	29.39	0	2.84	26.56	0		90.35	0
ec2-107-22-44-43	2	In Service	29.39	0	2.84	26.56	0		90.35	0
ec2-107-23-175-239	2	In Service	29.39	0	2.84	26.56	0		90.35	1
ec2-54-164-150-205	3	In Service	29.39	0	2.84	26.56	0		90.35	0
ec2-54-165-174-73	2	In Service	29.39	0	2.84	26.56	0		90.35	1
ec2-54-173-67-59	0	In Service	29.39	0	2.84	26.56	0		90.35	1
ec2-54-237-203-18	2	In Service	29.39	0	2.84	26.56	0		90.35	0
ec2-54-86-28-116	2	In Service	29.39	0	2.84	26.56	0		90.35	0

This is Apache Hadoop release 1.2.1

(b) Data nodes status

Figure 4. The Status of the Cluster and Data Nodes

Add the following script into the “mapred-site.xml” file of master node.

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>hdfs://ec2-54-209-74-13.compute-1.amazonaws.com:8021</value>
</property>
</configuration>
```

Copy the “hadoop-env.sh”, “core-site.xml”, “mapred-site.xml” files to the all of slave nodes.

Configure the “masters”, and “slaves” files. For the master node, its “masters” file contains the DNS URL of name node; its “slaves” file contains all the slave nodes’ DNS URL. For the slave nodes, the “masters” file is blank; the “slaves” file contains the itself’s DNS URL.

(5) Start up the Hadoop Cluster

Launch the master node, and input the following command. It will start the Hadoop cluster.

```
$ cd $HADOOP_CONF
$ start-all.sh
```

The status of the cluster and the data node can be verified by: http://MASTER_URL:50070/dfshealth.jsp (see Figure 4).

4. Performance Evaluation Results and Analysis

In order to evaluate the IO performance of the cloud storage scheme, we test the reading/writing throughput of different size of SEG-Y files on different scale of parallel (1, 2, 4, 8, 16 data node(s)) for 5 times. The reading and writing performance evaluation results are listed in Table 3 and Table 4, respectively.

Table 3. Reading Performance Evaluation (unit: gb/s)

	1 data node		2 data nodes		4 data nodes		8 data nodes		16 data nodes	
	avg	std	avg	std	avg	std	avg	std	avg	std
64MB	105.50	33.03	205.32	49.38	254.04	37.84	414.79	11.67	705.64	14.32
128MB	101.54	13.93	202.15	76.71	336.15	78.41	463.45	124.95	764.79	116.35
256MB	92.86	5.36	180.52	55.65	377.59	88.53	609.99	92.98	1017.62	87.19
512MB	126.02	24.69	232.04	76.05	334.72	46.02	769.74	104.79	1358.67	95.67
1GB	97.15	37.76	195.66	43.95	438.39	98.29	625.70	73.09	1087.35	68.24
2GB	105.76	13.08	154.98	27.03	309.70	38.82	459.58	59.53	749.56	56.27
4GB	80.76	18.43	162.27	12.65	269.43	17.35	413.24	39.89	698.35	35.21
8GB	81.82	15.33	146.42	13.46	245.23	10.07	386.25	33.83	556.19	29.64
16GB	56.32	23.00	116.22	43.72	238.15	48.22	341.24	27.91	521.73	24.33

From Table 3 and Table 4, we can see that the reading and writing performances vary slightly with the different the size of data file, while they vary obviously with different parallel scale (number of data nodes). In order to analysis the influence factors (data size and parallel scale) to the IO performance clearly, we draw the Figure 5 and Figure 6 using the data listed in Table 3 and Table 4 respectively.

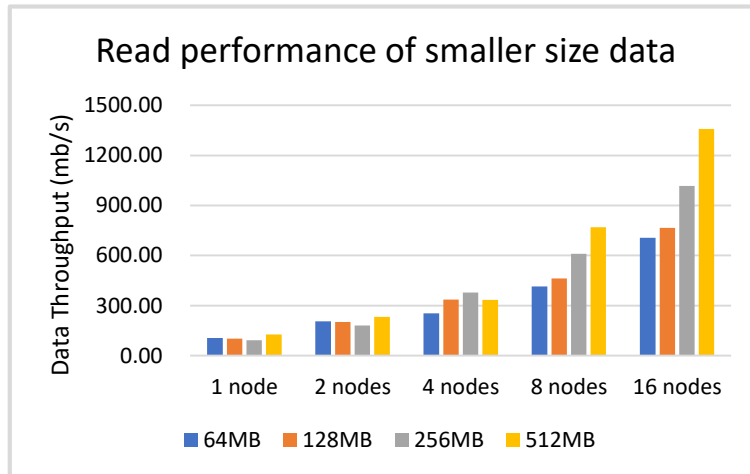
Table 4. Writing Performance Evaluation (unit: gb/s)

	1 data node		2 data nodes		4 data nodes		8 data nodes		16 data nodes	
	avg	std	avg	std	avg	std	avg	std	avg	std
64MB	25.17	5.61	36.78	8.61	53.68	10.80	89.21	19.73	132.60	18.24
128MB	26.84	4.49	38.18	15.97	58.87	17.11	62.85	20.89	92.34	19.17
256MB	24.45	3.82	36.10	6.52	53.19	5.71	76.53	9.04	102.38	8.99
512MB	22.99	0.32	38.82	2.34	56.54	8.69	75.30	11.10	101.19	10.37

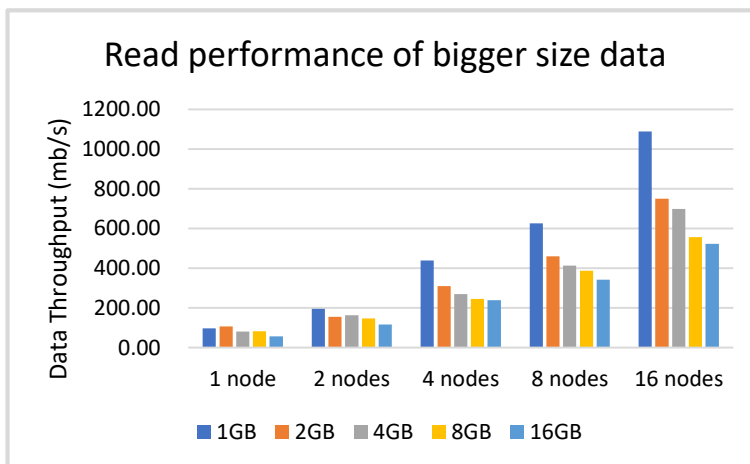
1GB	24.25	0.59	34.89	3.09	53.27	6.51	76.09	6.56	103.95	6.19
2GB	25.19	2.22	33.56	0.87	50.63	4.47	67.74	5.94	88.84	5.73
4GB	24.26	1.02	36.47	1.16	52.69	4.51	72.29	10.79	94.33	10.12
8GB	24.20	1.40	37.02	0.85	52.46	2.64	68.43	2.21	88.65	2.20
16GB	22.58	3.06	30.81	10.47	47.27	8.24	54.04	13.63	69.86	12.49

From Figure 5, we can see the reading throughput varies from 100 gb/s to over 1300 gb/s. Reading distributed data in parallel can improve the reading performance obviously. The reading performance for the data size of 512MB and 1GB is better than the others, especially in the case of 16 data nodes. For comparing, a typical access reading throughput of the distributed parallel storage system of industry is 700mb/s [4]. The public cloud storage for seismic big data is suitable for the oil and gas industry.

From Figure 6, we can see the writing throughput varies from 20 gb/s to over 130 gb/s. Writing distributed data in parallel can improve the writing performance to some extent. The writing performance for the data size of 64MB and 1GB is better than the others, especially in the case of 16 data nodes.

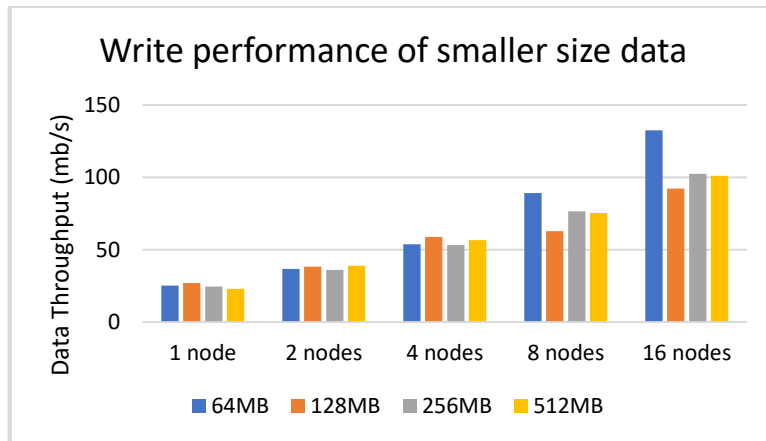


(a) smaller size data

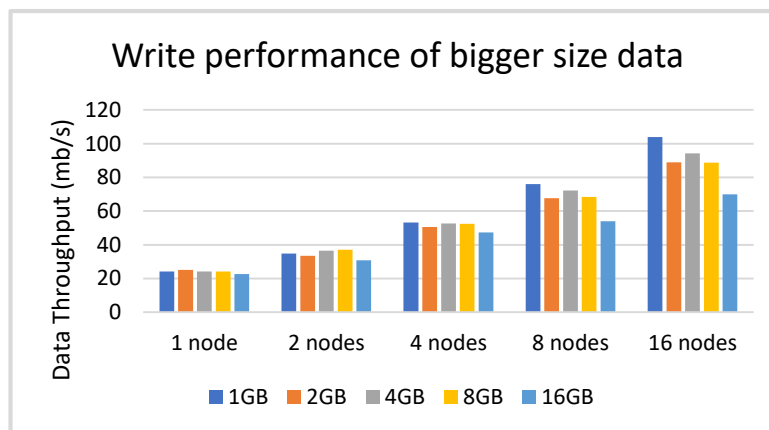


(b) bigger size data

Figure 5. Reading Performance of the Cloud Storage for Seismic Big Data



(a) smaller size data



(b) bigger size data

Figure 6. Writing Performance of the Cloud Storage for Seismic Big Data

To focus on the influence of parallel scale to the IO performance, we calculate the average and standard derivation of the reading/writing throughput of different size of data file (64MB~16GB). The results are listed in the Table 5.

Table 5. Average IO Performance of the Cloud Storage For Seismic Big Data (unit: gb/s)

	1 data node		2 data nodes		4 data nodes		8 data nodes		16 data nodes	
	avg	std	avg	std	avg	std	avg	std	avg	std
Read (mb/s)	94.19	19.71	177.29	35.63	311.49	67.51	498.22	139.88	828.88	272.58
Write (mb/s)	24.44	1.86	35.85	5.28	53.18	4.34	71.39	6.20	97.13	5.61

From Table 5 we can see the average reading throughput is much better than the writing, perhaps because the HDFS is designed for the aim of “Once write, read many times”. However, the standard derivation of reading throughput is worse than writing, which indicates the reading throughput is more rely on the IO balance within the cluster.

To investigate the scalability of the cloud storage for the seismic big data, we calculate the IO speedup and speedup efficiency defined by Equations (1)(2), and draw the result in the Figure 7.

$$Speedup_{N_nodes} = \frac{Throughput\ of\ N_nodes}{Throughput\ of\ one_nodes} \quad (1)$$

$$Speedup\ efficiency_{N_nodes} = \frac{Speedup_{N_nodes}}{N} \quad (2)$$



Figure 7. The Speedup and Speedup Efficiency of Reading and Writing Performance of the Cloud Storage for Seismic Big Data

From Figure 7 we can see the speedups of reading and writing both increased when more data nodes were used. It is good news for us that the reading speedup increased much faster than the writing because we read the seismic data much more frequently than write. However, the speedup efficiency decreased when more data nodes were used. This result indicates we may choose the best cluster size for our public cloud storage scheme for the seismic big data by balancing the throughput and speedup efficiency.

5. Conclusion

A public cloud storage scheme for the seismic big data is proposed and implemented based on the Amazon EC2 cluster and Hadoop. A comprehensive IO performance evaluation have been taken on the virtual cloud clusters which have different scale, say 1, 2, 4, 8, and 16 nodes. The results show that the Hadoop can run on Amazon EC2 cluster smoothly, and the proposed cloud storage scheme could provide the pretty good throughput for reading and writing, when only 16 low-performance data nodes employed in the cluster. Higher IO performance could be achieved by adding more data nodes into the cluster, which is convenient by renting more virtual servers from Amazon Cloud. The proposed public cloud storage scheme has advantages of high IO performance, good scalability, and is suitable for the seismic big data storage and access.

6. Future Work

The proposed public cloud storage scheme for the seismic big data does not optimized for the special structure of seismic data. It reads file header first, and then finding which part of data in a seismic file should be read. Introduction an index for each seismic data file may improve the reading performance further. Some slower data node slows down the

reading speed, which is indicated by the high standard deviation of reading performance. A smarter load-balance algorithm will be studied in our future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61273179, No.61673006), and the Science and Technology Research Project of Education Department of Hubei Province of China (No.D20131206, No.B2016034, No.20141304).

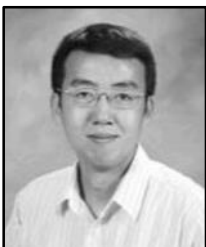
References

- [1] I. Tsvankin, J. Gaiser, V. Grechka, M.V.D. Baan and L. Thomsen, "Seismic anisotropy in exploration and reservoir characterization: An overview", *Geophysics*, vol.75, no.5, (2010), pp. 75A15-75A29.
- [2] J. Ba, J. M. Carcione, Q. Du, H. Zhao and T. Muller, "Seismic Exploration of Hydrocarbons in Heterogeneous Reservoirs: New Theories, Methods and Applications", Elsevier, Amsterdam, (2014).
- [3] M.W. Norris and A.K. Faichney, Editors, "SEG Y rev1 Data Exchange format", Society of Exploration Geophysicists, Tulsa, (2002).
- [4] I. Tayir, X. Yang, X. Song and X. Tao, "Analyzing for the storage effect to the big data seismic processing cluster efficiency", *Information Technology*, vol. 3, no. 3, (2016), pp. 195-198.
- [5] Y. Liu, Q. Shao and S. Peng, "Distributed Seismic Data File System", *Computer Technology and Development*, vol. 25, no. 11, (2015), pp. 209-212.
- [6] Y. Li, N. Zhou, G. Zhang and F. Wang, "Seismic Data Management and Service Applied in Cloud Computing Environment", *Technology for Earthquake Disaster Prevention*, vol. 10, Suppl., (2015), pp. 811-817.
- [7] D. Jin, X. Zhuang, Q. Wang, X. Cao and Z. Wang, "Application of Storage Framework Model in Seismic Big Data", vol. 25, no. 2, (2016), pp. 45-51.
- [8] F. Bugiotti, F. Goasdoué, Z. Kaoudi and I. Kaoudi, "RDF data management in the Amazon cloud", *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, New York, USA, (2012) March 26-30.
- [9] Amazon EC2, <https://aws.amazon.com/ec2/>.
- [10] S. Yi, A. Andrzejak and D. Kondo, "Monetary Cost-Aware Checkpointing and Migration on Amazon Cloud Spot Instances", *IEEE Trans. on Services*, vol. 5, no. 4, (2012), pp. 512-524.
- [11] Hadoop, <http://hadoop.apache.org/>.
- [12] Y. Lee and Y. Lee, "Toward Scalable Internet Traffic Measurement and Analysis with Hadoop", *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 1, (2013), pp. 6-13.
- [13] K. Shvachko, H. Kuang, S. Radia and R. Chansler, "Hadoop Distributed File System", 2010 IEEE 26th Symposium on Mass Storage System and Technologies, Incline Village, NV, USA, (2010) May 3-7.

Authors



Jie Xiong, he received his Ph. D. Degree in Geophysics and Information Technology from China University of Geosciences in 2012. He has been a visiting scientist in Department of Computer Science and Engineering, Mississippi State University, USA, and an associate professor in School of Electronics and Information, Yangtze University, China. His research interests include cloud computing, scientific visualization, and applied geophysics.



Song Zhang, he received his M.S and Ph. D. Degree in Computer Science from Brown University in 2000, 2006, respectively. He has been an associate professor in Department of Computer Science and Engineering, Mississippi State University, USA. His research interests include scientific visualization, data analysis, medical imaging, and computer graphics.