Resource management model based on cloud computing environment

Ahyoung Kim[1], Junwoo Lee[1] and Mucheol Kim[2]

Abstract

In this article, we propose the dynamic resources management model in a cloud computing environment. For monitoring the certain resource, we should utilize not only a cloud management module but also a network management module. However, it is difficult to check the duration time and to observe the digested information about the resources. To investigate these problems in a cloud computing environment, we designed and deployed the cloud service infrastructure based on open-source software, namely, CloudStack. The proposed model regularly stores the usage data for computing resources based on Hadoop and HBase. In addition, our model analyzes the raw data for virtual machines and makes an effective recommendation regarding the consumption of computing resources.

Keywords

Cloud computing environment, resource management, Hadoop, HBase, simple network management protocol

Introduction

The concept of a cloud computing is one of the emerging issues in the information technology (IT) industry.[1,2] It enables users to migrate their data and computation to a remote location with minimal impact on system performance.[3] Typically, this provides a number of benefits which could not be realized. Whether using a specific application, a set of tools, or Web services, Clouds provide access to a potentially vast amount of computing resources in an easy and user-centric way.[4] We have investigated such an interface within grid systems through the use of the Cyberaide project.[5] There are a number of underlying technologies, services, and infrastructure-level configurations that make cloud computing possible. One of the most important technologies is the use of virtualization.[6] Then, Gulati et al.[7] discuss resource management challenges and techniques with the perspective of VMWare's offerings. Some researchers also focused on the data center networking for seamless data transfer.[8,9]

Accordingly, major companies such as Amazon[10] and Google[11] have announced private or public cloud computing environments for fault tolerant and effective business environments.[12] Additionally, other companies such as HP,[13] IBM,[14] and Dell[15] are supporting hardware products and services related to cloud computing environment.[16]

In spite of the appearance of such commercial software and hardware based on cloud computing environment, some users have continuously requested the development based on open-source software. First, CloudStack[17] from Citrix[18] is a well-known product based on open-source software and is distributed by Apache License. CloudStack already has several reference sites and includes such strengthens as extensibility and centralized management for resources in a cloud computing environment. Citrix has decided on license

[1]Electronics and Telecommunications Research Institute, Daejeon, Korea
[2]Department of Media Software, Sungkyul University, Anyang, Korea

Corresponding author:
Mucheol Kim, Department of Media Software, Sungkyul University, 53, Sungkyuldaehak-ro, Manan-gu, Anyang, Gyeonggi-do 14097, Korea.
Email: mucheol.kim@gmail.com

policies for CloudStack from the General Public License (GPL) to an Apache License. It provides more opportunities for open-source developers to join this project. Alternatively, many major projects are based on the OpenStack project.[19] In addition, many open-source developers have been attending to these projects to develop the newest technologies related to each topic. We choose CloudStack for constructing resource management model which is a stable and fast platform. OpenStack has a number of sub-projects, so it needs several debugging when installing the system. CloudStack can offer stable and fast development environment for complex and large project services based on Citrix's practical skill.

In section "Cloud service infrastructure," we describe a cloud service infrastructure which should be providing small- to mid-sized companies with detailed information on a cloud computing environment. Section "Resource management model" introduces some problems that may be found while a cloud service infrastructure is serviced, and the proposed approach is described in section "Experiments." Finally, in section "Conclusion," the contents of this article are summarized and future work is discussed.

Cloud service infrastructure

The proposed mobile log management model which is based on cloud infrastructure should be aggregating the sensed data from typical mobile devices. The collected data should be transported to cloud frameworks, and it is analyzed with Hadoop-based big data system. It could provide convenience to network availability of mobile devices. Moreover, we can obtain a sensed data from user activities easily. Because of these benefits, it can be utilized in mobile application development and marketing initiatives.

The proposed cloud infrastructure consists of about 60 server nodes based on the CloudStack project, which is open-source software with an Apache License, with additional storage servers, firewalls, and monitoring servers. Storage servers generally have a parallel architecture for supporting the integration when abnormal accidents are occurred. Firewalls support secure operation among virtual machines (VMs) and have additional operations such as port forwarding and network address translation (NAT) to protect a private network from attacks. Monitoring servers allow a system manager to completely observe the status of the VMs and network resources in the cloud service infrastructure (Figure 1).

A cloud service infrastructure has a Gigabit network capacity by default to communicate with server nodes including the cloud manager. Nevertheless, for the speed of a Gigabit network, it supports a 10-Gb
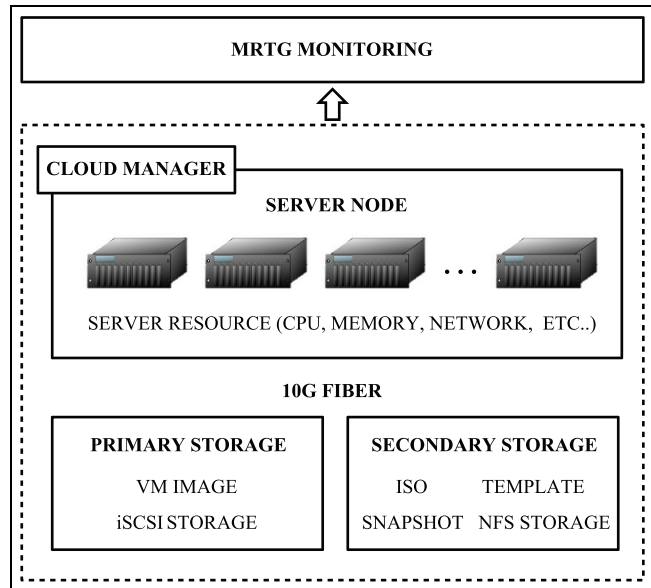

Figure 1. Cloud service infrastructure.

network for communicating between server nodes and the storage servers, particularly to prevent a bottleneck problem from network congestion in advance.

Server node

The proposed approach has a cloud manager and 60 server nodes according to the guidelines of the CloudStack project. A set of server nodes in the cloud service infrastructure supports VMs created by the user, allowing available computing resources to be allocated within the required time. CloudStack supports various types of VMs, such as ESX, XenServer, KVM, BareMetal, and OVM, by default. However, XenServer is recommended as a type of VM in a cloud service infrastructure since we have to consider the stability of the system itself. Table 1 shows the hardware specifications of a server node.

Storage server

The proposed approach has two storage servers for physically storing VMs created by users. First, the master storage server stores VMs and deals with the request for VMs from users in real time. Therefore, the master server has a network capacity of 10 Gb and multipath input/output (MPIO) capability to roll back the problems from the system or network as soon as possible (Table 2). Alternatively, a secondary storage server stores ISO files, templates, and VM snapshots to manipulate VMs easily using a network file system (NFS) technology.

Table 1. Hardware specifications of server node.

CPU	Intel Xeon CPU E5440@ 2.83 GHz
Memory	PC3-10600 4 GB * 4EA
HDD	147 GB SAS 10K
NIC	Gigabit Ethernet * 4EA

HDD: hard disk drive; NIC: network interface card.

Table 2. Hardware specifications of storage server.

HDD	600 GB (10K) * 8EA
iSCSI	Multipath input/output

HDD: hard disk drive; iSCSI: Internet Small Computer System Interface.

Monitoring system

Users are generally able to activate and deactivate VMs using a cloud management console. As an aside, it does not have a monitoring capability and cannot control VMs operated by the user. Thus, the system manager has to rely on an additional monitoring tool such as a Multi Router Traffic Grapher (MRTG)[20] with a simple network management protocol (SNMP) to look into the status of VMs. The proposed monitoring tool is for observing the traffic load on network links.

Resource management model

Basically, the proposed management console, which is supported by CloudStack, is able to create, delete, copy, and paste VMs in a cloud computing environment. However, the system manager cannot monitor computing resources which are used by VMs, but does check the status of VMs such as the on/off status at the cloud management console. A network management tool based on SNMP is able to report the total status of the systems to the system manager, as well as detailed information on the current status of the VM selected by the system manager, such as the usage rate of the CPU, disk, and network. While a cloud service infrastructure has been serviced, we have found the possibility that the information generated by the network management tool can leverage the cloud management console and control the VMs and the other resources dynamically for autonomic process, which it has previously done statically. The proposed model is able to show detailed information on the VMs, such as MRTG. Additionally, the cloud management console is able to make a decision on the limitation of resources that VMs can use within the availability of the computing resources. If the current usage rate for a certain VM is very close to the limitation set by the system manager, it could generate a policy for the autonomous process using a snapshot image. It is stored in the secondary storage server

Table 3. Hardware specifications of one-unit server.

CPU	Intel Xeon CPU 5130@ 2.00 GHz
Memory	PC3-10600 2 GB * 4EA
HDD	72 GB SAS 10K
NIC	Gigabit Ethernet * 2EA

HDD: hard disk drive; NIC: network interface card.

without a system manager interrupt to allow stable service for a VM.

The purpose of the resource management model is to analyze raw data and report the statistical information regarding the VM usage amount for the computing resources within the limitation of the resources.

For an analysis of the current status, the raw data for a certain VM must be gradually and completely collected and stored in a kind of database. As an aside, MRTG does not save the raw data collected using SNMP, and thus, we have no choice but to design a log analysis framework separately. A resource management model makes use of a log analysis framework to collect and save the raw data from communication among VMs.

Log analysis framework architecture

For a log analysis framework, we decided to use a distributed file system based on the Hadoop distributed file system (HDFS)[21] and a database based on HBase[22] since HDFS guarantees extensibility based on Google file system (GFS) and HBase is a database on top of HDFS for NoSQL application.[23,24] HDFS is the primary storage system used by Hadoop applications.[25]

HDFS is suited for the storage of large files, and HBase provides fast record lookups (and updates) for large tables. Finally, a log analysis framework (in Figure 2) prepares the MapReduce programming model[26] to support various types of statistical analyses for the usage amount of computing resources.[27,28]

For a resource management model, HDFS based on Hadoop, makes use of 80 one-unit servers with CentOS 5.5 to create a distributed file system. Table 3 shows the hardware specifications of a one-unit server.

For a statistical analysis, a resource management model creates two tables for storing a set of user uniform resource identifier (URI) data per VM URI and a set of collected raw data for each VM server such as CPU, disk, and network usage data; they are described in Tables 4 and 5.

User scenario

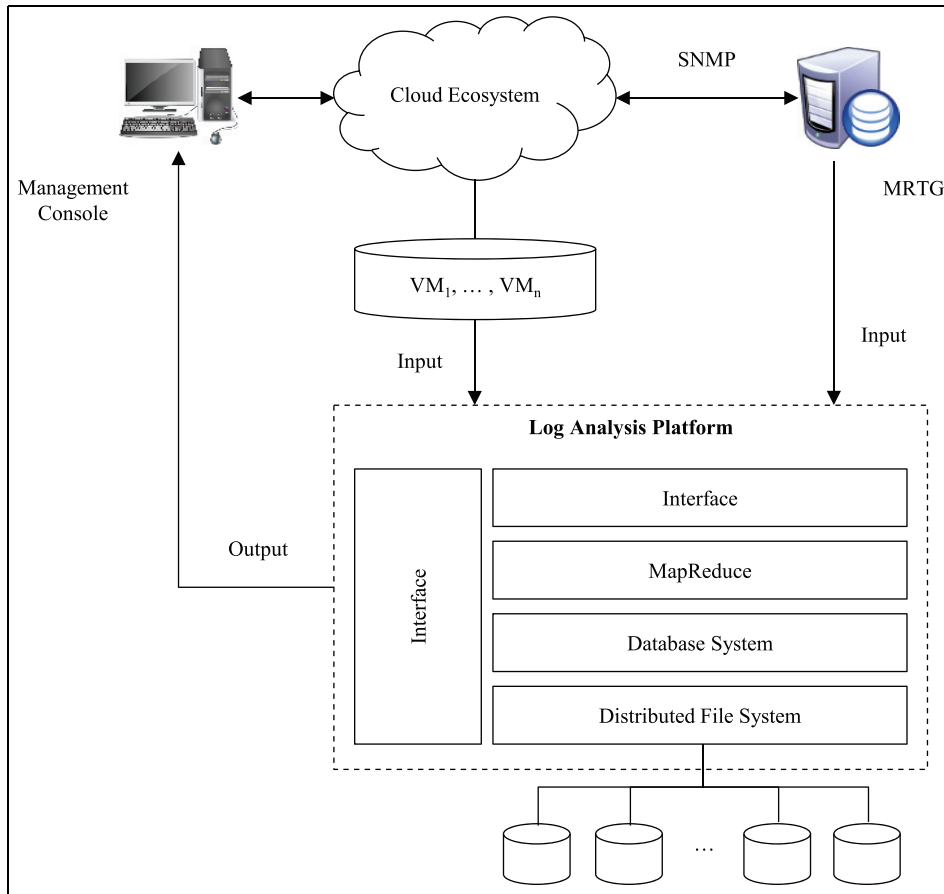The proposed approach suggests a process for resource management for maintaining stability of the mobile

Figure 2. Log analysis framework architecture.

Table 4. Set of user URI data per VM URI.

VM URI	Timestamp	User URI
$data_1$	t_1	$user_1$
…	…	…
$data_j$	t_j	$user_k$

VM: virtual machine; URI: uniform resource identifier.

cloud services. Specific descriptions are separated into four steps, which are shown in Table 6.

First, a resource management model allows a system manager to check the current status with statistical analysis from the log analysis framework. In this case, the system manager no longer has to look into the graph from the network management system, as in MRTG, if they want to see the CPU status of a certain VM.

Second, a resource management model helps a system manager to generate a rule for the current limitation of available computing resources from the log analysis framework. For example, if the user requests and creates VMs using a quad core CPU initially, the system manager can decide to scale down from a quad core to a dual core for the number of CPUs based on the statistical report from the log analysis framework. Additionally, the resource management model provides an opportunity for the system manager to use a script allowing the cloud management console to scale the VMs up or down automatically based on a firewall policy.

Finally, the resource management model allows the system manager to generate a policy for emergencies in terms of the service stability of the VMs. For example, the system manager can guess how much computing resources a certain VM consumes at peak and idle times based on statistical information from a log analysis framework. Normally, this VM operates quite well; however, the usage rate of this VM may increase rapidly owing to some company events. In such a case, the system manager may be unable to anticipate this occurrence, and therefore the service from this VM may cease owing to network congestion, such as from a denial-of-service (DDoS) attack. The resource management model allows the system manager to avoid this situation by automatically scaling out the VM having a problem without the system manager's order. Of course, the
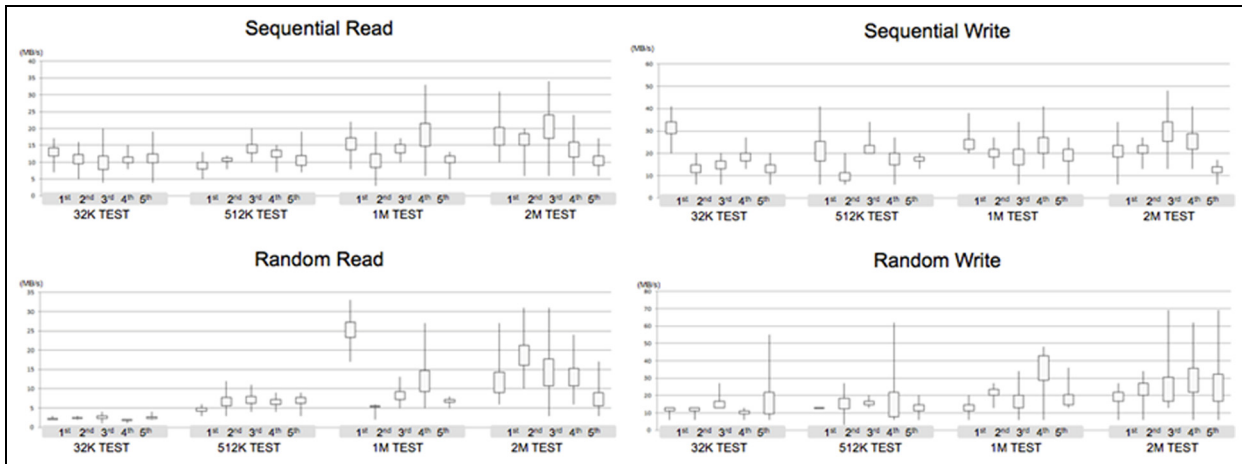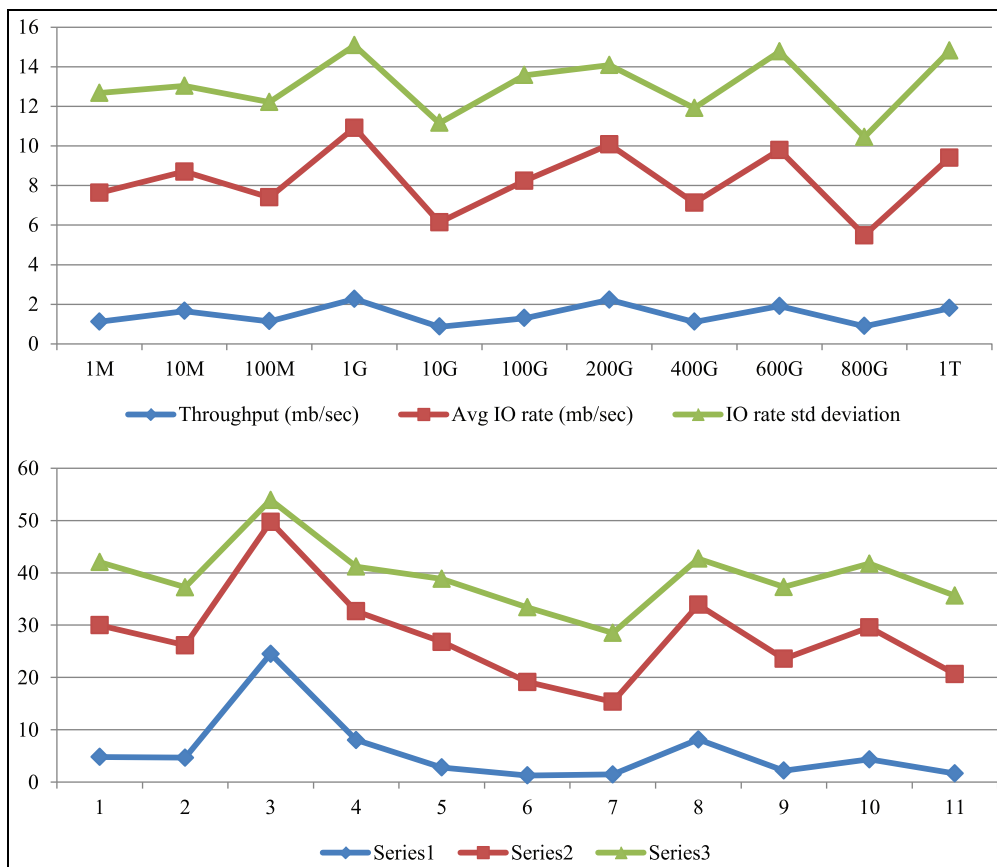
Figure 3. Performance of CloudStack.

Figure 4. Performance evaluations of Hadoop infrastructure.

performance graph shows that there is no time increasing according with the data size. We conclude that Hadoop system is sufficient to deal with the big data.

Conclusion

In this article, a cloud service infrastructure deployed by Electronics and Telecommunications Research Institute (ETRI) is introduced and a resource management model with a log analysis framework is described. The cloud service infrastructure based on the CloudStack project is supporting the cloud services to small- to mid-sized companies in a practical manner and to investigate the possibility of cooperation between a cloud service infrastructure and network management system.

The proposed approaches are mobile log aggregation and analysis framework which are based on cloud environments. It provides high availability with duplicated log aggregations.

By the way, relational database (RDB)-based log analysis system is not proper to real-time processes. So, we designed the analysis architecture with NoSQL-based MongoDB for sensed data management. Due to the replica sets of MongoDB, we could improve the availability with fail-over policy. So, the proposed approach provides scale-out free for extending the mobile log management frameworks.

In order to performance evaluation, we applied the proposed resource management model to the mobile log collection system. Because we need to aggregate the various raw data in the mobile cloud environments. The next phase of this research is to finalize the development of this model to operate VMs at the cloud management console without the system manager's interrupt. Additionally, we are going to survey a use case for cloud management and generate an experiment for the cloud service infrastructure.

Declaration of conflicting interests

Funding

References

1. Zhang Q, Cheng L and Boutaba R. Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl* 2010; 1(1): 7–18.
2. Armbrust M, Fox A, Grith R, et al. A view of cloud computing. *Commun ACM* 2010; 53(4): 50–58.
3. Wang L, Von Laszewski G, Younge A, et al. Cloud computing: a perspective study. *New Generat Comput* 2010; 28: 137–146.
4. Kim M and Park S. Trust management on user behavioral patterns for a mobile cloud computing. *Cluster Comput* 2013; 16(4): 725–731.
5. Von Laszewski G, Younge A, He X, et al. Experiment and workflow management using cyberaide shell. In: *Proceedings of the 4th international workshop on workflow systems in e-science (WSES 09) in conjunction with 9th IEEE international symposium on cluster computing and the grid*, Shanghai, China, 18–21 May 2009. New York: IEEE.
6. Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization. In: *Proceedings of the 19th ACM symposium on operating systems principles*, Bolton Landing, NY, 19–22 October 2003, pp.164–177. New York: ACM.
7. Gulati A, Shanmuganathan G, Holler A, et al. Cloud scale resource management: challenges and techniques. In: *Proceedings of the 3rd USENIX workshop on hot topics in cloud computing*, Portland, OR, 14–17 June 2011. New York: ACM.
8. Abts D and Felderman B. A guided tour of data-center networking. *Commun ACM* 2012; 55(6): 44–51.
9. Bari MF, Boutaba R, Esteves R, et al. Data center network virtualization: a survey. *IEEE Commun Surv Tutor* 2013; 15(2): 909–928.
10. Amazon Web Services, http://aws.amazon.com/ec2/
11. Google Cloud, http://cloud.google.com/
12. Singh S and Jangwal T. Cost breakdown of public cloud computing and private cloud computing and security issues. *Int J Comput Sci Inform Tech* 2012; 4(2): 17–31.
13. HP Cloud, http://www.hpcloud.com/
14. IBM Cloud, http://www.ibm.com/cloud-computing/us/en/
15. Dell Cloud Computing, http://content.dell.com/us/en/enterprise/cloud-computing
16. Younge A, Henschel R, Brown J, et al. Analysis of virtualization technologies for high performance computing environments. In: *Proceedings of the IEEE international conference on cloud computing*, Washington, DC, 4–9 July 2011. New York: IEEE.
17. Apache CloudStack, http://cloudstack.apache.org/
18. Citrix, http://www.citrix.com/lang/English/home.asp
19. OpenStack Project, http://openstack.org/
20. The Multi Router Traffic Grapher (MRTG), http://oss.oetiker.ch/mrtg/
21. Hadoop File System, http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
22. HBase, http://hbase.apache.org/
23. Madden S. From databases to big data. *IEEE Comput Soc* 2012; 16(3): 4–6.
24. Padhy RP, Patra MR and Satapathy SC. RDBMS to NoSQL: reviewing some next-generation non-relational database's. *Int J Adv Eng Sci Technol* 2011; 11(1): 15–30.
25. Ghemawat S, Gobioff H and Leung S. The google file system. In: *Proceedings of the 19th ACM symposium on operating system principles*, Lake George, NY, 19–22 October 2003. New York: ACM.
26. MapReduce Programming model, https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
27. Dean J and Ghemawat S. MapReduce: simplified data processing on large clusters. In: *Proceedings of the 6th symposium of operating systems design and implementation*, San Francisco, CA, 6–8 December 2004.
28. Lee KH, Lee YJ, Choi HS, et al. Parallel data processing with MapReduce: a survey. *ACM SIGMOD Rec* 2011; 40(4): 11–20.