International Journal of Smart Home
Vol. 8, No. 6 (2014), pp. 27-36
http://dx.doi.org/10.14257/ijsh.2014.8.6.03

Design of Simulator for Cloud Computing Infrastructure and Service

Changhyeon Kim, Junsang Kim and Won Joo Lee [*]

Dept. of Computer Science and Engineering,
Hanyang University (ERICA Campus), Korea
Dept. of Computer Science,
*Inha Technical College, Korea **
{ctcquatre, kjspbe}@hanyang.ac.kr, wonjoo2@inhatc.ac.kr

Abstract

Cloud computing that provides computational resource, platform and software as services is operated in cluster that is comprised of hundreds or thousands of servers. Thus, cloud computing providers may use various policies such as complex resource provisioning, allocation, configuration, and so on, to efficiently manage huge resources. In this environment, it is difficult to customize circumstance and to test and evaluate a hypothesis independently. To overcome these limitations, we provide SimCloudIS: simulator for cloud computing infrastructure and service. The SimCloudIS provides cloud system components such as the datacenter, host, VM, application, network and scheduling polices that are extendable and managed in independent layers. The network component supports packet based network communication which is based on the real network model, thereby facilitating simulation of internet based services. For evaluation, SimCloudIS provides a monitor that observes the status of all components and records them. The usefulness of SimCloudIS is demonstrated by involving a case study for dynamic allocation and relocation of VMs in a cloud environment and analyzing the result.

Keywords: *Cloud computing, Simulator, Simulation*

1. Introduction

Cloud computing provides services such as computation resource, platform, software, and so on based on the pay-as-you-go model. Therefore, a service user uses these services in needs without the burden of purchasing or operating hardware or software. For this advantage, various IT companies have moved on to cloud computing services provided by IT vendors such as Amazon, Google or Microsoft instead of operating their own computing center [1].

A cloud cluster providing cloud computing services is comprised of hundreds or thousands of servers in general. Resources of these servers are virtualized through a virtualization technique and managed in logical units that are unrestricted by boundaries of physical resources. For management of virtualized resources, various policies have been proposed for enhancing efficiency in terms of using computational resource, storage, network or energy. For examples, there are some policies; resource allocation to meet SLA (Service Level Agreement) specified by the user and resource placement to map virtualized resources with various size by user requirements into the minimum physical resources [2-3].

However, evaluating the performance of a policy is difficult to achieve in real cloud computing environment due to following reasons. First, it is difficult to ensure independent

[*] Corresponding Author

ISSN: 1738-9968 IJSH

execution of a policy to be evaluated. The majority of cloud service providers use their own resource and service management policies. Therefore, to independently perform a policy in cloud cluster without interference with other policies of cloud service providers, an individual cloud cluster is required. However, this requires a considerable amount of cost. Second, it is difficult to construct a customized environment required to evaluate the policy. Cloud clusters providing cloud computing service have different configurations for clusters and networks as well as various resource capabilities of servers, and the configurations and resource capabilities are not customizable factors in general. Third, it is difficult to play a certain scenario to test a policy. Some policies may need certain workload patterns, or I/O bottleneck occurrence of network, or storage in performance evaluation. In a cloud service cluster, causes that may lead to performance degradation, such as, an I/O bottleneck, can be handled by the cloud service manager or the administrator as soon as they are detected.

A viable alternative to overcome the above difficulties is to use the simulator. The simulator opens up the possibility of evaluating a policy or hypothesis in a controlled environment with lower cost and efforts than in real cloud clusters. Also, the simulator can be used as a test bed for tuning or optimizing the performance of a policy based on the characteristics of a circumstance before applying the policy to a real cloud cluster.

In this paper, we propose SimCloudIS : Simulator for Cloud computing Infrastructure and service. The SimCloudIS provides two main functions. First, network communication among datacenters, hosts, VMs(virtual machines) and applications are achieved by transmission/reception of network packets based on real network model. In such real network environments datacenters, hosts, VMs and applications have a unique network address in the hierarchical format, and network layers connected with them are disjoined according to the network address. The bandwidth and delay of each network can be controlled. These features allow constructing federated clouds that several data centers are connected through internet, and also facilitating implementation of cloud computing service on internet-driven method. Second, the application model is extendable. The application is the main agent that consumes VM resources and generates network traffics. Therefore, similarity between implemented application and real application has an impact on the reality of a simulation. SimCoudIS enables applications to perform operations that are similar to real applications by providing application scheduler, the periodic or aperiodic event timer and methods to use VM resources.

The rest of this paper is organized as follows. Section 2 reviews some simulators for cloud computing. Section 3 describes the architecture, model and details of SimCloudIS. In Section 4, we simulate a scenario that can happen in cloud computing environment and analyze its results. Section 5 concludes the paper.

2. Related Work

In cloud computing computation, memory and network resources are composed logically, and various techniques and policies are used to manage them. Additionally, a variety of services management policies can be considered for use since the cloud computing provides many different types of services. This makes it difficult to evaluate a new policy or hypothesis. Accordingly, a few simulators are proposed to relieve the difficulty in the simulation of cloud computing service or infrastructures.

CloudSim [4], a well-known cloud computing simulator, provides frameworks to construct cloud computing infrastructures and implement cloud computing services. It is composed with several layers that are the lowest layer including discrete event simulation engine and upper layers that include cluster, storage, networks, traffic profiles, resource provisioning and application service models. However, since its network model has some differences with the real network model, CloudSim may not be suitable for a network sensitive simulation such as the message passing application that needs to handle network packet. NetworkCloudSim [5] enhances network model of CloudSim by employing the concept of network packet. In NetworkCloudsim, the packet is classified into either the network packet or the host packet according to the packet delay. The network packet has some delay according to the packet size and the bandwidth of network connected with the hosts. The host packet, however, has little delay because it travels through the virtual network of a host.

GreenCloud [6] is an extension of NS2, a packet level network simulator, and similarly emulates packet processing of the real network model. Therefore, GreenCloud provides a highly accurate performance measurement in terms of network communication. Also, it provides a way to estimate the energy consumption based on the power model that reflects CPU load of servers and network equipments. However, elaborate packet processing of GreenCloud gives rise to computational overhead. In addition, there is a lack of consideration for virtualization andresource management.

Nunez *et al.* proposed iCanCloud simulator [7-8]. It is built on SimCAN, a simulation framework for modeling HPC architectures. iCanCloud can predict the trade-offs between the costs and the performances of applications running on a cloud. Since they focus on the simulation of Amazon EC2 service, the prediction for trade-off is based on the hardware, VM instance specifications of Amazon EC2 and pay-as-you-go manner.

3. Design and Implementation of SimCloudIS

SimCloudIS are designed as independent layers that models cloud system components to provide cloud developer with enhanced expandability. The cloud system components are operated based on events processed in the SimCloudIS core. The events are classified either into a packet or a timer. The packet event is a network event delivered only through the network layer and the timer event is a user-defined event that can occur in any component. In this section, we explain the layer design, event handling manner and other functions of the SimCloudIS in detail.

3.1. Layer Design

Datacenter consists of numerous hosts connected through the network. The host that has computational ability can host several VMs by resource virtualization. The VM provides independent operating environment with an application on top of virtualized resources.

SimCloudIS provides independent layers mapped into a datacenter, host, VM and application to enable flexible infrastructure modeling and service environment. These layers are connected through the network layer which is designed based on the packet-driven method to reflect characteristics of an internet-driven service in cloud computing. Figure 1 shows the multi-layered design of the SimCloudIS and its architecture.

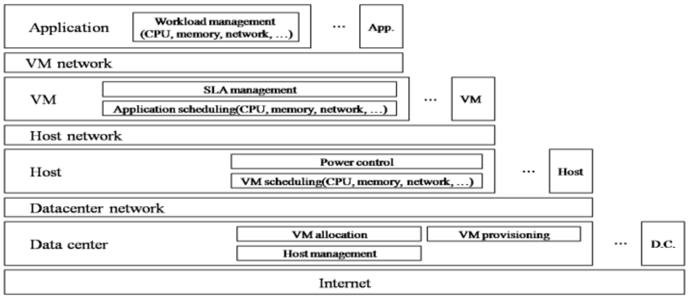

Figure 1. Design and Architecture of SimCloudIS

The Internet layer connects datacenters belonging to different domains. In general, the service provider of cloud computing and the service consumer are located on different domains. Also, federated cloud is composed of datacenters belonging to difference domains. Therefore, their communications between domains have some delay depending on the distance between domains and link bandwidth. The internet layer reflects this feature.

Datacenter layer includes functionalities that are host management, VM allocation and resource provisioning. The Datacenter connects hosts belonging to itself through network layer and commands VM migration or changes their power status by using a network message. And, it generates a map of VMs to PMs based on resource demands of VMs by a provisioning policy and resource capabilities of hosts, and relocates the VMs with migration message.

Host layer includes power controller and VM scheduler. Since the power consumption of a host depends on its resource utilization, it is desirable to calculate the power consumption based on the amount of resources used in the host. The power controller calculates the amount of resources used by VMs on a runtime and reflects it to the total power consumption of the host. The VM scheduler makes a plan for allocating host resources to VMs based on their resource demands. Although the SimCloudIS provides only time-shared VM scheduler, the VM scheduler can be extended based on the abstract VM scheduler model.

VM layer contains SLA (Service Level Agreement) management and application scheduler. The VM allocates its resources to applications based on their resource demands. If the amount of resources allocated to an application is less than the amount of its demand, SLA becomes violated. The resource allocation plan is determined by application schedule policy. The application scheduler is implemented with time-shared manner as implemented on the VM scheduler.

Application layer provides functionalities to generate workload. The workload can be generated in two manners. The first manner is use of workload traces measured on servers. Recently, several companies and research groups such as google [9] and planet lab[10] have provided workload traces measured on their servers. Therefore, application can imitate workload from these traces. Second, cloud application developer can produce workload by implementing a service application based on the abstract application model.

3.2. Implementation of SimCloudIS

3.2.1. Simulator Core: Figure 2 shows a class diagram of SimCloudIS core. SimCloudIS maintains an internal clock to represent the simulation time. The clock time increases gradually while events are processed. The type of event is classified into either a packet or a timer. The packet event is an internal event that can happen on entities (datacenter, host, VM, application, networkLayer) inherited from SimObject class. It is processed in the ProcessPacket() method at the estimated completion time calculated based on the bandwidth and the delay of network and processing time. The timer event is an external event to do pre-defined works at appointed time. The Expired() method of Timer class contains user-defined codes and is invoked at the expired time of a timer object.

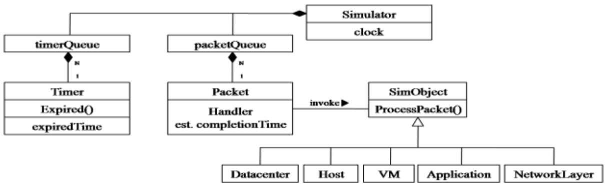

Figure 2. SimCloudIS Core Class Diagram

3.2.2. Communication: In SimCloudIS, datacenter, host, VM, application are linked through network layers and, thus, communications among them are permitted only by using a network packet. To route the packet, we assign a unique network address into each entities which needs communication. The network address is composed in hierarchal format using datacenter, host, VM and application IDs. For instance, assuming that IDs of a datacenter, a host in the datacenter, a VM in the host, and an application running on the VM are A, B, C, D, respectively, then, their network addresses become A.0.0.0, A.B.0.0, A.B.C.0 and A.B.C.D. Based on this network address format, network communication among entities are carried out in a similar manner to the real network.

The estimated completion time of packet transmission in a network layer is calculated as the equation below

$$est.completion\ time = \frac{packet\ size}{available\ BW} + delay + current\ time$$

where available BW means bandwidth that the packet can use in the network layer. A packet can travel several network layers if a source and destination of the packet belong to a different network domain. Therefore, the total transmission time of the packet is the summation of each estimated completion time calculated in all network layers where the packet has passed through.

A network layer updates available BW in real time when a new packet is detected. Therefore, as soon as a packet is generated, estimated completion time of all packets that pass through the network layers included in the route of the packet is updated.

In real virtualization environment, the hypervisor provides command set to control VM operations such as start, stop, shutdown, resume and migration. These control messages between a host and VM running on the host is delivered not through the network, but the signal channel. The SimCloudIS provides these signal channels. The signal channel is only allowed between the host and the hosting M and between the VM and the hosted application, being able to send the control message without any delay.

3.2.3. VM Migration: Figure 3 shows VM migration procedure implemented in the SimCloudIS. The VM migration procedure is divided into resource reservation, data transmission and VM restart phase.

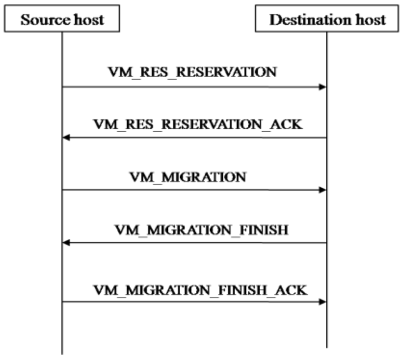

Figure 3. VM Migration Procedure

In the resource reservation phase, the destination host, into which a VM will migrate to, reserves the resources. To start migration of a VM, available resources should exist in the destination host. Therefore, in the beginning of this phase, the source host sends a VM_RES_RESERVATION message with resource demands of the VM to the destination host. On receiving the message, the destination host checks if there are sufficient number of available resources to be occupied by the VM, and then reserves the resources so that other VMs cannot occupy it. Then, the destination host sends a VM_RES_RESERVATION_ACK message with the reservation result to the source host. The source host starts data transmission phase only if the resource reservation results in VM_RES_RESERVATION_ACK message is success.

In data transmission phase, the source host sends a VM_MIGRATION message to the destination host. The VM_MIGRATION message contains information and data of the VM. The information contains status and detailed specifications of the VM. The data are memory pages of the VM. This phase generally consumes the most time in VM migration procedures because normally the memory size of VM reaches hundreds of thousands of Mbytes.

Finally, in the VM restart phase, the source host transmits dirty pages of the VM, and then changes VM status into suspended. The VM is running while memory pages are transmitted. Therefore, some of the transmitted memory pages may differ from current memory pages of the VM in terms of contents due to dirty pages. The dirty pages can continuously occur during running of the VM. Therefore, the source host suspends the VM and transmits all dirty pages that have occurred in the VM to the destination host. When all the dirty pages are transmitted, the destination host has a clone of the VM in the source host and sends a VM_MIGRATION_FINISH message to the source host. On receiving the message, the source host destroys the VM and sends a VM_MIGRATION_FINISH_ACK. After the host receiving the message resumes the clone VM, VM migration procedures is finished.

During VM migration, the source host and destination host have performance degradation due to intensive I/O operations of memory and network device. Also, available network bandwidth between the two hosts reduces drastically due to the transmission of memory pages. Therefore, the SimCloudIS diminishes the calculation capabilities of the two host’s CPU and reduces their available network bandwidth as much as network traffic that reflects the amount of memory pages and dirty pages to be transmitted.

3.2.4. Monitor: The correct and detailed performance measurement is one of several important functions that should be provided in a simulator. The SimCloudIS provides the monitor for the performance measurement. The monitor observes resource usage, allocation, status, and etc. of all registered entities and records them in detail in a certain format. However, recording all information of the entities in detail may lead to simulation overhead or waste of storage space. To prevent such problem, the monitor records information of an entity only when the information of the entity is changed. Also, the recording format can be customized to reduce information size.

4. Simulation and Evaluation

In this section, we simulate what can happen in the cluster providing the cloud computing service by using SimCloudIS and analyze the result of the simulation.

4.1. Simulation Scenario

In cloud computing, VM instances may be created dynamically by user requirements, and these instances can be scattered on hosts. If the creation and placement of VM instances are done repeatedly, then eventually, many hosts can be put in low utilization status due to inefficient placement of VMs. For this reason, some cloud providers periodically carries out server consolidation that vacate some hosts through VM placement, and turns off their power to save energy.

We simulate a scenario that VM instances are created in runtime and are relocated periodically on hosts. We operate cloud cluster for 3 hours and create a VM instance for every seven second intervals. The new VM instance is located on an arbitrary host that can accommodate it. We carry out server consolidation based on First fit decreasing algorithm for every half an hour. For evaluation, resource usage of hosts, VMs and applications and network traffic are measured at an interval of ten seconds.

In simulation, all hosts of a cluster are connected with gigabits switch and equip16 CPU cores with 5600 MIPS per core and 16 GBytes of RAM. The power model of the host is based on the power consumption measurement values [11] of Hewlett Packard's ProLiant DL580G3 equipped with eight Inter Xeon Processor 7020 and 16 GBytes of RAM. Figure 4 shows the amount of power consumption per second of the host with changes in CPU load in the power model. VM instances have different compute units and memory size according to the instance type. We set the number of compute units and memory size for VM instance types equal to them of each VM instance types provided from Amazon EC2[12], as shown in table 1. In our simulation, the type of a VM instance to be created is determined by a random variable with uniform distribution. All VM instance runs a single application that has a computational workload generated from workload trace provided from PlanetLab [10].

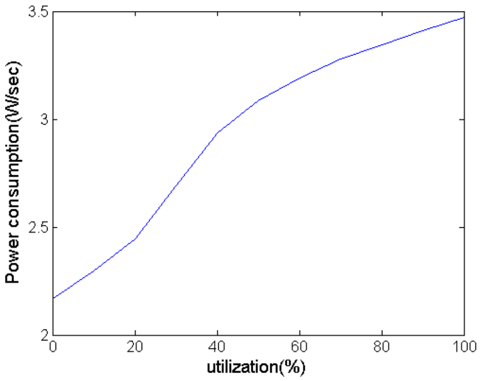

Figure 4. Power Consumption of a Host

Table 1. VM Instance Type

resources / instance type	Compute units(virtual cores)	Memory size in GBytes
Small	1	1.7
Large	4	7.5
X-large	8	15
High-CPU medium	5	1.7
High-CPU X-large	20	7

4.2. Results

The graphs in this section show total consumption, network traffic and SLA violation ratio during simulation time.

Figure 5 shows the total power consumption of hosts in the cluster. In our scenario, a single VM instance is created at every interval and is placed on an arbitrary host. Accordingly, the total power consumption increases gradually. In particular, the total power consumption increases drastically at every interval of 1800 seconds. It is because of an increase in the number of hosts powered on and increases in their CPU loads during the server consolidation time. During the server consolidation time, many VMs are migrated from the source hosts to the destination hosts, simultaneously. Thus, the number of hosts powered on is the sum of the number of source hosts and destination hosts. Also, intensive memory I/O and network I/O occur in the source and destination hosts, and accordingly, their CPU load increases. After server consolidation, the total power consumption becomes less than before the server consolidation because the source hosts are powered off.

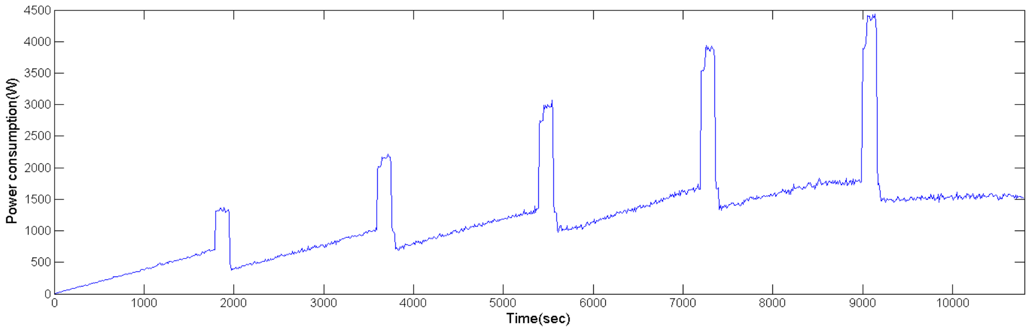

Figure 5. Total Power Consumption of Hosts in the Cluster

The total network traffic generated in the cluster is shown in figure 6. Since, in our scenario, applications generate only the computational workloads, network traffic does not occur except in those traffic related with VM migration. Therefore, the network traffic is generated only during server consolidation time. The amount of network traffic increases as repeating server consolidation. It is caused by an increase in VMs to be migrated.

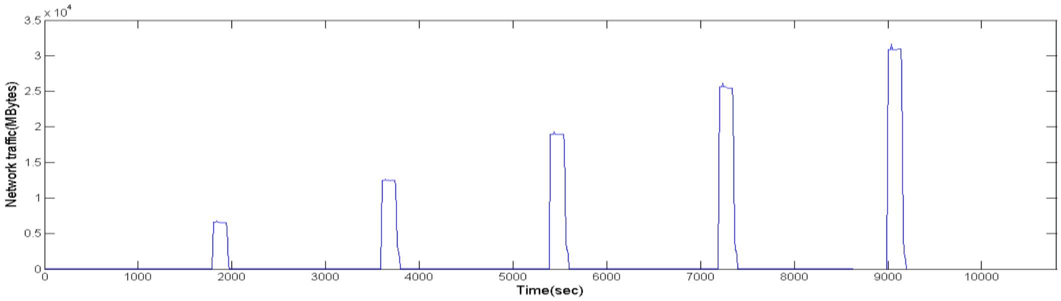

Figure 6. Total Network Traffic of the Cluster

Figure 7 shows the ratio of SLA violated VM instances to the total VM instances. The SLA violation in a VM occurs when the VM does not have enough resources to meet application’s resource demand. In figure 7, the causes of SLA violation with respect to the time of occurrence may fall into the two. During the server consolidation, SLA violation occurs due mainly to migration overhead. While a VM is migrated, the source and destination hosts consume some of their resources to process intensive I/O. Thus, these two hosts may have difficulties allocating resources demanded by their hosted VMs. Thus, there is a high possibility that SLA is violated in the VMs. During the non-server consolidation time, SLA violation occurs due to changes in the application's workload. The resource demand of a VM depends on the workload of the application running on the VM. If the VM's resource demand due to an increase in application workload exceeds total available resources of its host, then SLA is violated.

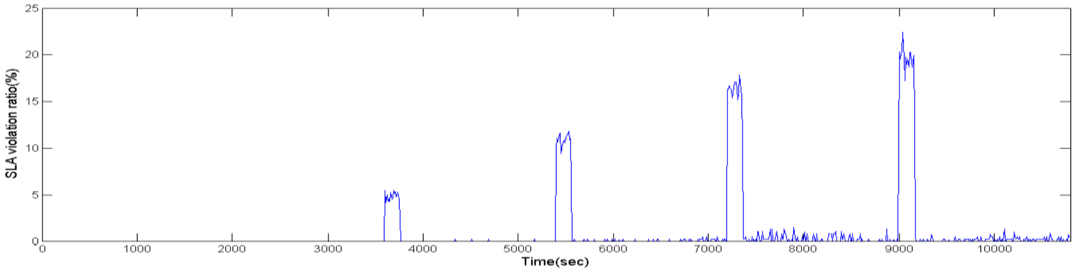

Figure 7. Ratio of SLA Violated VMs to total VMs

5. Conclusion

SimCloudIS for the simulation of cloud computing infrastructure and service was presented. The SimCloudIS provides extendable and customizable cloud system components managed in independent layers. Also, it supports packet based network communication that is based on the real network model, thereby facilitating the simulation of internet based services. For performance evaluation, a monitor that observes the status of all components and recording them is provided.

The usefulness of SimCloudIS is demonstrated by involving a case study for dynamic allocation and relocation of VMs in a cloud environment. In analyzing the result of the case study, changes in trend of the total power consumption, network traffic and SLA violation ratio in cloud cluster are observed while a VM instance is created dynamically.

For more measurements for detailed evaluation, the functionalities of the monitor can be extended. Therefore, it is expected that the SimCloudIS can be used for simulating a resource management policy and evaluating a hypothesis.

References

[1] A. M. Armbrust , A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing", Communications of the ACM, vol. 53 no .4, (**2010**), April, pp. 50-58.

[2] H. Abdelsalam, K. Maly and D. Kaminsky, "Analysis of energy efficiency in clouds", in proc. of Computation World, in proc. of Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, (**2009),** pp. 416–421.

[3] R. N. Calheiros, R. Ranjan and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environment", in proc. of 2011 International Conference on Parallel Processing (ICPP), (**2011),** pp. 295-304.

[4] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,", Software: Practice and Experience, vol. 41, no. 1, January (**2011),** pp. 23-50.

[5] S. Garg and R. Buyya, "Networkcloudsim: modelling parallel applications in cloud simulations", in proc. of the 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011), (**2011),** pp. 105-113

[6] D. Kliazovich, P. Bouvry, Y. Audzevich and S. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers", in proc. of the Global Telecommunications Conference (GLOBECOM 2010), (**2010),** pp. 1–5.

[7] A. Nunez, J. L. Vazquez-Poletti, A. C. Caminero, G. G. Castane, J. Carretero and I. M. Llorente, “iCanCloud: a flexible and scalable cloud infrastructure simulator”, Journal of Grid Computing, vol. 10, no. 1, (**2012),** pp. 185–209.

[8] A. Nunez, J. L. Vazquez-Poletti, A. C. Caminero, J. Carretero and I. M. Llorente, “Design of a new cloud computing simulation platform”, in proc. of the International conference on Computational science and its applications(ICCSA’11), (**2011),** pp. 582–593.

[9] Google cluster data. Available at https://code.google.com/p/googleclusterdata/

[10] PlanetLab. Available at http://planet-lab.org/

[11] Standard performance Evaluation Corporation. Available at http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00336.html

[12] Amazon Elastic Compute Cloud (EC2). Available at http://www.amazon.com/ec2/

Authors

Changhyeon Kim received the B.S. degree in Computer Science and Engineering from Kyungil University, Daegu, Korea, in 2008, and the M.S. degree in Computer Science and Engineering, from Hanyang University(ERICA Campus), Ansan, Korea, in 2010. He is currently a Ph.D. candidate at Hanyang University(ERICA Campus). His research is focused on MapReduce programming model and virtualized resource management on cloud computing environment.

Junsang Kim received the B.S. degree in Electronic and Computer Engineering from Hanyang University(ERICA Campus), in 2003, and the M.S. degree in Computer Science and Engineering, from Hanyang University(ERICA Campus), Ansan, Korea, in 2005. He is currently a Ph.D. candidate at Hanyang University(ERICA Campus). His research is focused on data management and file system on cloud computing and grid environment.

Wonjoo Lee received his B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University(ERICA Campus), Ansan, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of Computer Science at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science. He is currently a Professor in the Department of Computer Science, Inha Technical College. He has also served as the Vice-president of The Korean Society of Computer Information and the Editor-in-Chief for the Journal of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing.