Enhancing Performance of Applications in Cloud using Hybrid Scaling Technique

Madhukar Shelar
Ph.D. Research Scholar
Department of Computer Sci.
S. P. Pune University, Pune

Shirish Sane
K. K. Wagh Institute of
Engineering Education and
Research, Nashik

Vilas Kharat
Department of Computer Sci.
Savitribai Phule Pune
University, Pune

ABSTRACT
In Infrastructure as a Service (IaaS) model of cloud computing paradigm, users acquire computing resources such as CPU, memory, storage and network bandwidth from an IaaS provider and these resources are used to deploy and run their applications. Cloud service providers share computing resources of a physical machine by running isolated Virtual Machines (VM) for web applications. As the load on web application increases, the associated VM must be able to scale up resources to support the increasing load. At the same time, VM should also be able to scale-down resources at light load. In this paper the novel architecture is proposed that provides the hybrid solution of vertical followed by horizontal scaling techniques of resource management in cloud data center. As per the dynamic load on web applications, the proposed algorithm takes the appropriate scaling decision to allocate resources from available pool of resources. Generally dynamic scaling is achieved by the conventional live VM migration technique to create additional VM instances, but VM migration spends CPU time and consumes large amount of IO and network traffic. The proposed technique postpones live VM migration as long as possible with the help of vertical scaling technique to improve the performance of applications.

Keywords
Cloud Computing, Infrastructure as a Service, Virtualization, Cloud Resource Management, Virtual Machine Migration, Horizontal Scaling, Vertical Scaling

1. INTRODUCTION
In recent years, cloud computing has become very popular computing model in which virtualized and scalable resources are provided as services to the clients [1]. Wide range of cloud products and applications are used today by internet users such as Amazon EC2 [2], Google App Engine [3], Microsoft Azure [4], IBM Blue Mix [5] and many more. Many users and organizations also deploy their web applications on cloud data centers so that it gets available to them through internet. Data centers provide facilities that support huge data processing. The performance of a data center directly affects the quality of service provided [6]. The performance of a data center can be improved by increasing the resource utilization. Cloud data centers share their physical resources such as CPU cores, memory, storage and network bandwidth among these user products and web applications. Server virtualization is the key technology that can be used to increase resource utilization. It is one of the most prominent technologies used in cloud computing that provides on-demand and scalable computational resources to these cloud applications. Server virtualization technology creates an abstract layer above physical infrastructure available on host servers. Cloud applications are provisioned with a set of Virtual Machines (VMs) sharing physical resources as shown in figure 1 [7]. Each VM hosts guest operating system, middleware software, applications and given a partition of underlying resource capacity of Physical Machine (PM) [8]. Statistically partitioning the physical resources into VMs as per peak demand of applications may lead to poor utilization of resources. Over-provisioning of resources results in lower profit margin while under-provisioning certainly results in customer dissatisfaction [9]. Obviously, the solution is to dynamically scale the resources based on workload demand without any service interruption. Hypervisor or Virtual Machine Monitor (VMM) software plays a key role for the management of such physical resources among VMs. Xen[10], KVM[11], VMware[12] are widely used hypervisor products used by cloud data centers.

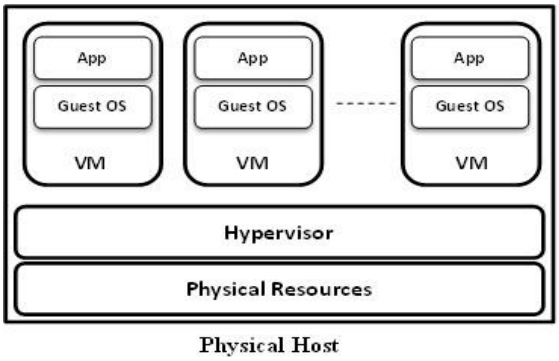

Fig 1: Server Virtualization

Server virtualization technology integrates physical infrastructure into pool of virtual resources. Cloud data center tries to satisfy users' requests and provide services by sharing virtualized infrastructure. As the workload on web applications is dynamic in nature, the resource allocated to VM may be under provisioned or over provisioned. When an allocated resources for virtual machine are unable to satisfy user requests, additional resources are allocated by VM-based scaling technique. Scaling is the ability of an application to make optimum utilization of resources at different workload levels by avoiding over-provisioning, under-utilization and under-provisioning [13]. A large fraction of cloud applications (called tenants) may cause its workload to grow sharply over a period of sometime or days or week. To deal with such unpredictable load patterns, there is a need of dynamic scaling or elasticity without any service interruption. Cloud computing provides pay-per-use models as services that can be scaled on-demand. However, it increases the overhead of managing virtualized resources among VMs. As per the users' requirement, VMs are to be created, deployed, configured and managed properly. When the large numbers of VMs are requested, new tools and methodologies for managing VMs are become necessary in order to automate all steps of cloud

services. Dynamically creating and configuring thousands of VMs for a particular purpose is an open issue for cloud users [14]. The main aim of this paper is to propose the architecture of an efficient dynamic resource scaling in server virtualization with respect to responsiveness, availability and commercial profits of web applications.

The remainder of the paper is organized as follows. Section 2 describes the resource scaling techniques and related parameters. Section 3 reviews the related work. In section 4, the novel architecture of scaling decision maker and its algorithm are proposed. Experimental results of simulation model are presented in section 5. Section 6 concludes the work and lights up the future direction.

2. RESOURCE SCALING

The responsiveness, availability and commercial profit of any web application highly depends upon resource allocation strategy applied. The resource demands of cloud applications are satisfied using the virtualized infrastructure available in data center. As the load on cloud applications is dynamic, the allocated resources need to be scaled up or scaled down as per their requirement. VM-based dynamic scalability is the widely used technique to achieve these requirements of web applications. VM-based scaling can be implemented by either changing the partition of resources (e.g. CPU, memory, storage) inside a VM or adjusting the number of VM instances [7]. These kinds of scaling techniques are referred to as vertical and horizontal scaling respectively. Horizontal scaling obtains more computing power by adding more virtual machines whereas vertical scaling obtains more computing power by adding more resources in virtual machine [6].

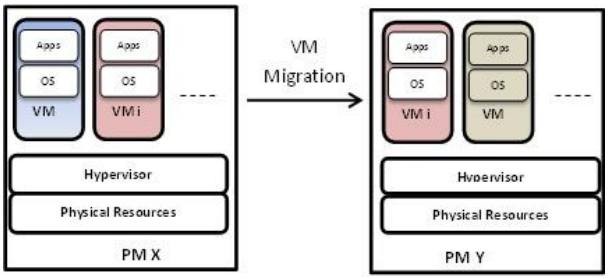

Fig 2: Horizontal Scaling of Virtual Machine

Horizontal scaling creates additional instance of VM and load is distributed among those instances. As shown in figure 2 when the allocated resources for VMi in PM X are insufficient to satisfy the demand of underlying applications, new instance of VMi is created on another PM Y and the load is distributed among these multiple instances with the help of load balancer. Amazon EC2 provide auto-scaling feature that automatically increase the number of VM instances and balancing load among replicas during high load to maintain the performance [2].

In vertical scaling, allocated resources of an already running VM instance are scaled up on-the-fly. As shown in figure 3, to satisfy the resource demand of applications in VMi, the allocated resources are vertically scaled up. Unfortunately, most common operating systems does not support for on-the-fly changes on allocated resources without rebooting [15].

These two kinds of scaling techniques affect various parameters such as performance, availability, cost, physical limitation, power consumption etc.

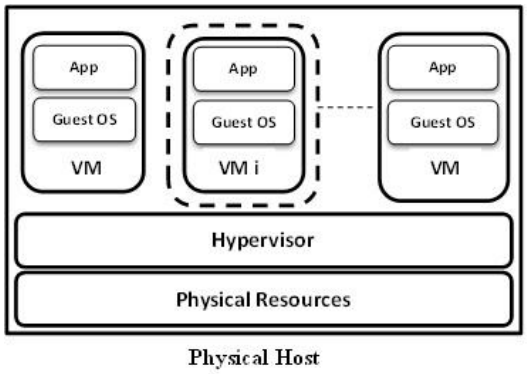

Fig 3: Vertical Scaling of Virtual Machine

2.1 Performance

The performance of applications can be improved by creating their replicas and distributing the load equally among those replicas. This is achieved with the help of horizontal scaling technique that creates another instance of the VM in which applications are deployed. However, this technique spends more CPU time and consumes network bandwidth during live VM migration from one host to another. Vertical scaling technique improves performance by scaling up allocated resources to VMs within the host server itself, hence no CPU time and extra bandwidth is consumed. But vertical scaling has limitations to scale up resources as per the host server capacity.

2.2 Availability

Availability of cloud applications is equivalently crucial as cost saving and performance. Availability constraint provides overall availability of applications by deploying VMs across different isolation levels in the data center. As horizontal scaling creates multiple VM instances, it enhances the overall availability of applications. However, vertical scaling does not create any additional VM instances; therefore reliability and availability is lesser in vertical scaling as compare to horizontal scaling.

2.3 Cost

The VM migration time, cost of keeping consistency between multiple instances, cost of additional software licenses, cost of network and IO traffic are the major issues in resource scaling. Vertical scaling does the metric adjustment of resources within the host itself. Hence, it does not involve above issues related to cost. It is time-efficient than horizontal scaling technique, because horizontal scaling involves time duration for creating copy of VM and its initialization. The cost of maintaining consistency in multiple copies of data is another issue in horizontal scaling technique; however it is not the case in vertical scaling.

2.4 Resource Limitations

In vertical scaling, the available free resources in host server are the upper limits of resource allocation in VM. When the resources that VM demand exceeds the available free resources of host server, vertical scaling fails. However, in horizontal scaling, the upper limit of resource allocation is the total resources available inside whole cloud data center. If the resources in data center are exhausted, inter-cloud could be used.

2.5 Consistency

Maintaining the consistency between all instances of VM is the key challenge in horizontal scaling. To maintain the

consistency among all instances of VM, the traffic load is increased. In order to reduce the traffic load, instances are assigned to host machines in close proximity. However, the consistency is not the issue in case of vertical scaling, as there is a single instance of VM.

2.6 Load Balancing

Horizontal scaling needs a gateway and workload balance that can forward requests to multiple instances. Therefore, it provides more throughputs at expense of complexity. As in vertical scaling, all requests are handled by a single instance of VM, therefore it provides fewer throughputs.

3. RELATED WORK

At the time of application deployment new Virtual Machine (VM) is created that hosts guest operating system, middleware, application etc and given a partition of underlying resource capacity (CPU, RAM, bandwidth, storage etc.) of Physical Machine (PM) or host server. Resources for VM are allocated as per application requirement and Service Level Agreement (SLA) between client and cloud service provider. The resource allocation during application deployment in data centers is modeled through mapping a set of applications on to set of host servers with the help of virtualization. This problem is incorporated by many researchers and addressed various solutions to maximize some utility function under resource constraints. Hyser et al. [16] proposed a system architecture design of an autonomic VM placement for better utilization of computing resources and cost savings. The data center load is distributed among all available physical servers and resource usage is balanced as much as possible across all resource types. The cost for resource provisioning is reduced by minimizing the number of used physical servers [17, 18, 19]. In addition to the cost and performance, availability of cloud applications is the important factor to be considered during VM deployment. Availability of cloud applications can be maintained by keeping multiple copies of VMs on different physical servers [20]. This feature can also improve the performance of applications by distributing load among multiple copies. However, this research does not focus on consistency and cooperation between multiple copies.

The loads on applications are dynamic in nature therefore the resources allocated for VM can be scaled up or scaled down to satisfy its demand. There are several research works related to scalability of resources in cloud computing environment have been published in recent years. Resource scalability can performed using either vertical or horizontal scaling techniques. Wang et al. [7] has compared the influence of vertical and horizontal scaling techniques on cloud resource management. Vertical scaling technique has the advantage of performance whereas horizontal scaling enhances the overall availability of applications. In horizontal scaling technique VM is migrated to create an additional instance and the load is handled by multiple instances. VM migration is the process of transferring VM from one physical server to another. As VM migration consumes network bandwidth, increases IO traffic and spends much of CPU time, many research works have been proposed to save this migration cost. Liu et.al [21] proposed a novel approach, CR/TR-Motion to provide fast and transparent VM migration. It achieves negligible downtime and reasonable network bandwidth consumption. Isci et.al [22] presented direct memory access based VM migration technique which significantly reduces migration overheads. A post-copy algorithm proposed in [23] reduces the total migration time and network traffic. It first replicates CPU execution state and then transfers memory state. Vaquero et.al [15] reviewed the work related with scalability at different levels by giving more focus on VM migration techniques. VM migration process gets triggered when resource conflict occurs in VM. However, it would be too late to trigger the VM migration process on resource conflict, since migration may take long time to finish and during that time performance of migrated applications is substantially reduced. Shen et.al [24] presented CloudScale, a technique that uses a predication migration which can start the migration process before conflict happens to minimize the impact on migrated applications. If the number of VMs and physical resources increases it would be too difficult to analyze, scale and manage resource provision for centralized decision maker. Nguyen et.al [8] also proposed a system which has local and global decision modules that interact with each other and take decisions for VM migration on same or other physical hosts.

However, live VM migration causes significant impact on performance of applications. Hence there is need of further research that avoids VM migration process. The proposed approach reserves certain amount of free resources in every host server. These free resources can be utilized by VMs at peak load with the help of vertical scaling. VM is migrated to another server only when there are insufficient free resources are available in the host server. Thus with the help of hybrid scaling VM migration process is postponed as long as possible without degrading the performance of applications.

4. PROPOSED ARCHITECTURE

Clients, users and cloud service providers are three main actors involved in cloud computing services. There can be difference between clients and users in the cloud service. The clients can be any organization deploying web applications and users access those applications by providing variety of load as per their requirement. Service Level Agreement (SLA) signed between client and service provider defines various parameters related to performance, cost and resource allocation. Cloud service is provided for clients applications by network of data centers with the help of virtualized resources so that users can access applications from anywhere through the internet. Users and clients acquire computing resources (CPU, memory and storage) from cloud service provider and use those resources to deploy and run their applications. Service provider creates isolated Virtual Machines (VM) for applications and allocates physical resources to them. However, during peak load, the allocated resources for an application would be insufficient and that affects its performance. Therefore cloud service provider may statistically allocate physical resources to VMs as per the peak demand of underlying applications. But the allocation of maximum resources to VM may leads to poor resource utilization during the light load. Statistically over-provisioning of resources results in lower profit margin while under-provisioning certainly results in customer dissatisfaction [9]. The solution is to scale the resources dynamically based on workload demand without any service interruption. The applications must be able to scale-up or scale-down allocated resources as per users load.

So due to the unpredictable nature of workload patterns, achieving dynamic scalability is a challenge for applications in the cloud. The proposed architecture for dynamic resource allocation with the help of vertical as well as horizontal scaling techniques is shown in figure 4. It involves the following entities to provide QoS (Quality of Service) requirement as per the clients or brokers demand.

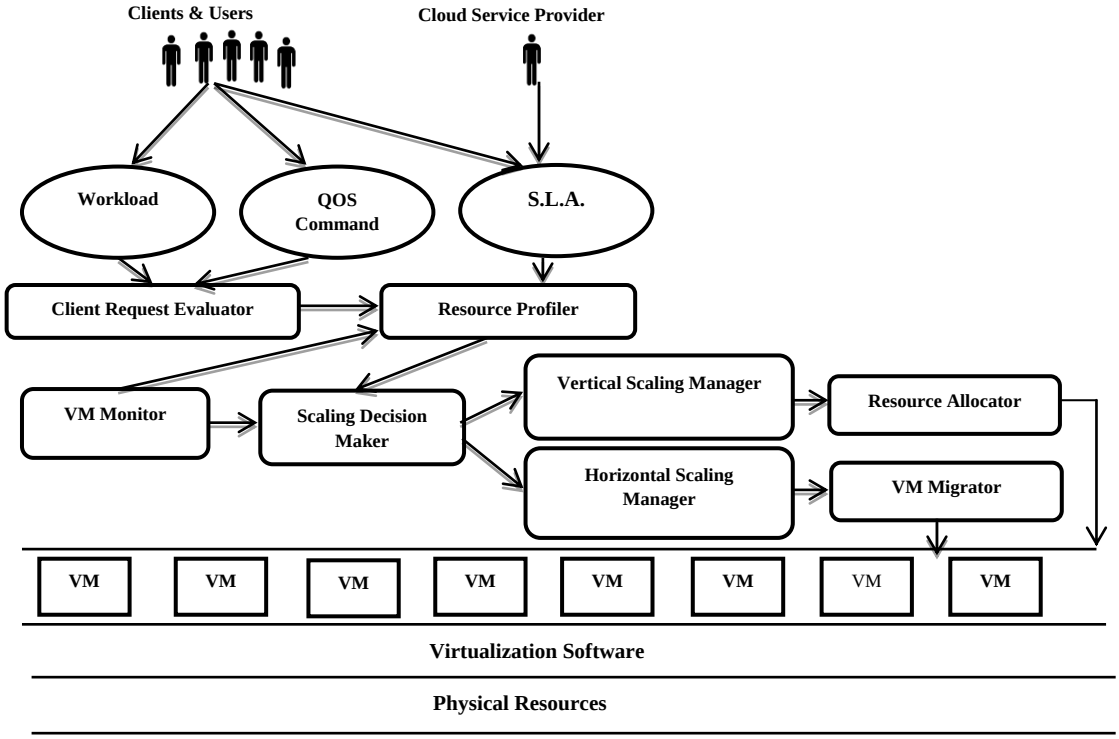

Fig 4 : Proposed Architecture of Hybrid Scaling

4.1 Client Request Evaluator

This entity evaluates users' workload on application with the QoS requirement provided by client. QoS requirements specify performance, availability, scalability and reliability of applications. It generates evaluative inputs of resource demand for resource profiler.

4.2 VM monitor

VM allocates physical resources from resource pool of host server as per the workload of underlying application. VM monitor continuously keeps the track of resource usage and future demand of resources for VM. The appropriate inputs are provided to the resource profiler as well as to the scaling decision maker.

4.3 Resource Profiler

It collects the current resource usage and resource demand from VM monitor and client request evaluator respectively. Resource profiler estimates the resource usage cost with the help of SLA and interacts with scaling decision maker to allocate additional resources or free the existing allocated resources.

4.4 Scaling Decision Maker

This entity analyzes resource statistics allocated to VMs as shown in algorithm 1 and initiates the appropriate scaling. If additional resources required for VM are available in the host server itself then the vertical scaling is initiated towards vertical scaling manager. However, if there are insufficient free resources in the host server then horizontal scaling manager initiates VM migration.

4.5 Vertical Scaling Manager

It dynamically adjusts the logical partition of allocated resources (CPU, RAM etc.) to VM. Vertical scaling technique either logically attaches or detaches resources for VM without rebooting [25]. Vertical scaling does not perform VM migration; therefore network bandwidth is not consumed.

4.6 Horizontal Scaling Manager

It triggers VM migration that changes number of VM instances hosted on different servers. Horizontal scaling manager creates multiple instances of VM and the workload on deployed applications are kept balanced among all instances with the help of load balancer. Horizontal scaling consumes more bandwidth as compare to vertical scaling technique. Hence it may affect the performance of applications. Scaling decision maker as mentioned in algorithm 1 performs either scale-up or scale-down of allocated resources for n Virtual Machines loaded on host server.

Scaling decision maker takes the appropriate decision based on the current resource allocation state and new demand of resources in Virtual Machines. Some amount of resources say 30% are reserved in every host server to satisfy the demand of increasing load on applications using vertical scaling technique. For all n VMs loaded on host server, scaling decision maker performs either scale-up or scale-down based on the inputs provided by resource profiler and VM monitor. Scale-up process allocates additional resources to VM from the available pool of resources whereas scale-down process frees the resources from VM and put them into available pool of resources. If the resource demand of application exceeds the currently allocated resources then scale-up decision is triggered. Vertical scaling manager initiates the vertical scale-up, if the sufficient free resources are available on host server. Otherwise, the horizontal scaling manager creates additional instance of VM on another host server. However, if the allocated resources for VM exceed the actual demand then resource decision maker decides to scale-down the resources. The scale-down process can be either vertical or horizontal that depends on available number of instances. The proposed

technique introduces a hybrid approach that postpones VM migration process with the help of vertical scaling. Therefore it saves CPU time, network and IO traffic. Obviously the performance of application is enhanced.

Algorithm 1 : Scaling Decision Maker

Assumptions
V=V1,V2,,Vn set of n virtual machines loaded on host server
R : Indicates available resources in host server
Vki : Number of running instances of VM Vk
Vkc : Currently allocated resources by VM Vk
Vkd : Demand of resources by VM Vk
Vkd > Vkc : VM Vk demanding more resources than allocated
Vkd < Vkc : VM Vk demanding less resources than allocated

```
1.    for k=1 to n do
2.      if (Vkd>Vkc) //To scale up
3.        if (R > (Vkd-Vkc)) //Sufficient resources available
4.            VerticalScaleUp (Vk,Vkc,Vkd)
5.        else //Lack of resources on server
6.            HorizontalScaleUp(Vk,Vkc,Vkd)
7.        end if
8.      Else
9.        if (Vkd<Vkc) //To scale down
10.         if (Vki == 1) //Single instance of Vk
11.           VerticalScaleDown(Vk, Vkc, Vkd)
12.         else //multiple instances of Vk
13.           HorizontalScaleDown(Vk, Vkc, Vkd)
14.         end if
15.     end if
16.   end for
```

5. EXPERIMENT RESULTS

The proposed algorithm for scaling decision maker is based on Infrastructure as a Service (IaaS) model and it is essential to evaluate it on a virtualized data center infrastructure. However, it is extremely difficult to conduct scaling experiments on a real infrastructure in cloud data center. Therefore to test the performance of proposed algorithm, a simulation model is developed in Java. A data center with 20 PMs or host servers with four different configurations (mentioned in table 1) is simulated. The load of 500 VMs each require 100 to 500 MB RAM is assumed in a data center. Initially all VMs are randomly placed on appropriate PMs by considering the required resources for VM and available resources in PM.

Table 1 : PM or Host Server Configurations

PM Configuration#	No. of CPU cores / MIPS	RAM Size
1	8	64 GB
2	12	128 GB
3	16	192 GB
4	24	256 GB

For comparative study of proposed approach of hybrid scaling with conventional horizontal scaling method, the loads on applications are dynamically increased. As the load on applications increases, VMs underlying such applications demands for more resources and if the existing resources allocated of VM are insufficient then allocated resources are scaled up as per load. The figure 5 shows the comparative result obtained from the simulation model on VM migration count between conventional horizontal scaling and proposed hybrid scaling approach.

In conventional horizontal scaling, VMs are migrated to create additional instances on the same or another host for load balancing. The horizontal scaling approach increases the VM migrations that consumes network bandwidth and degrades the performance of application. In proposed hybrid scaling technique, initially additional demands of resources are satisfied from reserved quota with the help of vertical scaling within local PM itself and hence migration is avoided. If the resources on local PM are insufficient then only VM is migrated to create additional instances. The proposed approach uses vertical scaling followed by horizontal scaling that tries to postpone VM migration, hence migration count is reduced.

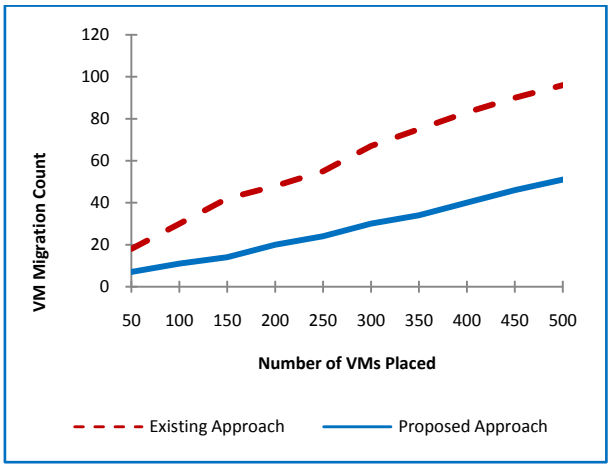

Fig 5: Comparative result of Existing and Proposed approaches on number of VM migrations

6. CONCLUSION

In server virtualization, dynamic resource scaling for web applications is the key challenge of cloud service providers. This paper presents an efficient scaling decision maker that postpones VM migration as long as possible, because VM migration spends CPU time and consumes network and I/O traffic. Hence it would results in higher resource utilization and improving the performance of applications. The proposed architecture is the hybrid of vertical and horizontal scaling techniques that considers current workload, allocated resources of application and available resources in host servers. In experimental evaluation the simulation model for scaling decision maker is developed. The results of simulation proves that proposed approach of hybrid scaling reduces the number of VM migrations as compare to the existing approach of horizontal scaling. Therefore the network bandwidth consumption for VM migration is also reduced that improves the performance of running applications.

As a future work, the experimental evaluation of proposed scaling decision maker algorithm is to be performed to measure the VM migration cost as well as bandwidth consumption. The simulation model will be also be extended to study the other positive side of existing horizontal scaling technique which improves parallelism and reliability due to multiple VM instances.

7. REFERENCES

[1] Li, Bo, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong. "Enacloud: An energy-saving application live placement approach for cloud computing environments." In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, pp. 17-24. IEEE, 2009.

[2] Amazon Elastic Compute Cloud, https://aws.amazon.com/ec2/

[3] Google App Engine, https://cloud.google.com/appengine/

[4] Microsoft Azure, https://azure.microsoft.com/en-in/services/sql-database/

[5] IBM Blue Mix, http://www.ibm.com/ cloud-computing/bluemix/

[6] Liu, Chien-Yu, Meng-Ru Shie, Yi-Fang Lee, Yu-Chun Lin, and Kuan-Chou Lai. "Vertical/Horizontal Resource Scaling Mechanism for Federated Clouds." In Information Science and Applications (ICISA), 2014 International Conference on, pp.1-4. IEEE, 2014

[7] Wang, Wenting, Haopeng Chen, and Xi Chen. "An availability-aware virtual machine placement approach for dynamic scaling of cloud applications." In Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on, pp. 509-516. IEEE, 2012

[8] Nguyen Van, Hien, Frederic Dang Tran, and Jean-Marc Menaud. "Autonomic virtual resource management for service hosting platforms." In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, pp.1-8. IEEE Computer Society, 2009

[9] Chieu, Trieu C., and Hoi Chan. "Dynamic resource allocation via distributed decisions in cloud environment." In e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on, pp. 125-130. IEEE, 2011

[10] XenServer Open Source Virtualization, http://www.xenserver.org

[11] Kernel Virtual Machine (KVM), http://www.linux-kvm.org/page/Main Page

[12] VMware, http://www.vmware.com/in

[13] Bellenger, Dominique, Jens Bertram, Andy Budina, ArneKoschel, Benjamin Pfnder, Carsten Serowy, Irina Astrova, Stella Gatziu Grivas, and Marc Schaaf. "Scaling in cloud environments." Recent Researches in Computer Science (2011)

[14] Kirschnick, Johannes, Jose M. Alcaraz Calero, Lawrence Wilcock, and Nigel Edwards. "Toward an architecture for the automated provisioning of cloud services." Communications Magazine, IEEE 48, no. 12 (2010): 124-131

[15] Vaquero, Luis M., Luis Rodero-Merino, and Rajkumar Buyya. "Dynamically scaling applications in the cloud." ACM SIGCOMM Computer Communication Review 41, no. 1(2011): 45-52

[16] Hyser, Chris, Bret McKee, Rob Gardner, and Brian J.Watson." Autonomic virtual machine placement in the data center." Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189 (2007): 2007-189

[17] Beloglazov, A., Buyya, R., Lee, Y. C., Zomaya, A.: Ataxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in computers. 82, no. 2 : 47-111 (2011)

[18] Chaisiri, Sivadon, Bu-Sung Lee, and Dusit Niyato. "Optimal virtual machine placement across multiple cloud providers." In Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific, pp. 103-110. IEEE, 2009

[19] Shelar, M., Sane, S., Kharat, V., Jadhav, R. : Efficient Virtual Machine Placement with Energy Saving in Cloud Data Center. International Journal of Cloud-computing and Super-computing. SERSC, Vol.1, No.1, pp.15-26 (2014)

[20] Goudarzi, Hadi, and Massoud Pedram. "Energy-efficient virtual machine replication and placement in a cloud computing system." In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, pp. 750-757. IEEE, 2012

[21] Liu, Haikun, Hai Jin, Xiaofei Liao, Chen Yu, and Cheng-Zhong Xu. "Live virtual machine migration via asynchronous replication and state synchronization."parallel and distributed Systems, IEEE Transactions on 22, no. 12(2011): 1986-1999

[22] Isci, Canturk, Jiuxing Liu, Blent Abali, Jeffrey O. Kephart, and Jack Kouloheris. "Improving server utilization using fast virtual machine migration."IBM Journal of Research and Development 55, no. 6 (2011): 4-1

[23] Hines, Michael R., and Kartik Gopalan. "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning." InProceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp. 51-60. ACM, 2009

[24] Shen, Zhiming, Sethuraman Subbiah, Xiaohui Gu, and John Wilkes. "Cloudscale: elastic resource scaling for multi-tenant cloud systems." InProceedings of the 2nd ACM Symposium on Cloud Computing, p. 5. ACM, 2011

[25] Jadhav, R., Somani, P. : Method and System for Real Time Detection of Resource Requirement and Automatic Adjustments. U.S. Patent Application 13/495,906. US 20130047158 A1. (2013)

[26] Gong, Zhenhuan, Xiaohui Gu, and John Wilkes. "Press: Predictive elastic resource scaling for cloud systems." In Network and Service Management (CNSM), 2010 International Conference on, pp. 9-16. IEEE, 2010.

[27] Gupta, A., Milojicic, D., Kal, L. V. : Optimizing VM Placement for HPC in the Cloud. In Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit, pp. 1-6. ACM. (2012)

[28] He, Sijin, Li Guo, and Yike Guo. "Real time elastic cloud management for limited resources." In Cloud Computing (CLOUD), 2011 IEEE International Conference on, pp.622-629. IEEE, 2011

[29] http://aws.amazon.com/autoscaling/

[30] Li, Xin, Zhuzhong Qian, Sanglu Lu, and Jie Wu. "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center." Mathematical and Computer Modelling 58, no. 5 (2013): 1222-1235