JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 859-869 (2017)
DOI: 10.6688/JISE.2017.33.3.16

Efficient Virtualization Technique for Cloud to Achieve Optimal Resource Allocation

SUGUNA MARAPPAN[1] AND SHARMILA DHANDAPANI[2]
[1]*Department of Computer Science and Engineering*
Kumaraguru College of Technology
Coimbatore, Tamilnadu, 641049 India
E-mail: suguna.marappan@gmail.com
[2]*Department of Electronics and Instrumentation Engineering*
Bannari Amman Institute of Technology
Sathyamangalam, Tamilnadu, 638401 India
E-mail: hodei@bitsathy.ac.in

Cloud computing technique has become an unavoidable technology in current IT world and now all the IT enterprises have switched from client-server technology to cloud technology. The underlying architecture of cloud is virtualization technology which multiplexes the server resources and utilizes the server in maximum level. Though the virtualization technology is used by many cloud providers it is not efficient in worst case scenario. The worst-case scenario is when the cloud server is overloaded; resource need for an incoming task is insufficient in the cloud server. In this paper, we propose some algorithms and methodologies to make the cloud efficient in a worst-case scenario.

Keywords: cloud computing, big cloud, small cloud, virtualization, load rebalancing, dynamic resource allocation

1. INTRODUCTION

In recent years, there was a drastic change in the IT field resulting in the birth of many new technologies. Among them Cloud Computing is the dominant and fast developing technology [9]. Though cloud computing is new, lot of the techniques are borrowed from existing technologies like Distributed Computing, Utility Computing and Grid Computing. Cloud computing technique is very clear and adopted by many Enterprises and common people, but still it has some drawbacks concerning its operation. Here we address some of the main problems associated with cloud, such as, resource allocation, load-balancing and fault tolerance. Cloud environment is a web-based environment so it has to behave in dynamic manner. Incoming request for resources may vary in cloud. According to higher number of requests and lower number of requests it should change its behavior. This technique is known as rapid elasticity. The inefficiencies in the rapid elasticity technique are discussed and some algorithms and techniques to make it more efficient are introduced here.

Fig. 1 (a) shows the System Architecture for Dynamic Resource Allocator in which various clients try to access the cloud resources. In this method, Dynamic resource allocator is responsible for load balancing, resource mapping and dynamic resource allocation. Fig. 1 (b) shows the internal snapshot of a cloud server, where it shows how the physical machine is deployed to run various virtual machines (VMs).

Received July 25, 2016; accepted September 21, 2016.
Communicated by Balamurugan Balusamy.

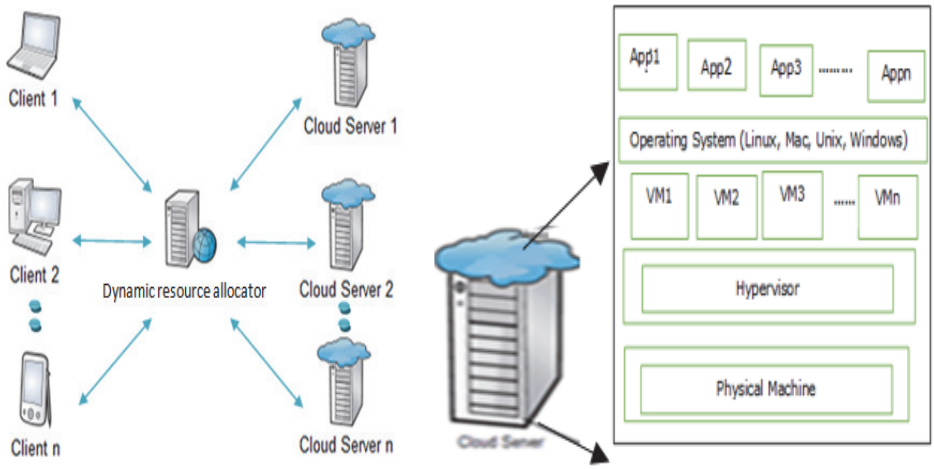

Fig. 1. (a) System architecture; (b) Internal snapshot of a cloud server.

1.1 System Architecture

In cloud computing environment, the end user or user requests for cloud resources so the process scheduler schedules the user to available clouds (resources), cloud monitoring and allocation is responsible for monitoring the cloud usage and allocating the resources to users. The process migration is responsible to migrate virtual machines (VMs) among clouds when migrating virtual machines (VMs) it will calculate cost and time.

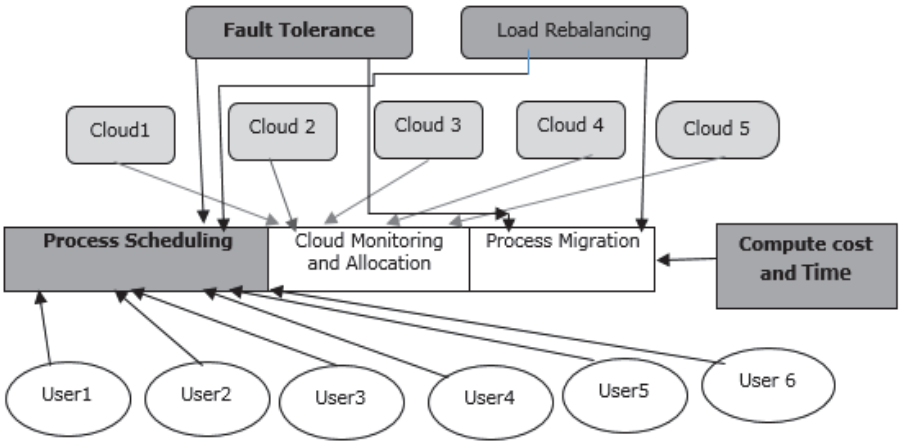

Fig. 1. (c) System architecture of proposed system.

Load Rebalancing mechanism in which the jobs are efficiently mapped to available resources and fault tolerance is achieved. By this the jobs are executed efficiently by having sufficient resources without worrying about the job failures.

2. PROBLEM 1: RESOURCE ALLOCATION

The Resource allocation is broadly classified into two types, that is, Static and Dynamic resource allocation [3, 4]. Dynamic resource allocation out performs static resource allocation in real time environment. In the cloud, virtualization technique is the heart of cloud architecture. Whenever a client raises request for resources, a Virtual Machine (VM) is allocated for each client [2]. There are chances for inefficiency during resource allocation in Cloud server (CS) when there is a high traffic. For example, if the cloud environment in which virtual machines (VMs) allocated for clients is given a maximum and minimum capability to handle, say, maximum 10VMs and minimum 3VMs. But if 15 VMs are allocated then it will be overloaded and at the same time if only 1 VM is allocated then it is underutilized. As shown in the (Fig. 2) CS1 is overloaded with jobs and likewise CS2 has few jobs but the CS3 and CS4 are severely underutilized, so it still consumes lots of energy even when it has no jobs running. Here the system is not utilized properly resulting in the wastage of energy and time.

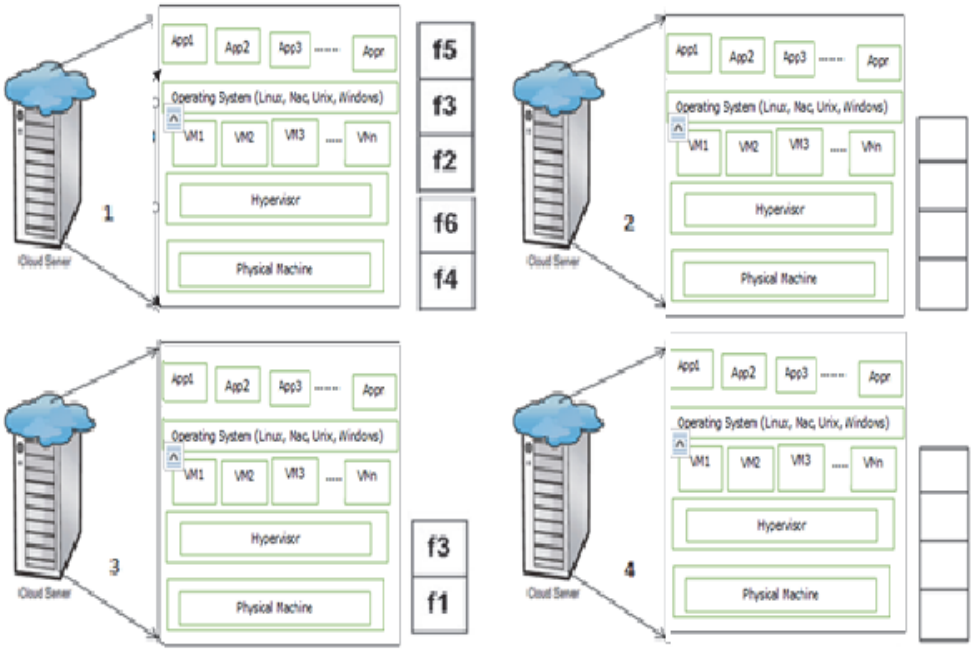

Fig. 2. General resource allocation in cloud server.

3. PROBLEM 2: RESOURCE INSUFFICIENT

The resource insufficient [6] problem occurs when the available resource is not enough. This degrades the performance of currently running Virtual machine and also to relinquish the execution.

As shown in the below (Fig. 3) the previously required resource is lower than the currently required resource and the currently required resource is three-fold higher than the previously required. If such a condition arises, when the virtual machine requests the cloud server to allocate the resource, CS checks if the sufficient resource is available, if not, then the cloud notifies the virtual machine and withdraws its execution. This degrades not only the performance of virtual machine and also the application running on the top of the virtual machine.

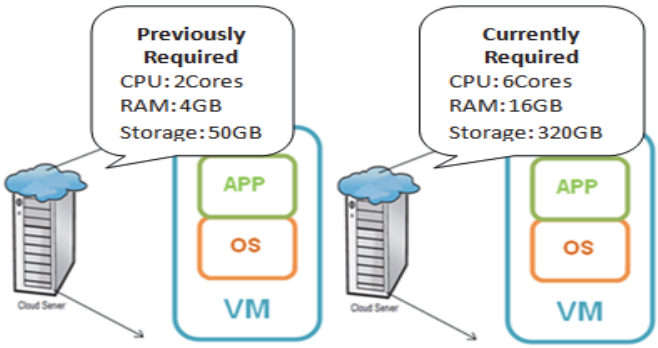

Fig. 3. Resource insufficient problem.

4. SOLUTION 1: DYNAMIC RESOURCE ALLOCATION

In dynamic resource allocation [3, 4] the resources are allocated only when there is a need for those particular resources arise, otherwise it won't allocate any resources in a cloud server. Similarly, when a cloud server is overloaded, instantaneously the VMs are migrated [1, 4] to other cloud server in order to avoid the overload and in the same way when a cloud server is underutilized the VMs are migrated to other cloud server to avoid underutilization. This is also known as Cloud server consolidation [7]. This helps in maximum utilization of cloud server and also supports green computing. The proposed model not only performs dynamic resource allocation but also responds instantaneously to the overload and avoid overload by VM migration process.

As shown in the (Fig. 4) the VMs in a cloud server exceeding maximum threshold or fall behind minimum threshold then the VMs are migrated from one cloud server to other cloud server and unutilized cloud servers are switched off and the running cloud servers have consolidated VMs that will utilize the cloud server properly.

4.1 Load Prediction and Overload Avoidance Algorithm

Load prediction and overload avoidance algorithm [3-5] is responsible for predicting the load in a cloud server and to avoid when it is overloaded. The algorithm is intended to check the available memory in the cloud server and whether it is capable of running the VMs or not. Load prediction [10] and overload avoidance algorithm not only avoids the load in a cloud server but also responds to the overload instantaneously.

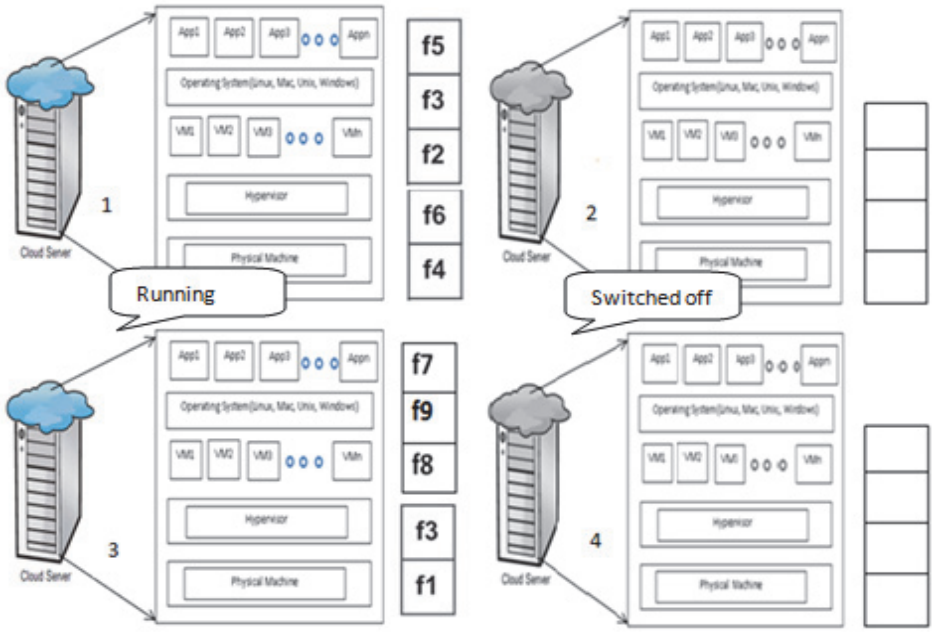

Fig. 4. Dynamic resource allocation in cloud server.

Algorithm 1: Load prediction and overload avoidance algorithm

Step 1: Initialize the Variables such as Input CS→Cloud Servers with VM→Virtual Machines associated with each Cloud Server AM→Available Memory, UM→ Utilized Memory L→Load for Virtual Machine, N→Number of User Process, U→ User Process.
Step 2: Assign priority to each Cloud Servers.
Step 3: for i=0 to N Identify the priority in Cloud and associated VM.
Step 4: if (UM < AM)
Step 5: Perform $V(M)$=U.
Step 6: else
Step 7: Identify the Free Resource on priority cloud to perform the allocation.
Step 8: else
Step 9: print Insufficient Resource.
Step 10: for i=0 to N
Step 11: if Load(VM) > Threshold(VM)| |Load(VM)<Threshold(VM)
Step 12: Print "Migration Required".
Step 13: End.

4.2 Overhead Computation Algorithm

While migrating virtual machines (VMs) [1] between cloud servers to avoid overload, the time delay and cost overhead are major aspects that should be taken into an account.

These overheads also are calculated in order to make virtual machine (VM) migra-

tion very efficient, so that the virtualization process can be done efficiently without hanging or crashing the server.

Algorithm 2: Overhead Computation Algorithm
Step 1: Initialize the variables such as $M \rightarrow$Cloud Servers,
$V \rightarrow$Virtual Machines.
Step 2: set VM cost = 0.10.
Step 3: for each (V: M)
Step 4: collect $t = dt$ in pmt.
Step 5: compute $c = \frac{VM_{mig} \rightarrow CS_i}{CS_n}$
Step 6: End.

5. SOLUTION 2: LOAD REBALANCING

Resource insufficient problem is avoided by load rebalancing [8] mechanism. When the VMs have sudden requirements, the loads in cloud servers are estimated and the best performing cloud server is also examined by Performance Estimation algorithm to fit the VMs requirement and 99.9% uptime of the cloud server. Load rebalancing is the process of rearranging the VMs by estimating the previous requirement and current requirement of resources. The group of the VMs which need only low requirement of resources are migrated to compact cloud servers called Small clouds and group of VMs which need high requirement of resources are migrated to Big clouds, which are larger cloud servers.

Fig. 5. Load rebalancing.

Table 1. Cloud performance evaluation.

Load	Server ID	Provider	Price	Successful interactions	Time outs	Average response time	Maximum response time
1000	X-Large	Go Grid	0.64	12405	243	741	14898
1000	M2.2xLarge	Amazon	1.12	11814	104	1256	15491
1000	C1.xLarge	Amazon	0.98	9039	451	2737	19798
1000	15GB	Rackspace	1.08	9187	323	2878	18358

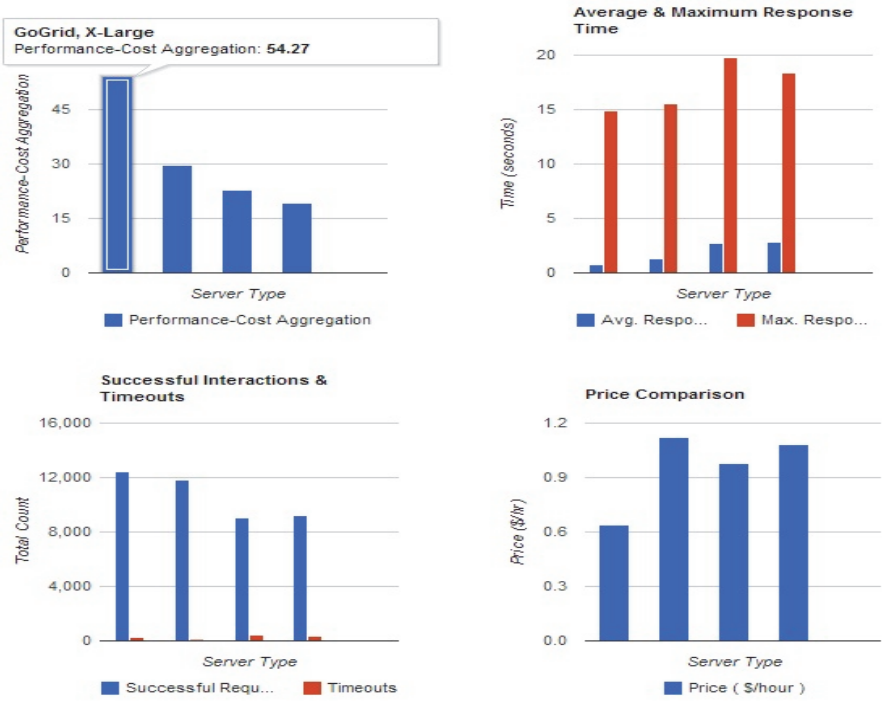

Fig. 6. Performance evaluations for the cloud servers.

As shown in the (Fig. 6) Small clouds and Big clouds will be predefined by estimating the cloud servers performance, that is, based on their previous statistics result like downtime percentage and fault tolerance. Best performing clouds are added to the category Big clouds and average performing clouds are added to the category Small clouds [12, 13].

Load rebalancing algorithm is used to rebalance the VMs based on their requirements to Small cloud and Big cloud. By this mechanism we can achieve a better utilization of cloud servers and also the insufficient resource problem can be solved. Table 1 shows an example cloud performance evaluation.

5.1 Performance Estimation Algorithm

Before load rebalancing the performance of the cloud servers should be estimated it is done by Performance estimation algorithm.

Step 1: Initialize the variables such as S→Small Cloud, B→Big Cloud, CS→Cloud Server.
Step 2: Collect Statistics T_s of CSn.
Step 3: if T_s == Good
Step 4: then CS_i→B.
Step 5: else if T_s == Average
Step 6: then CS_i→S.
Step 7: else reject from the list.
Step 8: End.

5.2 Load Rebalancing Algorithm

Once the Performance estimation is done then load rebalancing can be carried out in order to achieve efficient virtualization and resource utilization.

Step 1: Initialize the variables such as S→Small Cloud, B→Big Cloud,
R_{req}→Resource requirement, VM_{mig}→Virtual Machine Migration,
T_l→Threshold for Small Cloud, T_h→Threshold for Big Cloud,
T_l=4000bytes, T_h=10000bytes.
Step 2: if (R_{req}<=10000bytes)
Step 3: then VM_{mig}→B.
Step 4: else if (R_{req}<=4000bytes)
Step 5: then VM_{mig}→S.
Step 6: End.

6. SIMULATION RESULT

The output of the proposed system is shown in the above figure where (Fig. 6 (a)) shows the overloaded cloud server which is having peak loads that is before efficient resource allocation, (Fig. 6 (b)) shows, the peak loads are drastically reduced by the proposed algorithms, (Fig. 6 (c)) shows the equal resource utilization by all virtual machines (VMs) in a cloud server.

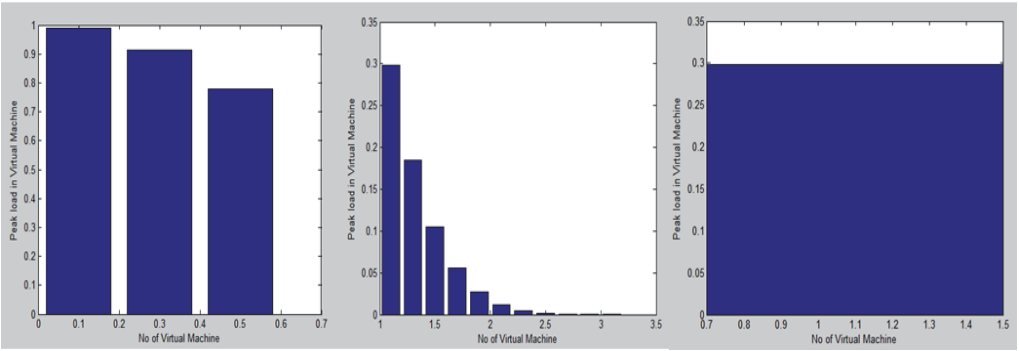

Fig. 6. (a) Before efficient resource allocation; (b) After dynamic resource allocation and load rebalancing; (c) Equal utilization of resources by the VMs.

6.1 Comparing Performance by Resource Utilization

Fig. 7 shows Comparing Performance by Resource Utilization in which the resource utilization for Existing System and Proposed System is compared. It is seen that the proposed system has proper resource utilization without any skew, whereas the existing system has improper resource utilization.

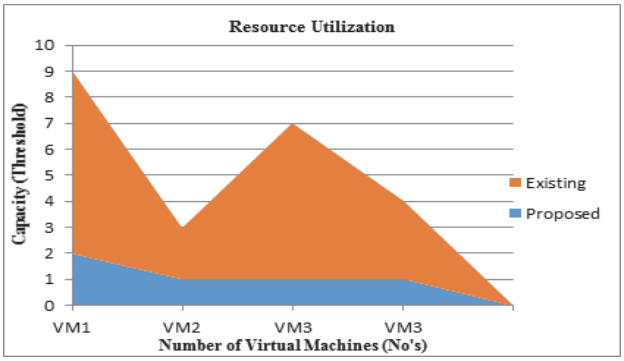

Fig. 7. Comparing performance by resource uilization.

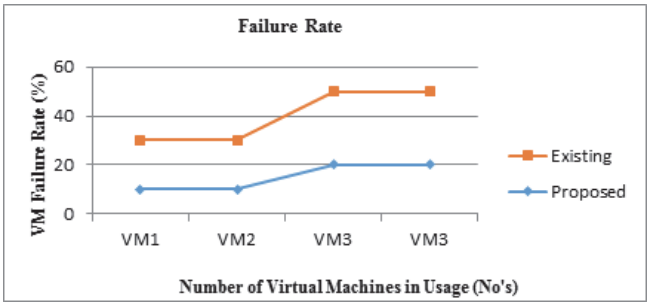

Fig. 8. Comparing performance by failure rate.

6.2 Comparing Performance by Failure Rate

Fig. 8 shows Comparing Performance by Failure Rate in which the resource utilization for Existing System and Proposed System is compared. It is seen that the proposed system has minimized Failure Rate to 10-20%, whereas in existing system the Failure Rate is 30-50%. when compared to other cloud resource allocation algorithms like best fit, first fit, load prediction algorithm has good load predicting ability and it can allocate resource dynamically and with overhead computation algorithm the cost and time taken to execute jobs is calculated. So with that the time, cost that happens for a particular process executed in the cloud environment can be estimated.

7. CONCLUSION AND FUTURE ENHANCEMENT

An effective Resource allocation system is proposed for cloud computing environment to make its underlying architecture-virtualization to behave in an efficient manner and also consolidation of virtual machines (VMs) is done to achieve green computing.

Load prediction and Overload avoidance algorithm is proposed to predict the load and avoid the overload. An overhead computation algorithm is proposed to compute overheads when virtual machines (VMs) are migrated between cloud servers and also load rebalancing mechanism is proposed to rebalance load between low resource utiliza-

tion and high resource utilization VMs. The implementation of load rebalance algorithm has cleared the resource insufficient problem and failures (fault) by mapping the jobs to resource available virtual machines to increase the job execution efficiency and fault is tolerated up to maximum level.

Future Enhancement will focus on fault tolerance as the cloud environment is prone to fault occurrence in cloud servers.

REFERENCES

1. C. Clark, K. Fraser, A. I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of Symposium on Networked Systems Design and Implementation*, Vol. 2, 2005, pp. 273-286.
2. P. Barham, B. Dragovic, T. Harris, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of ACM Symposium on Operating Systems Principles*, Vol. 37, 2003, pp. 164-177.
3. Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, 2013, pp. 1107-1117.
4. N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing SLA violations," in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, Vol. 1, 2011, pp. 119-228.
5. R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *Proceedings of International Conference on High Performance Computing and Simulation*, 2009, pp. 1-11.
6. S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Proceedings of IEEE Asia-Pacific Services Computing Conference*, 2009, pp. 103-110.
7. X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, C. Pu, and Y. Cao, "Who is your neighbor: Net I/O performance interference in virtualized clouds," *IEEE Transactions on Services Computing*, Vol. 99, 2012, pp. 314-329.
8. N. Bila, E. D. Lara, K. Joshi, and H. A. Lagar-Cavilla, "Jettison: Efficient idle desktop consolidation with partial VM migration," in *Proceedings of ACM European Conference on Computer Systems*, 2012, pp. 211-224.
9. H.-Y. Chung, C.-W. Chang, and Y.-C. Chao, "The load rebalancing problem in distributed file systems," in *Proceedings of IEEE International Conference on Cluster Computing*, 2012, pp. 117-125.
10. http://www.gartner.com/technology/research/cloud-computing.
11. J. Li, M. Qiu, J.-W. Niu, Y. Chen, and Z. Ming, "Adaptive resource allocation for pre-emptable jobs in cloud systems," in *Proceedings of the 10th International Conference on Intelligent System Design and Application*, 2011, pp. 31-36.
12. N. A. B. Mary and K. Saravanan, "Performance factors of cloud computing data centers using gdmodel queuing systems," *International Journal of Grid Computing & Applications*, Vol. 4, 2013, pp. 1-9.
13. F. D. Ross and R. N. Calheiros, "Non-invasive estimation of cloud applications per-

formance via hypervisor's operating systems counters," in *Proceedings of the 14th International Conference on Networks*, 2015, pp. 177-184.
14. S. Malik and F. Huet, "Adaptive fault tolerance in real time cloud computing," *IEEE World Congress on Services*, Vol. 10, 2011, pp. 280-287.
15. N. Bila, E. D. Lara, K. Joshi, and H. A. Lagar-Cavilla, "Jettison: Efficient idle desktop consolidation with partial VM migration," in *Proceedings of ACM European Conf. Computer Systems*, 2012, pp. 211-224.
16. T. Das, P. Padala, V. N. Padmanabhan, and R. Ramjee, "Lite green: Saving energy in networked desktops using virtualization," in *Proceedings of USENIX Annual Technical Conference*, 2010, pp. 33-48.
17. A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: Integration and load balancing in data centers," in *Proceedings of ACM/IEEE Conference on Supercomputing*, Vol. 10, 2008, pp. 1-12.

Suguna Marappan is an Assistant Professor (SRG) at the Department of Computer Science and Engineering in Kumaraguru College of Technology, Coimbatore. Her research interests include distributed computing and software project management. She is a member of ISTE, IAENG, IACSIT and CSTA. Currently, she is pursuing Ph.D. in Information and Communication Engineering at Anna University, Chennai, India.

Sharmila Dhandapani is a Professor at the Department of EIE, Bannari Amman Institute of Technology, Sathyamangalam, India. She received her Ph.D. in Wireless Security from Anna University, India. Her areas of interest include wireless security, low power VLSI, distributed computing and ADHOC networks. She has published more than 20 research articles in international journals and international conferences. She is member of IEEE, IAENG.