An Energy Efficient VM Allocation using Best Fit Decreasing Minimum Migration in Cloud Environment

Anurag Shrivastava[1], Vaibhav Patel[2], Sukanya Rajak[3]

[1]*Assist Prof, Head, CSE,* [2]*Assist. Prof., CSE,* [3]*PG, Scholar*

[1,2,3]*NIRT, Bhopal*

Abstract - Infrastructure as a Service (IaaS) has become one of the most dominant features that cloud computing offers nowadays. IaaS enables datacenter's hardware to get virtualized which allows Cloud providers to create multiple Virtual Machine (VM) instances on a single physical machine, thus improving resource utilization and decreasing energy consumption. VM allocation includes issues like determine load of host and also determine approach for selection of VMs for migration and placement of VMs to suitable hosts. VMs need to be migrated from over utilized host to guarantee that demand for computer resources and performance requirements are accomplished. Besides, they need to be migrated from underutilized host to deactivate that host for saving power consumption. In order to solve the problem of energy and performance, efficient dynamic VM consolidation approach is introduced in literature. In this work, proposed multiple redesigned VM allocation algorithms and introduced a technique by clustering VMs to migrate by taking account both CPU utilization and allocated RAM. We implement and study the performance of our algorithms on a cloud computing simulation toolkit known as CloudSim using PlanetLab, Bitbrains and Google Cluster workload data. Simulation results demonstrate that our proposed techniques outperform the default VM Placement algorithm designed in CloudSim.

Keywords: Cloud Computing, Dynamic consolidation, VM Allocation, CloudSim, PlanetLabs, BitBrains, GoogleCluster .

I. INTRODUCTION

With the limited amount of physical resources available, resource allocation becomes a challenging task for the cloud service provider. Since cloud computing is a multi-tenancy model, multiple users‟ requests for the cloud resources. So cloud service provider has to decide on how many virtual resources are to be created based on the cloud users‟ requests. Also which virtual machine (VM) has to be mapped onto which physical machine (PM) has to be taken care. That is, VM-PM mapping techniques have to be considered. At what instance VM migration has to be done is also based on identifying heavily loaded node and lightly loaded node. So the ultimate goal of the cloud service provider is to maximize profit and maximize resource utilization and the goal of cloud user is to minimize payment by renting the resources.

There are various parameters to be considered while allocating resources. While allocating resources to the cloud user, underutilization (wastage) of resources due to over provisioning and overutilization (due to under provisioning) should be avoided. Allocation of resources should consider various parameters like Quality of Service parameters like response time, performance, availability, reliability, security, throughput etc.

Performance: For some application demands, performance is one of the important criteria. The system should perform well to provide service to the cloud user.

Response Time: For interactive applications, response time is an important factor. The system must respond well for these kinds of applications.

Reliability: The system used should be reliable so that the cloud user has no head ache of changing the system.

Availability: Whenever cloud resources are requested the cloud service provider must be able to allocate within short span.

Security: For critical applications like online transaction applications, system used has to be secure. Otherwise it is not safe to use such a kind of system.

Throughput: No. of applications run per unit time should be more.

As an increasing number of complex applications leverage the computing power of the cloud for parallel computing, it becomes important to efficiently manage computing resources for these applications. Due to the difficulty in realizing parallelism, many parallel applications show a pattern of decreasing resource utilization along with the increase of parallelism. As the main aim of cloud computing is to provide resources as a service on demand to the user. In this work, there are going carried out two main problems.

1. *Reallocation of virtual machines.*

2. Allocation of virtual machines.

The process of selecting which virtual machines (VMs) should be located (i.e. executed) at each physical machine (PM) of a datacenter is known as Virtual Machine Placement (VMP). The VMP problem has been extensively studied in cloud computing literature and several surveys have already been presented. Existing surveys focus on specific issues such as:

(1) Energy-efficient techniques applied to the problem,

(2) Particular architectures where the VMP problem is applied specifically federated clouds.

(3) Methods for comparing performance of placement algorithms in large on demand clouds.

II. RELATED WORK

Ting et al [1], Cloud Computing refers to constructed data center or "super computer" by virtualization technology and provides computing and storage resources, as well as the application container environment of software running, to software developers in a manner of free or hiring.

Liu et al [2], propose priority-based method to consolidate parallel workloads in the cloud. We leverage virtualization technologies to partition the computing capacity of each node into two tiers, the foreground virtual machine (VM) tier (with high CPU priority) and the background VM tier (with low CPU priority).

Lugun et al [3], analysis the differentiated QoS requirements of Cloud computing resources users' jobs, we build the corresponding non-preemptive priority M/G/1 queuing model for the jobs.

Hsu et al [4], describes the important issue of energy conservation for data centers. We consider the problem of provisioning physical servers to a sequence of jobs, and reducing the total energy consumption.

Kaur et al [5], addresses parallel machine scheduling problems with practical Swarm Optimization (PSO). A PSO approach embedded in a simulation model is proposed to minimize the maximum completion time (make span).

Jung et al [6], In cloud computing, a service provider has to guarantee quality of service to offer stable services. For this, we should use scheduling algorithms.

III. PROPOSED METHODOLOGY

The basic algorithm of proposed work BFDMMT (Best Fit Decreasing with Minimum Migration Time) is as follows:

Step 1: Once it has been decided that a host is overloaded then select particular VMs to migrate from this host. For this purpose, we use Minimum Migration Time policy for virtual machine selection.

Step 2: After a selection of a VM to migrate, the host is checked again for being overloaded. If it is still considered as being overloaded, the VM selection policy is applied again to select another VM to migrate from the host. This is following steps repeated until the host is considered as being not overloaded.

(2.1) The Minimum Migration Time (MMT) policy migrates a VM v that requires the minimum time to complete a migration relatively to the other VMs allocated to the host.

(2.2) The Migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host j. Let V_j be a set of VMs currently allocated to the host j.

(2.3) MMT policy finds amount of RAM utilized by VM as per availability of network bandwidth.

Step 3: The VMs selected for migration are allocated to the destination hosts. For this purpose, perform following steps:

(3.1) Sort VM (1, 2 ...k) in order of their decreasing CPU utilization

(3.2) For every V_i in V(1,2,…,k) perform

manPower <- Max;

allocatedhost <- elist[];

(3.3) For every Mj in M (1, 2 ...n) perform

If M_j has enough resources for V_i then

Power <- estimatePower (M_j, V_i)

(3.4) if power < manPower then

allocatehost <- host

manPower <- power

(3.5) Elseif allocatehost= NULL then

Add (allocatehost, V_i) to NextVM

(3.6) return NextVM , goto step 3.1

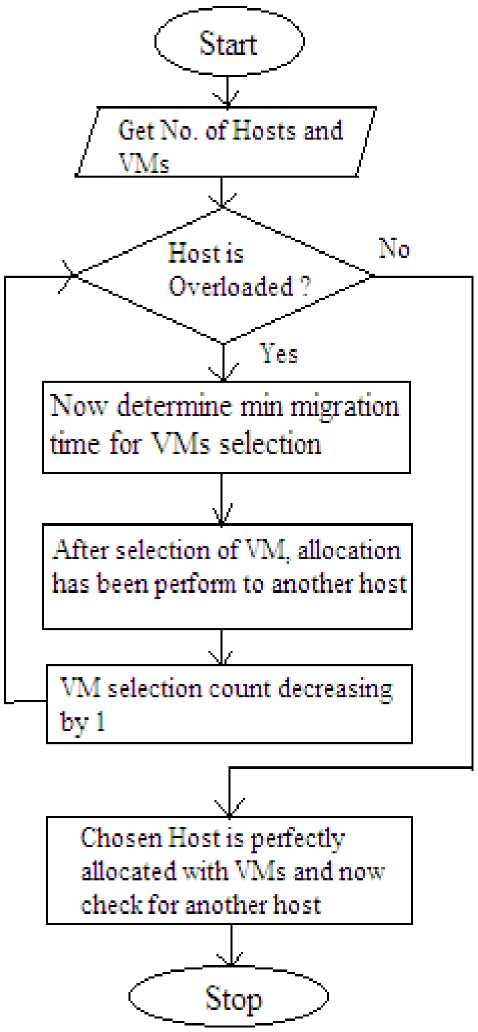

Fig: Flowchart of Proposed Methodology (BFDMMT)

Step 4: The system finds the host with the minimum utilization compared to the other hosts, and tries to place the VMs from this host on other hosts keeping them not overloaded.

Step 5: If this can be accomplished, the VMs are set for migration to the determined target hosts, and the source host is switched to the sleep mode once all the migrations have been completed.

Step 6: If all the VMs from the source host cannot be placed on other hosts, the host is kept active.

Step 7: This process is iteratively repeated for all hosts that have not been considered as being overloaded.

Step 8: Finally, Obtain hosts and virtual machine map.

IV. SIMULATION SETUP

The implemented extensions have been included in the 2.0 version of the CloudSim toolkit. The simulated a data center that comprises 800 heterogeneous physical nodes, half of which are HP ProLiant ML110 G4 servers, and the other half consists of HP ProLiant ML110 G5 servers. The simulation parameters are as follows:

SCHEDULING_INTERVAL = 300;

SIMULATION_LIMIT = 24 * 60 * 60;

CLOUDLET_LENGTH = 2500 * (int) SIMULATION_LIMIT;

CLOUDLET_PES = 1;

VM_TYPES = 4;

VM_MIPS = {2500, 2000, 1000, 500};

VM_PES = {1, 1, 1, 1};

VM_RAM = {870, 1740, 1740, 613};

VM_BW = 100000; // 100 Mbit/s

VM_SIZE = 2500; // 2.5 GB

Host types:

* HP ProLiant ML110 G4 (1 x [Xeon 3040 1860 MHz, 2 cores], 4GB)

* HP ProLiant ML110 G5 (1 x [Xeon 3075 2660 MHz, 2 cores], 4GB)

NUMBER_OF_HOSTS = 800;

NUMBER_OF_VMS = 1052;

HOST_TYPES = 2;

HOST_MIPS = {1860, 2660};

HOST_PES = {2, 2};

HOST_RAM = {4096, 4096};

HOST_BW = 1000000; // 1 Gbit/s

HOST_STORAGE = 1000000; // 1 GB

The DataCenter creation process can be defined as follows:

String arch = "x86"; // system architecture

String os = "Linux"; // operating system

String vmm = "Xen";

double time_zone = 10.0; // time zone this resource located double cost = 3.0; // the cost of using processing in this resource

double costPerMem = 0.05; // the cost of using memory in this resource

double costPerStorage = 0.001; // the cost of using storage in this resource

double costPerBw = 0.0; // the cost of using bw in this resource

V. RESULT ANALYSIS

There are three sets of workload traces namely as PlanetLab, GoogleCluster, BitBrains, that are publicly available and relevant for VM placement algorithms. In the following, we describe these and their integration into our test environment.

Origin	Virtualized	Size	Duration	PM Data	Static VM Data	Dynamic VM Data		
						CPU	Memory	Disk
Planetlab	Yes	1000 VMs	10 days	No	No	Yes	No	No
Google	No	12000 PMs	29 days	Yes	Yes	Yes	Yes	Yes
Bitbrains	Yes	1750 VMs	4 months	No	No	Yes	Yes	No

Table 1: Summary of the used real-world workload traces

The simulation is initialized by the Main class which creates instances of the scheduler, the job and machine loader, the failure loader and other entities as required by the standard CloudSim 3.0.2.

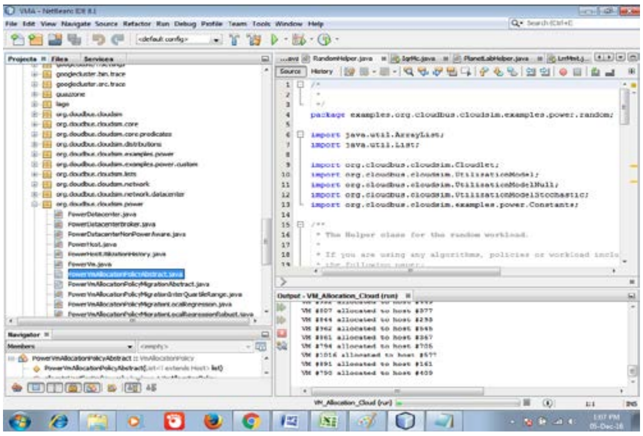

Fig 2: CloudSim 3.0.2 Environment in NetBeans IDE Environment

The energy consumption, VM migration and Hosts Shutdown can be evaluated through VMABS (Virtualization Migration with Abstract), ST (Static Threshold) and BFDMMT (Proposed Method) is as follows:

Dataset	Energy Consumption (KWH)			VM Migrations		
	VMABS	ST	BFDMMT (Proposed)	VMABS	ST	BFDMMT (Proposed)
PlanetLab	5.5	12.1	5.42	2382	2056	2050
Google	3.92	8.2	3.81	798	1312	706
BitBrains	10.89	12.14	10.83	1360	1771	1242

Table 2: Comparison of EC, VMM among VMABS, ST and BFDMMT

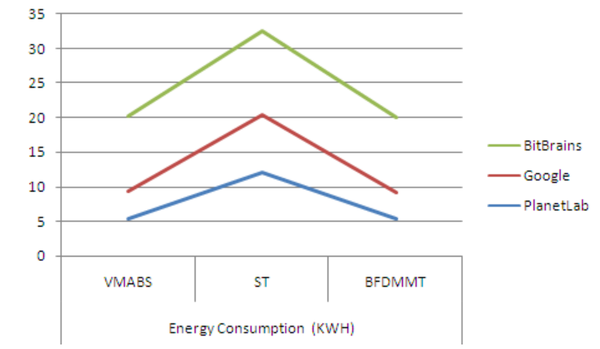

Fig 3: Energy Consumption among VMABS, ST and BFDMMT

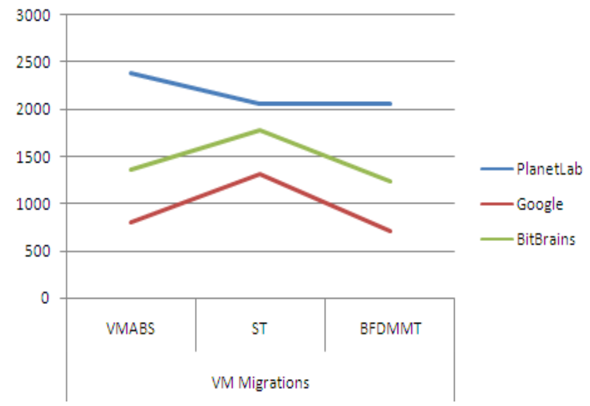

Fig 4: VM Migration among VMABS, ST and BFDMMT

Dataset	Number of Hosts Shutdown			SLATAH		
	VMABS	ST	BFDMMT (Proposed)	VMABS	ST	BFDMMT (Proposed)
PlanetLab	778	765	758	4.5	3.83	4.97
Google	773	774	770	11.38	10.27	11.89
BitBrains	766	887	740	7.22	6.07	8.43

Table 3: Comparison of NHS, SLATAH among VMABS, ST and BFDMMT

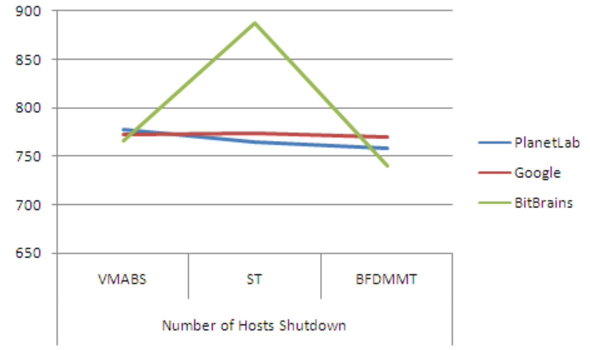

Fig 5: Number of Hosts Shutdown among VMABS, ST and BFDMMT

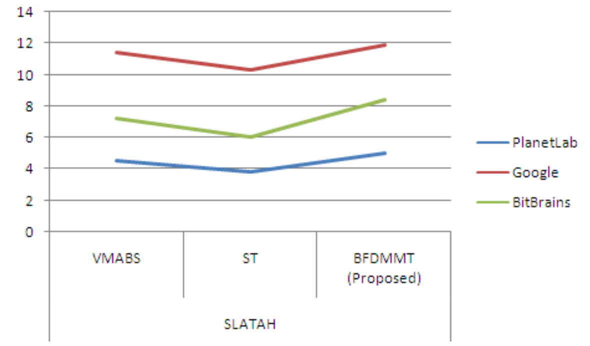

Fig 6: SLA Time per Active Hosts among VMABS, ST and BFDMMT

Dataset	Overall SLAV			Average SLAV		
	VMABS	ST	BFDMMT (Proposed)	VMABS	ST	BFDMMT (Proposed)
PlanetLab	0.16	0.13	0.11	14.42	10	9.85
Google	100	100	90	13	12.59	11.89
BitBrains	0.12	0.18	0.11	12.76	10.1	9.18

Table 3: Comparison of Overall SLA Violation, Average SLA Violation among VMABS, ST and BFDMMT

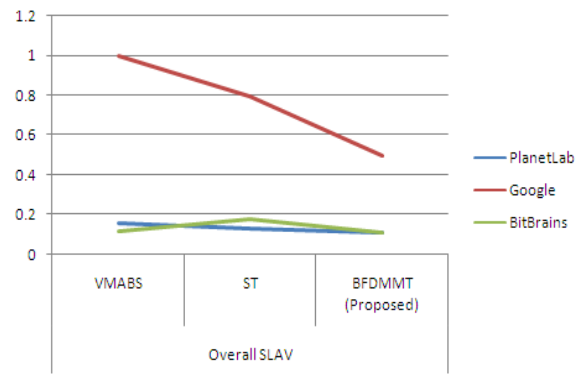

Fig 7: Overall SLAV among VMABS, ST and BFDMMT

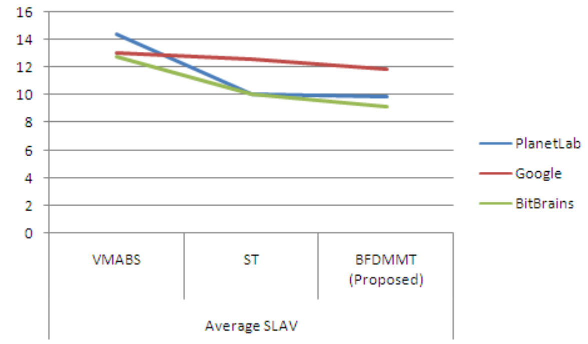

Fig 8: Average SLAV among VMABS, ST and BFDMMT

Dataset	PDM			ESV		
	VMABS	ST	BFDMMT (Proposed)	VMABS	ST	BFDMMT
PlanetLab	2.13	3.81	1.14	79.31	121	53.38
Google	1.05	2.17	0.34	50.96	103.23	45.3
BitBrains	2.17	4.6	1.8	138.95	122.61	99.41

Table 4: Comparison of Performance Degradation due to Migration, Energy and SLA Violation among VMABS, ST and BFDMMT

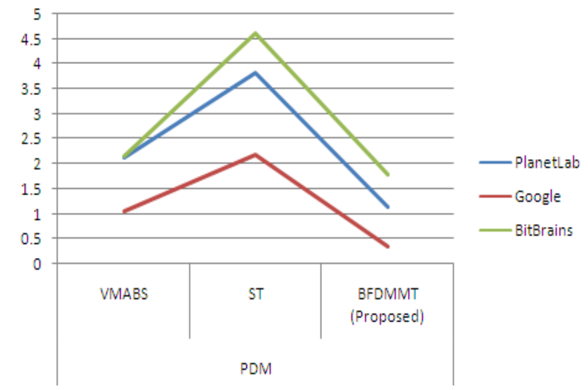

Fig 9: Comparison of PDM among VMABS, ST and BFDMMT

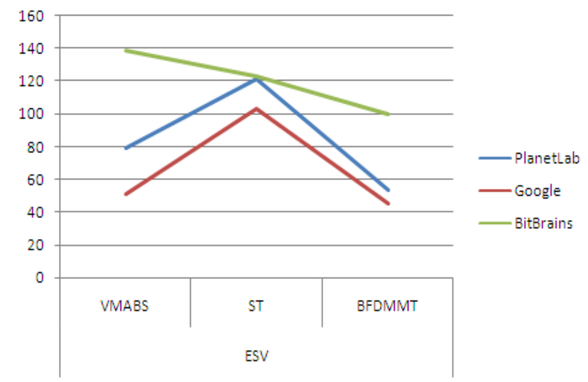

Fig 10: Comparison of ESV among VMABS, ST and BFDMMT

VI. CONCLUSIONS

Using different solutions of VM allocation problem, our proposed VM placement algorithm could make remarkable improvements over the existing solution. Our proposed techniques managed to get lower power consumption, less amount of SLA violation and less amount of performance degradation as compare than existing algorithm. We are also successful to show that VM placement is favored by higher virtual machine density which we proved by adopting allocation policy. From our result we also find out that best fit decreasing based algorithm equipped with the minimum migration time VM selection policy significantly outperforms other dynamic VM consolidation algorithms.

To maximize their ROI Cloud providers have to apply energy-efficient resource management strategies, such as dynamic consolidation of VMs and switching idle servers to power-saving modes. However, such consolidation is not trivial, as it can result in violations of the SLA negotiated with customers. We have conducted competitive analysis of the single VM migration and dynamic VM consolidation problems. We have found and proved competitive ratios for optimal online deterministic algorithms for these problems. We have concluded that it is necessary to develop randomized or adaptive algorithms to improve upon the performance of optimal deterministic algorithms. According to the results of the analysis, we have proposed novel Best Fit Decreasing with Minimum Migration Time (BFDMMT) adaptive heuristics that are based on an analysis of historical data on the resource usage for energy and performance efficient dynamic consolidation of VMs. We have evaluated the proposed algorithms through extensive simulations on a large-scale experiment setup using workload traces from more than a thousand PlanetLab, Google Cluster and Bit Brains VMs. The results of the experiments have shown that the proposed Best Fit Decreasing with Minimum Migration Time (BFDMMT) based algorithm combined with the MMT VM selection policy significantly outperforms other dynamic VM consolidation algorithms in regard to the ESV metric due to a substantially reduced level of SLA violations and the number of VM migrations.

VII. SCOPE OF FUTURE WORK

As a future work we plan to introduce fuzzy algorithm that could take advantages from different selection criteria and form a rule base for VM selection. We also suggest for making more eco friendly IT infrastructures with reasonable amount of on-demand operating cost to improve the quality of IaaS of cloud computing.

In order to evaluate the proposed system in a real Cloud infrastructure, we plan to implement it by extending a real-world Cloud platform, such as OpenStack. Another direction for future research is the investigation of more complex workload models, e.g. models based on Markov chains, and development of algorithms that will leverage these workload models. Besides the reduction in infrastructure and on-going operating costs, this work also has social significance as it decreases carbon dioxide footprints and energy consumption by modern IT infrastructures.

REFERENCES

[1] HUANG Qi-yi and HUANG Ting-lei, "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE Conf on Cloud Computing, Oct 2010.

[2] Xiaocheng Liu, Chen Wang, Bing Bing Zhou, Junliang Chen, Ting Yang and Albert Y. Zomaya, "Priority-Based Consolidation of Parallel Workloads in the Cloud", IEEE Trans on Parallel and Distributed Systems, Vol. 24, No. 9, Sep 2013.

[3] Luqun Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers", Third International Conference on Multimedia and Ubiquitous Engineering, 2009.

[4] Yung-Ching Hsu, Pangfeng Liu and Jan-Jan Wu, "Job Sequence Scheduling for Cloud Computing", International Conference on Cloud and Service Computing, 2011.

[5] Mandeep Kaur, Sugandha Sharma and Rajinder Kaur, "Optimization of Job Scheduling in Cloud Computing Environment", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 7, July 2014.

[6] Sung-Min Jung, Nam-Uk Kim and Tai-Myoung Chung, "Applying Scheduling Algorithms with QoS in the Cloud Computing" IEEE Conf on Cloud Computing jan 2013.

[7] Dr. Amit Agarwal and Saloni Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7– Mar 2014

[8] Seema Vahora and Ritesh Patel, "CloudSim-A Survey on VM Management Techniques", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2015.

[9] Yogita Chawla and Mansi Bhonsle, "A Study on Scheduling Methods in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science, Volume 1, Issue 3, September – October 2012.

[10] Gabriele D'Angelo, "Parallel and Distributed Simulation from Many Cores to the Public Cloud", IEEE International

Conference on High Performance Computing and Simulation (HPCS), July 2011.

[11] Kushang Parikh, Nagesh Hawanna, Haleema.P.K, Jayasubalakshmi.R and N.Ch.S.N.Iyengar, "Virtual Machine Allocation Policy in Cloud Computing Using CloudSim in Java", International Journal of Grid Distribution Computing, Vol.8, No.1, pp.145-158, 2015.

[12] Bhupendra Panchal and Prof. R. K. Kapoor, "International Journal of Advanced Research in Computer Science and Software Engineering", Volume 3, Issue 9, September 2013.

[13] Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud and Rashedur M. Rahman, "Implementation and performance analysis of various VM placement strategies in CloudSim", International Journal of Cloud Computing: Advances, Systems and Applications 2015.

[14] Anton Beloglazov and Rajkumar Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", International Journal of Concurrency and Computation: Practice and Experience 2012.

[15] Manan D. Shah and Harshad B. Prajapati, "Reallocation and Allocation of Virtual Machines in Cloud Computing", International Journal of Internet Services and Applications, 2010.

[16] A. Mu'alem and D. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM sp2 with Backfilling," IEEE Trans. Parallel and Distributed Systems, vol. 12, no. 6, pp. 529-543, June 2001.

[17] D. Jackson, Q. Snell, and M. Clement, "Core Algorithms of the Maui Scheduler," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 87-102, 2001.

[18] Y. Wiseman and D. Feitelson, "Paired Gang Scheduling," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 6, pp. 581-592, June 2003.

[19] A. Do, J. Chen, C. Wang, Y. Lee, A. Zomaya, and B. Zhou, "Profiling Applications for Virtual Machine Placement in Clouds," Proc. IEEE Int'l Conf. Cloud Computing (CLOUD), pp. 660-667, 2011.

[20] Y. Zhang, H. Franke, J. Moreira, and A. Sivasubramaniam, "An Integrated Approach to Parallel Scheduling Using Gang-Scheduling, Backfilling, and Migration," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 3, pp. 236-247, Mar. 2003.

[21] D. Tsafrir, Y. Etsion, and D. Feitelson, "Backfilling Using System-Generated Predictions Rather than User Runtime Estimates," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 6, pp. 789-803, June 2007.

[22] D. Feitelson, "Packing Schemes for Gang Scheduling," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 89-110, 1996.

[23] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riordan, "Modeling of Workload in Mpps," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 95-116, 1997.

[24] D. Feitelson, L. Rudolph, U. Schwiegelshohn, K. Sevcik, and P. Wong, "Theory and Practice in Parallel Job Scheduling," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 1-34, 1997.

[25] E. Von Puttkamer, "A Simple Hardware Buddy System Memory Allocator," IEEE Trans. Computers, vol. C-100, no. 10, pp. 953-957, Oct. 1975.

[26] K. Knowlton, "A Fast Storage Allocator," Comm. ACM, vol. 8, no. 10, pp. 623-624, 1965.

[27] http://www.netbeans.com

[28] http://www.gwa.ewi.tudelft.nl

[29] http://www.Plagiarismdetector.com