ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

VIRTUAL MACHINE PLACEMENT AND LOAD REBALANCING ALGORITHMS IN CLOUD COMPUTING SYSTEMS

T. Thiruvenkadam*, P. Kamalakkannan

* Part Time Research Scholar in Department of Computer Science, Periyar University, Salem, Working in Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode - 637215, Tamil Nadu, India

Department of Computer Science, Arignar Anna Govt., Arts College, Namakkal - 637002,Tamil Nadu, India

DOI: 10.5281/zenodo.59659

ABSTRACT

The goal of virtual machine placement in cloud environment is to provide better service to the cloud users, and also the effective use of available resources. In this paper a review of various algorithms has been made and We present an approach to optimal virtual machine placement within datacenters. A technique for finest load balancing is developed, based on the request intensity. This approach classifies the request based on the intensity into high, medium and low then three different enhanced algorithms are used to place the VMs in the respective queues to maximize the resource utilization rate and minimize the Number of PMs used, Response time, SLA Violation Rate, Power Usage, Load Imbalance Rate, and Migration Rate. The evaluation results propose that our algorithms are realistic, and that these can be used in the cloud environment for placing the VMs effectively to the PMs.

KEYWORDS: Physical machine, Virtual machine, VM Placement, Load Rebalancing, VM Migration, Load Monitoring.

INTRODUCTION

Cloud computing has gained popularity due to its various characteristics like cost economical and on-demand / pay as use services [1] that are independent of time and geographical locations. It is a general term used to describe a collection/group of integrated and networked hardware, software and Internet infrastructure. These platforms hide the complexity and details of the underlying infrastructure from users and applications by providing very simple graphical interface or API (Applications Programming Interface).

Cloud computing technology allows developers and IT professionals with the ability to focus on significant matters and frees them from works like maintenance, procurement and capacity planning. As cloud computing has grown in popularity, several different models and deployment strategies have emerged to help meet specific needs of different users [2].Each type of cloud service and deployment method provides different levels of control, flexibility, and management. There are three fundamental Service models in Cloud computing, namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [3]. All these three services can be deployed in four different ways, namely, private, public, community and hybrid cloud [4].

Cloud computing systems consists of several elements that includes virtualization, distributed computing, service oriented architectures, broadband networks, browser as a platform, servers, SAN/NAS (Storage Area Network/Network Attached Storage) and free and open source software. Each of these elements performs tasks with the main goal of offering better services to users. Out of the various elements, this research work is focused on enhancing the task of virtualization in order to improve the working of the cloud computing system.

Virtualization, a key concept of cloud computing, is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources. It is the process by which one computer hosts the appearance of many computers [5]. Without virtualization, all machines require same power, emit same heat, need same physical space, setup cost, maintenance overhead, support overhead, cost per hardware etc. are directly proportional to the number of machines. Usage of virtualization increases the resource utilization by sharing physical resources among multiple users and applications. Sharing of resources provides multi-advantages like Increased the resource utilization by sharing physical resources among multiple users and applications which also helps in help cost reduction, provides Isolation, encapsulation, hardware independence and Portability [6].

Virtualization technology has two main components, namely, Virtual Machine (VM) and hypervisor or Virtual Machine Manager (VMM). VM is an isolated runtime environment (guest OS and applications). Multiple virtual systems (VMs) can run on a single physical system. The Hypervisor or Virtual Machine Manager (VMM) is a program that allows multiple operating systems to share a single hardware host. Each guest operating system appears to have the host's processor, memory, and other resources all to itself. However, the VMM is actually controlling the host processor and resources allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other [7].
Virtualization in cloud computing involves three broad stages.

- Stage 1 - Application Profiling : In this stage, the applications are profiled in its physical environment in order to obtain its resource utilization.
- Stage 2 - Generation of VM Configurations : In this stage, the above obtained knowledge is used to generate configurations for VMs.
- Stage 3 - VM Placement : This stage uses the generated VM configurations to identify the optimal manner of mapping them onto physical machines (PMs).

RELATED WORK

Even though a lot of researchers have been studied this virtual machine mapping problem in the past we draw attention to some of the closest work in perspective of our point. In[8] the number of physical machines needed to deploy the requested virtual machine instances are reduced by combining time series forecasting techniques and bin packing heuristic but the model has not included the relationships between multiple resources, like CPU and I/O. In [9] the VM placement algorithms make use of the behavior of VMs to have some properties in general. In[10] for the placement of virtual machines to physical machines a two level control management system is used and it uses combinatory and multi-phase efficiency to solve potentially inconsistent scheduling constraints. In[11], VM scheduling constraints are considered as single dimension in a multidimensional Knapsack problem.

In[12], the VM scheduling policy is primarily dealt out from the viewpoint of network traffic and three common scheduling algorithms have been introduced for Cloud performing load balancing in data centers are intensively studied the heuristics has been used as a common approach among systems to enables the load balancing among physical servers. In[14] the performance variations have been identified and monitored in a physical server hosting VMs. A few simple VM placement algorithms like time-shared and space-shared were presented and compared in [15] and introduced a method to model and simulate Cloud computing environments, in which the algorithms can be implemented. In [16] pioneered methods for allocating and migrating virtual machines and proposed some migration techniques and algorithms based on the load imbalance level of the servers. [17] Evaluated most important load-balance scheduling algorithms for conventional Web servers. Vector Dot a novel load-balancing algorithm has been introduced in [18] to work with structured and multi-dimensional resources limitations by taking servers and storage of a Cloud into account. A countable measure of load imbalance on virtualized data center servers has been proposed in [19]. In[20] server consolidation was considered as a stochastic been packing problem and presented a VM sizing based algorithm which considers the cumulative resource demand of a host where the VM to be placed. An overloaded resource based VM placement approach has been presented in [21]. In our previous study [22] the comparison of various VM scheduling algorithm has been presented and demonstrated the necessity of new efficient placement VM placement algorithm. An algorithm for scheduling virtual machines have been presented in [23] based on user constraints and multi dimensional host load.

A genetic based simulated annealing algorithm for optimization of task scheduling in cloud computing has been proposed and implemented in [24]. This algorithm only considers the QOS necessities of various types of tasks. Some of the genetic operators that use the group-oriented structure lead the better results when compared to the non-grouping genetic based algorithms which are not use such grouping feature. In [25],[26] they used the grouping based genetic algorithm to reach better results than conventional methods and universal heuristic algorithms.

VM placement is defined as the process of mapping the VM requests to the PMs, according to the availability of resources in these hosts. VMs must be distributed in an efficient way such that no system or a request starves for the response from cloud [28]. The primary goal in VM placement task is to maximize the usage of the available resources.

Previously, when the number of VMs and PMS were small, mapping of VMs to appropriate PMs were possible manually. However, the current scenario faces a tremendous increase in the number of VMs and PMs, which makes automation of placement task mandatory. Existing automated solutions have to evaluate several number of possible mappings for a given set of VMs and PMs and thus, require improved intelligent placement heuristics to narrow down the search for a solution to obtain near-optimal placement plans. Moreover, the following issues were identified in the existing solutions.

- Issue 1 - Majority of the existing algorithms use single dimension during VM placement. However, today's' dynamic environment requirement involves multiple dimensions.
- Issue 2 - The VM-PM mapping task is not normally tuned to resource request details during scheduling
- Issue 3 - Determining which VMs to place on which PMs is a critical point that effect the performance of the system, which is still an open research problem.

The focal point of this research work is to find solutions to the above identified issues and incorporate them into the VM placement algorithm.[32] The system after incorporating solutions to the above issues is referred in this research work as "Virtual Machine Placement and Load Rebalancing Based on Multi-Dimensional Resource Characteristics in Cloud Computing Systems (VMP-LR)".

COMPONENTS OF PROPOSED VMP-LR

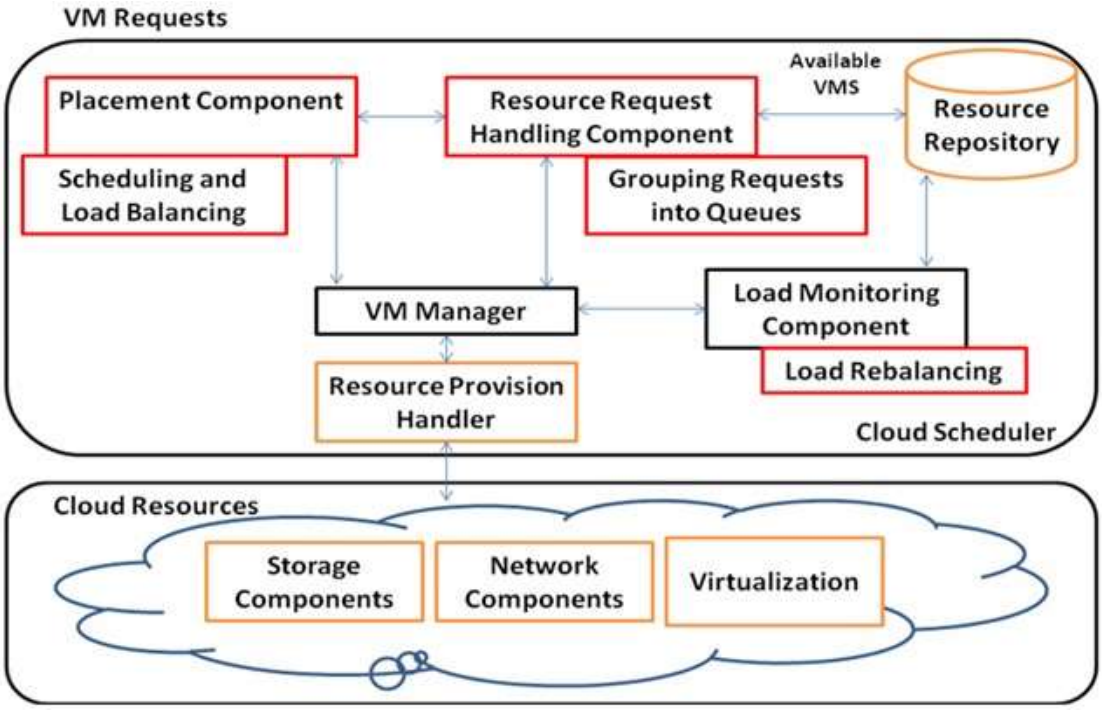


Fig 1: Components of VMP-LR

The proposed VMP-LR consists of three main components, namely, Resource Request Handling Component, Placement Component and Load Monitoring Component. The Resource Request Handling Component creates VM queues for each data centre using Queuing Algorithm based on Multi-dimensional Resource Characteristics. The VM placement Component uses traffic and load aware Scheduling Algorithms to map VMs to PMs efficiently. These algorithms are also fine tuned to efficiently handle high, medium and low resource requests. The third component, that is, Load Monitoring Component (LMC), performs monitoring the VMs periodically in terms of the selected resources and when usage of the PM drops below a threshold, performs rebalancing using an algorithm.This algorithm is based on ant colony optimization and can avoid over and under utilization of resources. All these components are interconnected and will be monitoring continuously by the VM manager.

Phase I : Queuing and Request Handling Components

As mentioned earlier, the first phase of the research methodology proposes algorithms to create request queues and methods to schedule them efficiently. This section presents details regarding the algorithms used for this purpose.

VM Queuing

The VQC collects the resource requests and classifies them into three different queues based on the current load and resource availability at a particular time, T. The three different queues generated are High Resource Queue (HRQ), Medium Resource Queue (MRQ) and Low Resource Queue (LRQ). In a dynamic environment, this grouping is performed for each Physical cluster of a datacenter, thus obtaining multiple queues under each category. These queues are created based on two thresholds, T1 and T2, which are estimated as 20% and 70% of current load of the Physical clusters respectively. All requests exceeding T2 are placed in HRQ and requests below T1 are placed in LRQ and the remaining requests are placed in MRQ.

Initially, all queues are considered empty and requests are placed into their respective category after classifying them using the thresholds T1 and T2. As the whole process is considered dynamic, after initial queue formation, the category of each new request is first identified and is then placed in a shortest queue using the Join Shortest Queue algorithm. The JSQA routes new requests to the shortest queues that is created from various data centres and also keeps track of queue length at all data centres.

Scheduling and Load Balancing

Model Design for VMP-LR System N number of virtual machines with resource requirements VR (CPU, Memory, N/W Bandwidth) to be placed on a set of M physical machines with resource capacities of PR(CPU, Memory, N/W Bandwidth) grouped in K number of physical machine cluster. Consider *PM* as a set of all the physical machines in the entire system, where PM = {PM_1, PM_2, PM_3 … PM_m }. m is total number of the physical machines and an individual physical machine can be denoted as *PM i*, where i denote the physical machine number and range of i is (1 <= i <= m). Similarly, the set of VMs on the physical machine i, can be {VMi1, VMi2….VMi n} here n is the number of VMs on the physical server i. If we want to deploy VM j on the PMi then the load of the CPU, RAM and bandwidth has to be calculated individually.[29] The CPU load of the PMi at the time interval ts is denoted as follows

$$\mathrm{PMi}(\mathrm{cpu},\mathrm{ts}) = \sum_{j=1}^{n} \mathrm{VMij}(\mathrm{cpu},\mathrm{ts}) \quad (1)$$

The amount of RAM utilized by all the VMs of PMi at the time interval ts can be denoted as follows,

$$\mathrm{PMi}(\mathrm{ram},\mathrm{ts}) = \sum_{j=1}^{n} \mathrm{VMij}(\mathrm{ram},\mathrm{ts}) \quad (2)$$

The amount of Network Bandwidth utilized by all the VMs of PMi at the time interval ts can be denoted as follows

$$\mathrm{PMi}(\mathrm{nbw},\mathrm{ts}) = \sum_{j=1}^{n} \mathrm{VMij}(\mathrm{nbw},\mathrm{ts}) \quad (3)$$

Where PMi represents the i^{th} physical machine of the Physical Machine Cluster k, VMij represents j^{th} virtual machine of the PMi and cpu, ram and nbw denotes the amount of CPU, RAM and Network Bandwidth utilized by all the VMs

of the PMi respectively.Hence derived from (1),(2) and (3) the weighted average load of the Physical Machine Cluster k at time interval ts can be denoted as follows

$$\mathrm{PMCk(WL, ts)} = \sum_{i=1}^{m} \mathrm{PMi\ (WL\ , ts)} \qquad (4)$$

Where PMCk represents the k^{th} physical machine cluster of the datacenter,WL represents the weighted load of physical machine cluster at time interval ts and PMi represents the i^{th} physical machine of the Physical Machine Cluster k.At any time interval the total VM load of a PM should not exceed the host capacity

$$\sum_{\text{resource}} \mathrm{PMi}\ W_{\text{resource usage}}\ (ts) \leq \text{TH value} \leq \sum_{\text{resource}} \mathrm{PMi}\ W_{\text{resource capacity}} \qquad (5)$$

Where resource € {CPU, RAM, Network Bandwidth} and $W_{resource}$ is the weight associated with each resource TH value is the threshold value set by the administrator if the load goes beyond this value the host can be considered as overloaded host and the selected VMs has to be migrated to other appropriate physical machines.

Algorithm Design for Dynamic VM placement

The objective is to place the VMs in PMs in a way that the total number of PMs required to place all the VMs is decreased. So we considered this a multi potential bin packing problem since this is a NP-hard problem, we provide a heuristic based on multiple policy. In the earlier stages of allocation most of the PMs are underutilized or not used so our heuristics works as like the first fit scheduler which is a simplest one to implement and which increases the response time of VM placement. As the number of VM grows in the datacenter the utilization level of PM is also being considered by our heuristic which really helps in maintaining the balanced load among servers. Towards the closing stages the heuristic works according to the nature of the VMs workload that is gathered from the user provided hints which helps in avoiding the bottleneck of a particular resource as well as avoiding the violence of any SLA agreements. The algorithm which is used to achieve these things is given below.

The focal point of VPC is to perform scheduling optimally in order to reduce time and energy and at the same time improve resource utilization through load balancing. For this purpose, this research work proposes a set of Hybrid Scheduling and Load Balancing algorithms. The VM request handling component performs the following steps in order to perform VM placement.

(i) Determine traffic load to identify rush and non-rush hour
(ii) Use appropriate algorithms to handle requests in each queue category and traffic load to perform placement.

The two most important aspects of scheduling are to reduce energy consumption and to effectively use the resources available. One way to incorporate these desirable aspects is using the knowledge regarding the traffic and by using effective scheduling algorithm. Scheduling of VMs has to be done correctly by using an appropriate load balancing technique, as it has a direct performance impact on the entire cloud system. In this research, in order to conserve energy, the second step of the proposed algorithm uses methods that can work efficiently during rush and non-rush hours.

To perform VM placement efficiently, the VMP-LR proposes four scheduling and load balancing hybrid algorithms (three for rush hour and one for non-rush hour) to optimally place the VM to an appropriate PM. During rush hour, in order to further optimize the process of scheduling, separate hybrid algorithms that is tuned up to work efficiently with HRQ, MRQ and LRQ are used. The three hybrid algorithms proposed are listed below.

- High Request Queue – Scheduling and Load Balancing Algorithm using Enhanced Max-Min, Ant Colony Optimization (ACO) and Artificial Bee Colony (ABC) (SLAM2A) algorithm
- Medium Request Queue – Hybrid Scheduling and Load Balancing algorithm based on First Fit, Best Fit algorithm and multi-level grouping genetic (SLAFBG) algorithm[30]
- Low Request Queue – Hybrid Scheduling and Load Balancing algorithm based on Enhanced Max-with Particle Swarm Optimizer (SLAMP) algorithm

The SLAM2A algorithm, designed to handle requests in HRQ, is a hybrid algorithm that combines the advantages of enhanced max-min algorithm, Ant Colony Optimization (ACO) and Artificial Bee Colony (ABC). The conventional max-min algorithm is designed to consider execution time and considers requests with maximum execution time first. In this research work, this algorithm is modified to consider both average execution time of the requests (et) and processing speed (ps). These two factors are selected so that the proposed algorithm is adaptable to both execution time and resource requests that produces minimum completion time. Based on these two factors, a resource makespan constraint is created as the product of et and ps. Using this factor, the scheduling is performed using the concept of max-min algorithm.

The Ant Colony Optimization algorithm is one of the most recent algorithms, which has been shown to be competitive to other algorithms[31]. The ACO algorithm is enhanced by combining it with ABC algorithm. This hybrid algorithm is designed to combine the advantages of the global search ability of ABC and the local search ability of ACO algorithm along with the advantages provided by max-min scheduling algorithm. The algorithm initially uses enhanced Max-Min algorithm for scheduling requests while load balancing is performed by ACO enhanced with ABC algorithm.

The SLAFBGA algorithm, designed for handling the requests in MRQ, is a hybrid method based on Dynamic Bin-Packing (DBP) is used. The DBP assumes that requests arrive and depart at arbitrary times, which is the scenario of non-rush hour. The bin packing problem is a combinatorial NP-hard problem. The proposed VM-placement algorithm combines two algorithms to minimize the number of PMs required to place a set of VMs, quick and correct placement of VMs, perform load balancing, increase resource utilization rate without violating any SLA (Service Level Agreements). Here, the PMs are considered as bins and the VM's to be placed are considered as objects to be filled in the bin. The two algorithms used are first fit and best fit algorithms. Initially, as most of the PMs are underutilized or not used, the proposed algorithm works like the first fit scheduler. This algorithm is selected because of its dual advantages of (i) being simple to implement and (ii) increases the response time of VM placement. Later, when the utilization level of PM increases, in order to balance load among server, the proposed algorithm uses best fit algorithm. A multi-level grouping genetic algorithm is used for load balancing among physical servers

The SLAMP algorithm, designed for handling requests in LRQ, enhances PSO algorithm through the use of enhanced Max-Min algorithm. Careful scrutiny of PSO algorithm revealed that the performance of the algorithm is dependent on the initial population and fitness function. In the proposed SLAMP algorithm, the initial population is automatically generated using the enhanced Max-Min algorithm and the fitness function is modified to minimize the makespan, thus improving the resource utilization.

During non-rush hour, all the requests are handled using the round robin algorithm. The algorithm mainly focuses on distributing the load equally to all the nodes. Using this algorithm, the scheduler allocates one VM to a node in a cyclic manner. The main advantage of this algorithm is that it utilizes all the resources in a balanced order and further equal number of VMs are allocated to all the nodes which ensure fairness.

PERFORMANCE EVALUATION

4.1 Experimental Setup

The presented algorithm is implemented in JAVA Net beans IDE. Then we use CloudSim simulator for simulation to appraise the execution and performance of our heuristics with some of the existing scheduling algorithm in terms of number of active PMs, Response Time, Resource Utilization, SLA violations and Power consumption. The performances of the proposed algorithm were examined from both users and service provider's perception.

Since it is difficult to access the real datacenters or cloud infrastructures we used simulation based evaluation which can be easily reproducible to compare the performance of the proposed algorithm with the existing works which is currently used by the majority of the cloud service providers. The simulated cloud environment contains a cluster of PMs the total resource capacity of PMs is expressed in percentage and randomly generated VM resource demand includes the number of CPU cores, amount of RAM and required network bandwidth.

ISSN: 2277-9655
Impact Factor: 4.116

4.2 Analysis

The investigations are done to analyze the effect our proposed algorithm in number of physical servers required to place a certain number of VMs, the time taken for placing a set of VMs, overall resource utilization rate of all the active servers, SLA violation rate of all the active VMs and the total power consumption of all the active PMs . The simulation results show that our proposed algorithm can use the less number of physical servers for placing a certain number of VMs which helps to improve the resource utilization rate. It is observed that the minimum time has been taken for placing a set of VM requirements when compared with other existing algorithms. The results shows that less percentage of SLA violations and less power consumption than the existing algorithms.

Stage 1 - Evaluation of Proposed Scheduling and Load Balancing Algorithm

The experiments were designed in three stages in the first stage the proposed scheduling and load balancing algorithms are compared with their existing counterparts. It is shown that in all the five parameters the proposed algorithms outperforms their existing counterparts.

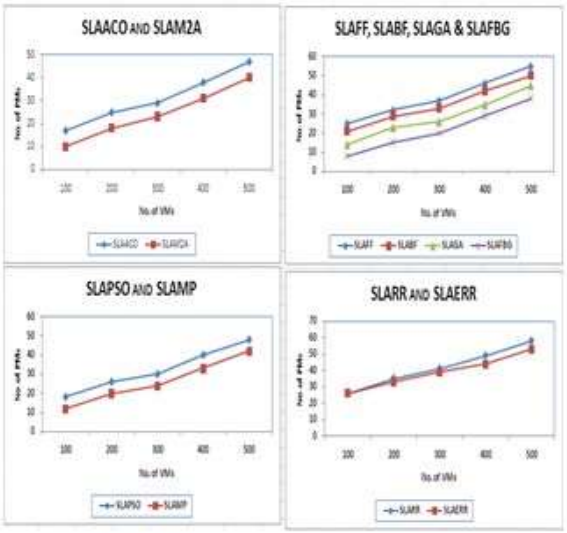


Fig 2: Number of Active PMs Used

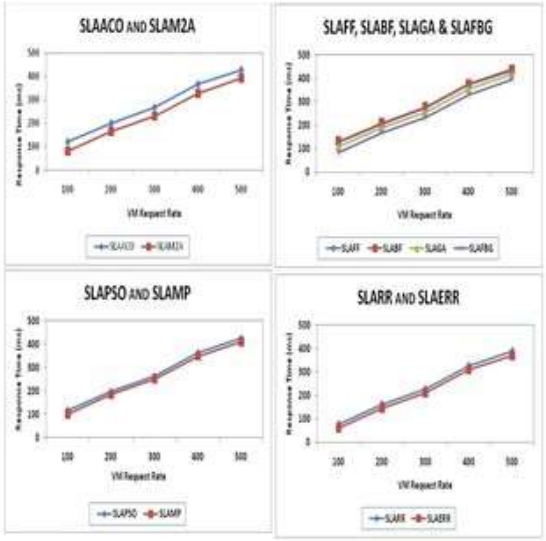


Fig 3: Response Time

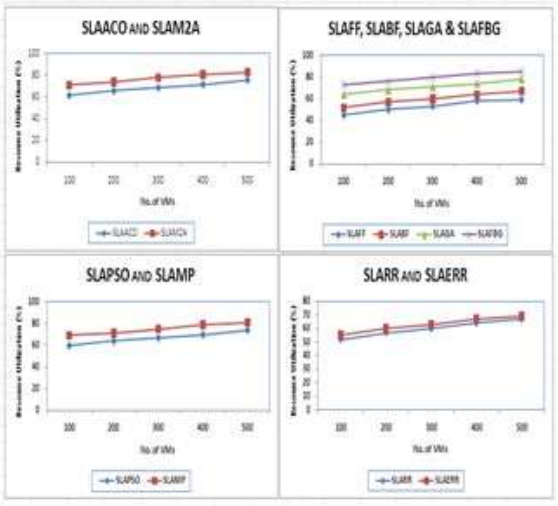


Fig 4:Resource Utilization

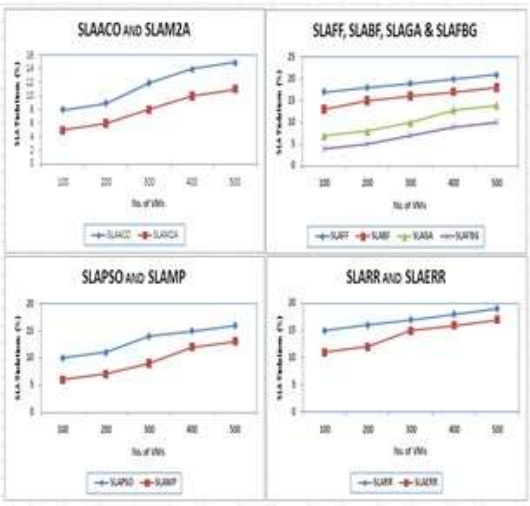


Fig 5: SLA Violations

ISSN: 2277-9655
Impact Factor: 4.116

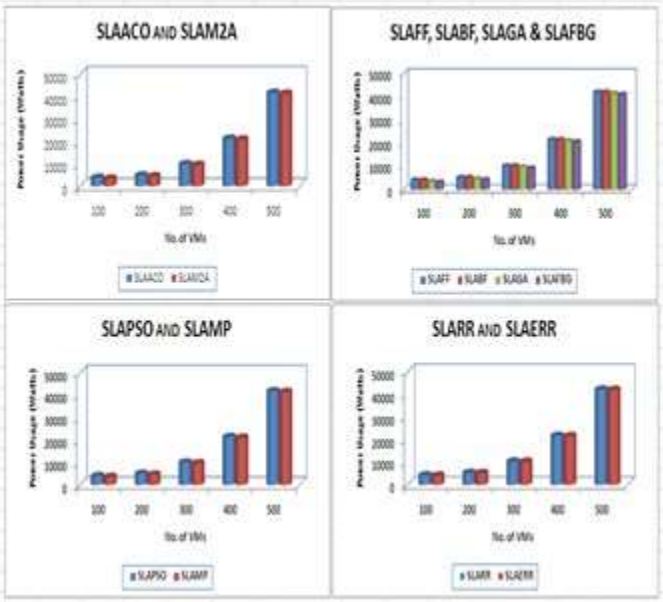


Fig 6: Power Usage

Stage 2 – Effect of Queuing algorithm

In stage two the experiments are done to identify the effect of applying queuing algorithm is examined for all the proposed VM placement algorithms by using the algorithm on the set of VM requests which comes in a single queue and also the VM requests which is classified into three different queues based on the proposed queuing algorithm. The results are shown that all the proposed VM placement algorithms produces good results in all the cases when it is used on the requests which are classified into three different queues.

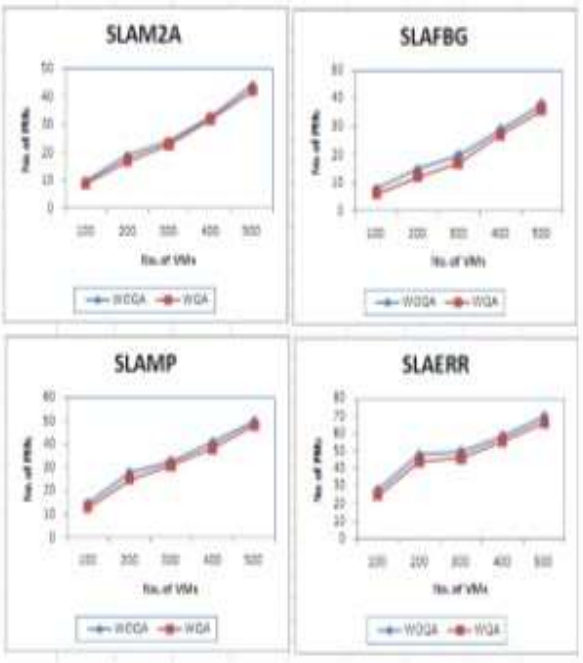


Fig 7: Effect of Queuing on No. of Active PMs

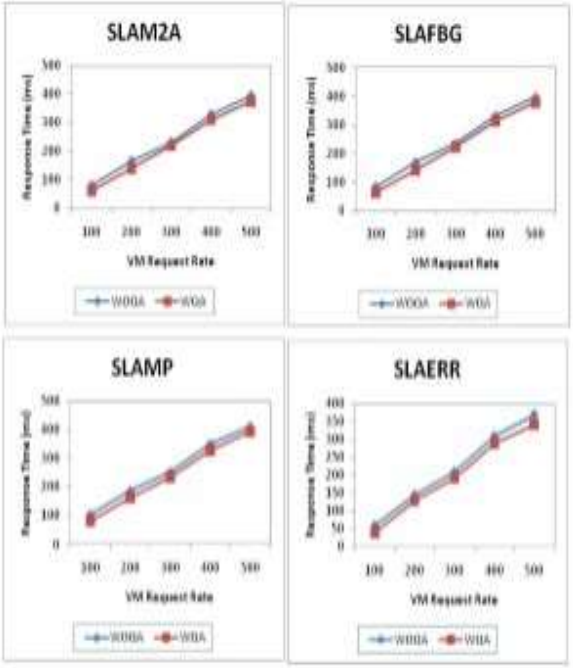


Fig 8: Effect of Queuing on Response Time

ISSN: 2277-9655
Impact Factor: 4.116

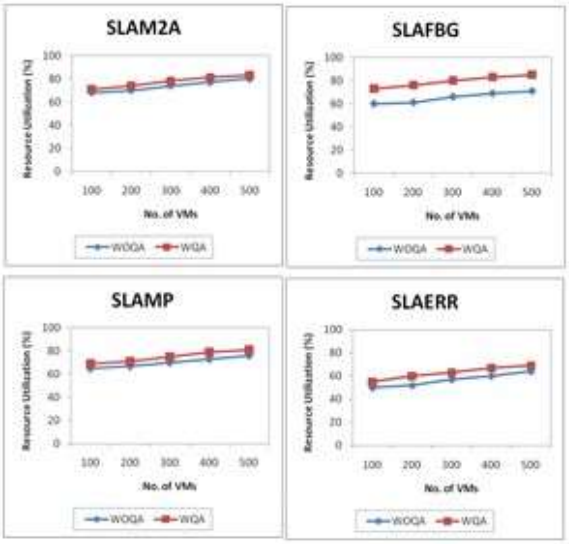


Fig 9: Effect of Queuing on Resource Utilization

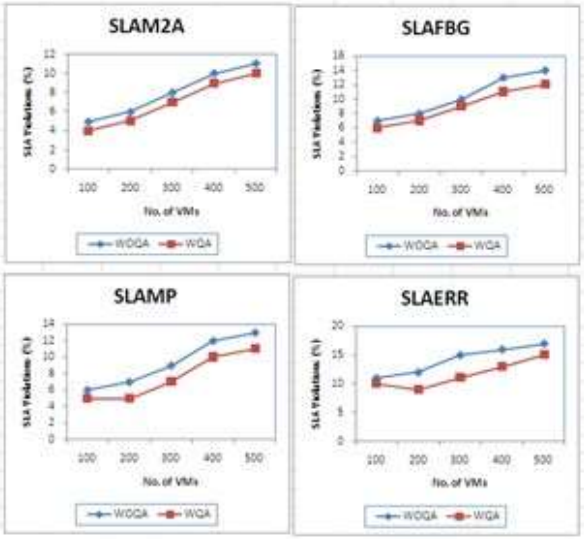


Fig 10: Effect of Queuing on SLA Violations

Fig 11: Effect of Queuing on Power Usage

Stage 4 – Analysis to Map Queues to any One of the Proposed Algorithms

As VM requests has been classified into high, medium and low resource requests queues, the suitable algorithms for these queues needs to be identified. In stage four the experiments were conducted to identify the suitable algorithms for the three different queues. The below results shows that The SLAM2A algorithm, provides better results in handling requests in HRQ. The SLAFBGA algorithm, is more suitable for handling the requests in MRQ. The SLAMP algorithm, performs well for handling requests in LRQ than the requests in other two queues.

Fig 12: Effect of VM request Rate on No. of Active PMs

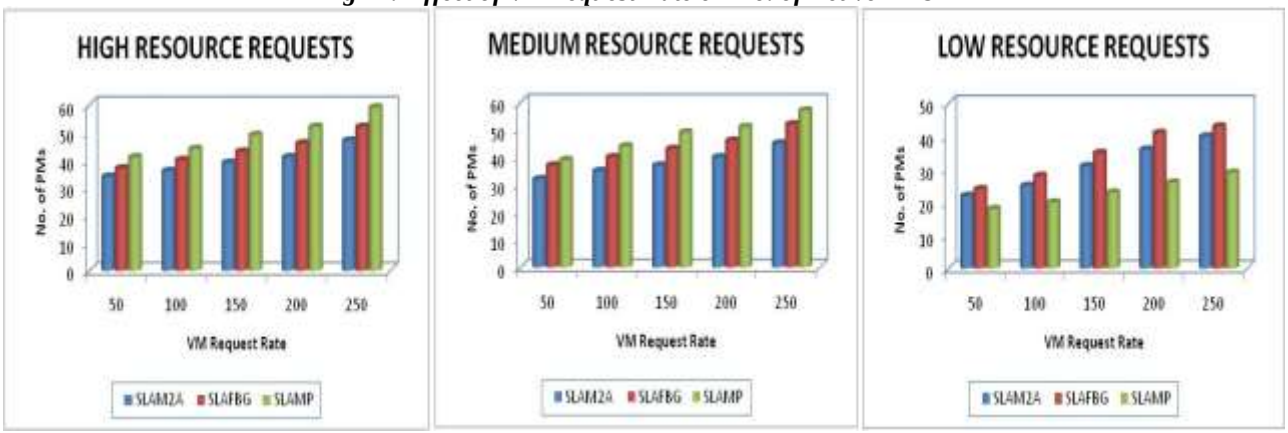
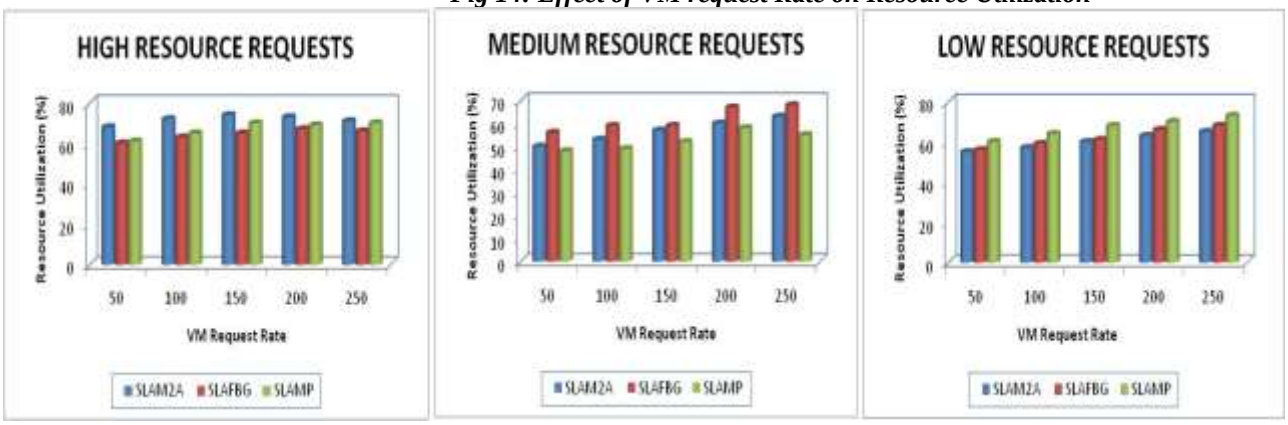


Fig 13: Effect of VM request Rate on Response Time

Fig 14: Effect of VM request Rate on Resource Utilization



ISSN: 2277-9655
Impact Factor: 4.116

Fig 15: Effect of VM request Rate on SLA Violations

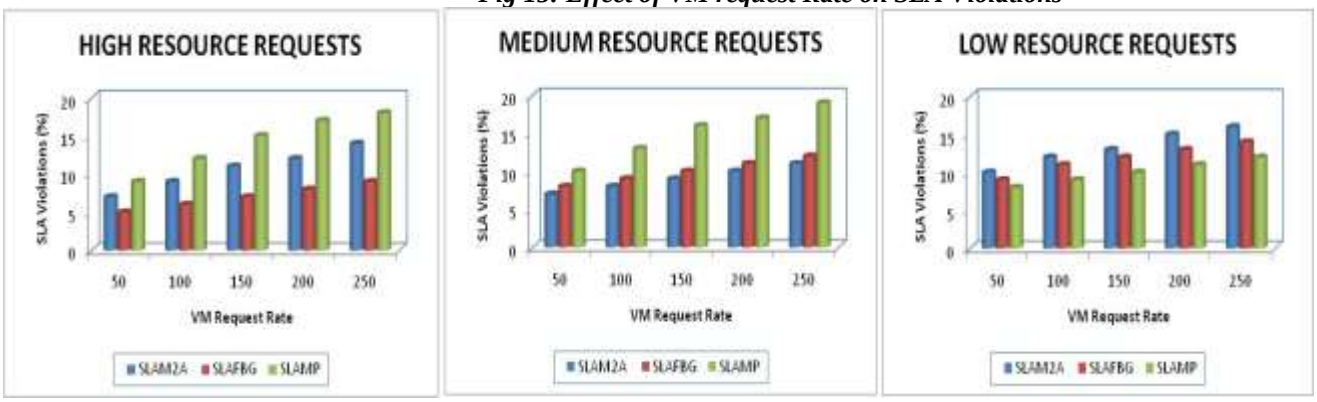
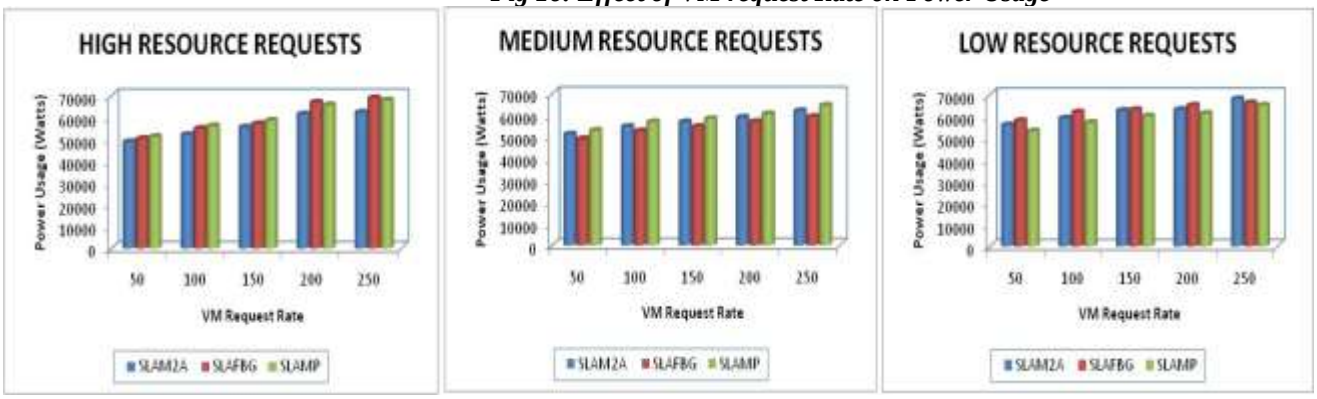


Fig 16: Effect of VM request Rate on Power Usage



Stage 5 – Analysis of Rebalancing Algorithm

Load rebalancing is a process of migrating services among hosts to ensure uniform resource distribution. Increased load imbalance factor results in resource fragmentation and thus leads to degradation in server resource utilization. Load imbalance also arises when existing services are stopped either by the cloud customers or in the event of host power cycling. This scenario can be managed through the use of load rebalancing algorithms

The Load Rebalancing using ACO Algorithm and Load Rebalancing using ACO and ABC Algorithms are separately analyzed against the VM placement done by all the existing algorithms and proposed VMP-LR algorithm. The results shows that there is a less load imbalance rate and less percentage of VM migrations rate are observed when using LR2A algorithm.

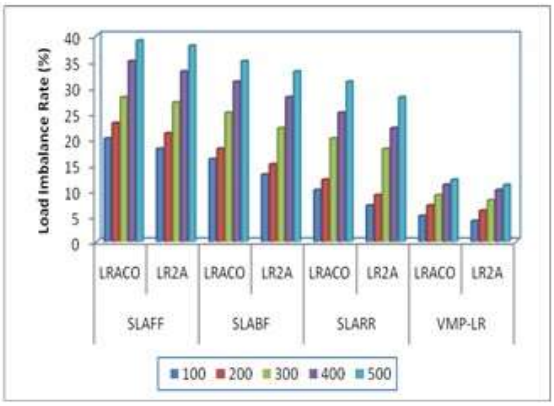


Fig 17: Load Imbalance Rate

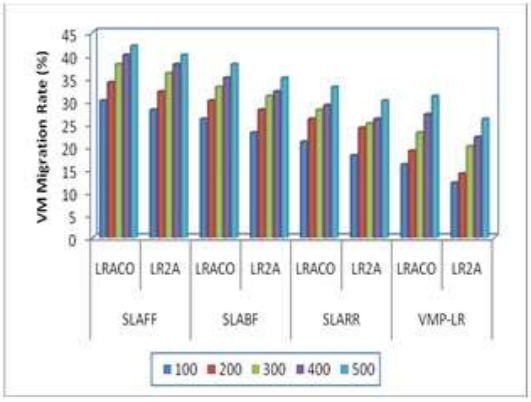


Fig 18: VM Migration Rate

Stage 3 – Evaluation of Proposed VMP-LR

In stage three the performance of the proposed VMP-LR is examined against the existing First fit, Best Fit, Round Robin, Genetic Algorithm, Ant colony Optimization, Particle Swarm Optimization algorithms. The results shows that the VMP-LR system can use decreased number of PMs, less response time, high resource utilization rate, lower SLA violation rate and less power consumption when compared to the existing algorithms

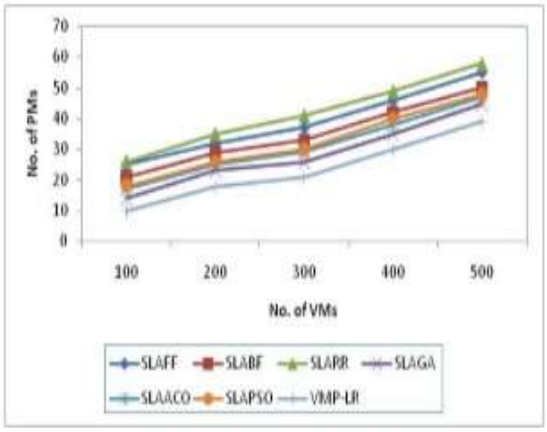


Fig 19: Number of Active PMs Used

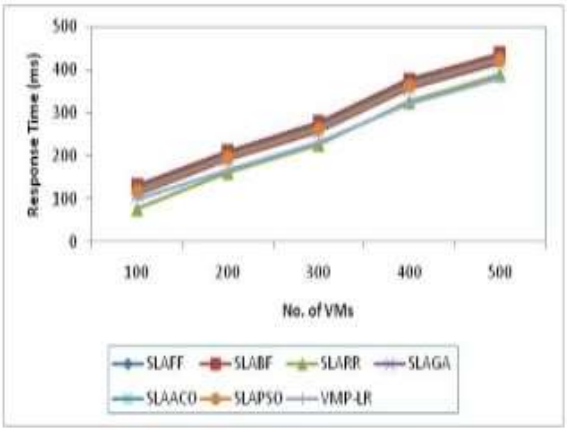


Fig 20: Response Time

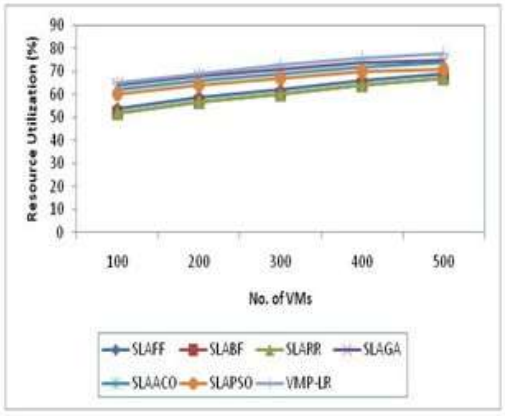


Fig 21: Resource Utilization

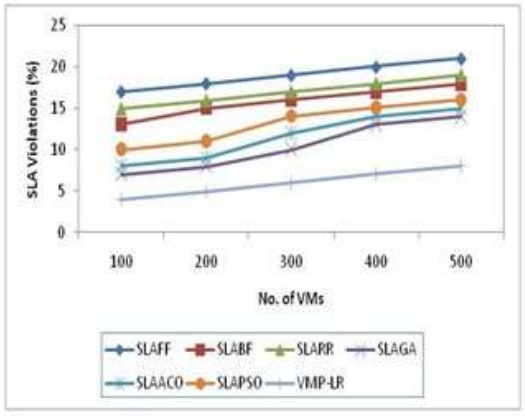


Fig 22: SLA Violations

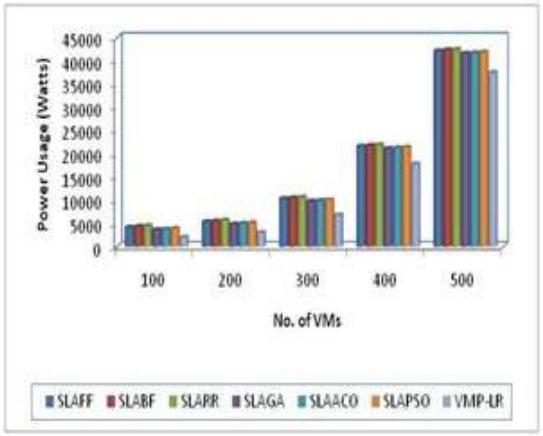


Fig 23: Power Usage

CONCLUSION AND FUTURE WORK

We presented our novel algorithm that considers user constraints of VM along with physical host load factor to address the problem of mapping the VMs into PMs such that the number physical host used is minimized, the overutilization and underutilization of the resources of a host can be identified and resolved at the same time without violating any SLA agreements. Since we consider this as a multi potential bin packing problem we combined three different heuristics which considers load factor of hosts along with user provided information at the various stages of placing the VMs in physical hosts. Based on our analysis we showed that our proposed algorithm utilizes minimum number of physical servers for hosting the set of VMs, which also reduces the energy consumption of the datacenter and it achieved high resource utilization rate by the way of using minimal number of physical servers. Another considerable enhancement in our algorithm is less percentage of load imbalance value and the percentage of VMs that violate their SLA.

As our future work we planned to incorporate the proposed algorithm with an open source cloud platform and test its efficiency against real time environment and also we would like to modeling the interconnection prerequisites that can correctly express the relationships between VMs consolidated in the same host which will be valuable for additional optimizations of VM scheduling in cloud infrastructure.

REFERENCE

[1] Ramesh, C., Shalini, S., Suganya, S. and Suvitha, N. (2016) An Approach for Efficient Dynamic Memory Allocation Using Skewness Algorithm in Cloud Environment, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 2, Pp. 1648-1653.
[2] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) Cloud computing and emerging IT platforms : Vision. Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, Vol. 25, No. 6, Pp. 599-616.
[3] Rani, B.K., Rani, B.P. and Babu, A.V.(2015) Cloud computing and inter-clouds - Types, topologies and research issues, Procedia Computer Science, Vol. 50, Pp. 24-29.
[4] Rajasekar, B. and Manigandan, S.K. (2015) An Efficient Resource Allocation Strategies in Cloud Computing, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 2, Pp. 1239-1244.
[5] Obasuyi, G.C. and Sari, A. (2015) Security challenges of virtualization hypervisors in virtualized hardware environment, International Journal of Communications, Network and System Sciences, Vol. 8, Pp. 260-273.
[6] Tamane, S. (2015) A review on virtualization : A cloud technology, International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3, Issue 7, Pp. 4582-4585.
[7] Durairaj, M. and Kannan, P. (2014) A study on virtualizaiton techniques and challenges in cloud computing, International Journal of Scientific and Technology Research, Vol. 3, Issue 11, Pp. 147-151.
[8] Norman Bobroff, Andrzej Kochut, Kirk Beaty Dynamic Placement of Virtual Machines for Managing SLA Violations
[9] Sindelar, M., Sitaraman, R.K., Shenoy, P.: Sharing-Aware Algorithms for Virtual Machine Colocation. In: Proceedings of the 23rd ACM Symposium on Parallelism in Algorithms and Architectures. San Jose, California, USA (June 2011)
[10] Xu, J., Fortes, J.A.B.: Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In: Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing. Hangshou, PR of China (Dec 2010)
[11] Singh, A., Korupolu, M., Mohapatra, D.: Server-Storage Virtualization: Integration and Load Balancing in Data Centers. In: Proc. of the 2008 ACM/IEEE conference on Supercomputing (SC'08). pp. 53:1–53:12. Austin, TX (2008)
[12] Meng, X., Pappas, V., Zhang, L.: Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement. In: Proceedings of IEEE INFOCOM. San Diego, CA, USA (March 2010)
[13] Kumar, S., Talwar, V., Kumar, V., Ranganathan, P., Schwan, K.: vManage: Loosely Coupled Platform and Virtualization Management in Data Centers. In: Proceedings of the 6th international conference on Autonomic computing. pp. 127–136. ACM, Barcelona, Spain (June 2009)

[14] Wood, T., Shenoy, P., Venkataramani, A., Yousif, M.: Black-box and Gray-box Strategies for Virtual Machine Migration. In: Proc of the 4th USENIX Symposium on Networked Systems Design and Implementation. Cambridge, MA (2007)
[15] Bobroff, N., Kochut, A., Beaty, K.: Dynamic Placement of Virtual Machines for Managing SLA Violations. In: Proc of the 10th IFIP/IEEE International Symposium on Integrated Network Management. Munich, Germany (May 2007)
[16] T. Wood et. al., Black-box and gray-box strategies for virtual machine migration, proceedings of Symp. on Networked Systems Design and Implementation (NSDI), 2007
[17] H. Zheng, L. Zhou, J. Wu, Design and implementation of load balancing in web server cluster system, Journal of Nanjing University of Aeronautics & Astronautics, Vol. 38, No. 3, Jun. 2006
[18] A. Singh, M. Korupolu, D. Mohapatra, Server-storage virtualization: Integration and load balancing in data centers, Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, 2008, 1-12
[19] E. Arzuaga, D. R. Kaeli, Quantifying load imbalance on virtualized enterprise servers, Proceedings of WOSP/SIPEW'10, San Jose, California, USA, January 28-30, 2010
[20] W. Tian, C. Jing, J. Hu, Analysis of resource allocation and scheduling policies in cloud datacenter, Proceedings of the IEEE 3rd International Conference on Networks Security Wireless Communications and Trusted Computing, March 2011
[21] T.Thiruvenkadam and Dr.V.Karthikeyani, An approach to virtual machine placement problem in a datacenter environment based on overloaded resource, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.6, June- 2014, pg. 837-842.
[22] T.Thiruvenkadam and Dr.V.Karthikeyani, A Comparative study of VM Placement Algorithms in Cloud Computing Environment, In proccedings of 07th SARC-IRF International Conference,August 2014,India.
[23] Guo G., Ting-Iei H., Shuai G., " Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60-63, 2010.
[24] Wilcox, D.,McNabb, A., Seppi, K.: Solving virtual machine packing with a reordering grouping genetic algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 362–369 (2011).
[25] Tarun goyal & Aakanksha agrawal, "Host scheduling algorithm using genetic algorithm in cloud computing environment," International Journal of Research in Engineering & Technology (IJRET) Vol. 1, Issue 1, June 2013, 7-12.
[26] M. Srinivas, and L. M. Patnaik, Fellow, IEEE. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 4, April 1994.
[27] P. Rohlfshagen and J. Bullinaria. A Genetic Algorithm with Exon Shu_ing Crossover for Hard Bin Packing Problems. Proceedings of Genetic and Evolutionary Computation Conference, 9:1365{1371, 2007.
[28] Janani, N., Shiva, R.D., and Prakash, P.(2015) Optimization of Virtual Machine Placement in Cloud Environment Using Genetic Algorithm, Research Journal of Applied Sciences, Engineering and Technology, Vol. 10, No. 3, PP. 274-287
[29] T.Thiruvenkadam and Dr.V.Karthikeyani, Multi Dimensional Host Load Aware and User Constraints Based Algorithm for Scheduling Virtual Machines, International Journal of Advanced Computing Technology (IJACT) ISSN: Volume 7, Number 1, January 2015,pg 56 – 66.
[30] T.Thiruvenkadam and Dr.P.kamalakkannan, Energy Efficient Multi Dimensional Host Load Aware Algorithm for Virtual Machine Placement and Optimization in Cloud Environment" in Indian Journal of Science and Technology Vol 8(17), 59140, August 2015, ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645,page 1 – 11.
[31] Majid, Y., Farzad, D. and Farhad, R. (2013) Modification of the Ant Colony optimization for solving the multiple traveling salesman problem, Romanian Journal of Information Science and Technology, Vol. 16, No. 1, Pp. 65-80.
[32] T.Thiruvenkadam and Dr.P.kamalakkannan, Virtual Machine Placement using Enhanced Scheduling and Load Rebalancing using Hybrid Algorithms Based on Multi-Dimensional Resource Characteristics in Cloud Computing Systems, International Journal for Scientific Research & Development| Vol. 4, Issue 05, 2016 | ISSN (online): 2321-0613, pg 268 – 276