ISSN No. 0976-5697

Volume 7, No. 3, May-June 2016

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A STUDY ON LOAD BALANCING, SLA MANAGEMENT AND POWER OPTIMIZATION TECHNIQUES IN CLOUD COMPUTING

Venkatesh K nayakodi
Department of computer science and engineering
BMS College Of Engineering
Bangalore, India

Somshekhar Tubachi
Department of computer science and engineering
BMS College Of Engineering
Bangalore, India

Sudhanwa G S
Department of computer science and engineering
BMS College Of Engineering
Bangalore, India

SunilKumar K
Department of computer science and engineering
BMS College Of Engineering
Bangalore, India

Rajeshwari B S
Assistant Professor
Department of computer science and engineering
BMS College Of Engineering
Bangalore, India

Abstract:Cloud computing is a service provisioning technique where computing resources such as servers, storage devices, software's and complete platform for developing applications are provided as a service by the cloud providers. Customers can use these resources as and when needed, can increase or decrease resource capacities dynamically according to their requirements and pay only for how much the resource were used. Because of these reasons cloud computing is gaining popularity. Before the customer actually starts the service with the cloud providers, they make an agreement called Service Level Agreement (SLA). SLA is a contract made between providers and customers that include service quality, resources capability, scalability, obligations and consequences in case of violations. Managing the SLA, avoiding SLA violations and retaining the customers are very challenging for the providers. Another major issue in cloud computing is power consumption. Since in the cloud servers are running all the time, huge amount of heat is released from servers. This heat may cause system failure and also large amount of CO2 gases are released which leads to increase in carbon emission rate that is polluting our natural environment. Thus, reducing power consumption and providing a green computing is another challenging issue for the cloud providers. Thus distributing the load optimally that satisfies SLA as well as effective utilization of provider's resources and optimizes the power consumption in the Servers is very challenging issue for the cloud providers. In this paper, several load optimization techniques, SLA management techniques and power reduction techniques are discussed.

Keywords: Service level agreement, Cloud computing, Load balancing, Power optimization.

I. INTRODUCTION

Cloud technology is being used everywhere in today's world. The vendor strategically places his data centers at different location around the world to better serve the customers. Cloud computing comes in providing three categories of services to the customers [1].

- Software as a Service (SaaS) is a cloud service wherein cloud provider is providing software to the customer on subscription basis, eliminating need to buy, install, maintain, update on their local machine. E.g. Google Docs, Customer Relationship Management software, accounting software etc.
- Platform as a Service (PaaS is a cloud service in which the cloud provider provides a complete development platform for the customers to design, develop, debug, test and run an applications. E.g. Google App Engine.
- Infrastructure as a Service (IaaS)) is a cloud service where the cloud provider provides servers and storages devices to the customers on pay per use basis. E.g. Amazon Elastic Compute Cloud (EC2) and Go Grid.

SERVICE LEVEL AGREEMENT

Service Level Agreement (SLA) is an agreement between the customers and the cloud provider that includes both functional requirements like service quality, resource capability, and scalability and also non-functional requirements like security, privacy, trust etc. If the provider fails to meet these parameters, a SLA violation occurs. When SLA violations happen, provider has to pay penalty to the customers. SLA violations also degrades the reputation of the provider, may lose customers. Thus it is very important for the provider to make sure that SLA is never violated. SLA based scheduling includes scheduling task to the appropriate machine based on SLA, monitoring of resource usage, customer service execution status, periodically checking for SLA violation [2].

SLA mainly includes

i. Responsibilities of both providers and customers.

ii. List of services and its description that is being provided to the customer by the provider.

iii. Agreement on functional and nonfunctional requirements provided by the provider. Legal context that has been negotiated between the provider and customers.

II. SERVICE LEVEL AGREEMENT TECHNIQUES

Xiaomin Zhu et al [3] proposed "QoS-Aware Fault-Tolerant Scheduling Technique QAFT" for real-time systems. For real-time systems, the correctness of system depends not only on the logical correctness of solution but also on the time interval in which it is generated. Fault tolerant scheduling is mainly concerned with making the system more reliable, which is of great importance for mission critical tasks. The QAFT works by advancing the start time of primary copy of task and by delaying the start time of backup copy so as to reduce simultaneous execution time and help in better resource utilization. The Quality of Service (QoS) level of task is determined by its start time, execution time and deadline. The task is then scheduled on a processor which has idle time-slot and scheduling the task on that processor produces a finish time which is less than the deadline for the task. All the available processors are examined and the processor that provides the best QoS is selected and allocated to the task. This makes sure that QoS requirement of the task are satisfied making the system QoS-aware. The downside of the QAFT scheduling algorithm is it works effectively when one of the systems fails in real time tasks but not considered for multiple systems failure.

Rajeshwari B S et al [4] proposed architecture consisting of two scheduling algorithms at two stages that offer both balancing the load among the servers as well as providing a quality service to the customers. The paper comes up with the idea of grouping the servers into clusters based on processing power, thus presenting a clustered approach for priority based load optimization. Based on servers processing capability, High processing power servers' cluster, Medium processing power servers' cluster and low processing power servers' clusters are created. Two scheduling algorithm at two stages were used: i) SLA based scheduling algorithm ii) Idle server monitoring algorithm. At the first stage, when the task enters, SLA based scheduling algorithm based on task length, deadline to finish task and cost paid by the user computes the priority of the task and then assigns the task to the respective cluster. In the second stage, the idle-server monitoring algorithm running with in each cluster finds a free server in its cluster and assigns the task to the idle server. When the task receives, Idle server monitoring algorithm running within medium processing power servers cluster first finds whether any server is idle in high processing power servers cluster, if so assigns its task to identified idle high power server otherwise schedules the task within its cluster. Similarly When the task receives, Idle server monitoring algorithm running within low processing servers cluster first finds whether any server is idle in medium processing power servers cluster, if so assigns its task to identified idle medium power server otherwise schedules the task within its cluster. Author has proved that by scheduling medium priority tasks to high power servers if it is idle and by scheduling the low priority task to medium power servers if it is idle, high power servers and medium power servers can be utilized effectively and improves the response time. The downside of the proposed framework is that the presented model is not evaluated against SLA violations.

Ivona Brandic, Vincent C et al [5] proposed a novel approach of mapping low-level resources matrix to SLA parameters necessary for identification of failure resources and a layered architecture for bottom up propagation of failure to the layers, which react to sensed SLA violation threats for a self-manageable cloud. A self-manageable cloud is one in which the infrastructure automatically responds to the changing components, workload, and external environmental condition. The proposed approach LAYSI, Layered Approach for prevention of SLA-Violations in Self-manageable Cloud Infrastructures is one of the building blocks of FoSII (Foundations of Self-governing ICT Infrastructures) and helps in advance detection of SLA violations threats and propagation of threat to the appropriate layer of cloud infrastructure. The approach uses two important components for its working. First is the Knowledge database, which is used to provide reactive actions for the detected SLA violation threat based on Case Based Reasoning (CBR). CBR works by looking at similar cases in the past and reusing those solutions for current case. Second component is the SLA manager, who propagates the threat to the appropriate layer for preventive actions. The SLA manager has two main parts: **i)** Automatic manager which receives notification from the lower layers. **ii)** Notification Broker which provides interfaces to use the notification design pattern with services. The simulation study shows that architecture effectively prevents SLA violations by using layered cloud architecture.

Vincent C et al [6] proposed scheduling strategies by considering multiple SLA parameters and efficient allocation of resources. The proposed method schedules and deploys service requests by considering multiple SLA parameters such as amount of CPU required, network bandwidth, memory and storage by using heuristic approach. Scheduling heuristic has three layers: IaaS layer which manages the physical resources utilization, PaaS layer where the VMs are deployed and maintained so as to meet the SLA requirements of the customer and SaaS layer where the user applications are deployed. Scheduling heuristic aims to schedule applications on VMs based on the agreed SLA terms and deploys VMs on physical resources based on resource availabilities. With this strategy, application performance is optimized while the possibilities of SLA violations are reduced. The input for the algorithm includes the customers' service requests which include SLA

terms and application data. The SLA terms are extracted first. These SLA terms are then used to find a list of VMs which are capable of providing required services. The load-balancer then selects a particular VM to deploy the application to optimize the load in the datacenter. If no VMs are free or cannot satisfy the SLA terms, the scheduler checks global resources to see if new VMs with required resources can be hosted. If not then the scheduler queues the task. Author discussed that recent work considers various strategies with single SLA parameters. However, those approaches are limited to simple workflows and single task applications, but scheduling and deploying service requests by considering multiple SLA parameters are still open research challenges.

Asma Al Falasi et al [7] proposed a framework which overcomes cloud inherited issues and enables dynamic specification of SLA. Dynamic specification of SLA is facilitated by Declarative Genetic Algorithm and with effective mode of communication between the two parties. The architecture uses a third party broker who settles the SLA issues between the user and service provider. The user looks up the provider repository and specifies the QoS requirements, the service provider looks into his web services for service candidates that match the requirements of the user and submits the list to the broker who verifies and validates if the candidates satisfy the QoS requirements. The broker then communicates the new SLA to the user, who can either agree on the terms, or renegotiate or can choose another service provider. The web service verification conducted by verifier has two phases: the first phase generates test cases to verify test cases specified in WSDL(Web Service Description Language) and the second phase QoS verification test cases are generated using information depicted in the SLA. Only the services that pass the verifications are passed as candidates to the user. The architecture also has cloud directory and cloud broker. Cloud directory gives information about the service provider, SLA terms and the broker is concerned with verifying SLA.

Chandrashekhar S. Pawar et al [8] proposed a dynamic resource provisioning algorithm that uses SLA to dynamically allocate resources. Efficient provisioning consists of two steps: first is static planning step, where grouping the VMs and deploy them on physical machine and the second step is dynamic resource provisioning which includes additional resource allocations, creation and migration of VMs. The algorithm extensively uses the power of parallel processing. The algorithm considers multiple SLA parameters, uses resource allocation by pre-emption to improve resource utilization in cloud. The algorithm explains the use of parallel processing based on three issues. 1) How to allocate resources to tasks. 2) Tasks are executed in what order in cloud. 3) How to schedule overheads when VMs prepare, terminate or switch. In the algorithm initially static resource allocation is done, then two greedy algorithms are used namely, the cloud list scheduling (CLS) and the cloud min-min scheduling (CMMS). The CLS considers the earliest start time(EST) and latest start time (LST) and prepares a list of tasks based on priority. The resources are then allocated to the tasks in the order of formed list. CMMS uses best effort method to schedule the tasks. CMMS considers the inter-dependencies between the tasks before selecting any of them, thus overcoming the disadvantage of min-min scheduling algorithm.

Ahmed Amamou et al [9] proposed SLA based Dynamic Bandwidth Allocator (DBA) algorithm in virtualized environment (DBA). The DBA algorithm is associated with the job of allocating bandwidth dynamically to each virtual machine. In DBA algorithm, a virtual machine is instantiated based on the requirements specified in SLA and the remaining resources are shared between the concurrent VMs based on their priority. The main aim of the algorithm is not only to make sure that VM's get the required minimum bandwidth but also see to it that other machines do not have to compromise. Physical resources like CPU cycles and memory are given to virtual machines based on SLA. Virtual machines are always given optimum bandwidth such that QoS is never compromised. If the allotted bandwidth is greater than the max bandwidth for the VM, then it is readjusted so as to avoid violation of QoS of other machines. If the allocated bandwidth for the virtual machine is below the specified range, then algorithm finds if any bandwidth available. If so, bandwidth will be added or readjust the bandwidth of all VM and bandwidth will be added to VM, readjust to minimum range so as to avoid SLA violations.

Mehak Chaudhary et al [10] proposed an architecture for load balancing and minimizing the response time based on SLA. The architecture uses Join Idle Queue (JIQ), which is one of the most efficient algorithms for load balancing. Balancing the load is done in JIQ by allotting the jobs to the VM's such that waiting queue length should be minimum. The proposed methodology works as follows: The incoming task is assigned by the broker to one of the free VM in the VM list. The allocation is done such that the execution time and threshold is minimum. Threshold value depends on the number of tasks and number of VMs. The threshold value makes sure that minimum queue length is maintained among all VMs. After every assignment of task, the queue lengths are checked if they are less than threshold. If not the tasks are migrated to suitable VM.

Dr. H S Guruprasad et al [11] proposed an approach for SLA compliance implemented at client end. The process is composed of two phases. First is the information fetching phase, followed by evaluator phase. In the information fetch phase, a task is generated which consists of instructions that can be used to fetch relevant data from the cloud regarding SLA. SLA itself is used as input for this phase. In the evaluator phase, the algorithm calculates the percentage of SLA breach that has happened. The user uses the information fetch function to generate information fetch task which is passed to the cloud along with the tasks. Cloud executes the tasks and sends back the output to user. The user then using evaluator function can find out percentage of SLA violations. The two staged process makes SLA monitoring effectively.

etc.

III. LOAD BALANCING TECHNIQUES

Qiaomin Xie et al [12] proposed "Join-Idle-queue", a distributed load balancing algorithm. The author discussed that because of elasticity, horizontal scaling has gained huge importance but the existing algorithms like Join-Shortest-Queue (JSQ) incurs high communication overhead. The proposed distributed Join-Idle Queue (JIQ) algorithm does not incurs communication overhead. In the proposed algorithm, the

idle servers inform the dispatcher about their idleness without interfering with job arrivals. In distributed load balancing for large systems, algorithm informs dispatcher about the idle processors, but if large number of dispatchers informed then queuing plus one dispatcher will waste there cycle at idle processors which will affect the response time. To avoid this join idle queue uses two level load balancing. In first level average queue length should be minimized and in second level load is balanced at each dispatch on basis of idle processors. It helps in reducing load on systems and improves response time. I-queue data structure is used to communicate between primary and secondary load balancing. The problem with this arises when the algorithm is used with distributed system which has multiple dispatchers. The algorithm looks into this issue also by assigning the servers to dispatchers.

Mayank Mishra et al [13] discussed about the virtual machine related issues like flexible resource provisioning, virtual machine migrations etc. A live migration is a type of VM migration wherein state of a VM is migrated from one physical machine to another physical machine without interruption. The author explains that using dynamic resource management, the virtualization efficient resource management is a very critical component in cloud-based solutions to prevents over commitment and underutilization of resources to a particular VM. This is of great help for the cloud service provider because over commitment of resources leads to SLA violations and underutilization of resources leads to loss of revenue. The approach also has a VM monitor that manages physical resources. Sometimes necessary to migrate VM's to prevent server sprawl. The paper also explains three important issues. 1) When to migrate? 2) Which VMs to migrate? 3) The set of destination host machines for migration.

Argha Roy et al [14] discussed on fault tolerance in cloud computing. When many clients sends request to the servers simultaneously, server will be overloaded which causes fault. The load balancing technique is used to avoid fault in existing cloud computing. Self-healing, job migration, static load balancing and replication are various fault tolerance techniques. There are some drawbacks in this technique. The dynamic load balancing technique is used to avoid this fault tolerance. The proposed algorithm is dynamic in nature and considers only the current state of the system. The user sends the request to the intermediate node, which monitors the load on each VM and then assigns it to the VM with least load. The intermediate node acts as the load balancer. In this technique the status of server like CPU utilization and the other resources used are considered, the resources are allocated if required or the requests are sent to other server if it is overloaded thus load balancing is achieved.

Shu-Ching Wang et al [15] proposed a two stage scheduling algorithm under a three level cloud network. At the first level, root manager, at the second level set of service managers and at the third level set of service nodes under each service manager. The proposed framework uses two scheduling algorithms: OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms at two levels. The root manager receives the incoming task and assigns it into suitable service manager using OLB algorithm. The service manager divides the tasks into subtasks assigns them to the service nodes using LBMM algorithm. OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms are combined so that the load is balanced effectively. OLB algorithm distributes incoming tasks to service manager in arbitrary order without considering current load of service manager. Further service manager splits task into subtasks. LBBM scheduling algorithm computes execution time of each subtask on each service node and assigns the subtasks to the service node that takes minimum execution time. The proposed two phase scheduling algorithms result in better execution efficiency and maintain good load balancing of a system.

Shu-Ching et al [16] proposed a three phase scheduling for better utilization of cloud resources and make each task obtain the resources in shortest time. In the proposed architecture, first level has a request manager, the second level has a set of service managers and the lowest level has a set of service nodes under each service manager. The proposed model uses three scheduling algorithms at three different phases: 1) BTO (Best Task Order) 2) EOLB (Enhanced Opportunistic Load Balancing) 3) EMM (Enhanced Min-Min). At the first stage, BTO scheduling algorithm determines the execution order for each task request. At the second stage, EOLB scheduling algorithm assigns a task to the suitable service manager for allocation of the service nodes. The service manager divides the task into subtasks. At the third level, The EMM scheduling algorithm assigns a task to the suitable service node that executes the task in minimum execution time. The experimental result shows that by combining Enhanced opportunistic Load balancing with Enhanced Min-Min algorithm improves performance about 50% when compared with the combination of opportunistic load balancing with Min-Min algorithm and about 20% when compared with the combination of Enhanced opportunistic load balancing with Min-Min algorithm.

IV. POWER OPTIMIZATION TECHNIQUES

Poulami Dalapati1 et al [17] proposed "Green Scheduling" algorithm integrating a neural network predictor for optimizing server power consumption. The algorithm sends unused servers into sleeping mode. The proposed green scheduling algorithm uses Bee Colony Optimization algorithm and Ant Colony Optimization algorithm. Bee Colony Optimization detects underutilized hosts, then migrate all VM's which have been allocated to these hosts to the other hosts while keep them not overloaded and then switch hosts to sleep mode. Ant Colony optimization for power consumption management identifies the idle CPU's and turns them off. In the proposed scheduling algorithm, the client sends the job to the job scheduler, which checks the availability of resources through VM monitor. If sufficient amount of resources are available, the job is accepted. After the job is accepted, the VM assigner checks for any under loaded CPU of VM. If found, the task is assigned to that VM and database is updated to reflect the same. Else the task is put to waiting queue. Only the required number of VMs are active during any time and rest are put in sleep mode.

T.R.V. Anandharajan et al [18] proposed "**Co-operative Scheduled Energy Aware Load Balancing Technique**" for an efficient computational cloud. The author discussed in the paper that cloud computing is used more for the commercial purposes and in the scientific applications. Power consumption

and Load balancing are important problem in computational cloud. Computational cloud differs from traditional high-performance computing systems in the heterogeneity of the computing nodes as well as the communication links. The author argued that it is necessary to develop an algorithm that captures the complexity and to solve load balancing scenarios in a data and computing intensive applications. The Dynamic Scheduling using Boundary Value Approach algorithm for centralized and energy efficient fault-tolerant nature of the distributed environment like cloud. In the proposed load balancing algorithm, all the job information like job id, required free memory etc. is placed in job pool. All the resource information is placed in resource pool. Unique boundary value is set as threshold value for under loaded and over loaded resources. These threshold values are used when a particular task is being assigned to resource. Usually tasks are also moved from over loaded resources to under loaded resources for effective utilization of resources.

Rey M. Galloway et al [19] presents a "**Power Aware Load Balancing (PALB)**" algorithm that resolves most problems of Eucalyptus, a system by Eucalyptus system which addresses on cloud infrastructure aware of the wasted power consumed by the unutilized resources. Eucalyptus system consists of a front-end cloud controller, a cluster controller for controlling compute nodes, a virtual machine image repository, a persistent storage controller, and many compute nodes. The problem with eucalyptus is that it is not power aware and uses same load balancing technique for SaaS and IaaS. PALB clearly distinguishes SaaS and IaaS load balancing techniques. The load balancer maintains a set of free compute nodes and assigns tasks to them. Also it dynamically calculates the number of nodes that must be active, thus providing a power aware solution. PALB algorithm has three basic sections. **i)** Balancing section determines where the virtual machine should be instantiated; it gathers utilization percentage of each active node. If all compute nodes n are above 75% utilization, PALB instantiates new virtual machine on the compute node with lowest utilization number. Otherwise, the new VM is booted on the compute node with highest utilization. ii**)** Upscale section of algorithm is used to power on additional compute nodes. **iii)** Downscale section is responsible for power down idle computer nodes. The downside of the algorithm is that the solution suits only for the small and medium sized local clouds.

Nawfal A et al [20] proposed Datacenter Load and Power consumption (DLP) method. The author discussed that reducing power consumption is very important for cloud providers which reduces the operational costs and improve the system reliability. Mapping task to resources with minimum power consumption and better datacenter load management are very important. By assigning tasks to the server which takes minimum execution time, the power can be saved. Huge amount of power may be consumed at peak load or when tasks are not distributed efficiently across the servers. Algorithms are built for task scheduling which lowers power consumption and reduce load on data centers. The algorithm deals with execution of tasks within certain deadline. The algorithm goes through the input tasks and maps them to the hosts provided that tasks meet their deadlines. Else the task is rejected as its deadline cannot be met. Before assigning the task, the estimated response time for task on each VM is calculated and task is assigned to VM which provides the best estimated response time.

V. CONCLUSION

With the increasing popularity of cloud computing, the competition among the vendors is growing. Thus it becomes very important for the cloud providers to provide the required resources and handling requests in an efficient manner. A small delay may result in loss of the customers as well as affects cloud provider's market reputation. Proper load optimization helps in making sure that the resources are used efficiently and helps in avoiding overloading of servers and improves response time. The datacenters consumes huge amounts of power. Thus it is necessary in today's situation to make the datacenter as green computing. In this paper, various SLA based scheduling strategies, load balancing techniques and power optimization techniques proposed by different authors are discussed.

VI. ACKNOWLEDGMENTS

The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India

VII. REFERENCES

[1] Anthony T.Velte, Toby J.Velte, Robert Eisenpeter, "Cloud Computing: A Practical Approach", Tata McGraw-Hill Publishers, 1st Edition, 2009, ISBN: 0071626948.

[2] Rajeshwari B.S, M. Dakshayini, H.S. Guruprasad, "Service Level Agreement based Scheduling Techniques in Cloud: A Survey", International Journal of Computer Applications, Volume 132, No.5, pp: 20 26, December 2015, Digital Object Identifier: 10.5120/ijca2015907358.

[3] Xiaomin Zhu,Manhao Ma, "A QoS-Aware Fault-Tolerant Scheduling Algorithm for Real-Time Tasks in Heterogeneous Systems",IEEE Transactions on Computers July 2011.

[4] Rajeshwari B S, M Dakshayini, "Optimized Service Level Agreement Based Workload Balancing Strategy for Cloud Environment", Advance Computing Conference (IACC), 2015 IEEE International, PP: 160-165, 12-13 June 2015, Print ISBN: 978-1-4799-8046-8, DOI: 10.1109/IADCC. 2015.7154690.

[5] Ivona Brandic, Vincent C. Emeakaroha, Michael Maurer, Schahram Dustdar, Sandor Acs, Attila Kertesz, Gabor Kecskemeti, "LAYSI: A Layered Approach for SLA-Violation Propagation in Self-manageable Cloud Infrastructures", 34th Annual IEEE Computer Software and Applications Conference Workshops, Seoul, 19-23 July 2010, PP:365–370,ISBN:978-1-4244-8089-0,DOI:10.1109/ COMP SACW.2010.70.

[6] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Ivan Breskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual, Munich 18-22 July 2011, PP:298-303,PrintISBN:978-1-4577-0980-7, DOI:10.1109/ COMPSACW.2011.97.

[7] Asma Al Falasi, Mohamed Adel Serhani, "A Framework for SLA-Based Cloud Services, Verification and Composition", International Conference on Innovations in Information Technology, Abu Dhabi, 25-27 April 2011, PP: 287–292, ISBN: 978-1-4577-0311-9, DOI: 10.1109/ INNOVATIONS.2011.5893834.

[8] Pawar, C. S., & Wagh, R. B, "Priority Based Dynamic Resource Allocation in Cloud Computing", IEEE International Symposium on Cloud and Services Computing (ISCOS), pp: 1-6, December 2012.

[9] Ahmed Amamou, Manel Bourguiba, Kamel Haddadou ,Guy Pujolle, "A Dynamic Bandwidth Allocator for Virtual Machines in a Cloud Environment", 9th Annual IEEE Consumer Communications and Networking Conference- Multimedia & Entertainment Networking and Services, Las Vegas, 14-17 January, 2012, PP: 99 – 104,ISBN:978-1-4577-2070-3,DOI: 10.1109/CCNC.2012.6181065.

[10] A Review On Sla Aware Load Balancing Algorithm Using Join-Idle Queue In Cloud Computing. by Mehak Choudhary presented at International Journal of Advanced Research in Science and Engineering, Volume 4, Issue 8, ISSN: 2319-8354, pp: 174-179, August 2015.

[11] Suneel K S, H S Guruprasad, "A Novel Approach for SLA Compliance Monitoring In Cloud Computing", International Journal of Innovative Research in Advanced Engineering, Volume 2, Issue 2, February 2015, PP: 154-159, ISSN: 2349-2163.

[12] Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R Larus, Albert Greenberg, "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", Elsevier, Journal on Performance Evaluation, pp 1056-1071, Vol 68, Issue 11, Nov 2011, DOI: 10.1016/j.peva.2011.07. 015.

[13] Mayank Mishra, Anwesha Das, Purushottam Kulkarni, Anirudha Sahoo, "Dynamic Resource Management using Virtual Machine Migration", IEEE Communications Magazine, pp 34-40,Volume 50, Issue9,September2012,DOI: 10.1109/MCOM.2012.6295709.

[14] Argha Roy, Diptam Dutta, "Dynamic Load Balancing: Improve Efficiency in Cloud Computing", International Journal of Emerging Research in Management Technology, pp 78-82, Vol 2, Issue 4, ISSN:2278-9359, April 2013.

[15] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", 3rd IEEE Conference on Computer Science and Information Technology[ICCSIT], Taiwan, Volume 1, pp 108-113, 9-11 July 2010, DOI: 10.1109/ICCSIT. 2010.5563889.

[16] Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng, Wang, Ching-Wei, Chen, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network", Third International Conference on Communications and Mobile Computing [CMC], Taiwan, pp 114-117, 18-20 April 2011, DOI: 10.1109/CMC.2011.28.

[17] Poulami Dalapati, G.Sahoo, "Green Solution for Cloud Computing with Load Balancing and Power Consumption Management," International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3,Issue 3, pp. 353-359, March 2013.

[18] T.R. Anandharajan, V., and M. A. Bhagyaveni, "Co-operative Scheduled Energy Aware Load Balancing Technique for an Efficient Computational Cloud", International Journal of Computer Science Issues, Volume. 8, Issue 2, pp: 571-576, ISSN (Online): 1694-0814 March 2011.

[19] Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky, "Power Aware Load Balancing for Cloud Computing", Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I WCECS 2011, October 19-21, 2011, San Francisco, USA.

[20] N. A. Mehdi, A. Mamat, A. Amer and Z. T. Abdul-Mehdi, "Minimum Completion Time for Power-Aware Scheduling in Cloud Computing," Developments in E-systems Engineering, 2011, Dubai, 2011, pp. 484-489.DOI: 10.1109/DeSE.2011.30.

