International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 10, Issue 1, January 2021)

Load Balancing in Cloud Computing by Efficient Utilization of Virtual Machines

Shruti Tripathi[1], Prof. Pradeep Tripathi[2]

[1,2]*Department of Computer Science & Engineering, Vindhya Institute of Technology and Science - [VITS, Satna], India*

***Abstract*–Load unbalancing problem is a multi-variant, multi-constraint problem that degrades performance and efficiency of computing resources. Load balancing techniques cater the solution for load unbalancing situation for two undesirable facets- overloading and under-loading. In contempt of the importance of load balancing techniques to the best of our knowledge, there is no comprehensive, extensive, systematic and hierarchical classification about the existing load balancing techniques. Further, the factors that cause load unbalancing problem are neither studied nor considered in the literature. This paper presents a detailed encyclopedic review about the load balancing techniques. The advantages and limitations of existing methods are highlighted with crucial challenges being addressed so as to develop efficient load balancing algorithms in future. The paper also suggests new insights towards load balancing in cloud computing.**

Index Terms--CloudSim, cloud computing, load balancing, Virtual Machine (VM), refresh period.

I. INTRODUCTION

Cloud computing is a technology based on distributed computing which helps users to utilize computing resources online. Cloud provides services in pay as you use fashion. As cloud based services are utilized by users across the globe, it is essential to have mechanisms to optimize resource utilization through balancing load. This is achieved using a 'Load Balancer'. Cloud servers experience sudden bursts in requests. In order to process such requests, it is important to have dynamic resource allocation and also load balancing provisions. When more resources are allocated, it results in wastage of resources.

When less resources are allocated, it causes server quality problems or inability to serve client requests. Therefore a robust load balancing mechanism is required that helps in optimization of resource utilization. The existing algorithm proposed by Xu et al. [26]explored load balancing problem in cloud computing. They employed a methodology that makes use of cloud partitions and the status information such as IDLE, NORMAL AND OVERLOADED in order to make load balancing decisions at server side.

Requests are processed by the server based on the balance of load on the servers. It does mean that requests that come from user are sent to different partitions based on the load degree of the servers involved in the partitions.

There is no provision for cloud partition rules.

Cloud servers with associated load balancer can help in optimizing the load balancing process in cloud computing.

The refresh period is an issue. Further investigation is required in order to find best refresh period to update the load status. No evolution strategy for load balancing.

We implemented a load balancing model where each partition is aware of load balancing model. Based on the status information the load balancing decisions are made. The status information provides needed load degree which is crucial to make well informed decisions. Among different partitions and servers, load balancer programs will help in bringing about coordination for making load balancing decisions dynamically at run time. The servers which are less loaded are given more requests while the servers which are overloaded are not given further requests. These decisions are implicitly made by load balancers as they do have intelligence and statistics used to maintain load in optimal fashion. Grid computing is predecessor of cloud computing [1], [2]. Quality of Service (QoS) is one of the requirements that are associated with load balancing in cloud computing [3].

With load balancing in place, the servers in the cloud can perform better thus improving availability of cloud and scalability. Scalability refers to the ability to serve increased number of clients without degrading performance. Availability refers to the measure that indicates the availability of server in a given year. These metrics also help in assessing overall performance of the cloud. The public cloud considered in this project is capable of improving performance in terms of latency, CPU utilization and speedup. Technologies play vital role in the cloud computing paradigm in future [4]. Resource provisioning can be improved in cloud using certain benchmark practices as explored in [5]. Cloud resources are very useful in applications related High Performance Computing (HPC).

In the existing solution for cloud load balancing proposed in [26], there is no provision for cloud partition rules. Cloud partition rules can help in optimizing the load balancing process in cloud computing. The refresh period is an issue. Further investigation is required in order to find best refresh period to update the load status.

There is no evolutionstrategy for load balancing. Therefore, a new algorithm for cloud partitioning is required in order to optimize the performance of cloud.Refresh period can serve the cloud as catalyst for determining ideal state of the partitions and make well informed decisions pertaining to load balancing. This paper takes care of optimization of load balancing and user job scheduling. The remainder of the paper is organized into different sections. The second section provides review of literature on load balancing issues in cloud computing. It throws light into different methods for load balancing. Section 3 provides problem for providing details of the proposed load balancing mechanism. Section 5 provides the details of our implementation. Section 6 shows results of experiments made. Section 7 provides conclusions on the proposed load balancing approach. It also includes the possible future enhancements to our work.

II. Related Works

This section provides review of literature on load balancing mechanisms used for public cloud. Due to the huge number of requests web applications generate high workloads that needs the services from cloud computing. Therefore load balancing is required in order to have better performance of cloud [6]. Cloud computing resources are being shared to users across the globe. The VMs used in the cloud computing are to be managed. Best Fit scheduling is one of the scheduling algorithms that can be used to optimize resource allocation and improve overall performance of the cloud [7]. Since scheduling and resource allocation are NP-hard, they need load balancing mechanism for optimal machine utilization. Honey Bee Behaviour Inspired Load Balancing (HBB-LB) can be used to achieve load balancing across VMs being used in cloud infrastructure. Based on priorities the system can schedule tasks and perform better resource utilization [8].

Conventional computing is being replaced by cloud computing which needs resource utilization in best way in order to provide services to users at affordable prices [9]. There is need for associating cloud with ICT and ensure that the resource utilization is done ideally. Energy consumption is also given importance in the research circles. Cloud computing energy consumption needs to be optimized for efficiency. With load balancing it is possible to avoid certain unnecessary things so as to improve the performance of cloud computing [10]. When requests arrive to cloud infrastructure, the selection of node which can serve the request needs to be determined. This task is left to load balancing mechanism that supports large-scale request processing at any given time besides exploiting effectiveness in resource allocation.

CloudSim based approaches can be used to demonstrate the proof of concept pertaining to load balancing [11].

As the cloud computing requests come in bursts it is essential to have robust mechanism to handle such situations in cloud computing. Here comes the need for load balancing to improve robustness. Service request routing problem is one of the mechanisms that is grid-aware which can improve load balancing efficiency and can withstand different load variations [12]. Utility oriented services are provided by cloud computing in pay per use model. It contains huge computing resources that are very valuable. They are to be utilized optimally. Towards this Green Cloud Computing solution came into existence. This solution operates with minimum cost and it is energy efficient. In tandem with it load balancing can optimize the performance of resource allocation in public cloud [13]. All kinds of businesses started using cloud computing. E-learning domain is not exception [14]. As cloud allows business to scale up to requirements of their clients, the resource utilization I the cloud needs to be optimized. Skewness measure can be used in order to optimize resources. Skewness provides the usage balancing approach [15].

Cloud data centers are being used increasingly for storage. However, the storage infrastructure with multiple layers can be optimally utilized for best results. Energy aware cloud computing simulations with provision for load balancing in the presence of cloud data centre with multiple tiers can help in optimizing resource allocation. This research can help in finding dynamics of resource allocation and take steps to realize it in the real world [16]. Certain sensor networks can also be used to connect to cloud computing infrastructure. Such integrated networks consume more resources. Therefore there should mechanism for balancing load. Environmental end to end monitoring is designed and implemented to ensure best results [17]. There are many cloud related products for infrastructure such as Cloud SQL Server, Microsoft SQl Server, SQL Azure and so on [18]. The under-utilized resources can be identified and load can be balanced in a heuristic approach. The under-utilized resources can be avoided and thus energy efficiency also can be improved besides achieving load balancing in optimal way [19]. However, there are many challenges in resource provisioning and load balancing [20]. Cloud computing has its drawbacks. For instance security issues threaten users and discourage them to use cloud in the real world. Cloud data security needs to be given paramount importance besides helping the optimal load balancing which improves the efficiency of cloud computing [21]. Distributed data management with smart grid is also used to have energy efficient solution for real time applications in cloud computing.

Such applications produce huge number of requests that are to be processed in a scalable fashion. Towards this end cloud data centers needs to optimize resource allocation in energy efficient fashion as explored in [22]. Elasticity needs to be improved in cloud computing. This feature is somehow associated with flexibility in resource allocation and the balance of load as well in order to optimize the performance of cloud infrastructure. Though there are certain challenges and issues with this mode, the bottom line is that resource allocation needs to be made in optimal fashion so as to improve the performance of cloud computing [23]. Not only load balancing but also scalability is an essential feature of cloud computing. Scalability is not possible unless resources are optimally utilized. The usage of data centers in pay per use fashion by users across the world causes much load on data center that needs to be distributed properly. Therefore load balancing ad load rebalancing algorithms were proposed and implemented. This has proved the cloud to have characteristics like high availability, scalability and cost-effective [24]. Local infrastructure integration with global cloud infrastructure can also be used as a strategy for load balancing. The best utilization of resources can help cloud to offer services in affordable way besides ensuring that the cloud data centers are optimized [25].Different load balancing algorithms for cloud computing are found in [27]-[31]. However, they need improvements. In this paper, weproposed a methodology for optimized load balancing and job scheduling for better performance. We considered refresh period and status of load in different partitions in our implementation.

III. Problem Definition

The load balancing algorithm proposed by Gaohao Xu [1] helps in balancing load in cloud computing due to its partitioning approach with status information like NORMAL, OVERLOADED and IDLE. The states helped them to make good decisions on the incoming requests. The requests are sent to servers based on the load particulars. IDLE state refers to the state in which a server is free which can take requests. NORMAL refers to the state in which a server is having NORMAL load. It may be able to take some more jobs. The OVERLOADED state indicates that the server is no longer able to take new requests unless existing requests are completed. The load status maintained by the application is crucial for making load balancing decisions. The existing load balancing system proposed in [26] as certain drawbacks. They are as follows.

a. There is no provision for cloud partition rules. Cloud partition rules can help in optimizing the load balancing process in cloud computing.

b. The refresh period is an issue. Further investigation is required in order to find best refresh period to update the load status.

c. No evolution strategy for load balancing.

IV. Proposed Load Balancing Technique

In order to overcome the problems of existing load balancing mechanism described in the preceding section, we designed and implemented a load balancing algorithm which on top of cloud partitioning. The proposed algorithm takes care of the limitations of the existing one. The partitions in cloud can have multiple virtual machines running over physical machines. A load balancer is associated with each partition. The state information is maintained by each partition. The state information is used by the algorithm. A controller program is used to coordinate all partitions. There is coordination between load balancing programs and the controller for making good decisions. The proposed system is graphically illustrated in Fig. 1.

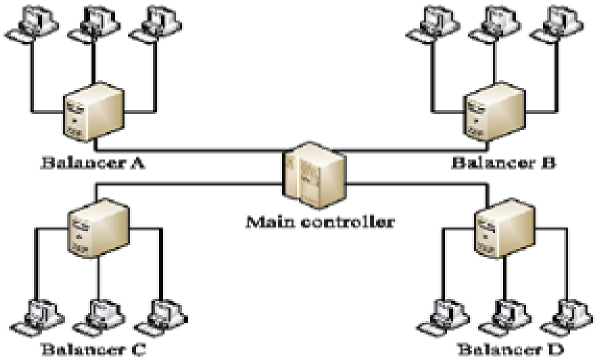

Fig. 1.Overview of partitioning based load balancing

As shown in Figure 1, it is evident that the controller has interaction with different load balancer programs.

The load balancer programs associated with each partition can updatethe load of the partition in the form of status messages aforementioned. Though status information is useful for improving load balancing, the proposed system considers additional features. Towards this end, two algorithms are proposed and implemented. More details are provided in the following sub sections.

A. *Partition Based Load Balancing Algorithm*

This algorithm is meant for load balancing in cloud. It takes number of partitions, number of virtual machines and number of jobs. The algorithm searches for best partition in which jobs are allocated to virtual machines. The main controller takes care of allocation of jobs as per the procedure provided in the algorithm.

```
Algorithm: Cloud Partition Based Load Balancing
Input: Set of jobs used to simulate the load balancing mechanism
Output: Load balanced across partitions and VMs
Create partitions
Create VMs in each partition
Assign jobs to the controller program
For each job in given jobs
  Find best partition based on job details
  IF state=IDLE or state=NORMAL THEN
    Send job to the partition
    Job processing
  ELSE
    Find another partition
  END IF
End For
```

Algorithm 1. Partition based load balancing algorithm

The algorithm has an iterative process that searchers for right partition and checks the state of the partition. The possible partitions include NORMAL, IDLE and OVERLOADED. When the partition is already overloaded, other partitions are preferred for job allocation. There is another mechanism known as refresh period which is not considered by this algorithm. When optimal refresh period is computed, the proposed methodology is improved further to achieve optimal load balancing and job scheduling. The following sub section throws light on this.

B. *Determination of Refresh Period*

Determination of refresh period is very important for optimisation of load balancing and job scheduling. The rationale behind this is that the controller gets updated states in specific time interval. This time interval may be too big to have something gone unnoticed and true status is not reflected. Therefore finding ideal refresh period is very challenging in the dynamic cloud environment. The following procedure can help in finding ideal refresh period.

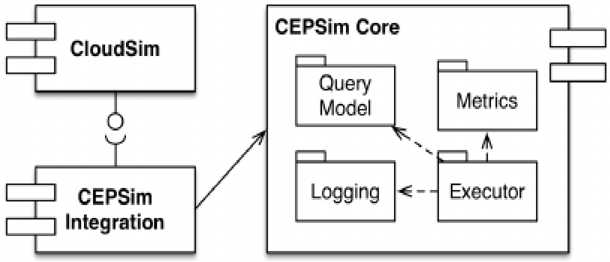

Fig. 2. Hierarchy of CloudSim classes

High frequency with less refresh period can reduce performance of the system. At the same time, if the refresh period is very high, it does not reflect true status of partitions in the cloud. Therefore there should be an ideal time period for refresh to strike balance between the two ends.

V. Implementation

A. *CloudSim*

Experiments are made with CloudSim tool kit as it is meant for simulating cloud computing solutions. It provides API for modeling cloud and having interactions among different components. It supports behavioral modeling and system modeling. It has API for creating data centers, cloud brokers, virtual machines, and resource provisioning related things. As CloudSim enables us to have different resource provisioning and scheduling approaches, it is useful to have experiments with it. The default mechanisms of CloudSim can be altered by writing our own algorithms. That is the rationale behind the usage of CloudSim in this paper. Moreover to have proof of concept, CloudSim is very feasible prior to implementing in the real world.

In addition to this, CloudSim provides benefits with respect to having interfaces that can have different implementations. Therefore every aspect of cloud computing mechanisms can be improved further. Researchers across the globe including HP Labs in the United States explored CloudSim for experimental study. The investigations in cloud computing can be simulated using the simulation framework.

B. *Class Hierarchy of CloudSim*

CloudSim has a hierarchy of classes that are used to simulate the proposed load balancing mechanisms to demonstrate proof of the concept.

As can be seen in Figure 2, CloudSim is the top level interface that is used as basis for simulation functionalities. Different classes are used in the implementation for creating data centers, partitions, resource provisioning, virtual machines, job allocation, and so on.

C. *Prototype with Visualization of Optimized Load Balancing and Job Scheduling*

The prototype application is used to simulate creation of cloud partitions, allocation of virtual machines and allocation of jobs. Then the simulation moves to allocation of jobs to different partitions based on the proposed algorithms in this paper. The results are presented in Figure 3 and Figure 4.

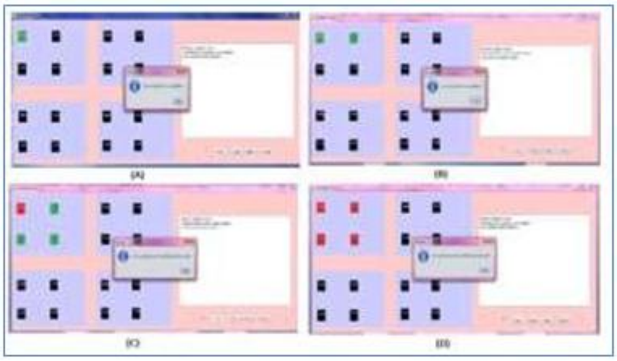

Fig. 3. states of partitions and controller is considering optimal load balancing

As shown in Figure 3, it is evident that there algorithms started working according to the states of partitions and controller is considering optimal load balancing and job scheduling. The results also show the visualization of VMs with different colours. Greencolour shows VM can take jobs and red indicates that VM cannot take jobs anymore.

Fig. 4. optimal load balancing

As shown in Figure 4, it is evident that the algorithms proved to be useful and effective as there is provision for optimal load balancing and job scheduling. This is reflected in the patterns in which jobs are allocations in (a), (b), (c), and (d) of Figure 3 and (e), (f), (g) and (h) of Figure 4.

VI. Experimental Results

Observations are made when simulations are carried out in terms of execution time and CPU cycles consumed. These observations are made with different virtual machines are in place in the cloud partitions.

TABLE 1.
Performance Comparison In Terms Of Execution Time

Number of Virtual Machines	Execution Time Comparison (Milliseconds)	
	Proposed	*Existing*
14	520	500
12	720	700
10	1020	1000
8	1520	1500
6	1820	1800
4	1920	1900
2	2020	2000

As shown in Table 1, the results of experiments in terms of execution time for given jobs in the presence of different number of virtual machines are presented.

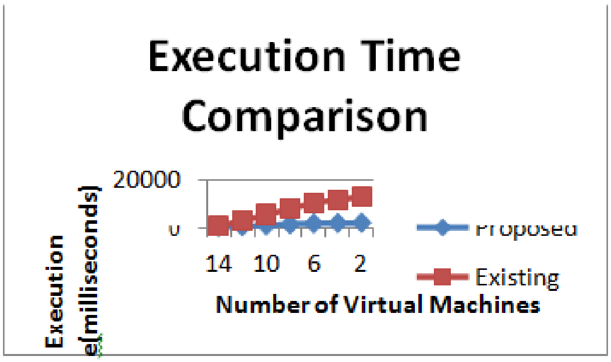

Fig. 5. Performance comparison in terms of execution time

As shown in Figure 5, the values taken under horizontal axis are number of virtual machines while the execution time is presented in vertical axis. The results revealed that the proposed system shows better performance with respect to execution time. As the virtual machines are increased, the execution time taken is decreased.

TABLE 2:
Performance Comparison In Terms Of Cpu Cycles

Number of Virtual Machines	CPU Cycles Comparison (Milliseconds)	
	Proposed	*Existing*
1	2	3
2	4	6
3	5	8
4	7	10
5	9	13
6	10	16
7	12	19

As shown in Table 1, the results of experiments in terms of execution time for given jobs in the presence of different number of virtual machines are presented.

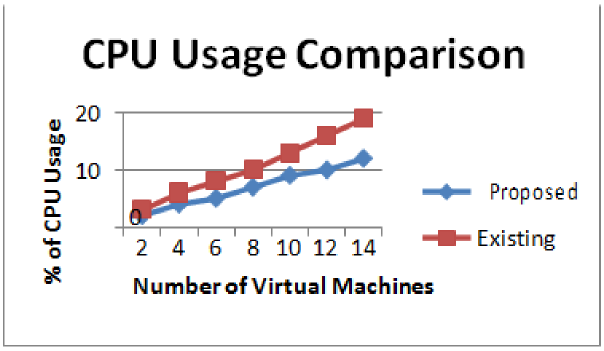

Fig. 6. Performance comparison in terms of CPU cycles

As shown in Figure 6, the values taken under horizontal axis are number of virtual machines while the CPU usage percentage is presented in vertical axis. The results revealed that the proposed system shows better performance with respect to consuming CPU cycles. As the virtual machines are increased, the CPU cycles consumption is increased.

VII. Conclusions And Future Work

In this paper, a new load balancing approach is proposed and implemented based on cloud partitioning. Cloud partitioning rules are used and the refresh period is optimized for perfect load balancing and resource optimization.The implementation is made using CloudSim which is one of the simulators to demonstrate the proof of concept of cloud based mechanisms.The results reveal that the proposed solution is better than the existing load balancing mechanism. We built a prototype application to demonstrate proof of the concept. However, as CloudSim does not provide visual representation of cloud activities, we considered providing graphical user interface to visualize simulations. These visualizations are implemented using Swing API using Java programming language. The results revealed that the proposed system is capable of optimizing load balancing and job scheduling in cloud computing. The drawback in the proposed system is that it does not consider formally defined partitioning rules. In future, we intend to improve the algorithms further with more partitioning rules and provision for fine grained pricing to bring about equilibrium of benefits for cloud service providers and consumers.

REFERENCES

[1] Chris Develder,. (2012). Optical Networks for Grid and Cloud Computing Applications. IEEE.p.25-34.

[2] Dr. Christof Weinhardt. (2009). Cloud Computing – A Classification, Business Models, and Research Directions. Business & Information Systems Engin p.25-34

[3] ChrysaPapagianni, ArisLeivadeas, SymeonPapavassiliou, Vasilis Maglaris, Cristina Cervello´ -Pastor, and Alvaro Monje. (2013). On the Optimal Allocation of Virtual Resources in Cloud Computing Networks.IEEE. 62 (6), p.25-34.

[4] Sean Marston a, Zhi Li a, SubhajyotiBandyopadhyay a, Juheng Zhang a, AnandGhalsasi. (2011). Cloud computing — The business perspective. ELsevier. 51 (1), p.176–189.

[5] Zhenhuan Gong, XiaohuiGu. (2010). PAC: Pattern- driven Application Consolidation for Efficient Cloud Computing. IEEE. p.213-313.

[6] Javier Espadas a, Arturo Molina b, Guillermo Jiménez a, Martín Molina b, RaúlRamírez a, David Conchaa. (2013). A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures. ELsevier. 29 (1), p.273–286.

[7] Philip A. Bernstein, IstvanCseri, Nishant Dani, Nigel Ellis, Ajay Kalhan, Gopal Kakivaya,. (2011). Adapting Microsoft SQL Server for Cloud Computing. IEEE. p.23-33.

[8] Young Choon Lee • Albert Y. Zomaya. (2012). Energy efficient utilization of resources in cloud computing systems. Springer Science+Business Media. p.56-60.

[9] Johan Tordsson a, Rubén S. Monterob, Rafael Moreno-Vozmedianob, Ignacio M. Llorente. (2012). Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. ELsevier. 28 (1), p.358–367.

[10] Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S. Tucker. (2011). Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport. IEEE.p.213-313.

[11] BrototiMondala, KousikDasguptaa, ParamarthaDuttab. (2012). Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach. ELsevier. 4 (1), p.783 – 789.

[12] Amir-HamedMohsenian-Rad and Alberto Leon-Garcia. (2010). Coordination of Cloud Computing and Smart Power Grids. IEEE.p.12-17.

[13] Anton Beloglazov a, JemalAbawajyb, RajkumarBuyyaa. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. ELsevier. 28 (1), p.755–768.

[14] Bo Dong, Qinghua Zheng Mu Qiao Jian Shu and Jie Yang. (2009). BlueSky Cloud Framework: An E- Learning Framework Embracing Cloud Computing. Springer-Verlag Berlin Heidelberg. p.577–582.

[15] Zhen Xiao,Weijia Song, and Qi Chen. (2013). Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment.IEEE. p.12-17.

[16] DzmitryKliazovich • Pascal Bouvry. (2012). GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. Springer-Verlag Berlin Heidelberg.p.23-33.

[17] Kevin Lee1, David Murray, DannyHughes, WouterJoosen. (2010). Extending Sensor Networks into the Cloud using Amazon Web Services.IEEE.p.25-34.

[18] Philip A. Bernstein, IstvanCseri, Nishant Dani, Nigel Ellis, Ajay Kalhan, Gopal Kakivaya. (2011). Adapting Microsoft SQL Server for Cloud Computing. IEEE. p.25-34.

[19] Young Choon Lee • Albert Y. Zomaya. (2012). Energy efficient utilization of resources in cloud computing systems. Springer Science+Business Media, LLC 2010. 60 (1), p.268–280.

[20] Qi Zhang • Lu Cheng • RaoufBoutaba. (2010). Cloud computing: state-of-the-art and research challenges. Springer. p.7–18.

[21] Flavio Lombardi a, RobertoDiPietro. (2011). Secure virtualizationforcloudcomputing. ELsevier. 34 (1), p.1113–1122.

[22] SebnemRusitschka, Kolja Eger, Christoph Gerdes. (2010). Smart Grid Data Cloud: A Model for Utilizing Cloud Computing in the Smart Grid Domain. IEEE. p.32-44.

[23] GuilhermeGalante and Luis Carlos E. de Bona. (2012). A Survey on Cloud Computing Elasticity. IEEE.p.32-44.

[24] BorkoFurht. (2010). Cloud Computing Fundamentals. Springer Science+Business Media, LLC 2010. p.23-33.

[25] Marcos Dias de Assunção and Alexandre di Costanzo. (2010). A cost-benefit analysis of using cloud computing to extend the capacity of clusters. Springer. 13 (1), p.335–347.

[26] Gaochao Xu, Junjie Pang, and Xiaodong Fu. (2013). A Load Balancing Model Based on Cloud Partitioning for the Public Cloud. TSINGHUA SCIENCE AND TECHNOLOGY. 18 (1), p1-6.

[27] QiangLiu ,Yujun Ma ,MusaedAlhussein ,Yin Zhang and Limei Peng. (2016). Green data center with IoT sensing and cloud-assisted smart temperature control system. Computer Networks. 10 , p104-112.

[28] Michael T. Kriegera, Oscar Torreno, Oswaldo Trelles and Dieter Kranzlm uller. (2016). Building an open source cloud environment with auto-scaling resources for executing bioinformatics and biomedical workflows. Preprint submitted to Future Generation Computer Systems. p1-27.

[29] FatemehJalali,KerryHinton,RobertAyre, TansuAlpcan and Rodney S. Tucker . (2016). Fog Computing May Help to Save Energy in Cloud Computing. journal on selected areas in communications(ieee). 34 (5), p1728-1739.

[30] Ayman E. Khedr and Amira M. Idrees. (2017). Adapting Load Balancing Techniques for Improving the Performance of e-Learning Educational Process. Journal of Computers. 12 (3), p250-256.

[31] ChenhaoQu , Rodrigo N. Calheiros and RajkumarBuyya. (2016). Mitigating Impact of Short-term Overload on Multi-Cloud Web Applications through Geographical Load Balancing. concurrency and computation: practice and experience. p1-22.