Middle-East Journal of Scientific Research 24 (9): 2866-2872, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.09.23831

LBRC: Framework for Load Balancing Based on Resource Control in Cloud Environment

[1]Suriya Begum and [2]C.S.R. Prashanth

[1]Senior Assistant Professor, Department of CSE, NHCE, Bangalore,
Visvesvaraya Technological University, Belgaum, India
[2]HOD and Dean of Academics, Department of CSE, NHCE, Bangalore,
Visvesvaraya Technological University, Belgaum, India

Abstract: Cloud computing offers a robust backbone for live and dynamic networks with a large range of its services. However, it is quite a challenging task to determine the uncertain and dynamic behavior of online users. Hence, various virtual machine and processing element have a higher dependency of a large range of resources to process the incoming jobs to maintain better queue management or zero downtime. However, this is far from reality where occurrences of downtime due to traffic load are quite witnessed by various applications running on cloud unnoticed. We review existing mechanism and techniques towards addressing the load balancing problems to find that still there is a large scope of enhancement. Hence, we proposed a simple framework called as LBRC or Load Balancing using Resource Control, where we emphasize more on cluster's resource management to balance the load on VMs and processing elements. The technique introduced three operational states of the cluster and presented its scheme of traffic management based on it. Compared to the existing system, the outcome of LBRC is found to posses better on minimal resource consumption, the better capability of processing jobs and reduced CPU utilization.

Key words: Cloud Computing · Task Scheduling · Resource Availability · Load Balancing

INTRODUCTION

With the increasing number of users over mobile networks, there has also been an exponential growth of mobile application. Such applications give rise of a very dynamic shape of traffic which is sustained by cloud environment to maintain pervasiveness [1, 2]. Cloud computing is all about storing and entrance data and programs on the internet as an alternative of your computer's HDD. Cloud is presently a symbol for the internet and also enables delivery of host service over the internet. It enables companies to consume compute resource as utilities rather than having to build and maintain computing infrastructure [3]. Although, cloud also offers a better traffic management for its existing customers, but there are certain uncertainties which cannot be handled by the cloud e.g. sudden rise of online users, no standard stereotyped schedule for peak time and off time. Due to such issues, there are many cases of downtime too resulting in a violation of SLA [4]. Hence, load balancing is highly essential to managing and controls such massively growing traffic over the cloud [5]. In a very general way load balancing approaches divides the amount of work that a cluster has to do between two or more clusters so that more work get done in the same amount of time. It is also about distributing workload and computing resource in a most dynamic environment allowing an organization to manage application by allocating resource among multiple clusters, network or server [6]. It also involves hosting the distribution of workload traffic and demands that reside over the internet and thereby helps the organization to achieve a high-performance level for the potentially lower cost. However, there is no denying the fact that existing load balancing techniques in the cloud are not sufficient enough to cater up the dynamic demand of the uncertain

Corresponding Author: Suriya Begum, Senior Assistant Professor, Department of CSE, NHCE, Bangalore, Visvesvaraya Technological University, Belgaum, India.

traffic from various parts of the world. Therefore, the proposed study reviews the existing load balancing techniques identify problems and provides a simple and cost-effective solution just by resource availability. The organization of the proposed paper is as follows: - the second section discusses the related work carried out in past addressing the problem of load balancing over cloud followed by brief discussion of problem identification in the third section. The fourth section discusses the proposed methodology followed by algorithm implementation in the fifth section. The result discussion is carried out in the sixth section followed by conclusion in the seventh section.

Literature Review: This section discusses the recent studies being carried out to address the problem of load balancing in the cloud. Ningning *et al*. [7] discussed the atomization cloud technology and fog computing to make physical nodes in different level into virtual machine nodes. This designed system provides system network flexible and dynamic load balancing mechanism can form effectively system resource as well as it minimizes the consumption of node migration. Cao *et al*. [8] discussed the develop power and performance constrained load distribution technique for cloud computing in the present and feature large scale data center. This method designed to provide performance optimization and power management. Assi *et al*. [9] discussed the decomposition approach to overcome from VLAN mapping problem in cloud data center through column generation. This method designed in such a way that it uses both an exact and semi-heuristic decomposition with the objective to achieve load balancing by minimizing the maximum link load in the network. Lin *et al*. [10] discussed the more practical dynamic multi-service scenario in which server cluster only handle a specific type of multi-media task and client request different types of multi-media service at a different interval of time. This method designed it effectively supports genetic algorithm can efficiently cope with dynamic multi-service load balancing in CMS. Rao *et al*.[11] discussed the problem of the electricity cost management for internet service with a collection of spatially distributed data center. This method designed effectively to minimize the total electricity cost geared to QOS constraint as well as the location diversity and time diversity of electricity price under MEM. Mishra *et al*. [12] discussed detailed overview of virtual machine migration method and their usage toward dynamic resource management in a virtualization environment. This method designed effectively to reduce server sprawl, minimizing power consumption, load balancing across the physical machine. Simeone *et al*. [13] discussed operator to minimize and operating expense needed to deploy and maintain the dense heterogeneous network. This method designed effectively gives spectral efficiency, statistical multiplexing and load balancing. Deng *et al*. [14] discussed taxonomy of the state of the art research in applying renewable energy in cloud computing data center. This new research challenges involved in managing the use of renewable energy in data centers. Xu *et al*. [15] discussed better load balance model for the public cloud based on the cloud partitioning concept using the game. Liang *et al*. [16] discussed service decision-making system for inter-domain service transfer to balance the load among multiple domains. This designed system significantly improves the system reward and decreases service disruption. Luo *et al*. [17] discussed how to leverage both geographical and temporal variable of energy price to reduce energy cost for distributed IDCs. Gao *et al*. [18] discussed a game theoretical perspective and examined how it affects the behavior of mobile cloud application. This designed facilities further research in the design of the offload decision engine of mobile cloud application and load balancing in game theory.

Problem Identification: This section discusses the problems that are identified after reviewing the work carried out by the researcher in the prior section. Following are the problems identification:

More focus on Virtual Machine (VM) and less focus on Cluster: It has been seen that majority of the existing techniques emphasizes on a virtual machine which has defined resources to perform a huge list of task. Hence, incorporating execution of a load balancing algorithm will make such VM always busy at the cost of heavy resource utilization. Out of all the resource, power consumption will also increase.

Few Works Integrating Resource with Traffic Management: All the existing techniques have significant problems of automated provisioning of service on identification of appropriate resource and its allocation policies. The existing policies of allocation support less dynamicity resulting in congestion or downtime.

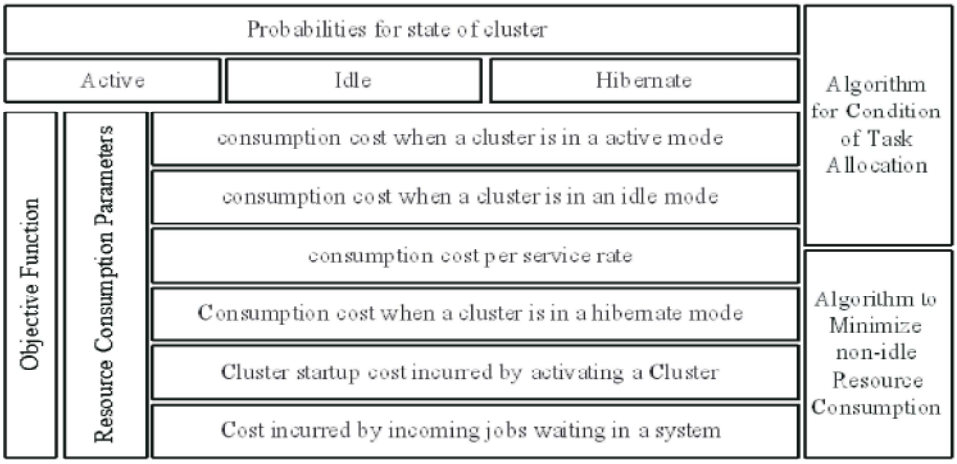

Fig. 1: Proposed Architecture of LBRC

Few Approaches to Consider Cluster State: A cluster resides in datacenters and has good availability of resources. It also has a defined states of its operation like active, idle, hibernate state to properly utilize the resources. However, few studies are found to formulate conditions of load balancing just by present states of operation of the cluster (on the contrary, the majority of the focus was laid on VM).

Proposed Methodology: The proposed study is a continuation of our past research work (Begum, [19-21). Using analytical modeling, the proposed system presents a framework that mainly focuses on proper utilization of resources to ensure better load availability. The study considers three different modes of cluster operation e.g. active state, an idle state and hibernating state to perform switching of the different clusters among the three defined states of operation. The prime motive of the proposed framework is to ensure better control strategies for unnecessary resource utilization for efficient load balancing. The study mainly uses probability theory to denote the states of clusters and consider multiple parameters responsible for resource consumption over the cloud. Fig. 1 highlights the architecture of the proposed system.

Algorithm Implementation: The algorithm design of the proposed system is carried out by considering mainly two different resource entities i.e. cluster (C) and its respective state (Cst). The algorithm considers that when the task arrives, it initially checks for the states of the cluster where the system considers three different states i.e. active, idle, hibernate [Line-1]. In case the state of the cluster is found to be active (Line-3) than the algorithm forwards the task j to the nearest cluster (Line-4). The next cluster is also checked for its state. If the next cluster is also found to be in active (or busy) state than the process of finding the alternative cluster repeats. If in case the currently exposed cluster is found to be in an idle state than it has to be allocated as fast as possible or else it will return to hibernate state. Hence, the study assumes that when the jobs arrive in idle state, the cluster is now ready to accept the job and process it (Line-5/6) and instantly switches itself to an active state (Line-7). After that, the cluster initiates its consecutive idle period once all the jobs are processed by it. In case the jobs arrive at the cluster in hibernating status than it has to take a decision based on a number of incoming jobs. If the cardinality of the incoming job is a less than certain controlled value (say N) than the cluster continues to be in hibernating state (Line-11). Once the incoming jobs keep on increasing the queue size more than N than only the cluster in hibernate state will be converted to an idle state to process the incoming job. The algorithm easily says that there is only two scenarios of initiating an active state of cluster i.e. i) initiating an active state when an incoming task comes in an idle state or ii) initiating an active state when the cardinality of incoming task over queue is found more than the certain threshold value.

Algorithm for Condition of Task Allocation (CTA)
Input: C (Cluster Node), C_{st} (State of cluster),
Output: Successful allocation of task (*j*)
Start

1. Init C, C_{st}
2. j→check(C_{st})
3. If (C_{st}=Active)
4. Forward j→near (C)
5. For (C_{st}=Idle)
6. process j
7. C→C_{st}(Active)
8. or else
9. Forward j→C(C_{st}=hibernate)
10. else if (C_{st} =hibernate)
11. if (car(j)>N)
12. process j
13. C→C_{st}(Active)
14. or else
15. C→C_{st}(hibernate)

End
End

A closer look at the above algorithm will show that there are certain amounts of resource being used in permitting the cluster to be in the idle state when it is a non-load time. However, the advantage of this algorithm is that the incoming task has fair chances to get processed instantly and thereby minimizing both dynamic loads along with minimization of cost of cluster start up.

To minimize the consumption of idle state of resources, it is essential that non-idle state of functioning to be considered which only assumes an active and hibernating state in its logical formulations. For minimizing load during idle resource states, the cluster should instantly go to hibernate state and not in an idle state when all the incoming jobs are done with processing. The algorithm allows the clusters to adhere to hibernate state once the entire queued tasks are processed. In case the cardinality of the incoming traffic is more than a specified number N than cluster switches to active from hibernate state (Line-5/6 of the second algorithm).

Algorithm to minimize non-idle Resource Consumption (MnIRC)

Input: C (Cluster Node), C_{st} (State of cluster),
Output: Successful allocation of task (j)
Start

1. Init C, C_{st}
2. j→check(C_{st})
3. If (car(j)=0)
4. C→C_{st}(hibernate)
5. if (car(j)>N)
6. C→C_{st}(active)

End
End

The above two algorithms are the core components of LBRC framework, a closer look into the above algorithms will show its dependency over the threshold value N. If N is kept more than it can conserve more resources but will result in a delay. On the other hand, if N is kept smaller than it may minimize delay but will generate reduced cycle of operation. Hence, the system algorithm also considers multiple resource consumption parameters due to multiple conditions e.g. when the cluster is active, idle, hibernate, etc. It also considers other reasons for resource utilization responsible for load balancing e.g. higher congestion, waiting for task in the queue, the cost incurred for waking up the cluster, etc. Hence, combining all these factors and then minimizing it will become an objective function. Hence the prime goal of the proposed algorithm of LBRC is to perform optimization of the cost of operation owing to resource variability. The system take the input as rate of arrival along with higher bounds of cluster rate as well as memory (or buffer) for waiting, resource consumption parameters (stated above). The framework LBRC initially checks for the rate of service for current and computes the utilization of the system. In such condition, it checks for the condition of the existing test parameters are found to satisfy the load balancing requirement based on current traffic. It also checks for computation of response time in this regards to take a decision. The challenge in constructing this algorithm is to maintain a better balance between resource optimization and enhanced response time to formulate better load balancing technique. This is because the operational cost is directly proportional to the rate of service in any cloud computing environment. Soon it starts engaging new virtual machines for this. Hence, this problem is sorted out by LBRC framework to meet the demands of SLA constraints on peak traffic condition. The interesting point of LBRC framework is that none of the decision for load balancing based n resource availability is not taken a virtual machine and thereby the system conserves a maximum amount of resources in VM. It is because VM will be the actual point of entry and exit of every requested and processed jobs via clusters. Hence, proposed LBRC framework does all this by super-imposing the proposed load balancing algorithm within clusters based on their respective states of operation. The next section discusses the outcomes accomplished from the study.

RESULTS

This section presents the discussion of the accomplished results from the proposed LBRC framework. As the prime motive of the study is focused on performing an effective load balancing in the highly distributed environment of cloud, hence its performance parameters are selected in such a way that it can measure its effectiveness in the presence of variable traffic. The study considered its performance parameters as overall resource consumption, the number of processed jobs and CPU utilization. For better benchmarking, we compare our work with that of similar kind of work carried out by Xu *et al*. [22] most recently, which was focused on task scheduling in connection with resource utilization. The author has presented a technique called as RAISM i.e. Resource Allocation using Improved Simulated Annealing Method to address the problems of load balancing over virtual technologies in the cloud. The authors used greedy approach and it outcome was assessed with multiple parameters; however, we choose to select only the relevant and significant parameters for comparative performance analysis.

Analysis of Overall Resource Consumption: The dependency of resource availability is quite high for VMs as they are the prime processing elements of incoming jobs in distributed cloud environment. All the jobs processed by VMs has again dependencies on core clusters over data centers. Therefore, it is quite necessary to testify the scalability of proposed LBRC technique with the recent approach of RAISM by measuring the amount of resource utilization over increasing number of clusters.

Fig. 2 shows that LBRC exhibits better performance over RAISM with increasing number of clusters.

The trend for RAISM is found to be increasing with the increase of clusters. The prime reason behind this is RAISM uses simulated annealing for optimization which fails to maintain a balance between operating cost and processing capabilities. RAISM also uses heuristics and thereby it requires too many numbers of information about each and every cluster. Evaluation of cluster-based heuristics consumes time and too much of resources, On the other hand, the significant contribution of LBRC is that it just uses states of clusters (which is not found in existing system) which is very simple, occupies less buff to compute a state and response time is faster in each cluster. Hence, the overall increment of the resource is quite in a controlled manner for proposed system.

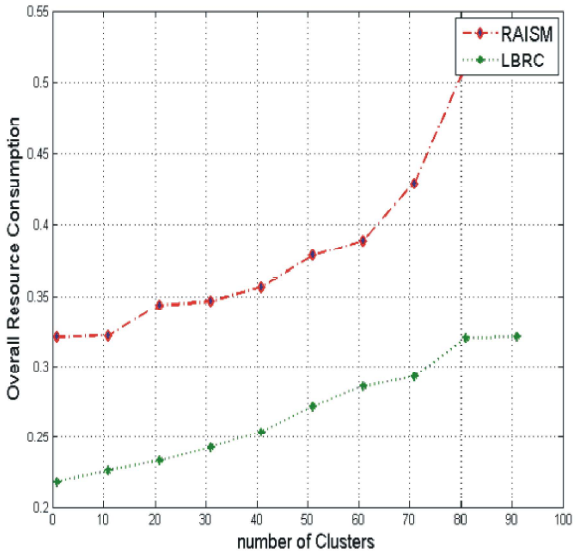

Fig. 2: Overall Resource Consumption

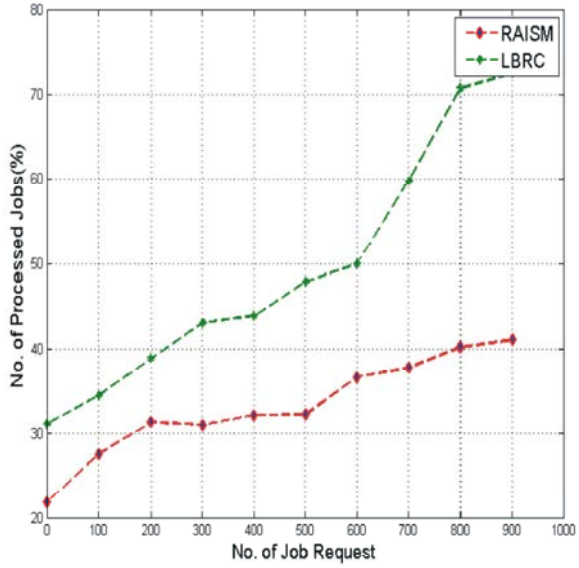

Fig. 3: Number of Processed Jobs

Analysis of Number of Processed Jobs: A total number of processed jobs is another indicator of a resource-efficient load balancing technique. Usually, with the rising uncertainty of the traffic, the VMs are more overloaded compared to the clusters and have the tendency to maintain a large queue. However, as cloud works on distributed system hence it can be expected that there will be a certain relaxation on the load as it will be too distributed over multiple Processing Elements (PE). However, it does so at the cost of the available resource. Hence, a reliable and sustainable load balancing algorithm should demonstrate faster processing capabilities even in

an increasing number of traffic. Fig. 3 shows that proposed LBRC offers increased capability of processing more number of jobs with increasing incoming job request.

The approach of RAISM is based on simulated annealing approach where the authors only focused on minimizing power consumption using predictive approach. The biggest problem with the presented predictive approach of RAISM is that prediction error doesn't get updates for which reason it has to re-perform the algorithm once again. Secondly, RAISM implements the technique over VM whereas LBRC uses it on clusters. The RAISM technique also claims of increasing number of active VM, which is very good for handling dynamic queues generated but unfortunately, it fails to keep number of nodes in hibernating condition. This phenomenon causes extensive drainage of resources for which reason capacity to increase more processing cannot be much enhanced. On the other hand, the significant contribution of the proposed LBRC framework is that it considers the non-idle state of resource utilization (which also includes power). It will mean that proposed system has multiple solutions on load balancing depending on the types of traffic, which increases its synchronicity with cluster and multiple VM to increase its capability of processing an increasing number of jobs.

Analysis of CPU Utilization: A CPU holds control over the resources and it the utilization of the CPU increase than it is quite obviously that resource utilization will also increase. CPU utilization is one of the important factor for any consumer as they pay for the cost of CPU utilization evaluated by their service provider. Hence, an effective load balancing technique must be able to keep a proper balance between this cost factor (i.e. CPU utilization) and efficient traffic management over cloud environment. Fig. 4 shows the analysis of CPU utilization of proposed LBRC technique with existing RAISM.

The first thing of RAISM implementation is that it uses Cloud Sim where always assumes that cloudlet is either in processing state or not in processing state. Because of this binary form of cloudlet condition, the system fails to understand the actual state of cloudlet. Cpu utilization can only be calculated accurately if a number of test-cases of cost incurred is considered in the modeling process. Unfortunately, RAISM uses only predefined configurations of cloudlet, which makes it less aware (or knowledge) about the cloudlet which can also have other possible discrete state. For this reason, there

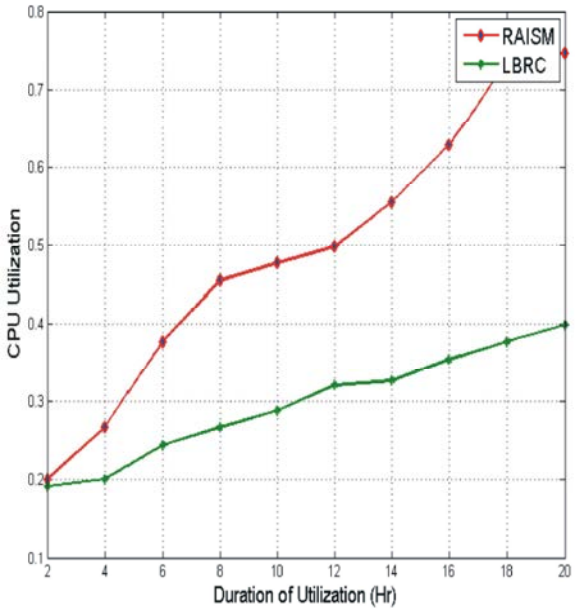

Fig. 4: Analysis of CPU Utilization

are less amount of heuristics available from each cluster that result in more CPU utilization. On the other hand, the proposed system of LBRC addresses this problem by considering multiple states of the cost incurred and highly flexible power management techniques. The first contribution of LBRC in this regards is that i) it uses 3 different states of clusters which provide more discrete information in order to take decision, ii) LBRC uses probabilities of such states of clusters over a queuing system making it more capable to identify the queues in critical and less-critical conditions, iii) the objective function for minimizing the cost is actually dependent on multiple resource consumption factors (viz. resource consumption due to active server, cluster in idle state, cost per rate of services, cost of hibernating, cost of waking up the server, number of incomings queued jobs, etc.). The modeling of LBRC is carried out considering more realistic variables compared to any studies over the existing system. This can be reflected in the trend of CPU utilization curve in Fig. 4.

CONCLUSIONS

The effect of proposed resource utilization has a direct affect on load balancing and vice-versa. The proposed study of LBRC addresses the unwanted resource consumption in its idle states. LBRC permits better-switching controls over the cluster (than to VM and PE) to enhance the decision-making strategies for

governing the rate of services. The design principle of study is also found to make a proper balance between cost of operation for a cluster and effective SLA (Service Level Agreement). The study outcome shows better load balancing capabilities with better resource management over the clusters. Our future work will be further in the direction of more improvement of the task scheduling and provisioning process in a dynamic cloud environment.

REFERENCES

1. Shah, S.I.A., M. Ilyas and H.T. Mouftah, 2016. Pervasive Communications Handbook, CRC Press.
2. Kumar, V., 2013. Fundamentals of Pervasive Information Management Systems, John Wiley & Sons.
3. Wenhong, T. and Z. Yong, 2014. Optimized Cloud Resource Management and Scheduling: Theories and Practices, Morgan Kaufmann.
4. Mahmood, Z., 2014. Cloud Computing: Challenges, Limitations and R&D Solutions, Springer.
5. Peitek, N., 2014. Algorithms for Energy Efficient Load Balancing in Cloud Environments, GRIN Verlag.
6. Antonopoulos, N. and L. Gillam, 2010. Cloud Computing: Principles, Systems and Applications, Springer Science & Business Media.
7. Ningning, S., G. Chao, A. Xingshuo and Z. Qiang, 2016. Fog computing dynamic load balancing mechanism based on graph repartitioning. IEEE China Communications, 13: 156-164.
8. Cao, J., K. Li and I. Stojmenovic, 2014. Optimal Power Allocation and Load Distribution for Multiple Heterogeneous Multicore Server Processors across Clouds and Data Centers. IEEE Transactions on Computers, 63: 45-58.
9. Assi, C., S. Ayoubi, S. Sebbah and K. Shaban, 2014. Towards Scalable Traffic Management in Cloud Data Centers. IEEE Transactions on Communications, 62: 1033-1045.
10. Lin, C.C., H.H. Chin and D.J. Deng, 2014. Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System. IEEE Systems Journal, 8: 225-234.
11. Rao, L., X. Liu, M.D. Ilic and J. Liu, 2012. Distributed Coordination of Internet Data Centers Under Multiregional Electricity Markets. Proceedings of the IEEE, 100: 269-282.
12. Mishra, M., A. Das, P. Kulkarni and A. Sahoo, 2012. Dynamic resource management using virtual machine migrations. IEEE Communications Magazine, 50: 34-40.
13. Simeone, O., A. Maeder, M. Peng, O. Sahin and W. Yu, 2016. Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems. Journal of Communications and Networks, 18(2): 135-149.
14. Deng, W., F. Liu, H. Jin, B. Li and D. Li, 2014. Harnessing renewable energy in cloud datacenters: opportunities and challenges. IEEE Network, 28: 48-55.
15. Xu, G., J. Pang and X. Fu, 2013. A load balancing model based on cloud partitioning for the public cloud. IEEE Tsinghua Science and Technology, 18: 34-39.
16. Liang, H., L.X. Cai, D. Huang, X. Shen and D. Peng, 2012. An SMDP-Based Service Model for Inter domain Resource Allocation in Mobile Cloud Networks. IEEE Transactions on Vehicular Technology, 61: 2222-2232.
17. Luo, J., L. Rao and X. Liu, 2015. Spatio-Temporal Load Balancing for Energy Cost Optimization in Distributed Internet Data Centers. IEEE Transactions on Cloud Computing, 3: 387-397.
18. Gao, B., L. He and S.A. Jarvis, 2015. Offload Decision Models and the Price of Anarchy in Mobile Cloud Application Ecosystems. IEEE Access, 3: 3125-3137.
19. Begum, S. and C.S.R. Prashanth, 2013. Review of Load Balancing in Cloud Computing. International Journal of Computer Science, pp: 10.
20. Begum, S. and C.S.R. Prashanth. 2013a. Investigational Study of 7 Effective Schemes of Load Balancing in Cloud Computing. International Journal of Computer Science, pp: 10.
21. Begum, S. and C.S.R. Prashanth, 2014. Mathematical Modelling of Joint Routing and Scheduling for an Effective Load Balancing in Cloud. International Journal of Computer Applications, pp: 104.
22. Xu, X., L. Cao and X. Wang, 2016. Resource pre-allocation algorithms for low-energy task scheduling of cloud computing. Journal of Systems Engineering and Electronics, 27: 457-469.