AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

Fotm: Framework Optimization of Traffic Management for Minimizing Load In Cloud

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

[1]Senior Assistant Professor, Department of CSE, New Horizon College of Engineering, Bangalore, Visvesvaraya Technological University, Belgaum, Karnataka, INDIA
[2]H.O.D and Dean of Academics, Department of CSE, New Horizon College of Engineering, Bangalore, Karnataka, INDIA

Address For Correspondence:
Suriya Begum, Senior Assistant Professor, Department of CSE, New Horizon College of Engineering, Bangalore, Visvesvaraya Technological University, Belgaum, Karnataka, INDIA
E-mail: suriyabegumvtu@gmail.com

ARTICLE INFO

Article history:
Received 26 July 2016
Accepted 21 September 2016
Published 30 September 2016

Keywords:
Load Balancing, Cloud Computing, Task Scheduling, Resource Availability.

ABSTRACT

Background: - With the increasing user-base from different geographic regions, it has become a challenging problem to maintain similar standard of service delivery by the cloud service providers. We reviewed the existing techniques of load balancing to find that they are highly symptomatic in nature by considered a much narrowed range of problem. We strongly feel that resource allocation, resource utilization, identification of dynamic states of Virtual Machines (VMs) are some important attributes for incorporating optimization in existing load balancing techniques, which are quite ignored in existing research work. ***Objective:*** - Hence, to present a novel modeling of FOTM or Framework Optimization of Traffic Management in this paper where the optimization is carried out on our prior framework of load balancing using analytical modeling approach. To develop explicit techniques for arbitrary job arrival, cloud system, load balancer, system to check resource status, allocation of incoming task, transition processing reserved area. ***Result:-***The study outcome is found to be better than most recently implemented load balancing with respect to resource consumption, number of processed jobs, and CPU Utilization.***Conclusion:*** - This paper is a continuation of our past work where we have illustrated a mathematical modeling to achieve an efficient load balancing.

INTRODUCTION

With the growing range of users over cellular networks there has also been exponential growth of mobile application. Such applications generate a very dynamic shape of site visitors which is continued with the aid of cloud environment in order to keep pervasiveness (Obaidat, 2011). Cloud computing is all about storage and processing with gaining access to information and applications over the internet instead of your local physical servers (Erl *et al*. 2015). The cloud is just a metaphor for the internet and also enables delivery of host provider over the net. It permits businesses to devour compute resource as utilities in place of having to build and maintain computing infrastructure (Wenhong, 2014). Although, cloud gives a better traffic management for its existing customers, but there are sure uncertainties which cannot be dealt with by cloud e.g. surprising rise of online users, no general stereotyped agenda for peak hours and idle time. Due to such problems, there are numerous cases of downtime too ensuing in violation of SLA (Murugesan, 2016). Hence, load balancing is tremendously essential to control such massively growing traffic over cloud (Norman, 2014). In a very preferred way load balancing strategies divides the amount of work that a cluster has to do among two or more clusters in order that extra work get performed inside the equal quantity of time. It is also approximately dispensing workload and computing resource in a maximum dynamic environment permitting an corporation to manage

ToCite ThisArticle: Suriya Begum and Dr. Prashanth C.S.R., Fotm: Framework Optimization Of Traffic Management For Minimizing Load In Cloud. ***Aust. J. Basic & Appl. Sci.,*** *10(14): 137-145, 2016*

utility by means of allocating aid among a couple of clusters, community or server (Gonzalez, 2015). It also involves hosting the distribution of workload traffic and demands that resides over the internet and thereby helps organization to achieve high performance level for potentially lower cost. However, there is no denying the fact that existing load balancing techniques in cloud are absolutely not sufficient enough to cater up the dynamic demand of the uncertain traffic from various parts of the world. Therefore, the proposed study presents a framework that performs optimization of the traffic management in order to ensure a proper load balancing mechanism over cloud environment. The organization of the proposed paper is as follows: -the second section discusses about the related work carried out in past addressing the problem of load balancing over cloud followed by brief discussion of problem identification in third section. The fourth section discusses about the proposed methodology followed by algorithm design principle in fifth section. Discussion of analytical modelling is carried out in sixth section. The result discussion is carried out in seventh section followed by conclusion.

Literature Review:

Simeone *et al*. (2016) discussed a technique to limit and running cost needed to install and preserve dense heterogeneous network. This technique designed effectively gives spectral efficiency, statistical multiplexing and load balancing. Assi *et al*. (2014) mentioned about the decomposition approach to triumph over from virtual networking environment mapping issue in cloud information with an aid of memory up scaling mechanism. This technique designed in this type of manner that it makes use of both a specific and semi-heuristic decomposition with the objective to use load balancing by means of minimizing the most hyperlink load in the network. Ningning *et al*. (2016) discussed the atomization of cloud era and fog computing to make physical nodes in one-of-a-kind degree into digital system nodes. This designed machine provides device community much flexibility and dynamic load balancing mechanism can shape effectively because it minimizes the intake of node migration. Xu *et al*. (2013) mentioned approximately higher load stability model for the public cloud primarily based on the cloud partitioning idea the use of probabilistic decision making model. Gao *et al*. (2015) have presented a similar model based on decision making principle and examined its affects on application. Luo *et al*. (2015) discussed about how to leverage both geographical and temporal variable of energy price to reduce energy cost for distributed IDCs. Lin *et al*.(2014) discussed about the more practical dynamic multi service scenario in which server cluster only handle a specific type of multi-media task and client request a 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. Deng *et al*. (2014) discussed about 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. Cao *et al*. (2014) mentioned about the broaden strength and performance confined load distribution approach for cloud computing in current and upcoming characteristic huge-scaled data center. This method is designed to offer performance optimization and energy management. Rao *et al*.(2012) discussed problem about the resource management for internet service with a collection of spatially distributed data center. This method designed effectively to minimize the total resources geared to QoS constraint as well as the location diversity and time diversity of resources under MEM.Mishra *et al*. (2012) mentioned about designated review of digital device migration method and their usage towards dynamic resource management in virtualization surroundings. This approach designed correctly to reduce server sprawl, minimizing energy consumption, load balancing throughout physical gadget. Liang *et al*.(2012) mentioned about provider selection for inter domain service transfer to accomplish the load stability amongst more than one domain. This designed principle substantially improves the device cost and reduces provider disruption.The next section discusses about problem identification of the proposed study.

Problem Identification:

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

- ***Less Emphasis on dynamic States of Virtual Machine (VM)****:*

Majority of the algorithms at present uses normal queuing approach to stack up the incoming jobs based on the availability of VM. However, there are few studies that searches for optimal state of VM for dynamic task allocation.

- ***Few Effective Scheduler Design:***

Existing studies considers schedulers to be a separate module residing in clusters over data centers that allocates resources based on VM availability. This process is quite less effective as it includes time and doesn't support many real-time application processing. Schedulers can be embedded properties within VM and PE (Processing Elements) too, which is less emphasized.

- ***Less benchmarked Techniques:***

The existing techniques are developed to solve limited problems over traffic. Moreover, till date, there are very few techniques of load balancing that are found to be benchmarked. In such case, it becomes quite non-trivial task to understand the robust load balancing algorithm till date.

Hence, the above mentioned points are some of the significant problems that are found unaddressed in existing research towards load balancing algorithms in cloud. The proposed system will discuss about its adopted technique to solve the problem of load balancing.

Proposed Methodology:

The proposed study is a continuation of our past research work (Begum, 2013, Begum, 2013a, Begum, 2014). Using analytical modeling, the proposed system presents a framework that mainly focuses on proper utilization of resources in order to ensure better load availability.The proposed technique of optimization is carried out by set of algorithms that performs optimization to attain better performance in traffic management over cloud.

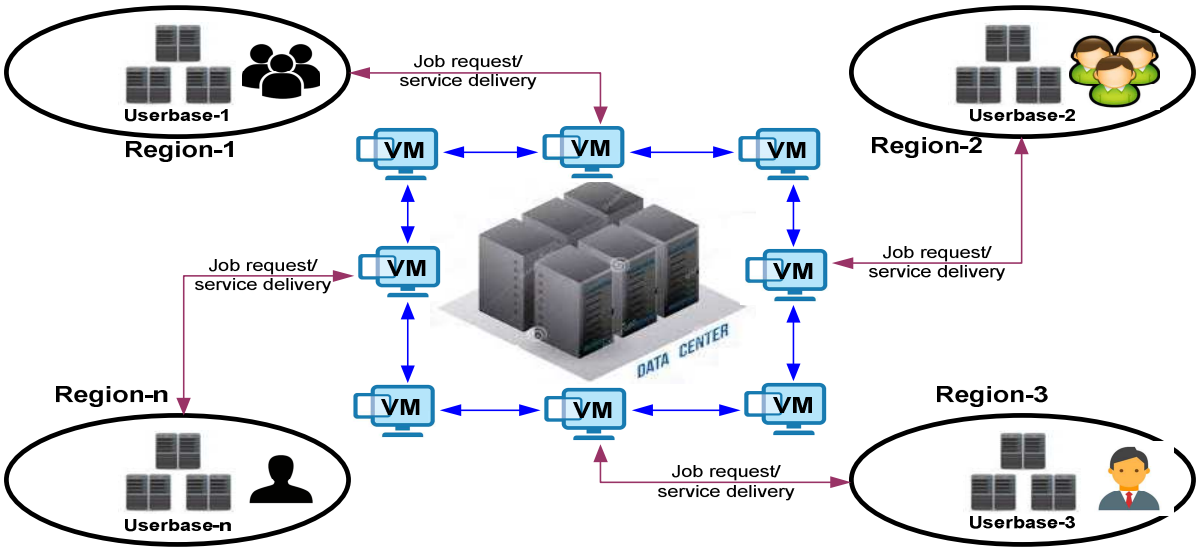


Fig. 1: Pictorial scheme of FOTM

The prime goal of the proposed algorithm design is to ensure a better traffic management over cloud. The underlying architectural model of proposed system is specifically developed for enhancing load balancing requirements over cloud. The study in this phase will emphasize on introducing a novel load-balancing technique in such a way that random job arrival should be allocated to available resources. The study also uses the concept of region in order to ensure that availability of service request (Fig.1). Each region consists of varied user bases. We also assume that each region is potentially unsynchronized with each other for better formulation of challenging traffic condition. All the virtual machines are considered to be highly synchronized with each other. The entire assessment of the study was carried out considering i) CPU usage, ii) Memory, and iii) Disk-space. The motive of the current technique is basically to accomplish zero waiting time for any randomly arrived jobs. Although, in reality, there are infinite number of users sending unlimited job request, which is really difficult to model. Hence, for better modelling, the proposed system will consider a threshold based schemes, where the system will consider a finite number of jobs that are randomly distributed and will act as input to the main system, whereas the output will be accomplishment of zero job waiting time in specified time-limits.

Algorithm Design Principle:

The proposed study will consists of following tentative components in the model:

- ***Random Job Arrived:***

The proposed system considers the arrival of the jobs in random fashion in order to map the real-time situation in cloud environment. The model used for designing random job arrival. Fig.2 highlights the process flow of the random job arrival system. The design will be initiated by considering maximum of the cloud resources details (e.g. CPU, maximum jobs, and disk usage). For spontaneous performance analysis, the design is controlled by time as threshold factor based on which an arrived jobs will be randomized and forwarded to the cloud system.

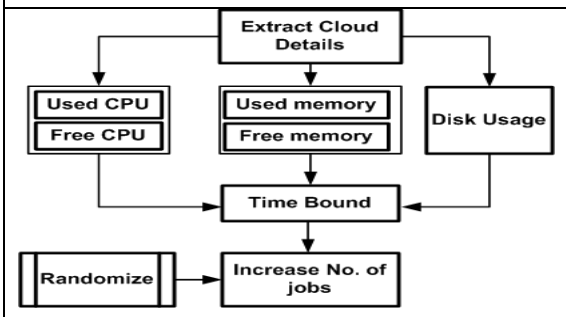
Process Flow	Algorithm Design
	**Algorithm For Random Job Arrival** **Input**: v_1(CPU Usage, v_2 (memory),and v_3 (disk-usage) **Output**: Generation of Random Jobs 1. Fetch Cloud Details 2. Define variables v_1, v_2, and v_3 3. Apply Time Bound 4. Use Random function 5. Increase number of request.

Fig.2: Design Flow of Random Job Arrived

- ***Cloud System:***

The design of the cloud system will consists of specific number of cloud servers where each server may have multiple end number of CPUs. The proposed cloud system will comprise of multiple number of request that are mapped with random job arrival system (Fig.3). In short, cumulatively it can be said to be a datacenter too. The request acts as both inputs for cloud system as well as proposed load balancer scheme. After the random jobs are arrived, the proposed system directs the traffic to the existing cloud server farm for further processing of the data.

- ***Proposed Load Balancer:***

The role of this module is to check for number of virtual machines that are free to process the incoming job. For availability of more number of VMs, the proposed system shares the work load equally among them. This ensures faster work execution and processing time. The technique makes use of the state transition factor in order to stores the state of the previous allocation of a VM to a request from a given user based. The proposed system uses a transition processing reserved area im resources so that once the job arrived (that is time stamped and indexed by job ID) by job allocator, it is instantly forwarded to the algorithm for ensuring minimal traffic.

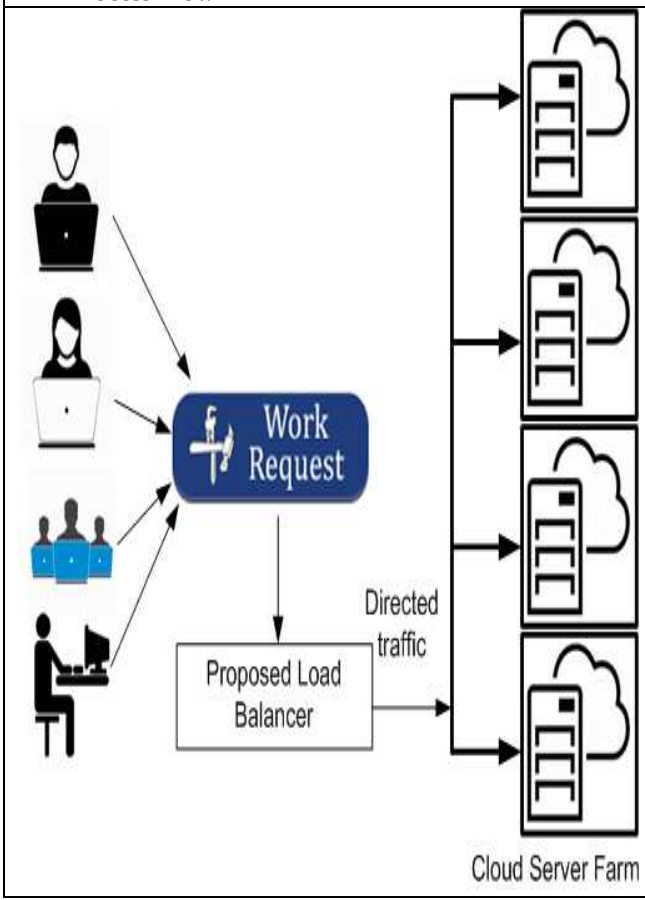
Process Flow	Algorithm Design
	**Algorithm for Cloud System** **Input**: C_i, r_m, h, H **Output**: Jobs allocated to VM 1. Initialize variable for cloud server (C_i) 2. Create connectivity among C_i. 3. User generates request r_m 4. Pass r_m to load balancer algorithm 5. While Schedule *U* is not empty (in an arbitrary order) 6. For each job r_m in meta-job *U* (in an arbitrary order) 7. For each VM h_j($h_j \epsilon H$) 8. Compute earliest completion time CT[i,j] on VM h_j 9. Compute ETT[i] on VM h_j //Estimated Transfer Time 10. Compute minimum ETPT [i, j]= CT[i,j]+ETT[i] 11. End for 12. For each job rm in meta-job set U 13. Find job r_m with maximum earliest ETPT (Estimated Total Processing Time) 14. End for 15. Assign job rm to the corresponding VM h_j with maximum earliest ETPT 16. Delete job r_m from all VM H 17. Update virtual network status 19. End For 20. End While

Fig.3: Design of Proposed Cloud System

- ***Resource Status Monitor:***

The proposed module also targets to keep a track of resource for effective check the quality of run-time evaluation of the proposed system. This module will also retain information about every VMs and the number of request currently allocated to which VM when a work request is allocated, a new VM with free state will arrive. This module also ensures that in case of availability of multiple VMs, the first identified free system is considered for resource status monitor. The core cloud system forwards the job request to the virtual machine using the specific ID and thereby notifies the proposed resource status monitor for the new allocation. It considers consider i) VM_ID, ii) Job ID, iii) No. of Jobs, iv) Transmission Regions, v) Execution Time, vi) Used CPU, vii) Free CPU, viii) Used Memory, and ix) Free Memory

- ***Job Allocator****:*

This module will be responsible for allocating the task from the proposed load balancer scheme and will formulate a matrix to sequentially arrange the incoming jobs to the next module TPRA.

- ***Transition Processing Reserved Area:***

The prime target of this component is to ensure that there is zero queue size and in-order to retain it, the proposed mechanism will introduce a middleware support that is termed as Transition Processing Reserved Area (TPRA).An arrived job acts as an input for TPRA, which maintains an intermediate memory to allocate resources to the incoming jobs for further processing. The prime objective of this module is to maintain a zero queue length. It further checks for free or busy system.

Hence, depending on the progress of time, the existing jobs are pushed into free systems. Using threshold-based mechanism, when the memory of the free system is over, it is then termed as busy system and hence the task allocations performs occasional transition from one to another system in order to maintain highly minimized waiting time.

Analytical Modelling:

Consider x_n be the no. of cloud server farms of n-type that are located in distributive manner. Consider r_m is the variable for user's request. Hence, for better dynamicity and challenging environment, the future work considers $m>>n$ where both m and n are natural numbers. The system will accomplish extracting information of cloud resources and can be mathematically represented as,

$C_{res}= [\alpha_1, \alpha_2, \alpha_3]^T$

Where α_1, α_2, and α_3 represents CPU utilization, maximum number of jobs, and disk usage that are recorded with respect to observational time (T). In the above equation, $\alpha_2 \approx \max(r_m)$ and hence, the proposed randomized process can be analytically represented as,Job-Arrived=Rand(α_2) $\approx$ Rand [$\max(r_m)$].The proposed system designs a component termed as cloud system which comprises of multiple sets of VMs. The job allocation process is done randomly to the VMs as shown in Fig.4.

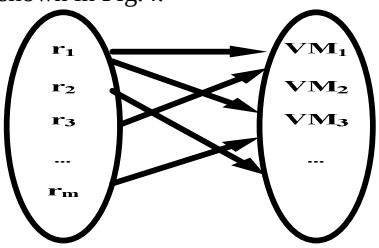


Fig.4: Proposed Job Allocation process to VMs

The cloud system will maintain a data-structure containing VM-IDF, Job ID of the jobs that has to be allocated to corresponding VM and VM status to keep track of load distribution. The VM status will represent percentage of utilization. The proposed study will also use TPRA to allocate the resources and distributes the load as per data structure. Consider β as the data-structure of proposed system that can be analytically represented as,

$B=[Job_{ID}, VM_{ID}, VM_{states}]$

Hence, the proposed study considers the design of data structure considering Job ID, VM ID and VM status ID. The system then evaluates server capacity of VM. For extensive analysis, the proposed study evaluated server capacity with an aid of multiple factors as analytically represented as,

Server-capacity=[*T, B, C*]

In the above equation, *T, B,* and *C*represent threshold-based time, busy-time, and completion time respectively. Therefore, the mean queue length can be determined as,

$$Q = \frac{\alpha_1}{1-\alpha_2}, \text{ where, } \alpha_1 = \frac{B}{T}$$

In the above mathematical representation, *B* is the amount of time that server was active during threshold-time *T*. Similarly, the service time can be evaluated as,

$$S = \frac{B}{C}$$

Where *C* is the number of transactions completed during threshold-period. Therefore, the objective function of the proposed system is expressed as follows

$$f_1(x) = \arg_{\min}(Q) \approx o \text{ and } f_2(x) = \arg_{\min}(C_{res})$$

In the course of processing, if any VM is overloaded (busy), than the jobs are migrated to the VMs that are underutilized (free). On completion of processing, the cloud system will update the entire data structure. This is the prime basis of the design of proposed load balancing scheme that will ensure that status of VM is always 100%, which will mean that it is completely utilized with no possible condition of deadlock (VM overrun). Hence, the algorithm for proposed load balancer is as follows:

Algorithm for Load Balancer:

Input: r_m

Output: Load balancing

1. Initialize the submitted jobs
2. Select a job for the job classes and create VMs for the chosen job size.
3. SORT in descending order
4. Select job category LN.
5. Set the current large job as LN.
6. Perform Resource Status Monitoring
7. Track all the events (i) Inter-arrival time, ii) Number of jobs, iii) Execution Time, iv) used CPU Time, v) Free CPU time, vi) used memory, vii) Free Memory, viii) Used Disk space, ix) Free Disk Space.)
8. β as the data-structure of proposed system

   $\beta = [Job_{ID}, VM_{ID}, VM_{states}]$
9. Define Server Capacity

   Server-capacity=[T, B, C]
10. Evaluate mean queue length

$$Q = \frac{\alpha_1}{1-\alpha_2} \text{ Where, } \alpha_1 = \frac{B}{T}$$

11. Evaluate Service Time

$$S = \frac{B}{C}$$

12. Define Objective Function

$$f_1(x) = \arg_{\min}(Q) \approx o \text{ and } f_2(x) = \arg_{\min}(C_{res})$$

13. Create one VM for LN with capacity C(LN).
14. Create VMs for next large jobs LN-1 with capacity allocated for LN is reached.
15. Set N=N+1
16. Repeat 1-3 until maximum capacity is reached.

Results:

This section presents the discussion of the accomplished results from the proposed FOTM modeling. As the core goal of the presented study was to establish an effective load balancing over highly dynamic traffic scenario over cloud, hence its performance parameters are selected in such a way that it can measure its effectiveness in presence of variable traffic. The study considered its performance parameters as overall resource consumption, number of processed jobs, and CPU utilization. In order to assess the outcomes, we choose to compare the outcomes of FOTM with the most recent work carried out by Aslanzadeh *et al*. (2015), Xu *et al*. (2016), and Zhao *et al*. (2016). Aslanzadeh *et al*. (2015) have developed a swarm intelligence based technique called as Endocrine PSO (Particle Swarm Optimization) in order to effectively manage traffic load. Xu *et al*. (2016)have presented their work on task scheduling in connection with greedy-based resource utilization called as RAISM i.e. Resource Allocation using Improved Simulated annealing Method to address the problems of load balancing over virtual technologies in cloud.Zhao *et al*. (2016) have applied a Bayes

concept using heuristic environmental attributes in order to perform load balancing. We consider all the work under similar performance parameters and test-environment to perform comparative performance analysis.

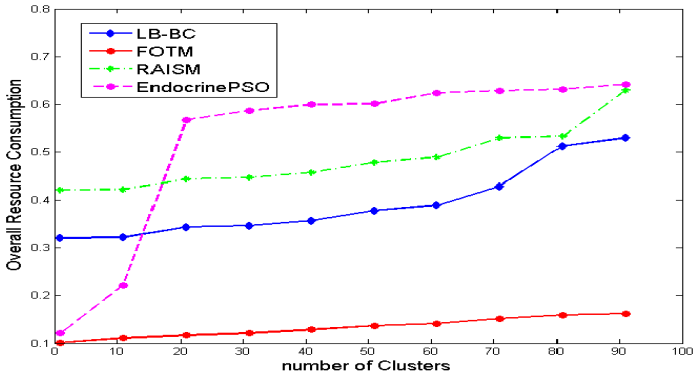


Fig.5: Analysis of Overall Resource Consumption

From Fig.5, it can be seen that proposed FOTM has lesser extent of resource utilization with respect to increasing number of clusters compared to other existing techniques. RAISMaddresses multi-objective constrained optimization problem for which reason the trend is somewhat predictable. However, usage of enhanced simulated annealing causes slower scheduling, which cannot match with the increases rate of incoming jobs causing faster drainage of resources. LB-BC overcomes the heuristic problems of RAISM by incorporating a deployment controller with the host cluster. Unfortunately, LB-BC was meant only for local area network to be working in chain in order to cater up the task scheduling over cloud. This causes the infrastructure (routers) to drain resource in order to get the task executed; however, it can successful save overall resource consumption for the host. Endocrine PSO seems to have a robust starting by searching for VM with overload task. As the VMs do have better synchronization among each other, hence enough amount of resource utilization is saved in this search till 20 clusters, however, with inclusion of more number of clusters, the Endocrine PSO algorithm generates recursive algorithm just to find the next overload VM, thereby causing massive drainage of resource after few rounds. However, proposed system purely works on state-transition matrix considering various region-based communication. All the sourced jobs that reaches VMs are then subjected to TPRA algorithm, which coordinates and sync itself with existing status of VM, which performs double time better than similar operation carried out using Endocrine PSO technique, but with lesser rounds of iterations. Hence, at any size of incoming traffic, the performance of FOTM is insignificantly affected.

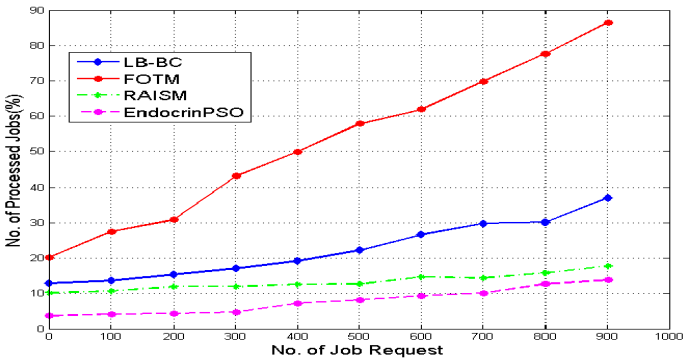


Fig.6: Analysis of Number of Processed Jobs

Fig.6 shows the analysis of number of processed jobs, where FOTM was found to have higher capability for faster processing based on incoming job from traffic. The prime reason behind this is the objective function which always attempts to queue to zero with less number of resources being involved in it. The mechanism is better and highly non-recursive for same job as compared to LB-BC, RAISM, and Endocrine PSO techniques.

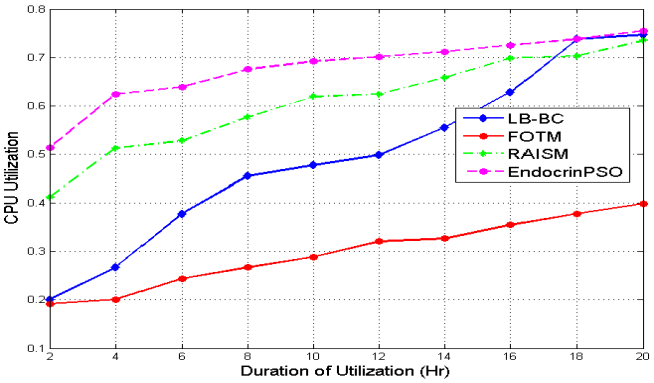


Fig.7: Analysis of CPU Utilization Time

Fig.7 highlights the performance of CPU utilization. A closer look will show that both proposed FOTM and LB-BC has nearly similar CPU utilization in initial few duration of system usage. This is due to the fact that both the model follows task deployment based on cluster. But, LB-BC posses increased loops to search for better physical host and this intermediate buffer will be always required to be stored resulting in sudden increased CPU utilization. RAISM computes capacity of host in 4 iterative and extra steps that results in excessive CPU utilization. Similarly, Endocrine PSO incorporates three extra steps of optimization in task scheduling resulting in excessive CPU utilization. Hence, the proposed system has better CPU utilization as compared to existing system.

Conclusions:

With the number of enterprise applications migrating to cloud-based services, the number of users are constantly increasing day-by-day. Developing a model to perfectly predict traffic and user behavior is far from reality at present and hence only way to control as well as manage traffic is to apply load balancing algorithm. However, after reviewing the existing system, it is found that existing load balancing techniques are developed on the ground of certain applications without considering traffic uncertainty factors e.g. unavailability of VM, service mirroring error, etc. This paper is a continuation of our past work where we have presented a mathematical modeling to perform load balancing. The current paper has presented a novel technique of optimization of our past framework by incorporating some of the novel features in it. The newly added features include algorithm design for random job arrival, cloud system, load balancer, system to check resource status, allocation of incoming task, transition processing reserved area etc. The study outcome was found to excel better performance in comparison to existing techniques with respect to load balancing performance over cloud.

REFERENCES

Obaidat, M.S., M.Denko and I.Woungang, 2011. Pervasive Computing and Networking. John Wiley & Sons

Erl, T., R.Cope and A.Naserpour, 2015. Cloud Computing Design Patterns. Prentice Hall

Wenhong, T andZ.Yong, 2014. Optimized Cloud Resource Management and Scheduling: Theories and Practices,Morgan Kaufmann

Murugesan, S andI.Bojanova, 2016. Encyclopedia of Cloud Computing, John Wiley & Sons

Gonzalez, J.U. and S.P.T.Krishnan, 2015. Building Your Next Big Thing with Google Cloud Platform: A Guide for Developers and Enterprise Architects. Apress

Simeone, O.,A.Maeder, M.Peng, O.Sahin andW.Yu, 2016. Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems. Journal of Communications and Networks, 18(2): 135-149.

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.

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.

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.

Gao, B., L.He andS.A.Jarvis, 2015. Offload Decision Models and the Price of Anarchy in Mobile Cloud Application Ecosystems. IEEE Access, 3: 3125-3137.

Luo,J., L.Rao andX.Liu,2015. SpatioTemporal Load Balancing for Energy Cost Optimization in Distributed Internet Data Centers. IEEE Transactions on Cloud Computing,3: 387-397.

Lin, C.C., H.H.Chinand D.J.Deng, 2014. Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System. IEEE Systems Journal, 8:225-234.

Deng, W., F.Liu, H.Jin, B.Li andD.Li, 2014. Harnessing renewable energy in cloud datacenters: opportunities and challenges. IEEE Network,28:48-55.

Cao, J., K.Li andI.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.

Rao, L.,X.Liu, M.D.Ilic andJ.Liu, 2012. Distributed Coordination of Internet Data Centers Under Multiregional Electricity Markets. Proceedings of the IEEE, 100: 269-282.

Mishra, M., A.Das, P.Kulkarni and A.Sahoo, 2012. Dynamic resource management using virtual machine migrations. IEEE Communications Magazine, 50: 34-40.

Liang, H., L.X.Cai, D.Huang, X.Shen andD.Peng, 2012. An SMDP-Based Service Model for Inter domain Resource Allocation in Mobile Cloud Networks. IEEE Transactions on Vehicular Technology, 61:2222-2232.

Begum, S.,and C.SR.Prashanth, 2013. Review of Load Balancing in Cloud Computing. International Journal of Computer Science Issues, 10.

Begum, S.,and C.S.R. Prashanth, 2013.Investigational Study of 7 Effective Schemes of Load Balancing in Cloud Computing. International Journal of Computer Science Issues, 10.

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, 104.

Aslanzadeh, S., and Z.Chaczko,2015.Load Balancing Optimization in Cloud Computing: Applying Endocrine-Particle Swarm Optimization.IEEE International Conference on Electro/Information Technology.

Xu, X., L.Cao andX.Wang, 2016. Resource pre-allocation algorithms for low-energy task scheduling of cloud computing. IEEE Journal of Systems Engineering and Electronics, 27: 457-469.

Zhao, J., K.Yang, X.Wei, Y.Ding, L.Hu and G.Xu, 2016. A Heuristic Clustering-Based Task Deployment Approach for Load Balancing using Bayes Theorem in Cloud Environment. IEEE Transactions on Parallel and Distributed Systems, 27(2): 305-316.