IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-5, Issue-5 E-ISSN: 2347-2693

Study and Analysis of Distributed Concurrent Models in Fog

P.P. Shaha [1*], Padma Priya.R.[2]

[1*] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
[2] School of Computer Engineering, Vellore institute of Technology, Vellore, India

Corresponding Author: pritamshah2011@gmail.com Tel.: +91-9503239108

Available online at: www.ijcseonline.org

Received: 22/Apr/2017, Revised: 30/Apr/2017, Accepted: 20/May/2017, Published: 30/May/2017

***Abstract*—** Fog computing is a developing area that offers numerous potential advantages to different organizations and common client. It is the extended type of distributed processing. It depends on on-demand-service model in which data, infrastructure, software and different service model are given according to the customer requirement at some instance of time. Load balancing is utilized to distribute workload among various cloud systems or node to get full resource utilization. It is the prominent means to achieve efficient resource sharing and utilization. Load balancing has turned into a challenge issue now in distributed computing frameworks. To meets the clients enormous number of requests, there is a need of distributed solution to handle client request concurrently. Servers can't be relegated to specific customers individually. Distributed computing involves a large network and different component that are available throughout the wide area. Hence, we design distributed concurrent model which handle multiple client request and reduce the load on the cloud and improve the performance. This research work is performed on student course registration in a University which shows how distributed concurrent models works in Fog when multiple request that are send by client. Our proposed Approach is use to complete tasks within deadline, efficient resource utilizations, fulfilling data consistency constraints.

***Keywords*—**Fog Computing; Dsitributed model; Course Registration; Resource Utilization; Concurrency; Load Balancin;MobileComputing

I. INTRODUCTION

There has been growing trust on the utilization of cell phones to do computation and get to information. These smarts devices can range in unpredictability from tablets, PC's, mobile phones in complicated embedded system. To fulfill user necessity for ubiquitous computing capabilities, new perfect models and idea have wound up accessibility to offer computational resources at anytime and anywhere, among which cloud computing has recently emerged as a standout among the most prominent processing ideal models to offer on demand computing capacity. The cloud computing has shown effective for a broad assortment of user needs, on-demand applications, providing basic information for data storage service, fully advanced platforms and virtual machines[1-7].

The cloud suppliers give such a large number of services over the internet or web, where users can access services that are available on server [8], [9]. These server farms are much of the time mid to huge workplaces that are scattered in several zones over the world. This arrangement makes customers inclined to organize delays when a service being utilized is not situated in a server farm geologically close them. The customer depends on the application; this delay can really degrade the quality of product or application when framework can't fulfills the basic requirement for service necessities of an application, bringing about service interference from the client perspective. The limitation posed by the last mile availability to end user has moreover been seen as a limitation by other framework providers. These relationship don't totally depend on upon huge data center it depends upon the network i.e. customer use handsets depending upon the available transmission capacity and latency. Rather, data sharing is regularly encouraged by master data servers that are discovered close to the users, and which frequently connect to station and handle important content.

To enhance service provisioning in the situation delineated above it is important to minimize the delay in network with a goal to improve user quality and reduce degradation of service quality. One of the methods for reducing the network delay is to convey the service closer to the user, while so far keeping up client portability. Unfortunately, with the current location and deployment data servers, it is impossible to provide low network delays; in this way another example is important to beat such issue. In such condition the Fog Computing perspective has designed to solve such

limitations and go about as a supplement to the present cloud computing model [10].

The Fog computing designed model relies on small distributed data server that is used to reduce the network delay for the application and reduce processing turnaround times for the users [11]. Fog server has less processing capability and is nearer to clients, but they can trust on huge cloud data server. In this way with various data server which are organized in hierarchically, has a suggestion on how clients' information and processing are figured out how to enhance service quality and in addition to decrease costs and new experiments to be handled. In this research work we deliver an how distributed concurrent models works in cloud when multiple request that are send by client by using android enabled mobile device. Our proposed Approach is use to complete tasks within deadline, efficient resource utilizations, fulfilling data consistency constraints.

In the second section we discuss the comparison between Cloud, Fog and Mobile computing. The section third we discuss some of existing algorithm. In Section Fourth we discuss our proposed methodology. In next section we discuss our system design of course registration system application. In sixth section we compare our proposed algorithm with existing algorithm. In Section sixth describe the results. In last section concludes the research work part.

II. CLOUD, FOG AND MOBILE COMPUTING

A. Comparison of Fog with Cloud

In this section we are considering some of the important parameter that differentiates between these two technologies which are shown in following table 1.

Parameter	Cloud Computing	Fog Computing
Latency	High	Low
Delay	High	Very Low
Location Awareness	No	Yes
Security	Less Secured	More Secured
Geographical Distribution	Centralized	Distributed
Mobility	Limited Support	Supported
Location Flexibility	Within the Internet	At the edge of local network
Maintenance	Daily	Single Hope
Vulnerability	High Probability	Very Low Probability
Number of Server Node	Few	Very Large
Attack of data transmission	Probable	Low Possibility
Target User	General Internet User	Mobile User

Table.1. Fog vs. Cloud Computing

B. Comparison Cloud with Mobile Computing

In this section we are considering some of the important parameters that differentiate between these two technologies which are shown in following table 2.

Parameter	Cloud Computing	Mobile Computing
Portability	Not Supported	Supported
Location Awareness	No	Yes
Security	Less Secured	Secured
Geographical Distribution	Centralized	Distributed
Mobility	Limited Support	Supported
Offline Usability	No offline Usability	Yes
Resource Availability	More	Less
Vulnerability	High Probability	Very Low Probability
Processing Power	High	Low
Attack of data transmission	Probable	Low Possibility
Storage	High Storage Space	Limited Storage Space
Target User	General Internet User	Any Mobile User

Table.2 Cloud vs. Mobile Computing

III. RELATED WORK

In this section the literature surveys that are mainly concentrations on the load balancing and distributed concurrent models in cloud computing is described in this section. In [1] researchers presented Bulk Synchronous Parallel algorithm which is a distributed computing model based on parallel iteration approach, in this methods total computation is divided into multiple sub tasks and the computation process contains the series of iterative super steps. Each and every super step can again forms three sub lower steps, like local calculation, synchronization and barrier. In the first step, the multiple clients can be used same function and that function is used for each and every sub task that are run concurrently, so that it effectively achieve parallelism. In the second step i.e. synchronization of data that are required other sub-task they can communicate each other. In the barrier step some of the task can need same of sub-task are depend on other task [1]. The model was planned to design for Multi-tenant approach, it uses SaaS system that assist multiple tenant though this system. This system also provides high availability and accessibility to the multiple tenants at a same time. It design such way that it satisfy the requirement of the SaaS system. The primary group understands the computing model of the SaaS structure by accommodating the system and its services as well as the tenant's services and applications [2]. In [3] researchers presented a novel model design which contains cloud as a database with information confidentiality and performing all the operation concurrently on the data which is available on encrypted format in database. This is the novel approach that support multiple user that are distributed over the world which are directly interface to cloud database and perform independent different operation on this database like updating database, modify database etc. In [4] researchers presented the Active monitoring load balancer approach, in this algorithm they use multiple VMs and maintain the details of every VMs and assign VMs to the user based on total number of request currently assign for VMs this algorithm reduced the load on the cloud environment. In [5]

researchers represent a load balancing algorithm which helps to minimize allocation time of user request and minimize the system overhead. They used the concept of virtual machine (VM) migration form overloaded VM to other in order to maintain efficient resource utilization. Authors in [6] proposed the hierarchical graph approach to decompose the linked jobs, which can improve the task execution concurrency and decrease the execution cost. Authors in [8] proposed a novel VM- assign algorithm for efficient allocation of incoming job on virtual machine existing in cloud computing systems. The presented algorithm focuses on finding such VM which have fewer loads and then allocate the incoming job to such VM this operation are performed concurrently. In [12] proposed Transparent computing which is highly visualized framework, which is mainly focuses to build up the processing framework that is transparent to clients with the support of cross-application and cross-platform. In [13] author proposed an Automated Course Advising System which guide the students for selecting proper course they design such model with the help of JAVA technology. A straightforward grid model for managing resources and scheduling the different task was proposed in [14]. This approach validates the scalability of ant colony algorithm and obtained good result like task allocation, resource utilization and reduced response time.

IV. METHODOLOGY

A. *System Architecture*

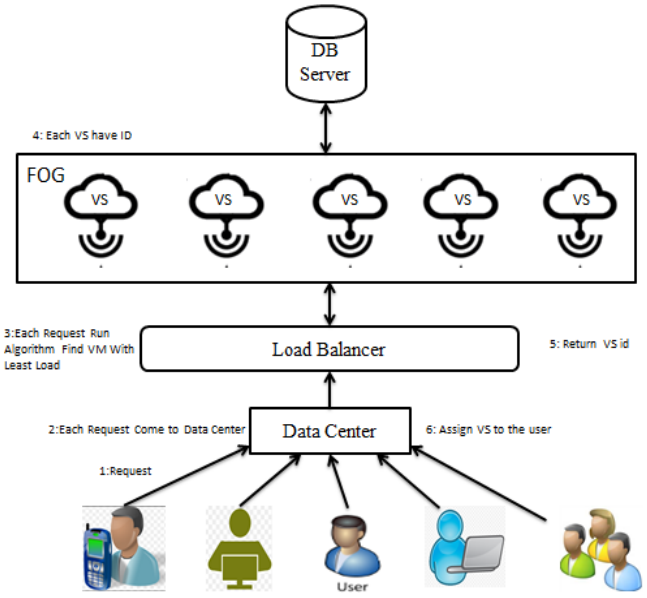

Fig.1. Architecture Diagram

In this methodology contain three models like student model, faculty model and third is fog server. In student model the student are available on different distributed location and connect nearest Fog server for performing course registration. Student can log in to the system by using their credentials like email id and password and check the availability of course and concern faculty. The fog server is providing the service to the students and multiple users are directly connected to fog server. In this research work we are creating multiple fog server and that are distributed geographically. Each student first connects any fog server and performs their course registration and each Fog server provides service to the multiple users those are available on various locations. In faculty model they can directly retrieve the list of student are register under them. In this architecture we are reduce the load of cloud using the multiple Fog server and complete tasks within deadline and database are share among the multiple Fog server fulfilling data consistency constraints with the help of multiple server and achieve concurrency which is shown in Figure 1.

B. *Course Registration System*

Every student has a one of a kind unique email id and a profile details. The profile contains first names, last name, sexual department, gender, username, password, and may have a profile picture. You can likewise include other important information. Every faculty has a one of a kind unique ID and a profile details. The profile must demonstrate the faculty department. A faculty may be work at one or more than one department. A department has a unique department name, department id. Each course has a multiple faculty, course number and other data you believe is necessary. We are keeping up every one of the courses a student has already enrolled or taken. This is utilized to check course essentials when registration. Student may login with their credentials like email id and passwords to the system to register courses and also to retrieve list of courses they have already registered under some faculty. Faculty may login with their credentials like username and password to the system and retrieve list of students are registered under them. A student can't register a course if course has exceeds the maximum capacity of limit and the course exceeds the maximum capacity of the faculty.
Following section contains the flow diagram of course registration process.

C. *Flow Diagram of Course Registration System*

In this section contains the flow of registering the multiple courses by using the student course registration portal. Student can login to the system by using their email id and password any check the list of availability of course with faculty details. Student can select department and check concern faculty is available or not, if not available then choose another faculty and allocate the course. Student can registered only one course at a time this process can be performed simultaneously by multiple students at same time with different geographical location. At the end of it shows

the grid view (timetable) which shows list of registered courses.

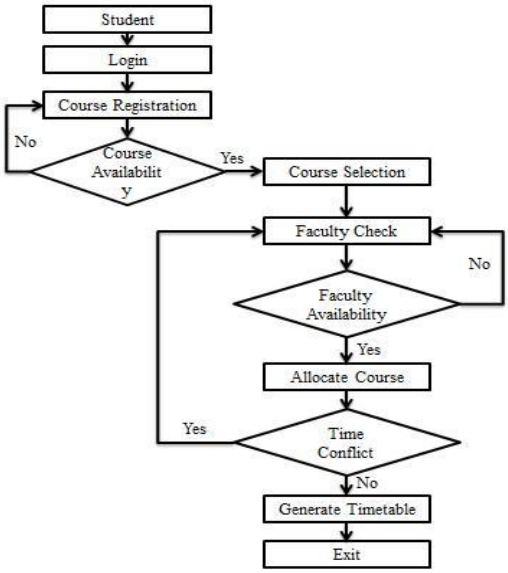

Figure.2.Flow Diagram of Course Registration

D. Proposed Task Scheduling Load Balancing Algorithm

In this proposed algorithm our main aim to find a virtual server (VS) i.e. fog server which has least load and allocate it to the current request The algorithm and flow of proposed algorithm shown in flowing figure.

Algorithm

Input: Number of incoming request R1,R2,R3……Rn
Available Virtual Server VS1, VS2, VS3,….VMn

Output: All incoming request R1,R2,R3……Rn are assign to least loaded virtual server

1. Initially all virtual server are free
2. Each request come to the Data Center it pass the request to the Load Balancer
3. Each virtual server have unique virtual server ID
4. The load balancer is find the least load on virtual server

Case: If it found

i. Check whether the chosen virtual server is not used by last incoming request

If **YES**

go to step 4 and find least virtual server

If **No**

least loaded virtual server is chosen

5. The load balancer return the virtual server Id to the data center
6. The incoming request assign to chosen virtual server

Figure.3.Proposed Algorithm

In this algorithm there are n number of student are available they send request and incoming request managed by central controller. The controller send request to the load balancer for assigns such virtual sever which have fewer loads. There are n number of virtual server are available and each virtual server have unique id. The request that is send by load balancer if it finds fewer loaded VS then it check the selected VS is not used by last incoming request if not, then assign such virtual to the current student otherwise it again find another virtual server. This process is performed continuously until we are not getting least loaded virtual server.

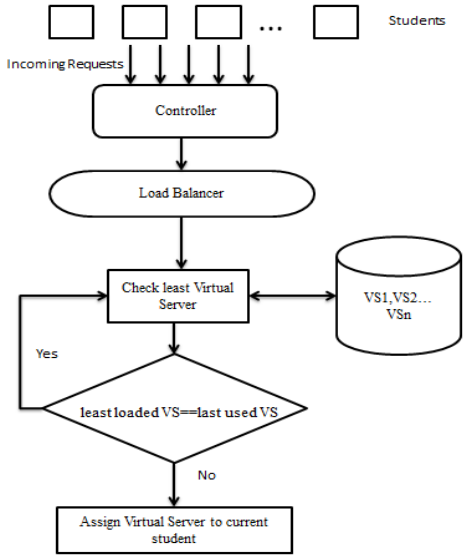

Figure.4. Flow of Proposed Algorithm

E. Pseudo Code

In the Following figure shows the pseudo code of proposed load balancing algorithm.

Pseudo Code of Proposed Algorithm

Input: Number of incoming request denoted by *Ii*
Available Virtual Server *VSn*

Output: Assign least loaded *VS* to each request

1. Initially d= 0; which has VS with zero allocation
2. For each request *j=1* to *n do*
3. For Each *j* run load balancer();
4. Find the Least load *VS* from All *VSn*;
   1. If selected *VS* is not = last incoming request *do*
      Allocate *Ij* to the *VSd*
      Else
      Finding next least load VS;
      d=d+1;
5. *End if*
6. *End for*
7. *End for*

Figure.5. Pseudo Code

V. SYSTEM DESIGN

A. Functinal Structure

In course registration system the first phase is selection of multiple courses, drop the course, view the total courses that are selected etc. Student can select course as first come first serve order. In second phase the faculty can see which students that are registered under her/ him. The faculty can

show the list of student registered using grid view. These implementations are done using angular2, JavaScript, PHP, HTML & CSS.

B. User Interface

In this section we design the two UI, one for student and another for faculty. In student user interface we design the system such a way that it can run in any mobile devices and browser. In this user interface student has to select the department and then choose the subject and type of subject like theory or theory plus practical. The student can select multiple courses but then can't select same course with different faculty. In second user interface for faculty, we design UI such way that faculty can see the lists of student are registered under her/him. In this way, the student and faculty user interface are designed.

C. Database Design

In database design schema we use MYSQL database for our real-time application. The database is store in server which is common to all virtual servers. The multiple users are connected to fog server which are near to them and perform their registration but the database is same for all servers so that it satisfies the date consistency and all operation are performed simultaneously.

VI. COMPARISION AND ANALYSIS

In this section we discuss some of the existing algorithm and compared with proposed algorithm. There are different types of load balancing algorithm are available that are used to distribute the work load into different server in the cloud environment. In any case, most regularly only one algorithm are use at a time. In [16] author proposed Round Robin algorithm, in this approach all the process are divided into processor and each process is assign to processor in the round robin fashion. The processing time of each job is not same for multiple processes so there is chance of some node are ideal and some nodes have more load. In this [17] researchers represent Opportunistic Load Balancing algorithm, it is mainly used for available node should not free at any time it continuously busy. In this algorithm, it takes task and assign to any random virtual machine. The processing time of this algorithm is very slow and current execution of virtual node is not calculated. In [18] author proposed Min-Min Load Balancing Algorithm (MMLB), it first finds such job which required minimum execution time and allocates such job to processor for complete operation. The job required maximum execution time is waiting in queue until processor is not ideal. This algorithm is good in case job having less execution time but main drawback is that starvation. In [18] author proposed Max-Min Load Balancing algorithm, it same as MMLB it first select such task that required maximum execution time. The job with minimum time is completed only once and remaining task that are in queue that are allocate to the ideal processor. It gives the better performance but some jobs are required more time that are kept in wait for more time. In this research work we proposed new task scheduling algorithm that overcome all disadvantages of existing algorithm. This algorithm can give better result and each and every task is assign to virtual server. The task required less or more execution time that is directly assigned to virtual machine so it overcomes the problem of starvation. In this algorithm the multiple request are tasks that first arrive to central controller and each tasks is assigned to such virtual server which has less load so that it give better execution results. The following graph represents the execution time and response time of existing and proposed algorithm.

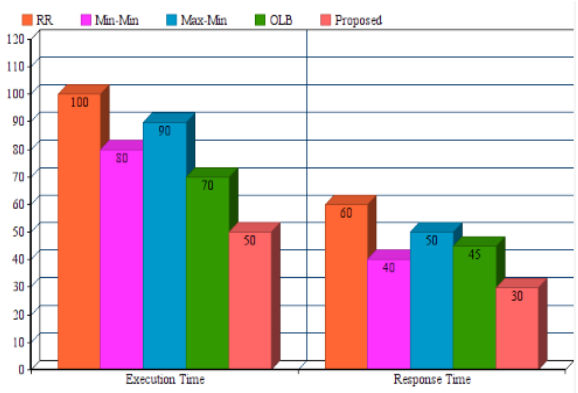

Figure.6. Comparison Graph

VII. RESULTS AND DISCUSSION

A. Home Page

It is the image in which the presence of the home page of our course registration system site.

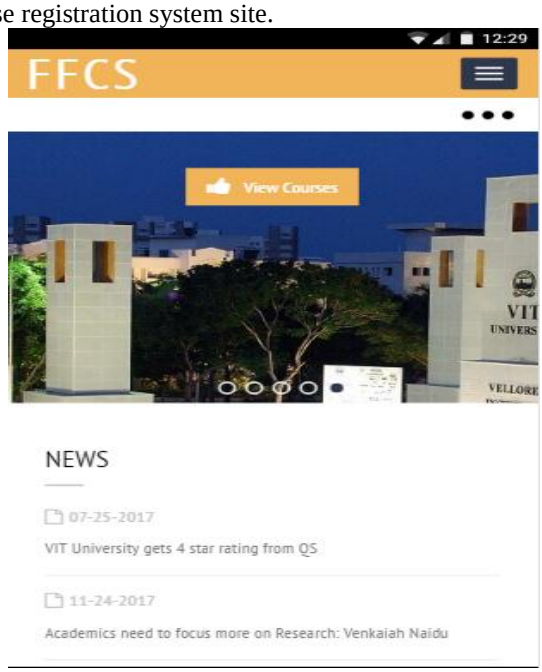

Figure.7. Home Page

B. *Login page*

It is the image in which student can fill their credentials for sign in to the system.

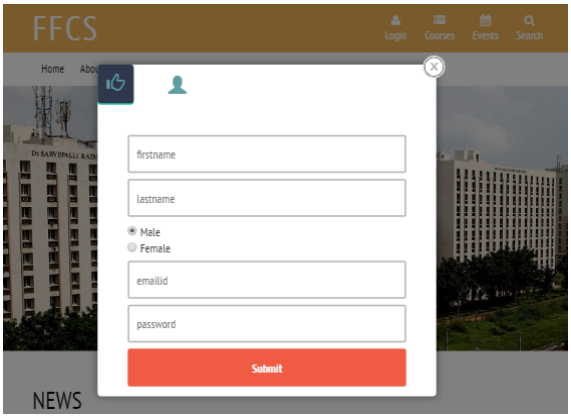

Figure.8. Student Sign in Page

C. *Student Home Page*

It is image in which the presence of the student home page which contains user profile, faculty details, course registration portal etc. of our course registration system.

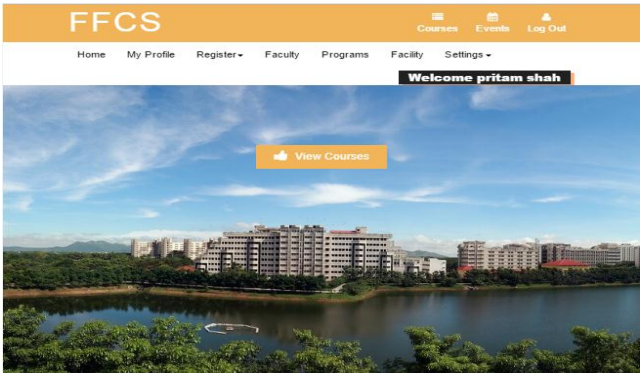

Figure.9. Student Home Page

D. *Course Details*

It is the image it contains course details like course ID, duration, intake and available vacancy etc.

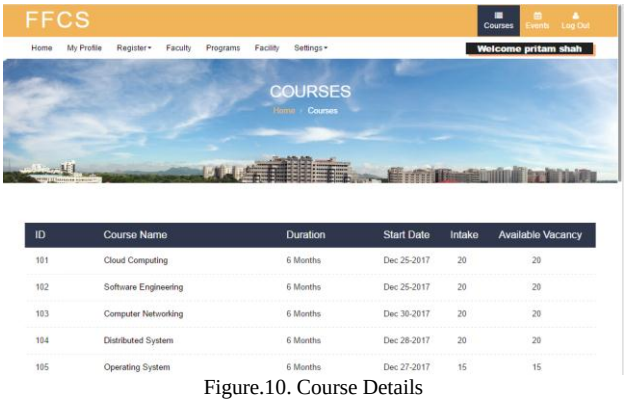

Figure.10. Course Details

E. *Course Registration Portal*

It is the image of course registration portal in which student has to select department, course, concern faculty and type of course in the dropdown for registering the multiple courses.

Figure.11. Course Registration Portal

F. *View Courses and Drop Courses Portal*

It is the image of list of courses student has registered which is shown like grid view and also it contains the drop option that are used to drop any of course that is already registered by students.

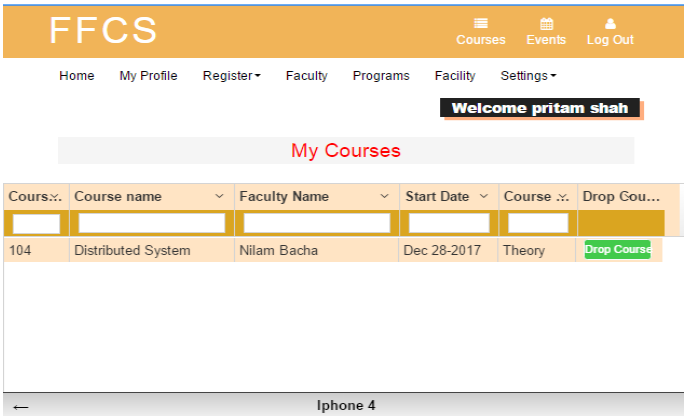

Fig.12. View Courses and Drop Courses Portal

VIII. CONCLUSION

In this paper, we have briefly introduced fog computing and Difference between cloud, fog and mobile computing. In this work we have designed and discussed the model that shows how distributed concurrent models works in Fog when multiple request that are send by client. We have design and implemented prototyping model in fog computing using Course Registration System Application. Our proposed approach is use to complete tasks within deadline, efficient resource utilizations, fulfilling data consistency constraints.

ACKNOWLEDGMENT

The author of this paper would like to thanks Asst. Prof. Padma Priya R. and SCOPE, Vellore Institute of Technology for providing us the resources and infrastructure to carry out this research work.

REFERENCES

[1] P. Jakovits, S. N. Srirama, I. Kromonov, "*Stratus: A Distributed Computing Framework for Scientific Simulations on the Cloud*", 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, pp. 1053-1059, 2012.

[2] P. Morakos, A. Meliones, "*Design and implementation of a Cloud SaaS framework for Multi-Tenant applications*", The 5th International Conference on Information Intelligence Systems and Applications, Chania, pp. 273-278, 2014.

[3] L. Ferretti, C. Michele, M. Mirco, "*Distributed, concurrent, and independent access to encrypted cloud databases*." IEEE transactions on parallel and distributed systems, Vol.25, Issue.2, pp.437-446, 2014.

[4] HS Mahalle, PR Kaveri, V Chavan, "*LoadBalancing On Cloud Data Centres*" in International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3, Issue.1, pp.1-9, 2013.

[5] Rachna Jain, Sushila Madan, Bindu Garg, "*Analyzing Various Existing Security Techniques to Secure Data Access in Cloud Environment*", International Journal of Computer Sciences and Engineering, Vol.3, Issue.1, pp.130-135, 2015.

[6] R.S Sajjan, R.Y. Biradar, "*Load Balancing and its Algorithms in Cloud Computing: A Survey*", International Journal of Computer Sciences and Engineering, Vol.5, Issue.1, pp.95-100, 2017.

[7] R.P. Steffi, S. Sundaramoorthy , "*Top-K Spatial Preference Query with Range Based Skyline Query in Mobile Environment*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.3, pp.46-50, 2014.

[8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica., "*A view of cloud computing*", Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[9] L.F. Bittencourt, E.R.M. Madeira, N.L.S. Da, "*Scheduling in hybrid clouds*",IEEE Communications Magazine, vol.50, no.9, pp. 42-47, 2012.

[10] Pranay Kujur, Kiran Gautam, "*Smart Interaction of Object on Internet of Things*", International Journal of Computer Sciences and Engineering, Vol.3, Issue.2, pp.15-19, 2015.

[11] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, "*The case for vm-based cloudlets in mobile computing*", IEEE Pervasive Computing, Vol.8, No.4, pp. 14-23, 2009.

[12] Y. Zhang, Y. Zhou, "*Transparent Computing: a New Paradigm for Pervasive Computing*", in Ubiquitous Intelligence and Computing, Berlin, pp. 1-11, 2006.

[13] M. S. Laghari, "*Automated Course Advising System*", International Journal of Machine Learning and Computing, Vol.4, No.1, pp.23-31, 2014.

[14] Z. Xu, X. Hou, J. Sun, "*Ant algorithm-based task scheduling in grid computing", thm-based task scheduling in grid computing*", presented at Electrical and Computer Engineering IEEE CCECE, CanadA, pp. 1107- 1110, 2003.

[15] L. Chen, "*Resource Scheduling Based on Ant Colony Optimization Algorithm in Grid Computing Environments*", Information Technology Journal, Vol. 12, No. 24, pp. 8010-8014, 2013.

[16] V.R.T. Kanakala, K.P. Kumar, S. Kavitha, "*A Hybrid Heuristic Algorithm to Enhance Load balancing in Cloud Environment*", International Journal of Computer Sciences and Engineering, Vol.4, Issue.11, pp.129-132, 2016.

[17] S. Kashyap, V. Jaydeep, "*A Survey Of Various Load Balancing Algorithms In Cloud Computing*", International Journal Of Scientific & Technology Research vol. 3, no. 11, pp.115-119, 2014.

[18] R. Bhaskar, "*Dynamic Allocation Method For Efficient Load Balancing In Virtual Machines For Cloud Computing Environment*", Advanced Computing: An International Journal, vol. 3, no. 5, pp. 53-61, 2012.

Authors Profile

Shaha Pritam Prashant is currently a second year post graduate Computer Science and Engineering Student in Vellore, India at the Vellore Institute of Technology. He will complete his post graduate in 2017 with a Master of Computer Science and Engineering. He has completed graduation from Mathoshri Prathishthan School of Engineering, SRT Marathwada University, Nanded, India. His research interest includes Big Data Analytics, Mobile Computing, Data Mining, Cloud Computing, Language Processing, Computer Networks, Semantic learning.

R. Padma Priya is currently working as assistant professor, in VIT University. She has published papers in reputed journals. She is a lifetime member of CSI. She has 14 years of teaching experience. She has completed Bachelor of Engineering from Madras University in the year 2001 and Master of Engineering in computer science from Anna University in the year 2012. She is a University rank holder and is a gold medallist of her college in ME computer Science. Her area of interest is parallel and distributed computing, optimization, mobile computing, and convergence of RFID, IoT and cloud technologies for smart cities.

