Ameliorated Analysis of Tradeoff and Round Robin in Mist Edge and Cloud Environments

Srinidhi Hiriyannaiah, Siddesh G. Matt[2], Shusmitha S[3]and Srinivasa K.GopalaIyengar[4]

[1,3]Department of Computer Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, India
[2]Department of Information Science and Engineering, M.S. Ramaiah Institute of Technology, Bangalore, India
[4]NITTTR, Chandigarh

ABSTRACT

Edge computing is the fastest growing technology in the world today, and it has clearly demonstrated to be the most optimal business model for all company requirements as the number of portable sensing devices grows tremendously. To deal with this, the technique of load balancing is required to reduce overhead and increase cloud throughput. With the use of a pure edge simulation, we examine the present execution load and performance in terms of average reaction time, hourly data centre response time, and Virtual Machine (VM) cost, among other metrics. Pure edge simulation model is the highest quality simulation for algorithm testing in a cloud infrastructure out of all the simulators. As the number of users grows, simulation findings show that round robin outperforms these methods.

Keywords

Fog computing, Cloud, Load balancing, Tradeoff

1. INTRODUCTION

The fast adoption of cloud computing, mobile broadband networks, and the Internet of Things (IoT) has resulted in increased demand for networking allocation of resources, data processing, and service management, resulting in a shift in traditional communication network. Figure 1 illustrates the idea of edge computing. Between IoT devices and the cloud, the edge network exists. Edge computing provides an extensive cloud computing methodology to the network edge in order to address the needs of IoT applications that require low latency and high compute. From the cloud datacentre to the CPE or micro data centre, the distributed architecture of edge computing nodes meets the computational needs of a variety of applications, data, and services. The edge computing network can be separated into cloudlets, mobile edge computing (MEC), and edge computing based on distinct domains of development. Cloudlets have been communication technologies that deliver micro services to the networking around the user and virtualize and compress computing resources so that they can be deployed closer to the user's end. Akamai and Microsoft are currently promoting cloudlet application technologies and services. MEC was developed by the European Telecommunications Standards Institute (ETSI) and is used and controlled by communication firms in the sphere of communication technology. This technology intends to assist mobile carriers build a unique mobile service model by reducing the increasing load on network equipment. Edge computing is a notion of enhanced cloud technology that focuses on the data processing function of the local network , which was first proposed by Cisco and is now pushed by the OpenEdge Consortium Alliance. Edge computing can be utilised in various networking gear, for personalized or company administration, and for offering relevant Iot applications in specific locations by

International Conference on Emerging Technologies: AI, IoT, and CPS for Science & Technology Applications, September 06–07, 2021, NITTTR Chandigarh, India

EMAIL:sushsri1496@gmail.com (A. 1); srinidhi.hiriyannaiah@gmail.com (A. 2); siddeshgm@gmail.com (A. 3); kgsrinivasa@gmail.com (A. 4)
ORCID: 0000-0002-9702-1603 (A. 1); 0000-0003-2304-5087 (A. 2); 0000-0025-6818-51X (A.31); 0000-0003-1022-8431(A. 4)

focusing approaching data exchange.

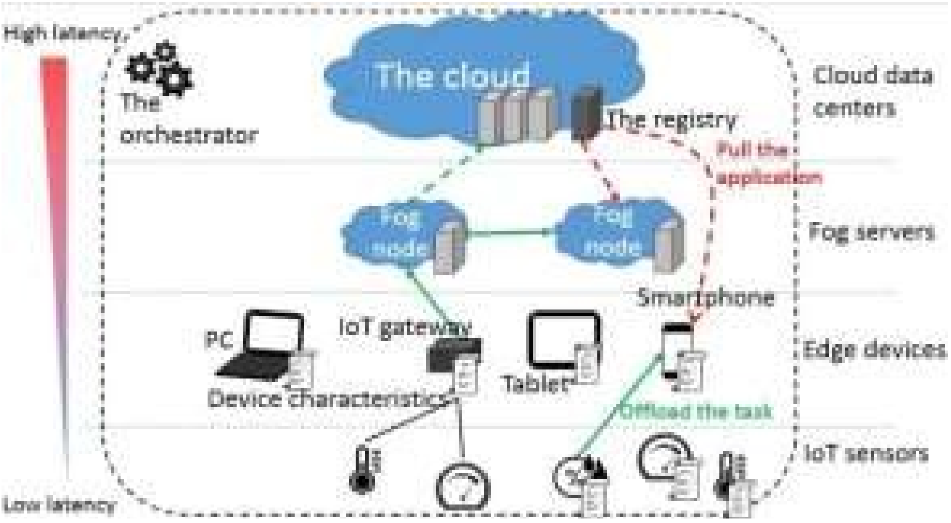

Figure1: Proposed architecture

2. RELATED WORK

We need to know about certain performance matrices of almost any load balancing algorithms in order to pick one for load balancing, and we need to compare algorithms based on respective performance matrices. Numerous studies, such as Volkova et al. [1], Muhammad et al. [2], and Pawan Kumar et al. [3], focus on load balancing.

Marco Dorigo [13] presented a load balancing system based on an ant colony, in which the resource provisioning node is discovered at the start of the search. Ant Colony Optimization (ACO) is a population-based technique to tackling combinatorial optimization problems that is inspired by ant foraging behaviour and their innate ability to locate the shortest path between a food source and their nest. ACO was inspired by the cooperative foraging activity of ant colonies.

Volkova et al. [1] study three load balancing algorithms, namely Round Robin, Throttled, and Active load balancing algorithms, and explain the study utilizing cloud analyzer with operating conditions such as arrival rate, overtime cloud centre reaction time, and so on. [14], after consulting with a cloud expert, concluded that Throttled is the best of the bunch, and therefore that Throttled has a faster response time than Round Robin and Active load balancing.

Again an author, Muhammad et al. [2,] focused on a certain algorithm analysis and performed the study for Round Robin, Widely Spreading Existing Load balancing. They ran simulations for a few clouds users and found that Round Robin is the better decision. Patel et al.[4] analyses load balancing techniques for various performance matrices and explains how they operate, stating that the efficiency of any technique is dependent on several aspects such as response time, higher processing, and many more. Using simulator, it was discovered that the performance of the proposed method alternative load balancing algorithms is influenced by a variety of performance parameters. From the perspective of the aforementioned researcher, we can infer that no one can assert that this algorithm is the best without evaluating their design parameters. Sometimes it is the finest in terms of response time, processing, or cost comparison; aside from these variables, some taken to reach is depending on the amount of clients.

3. PROBLEM DEFINITION

An edge computing task is often split among the client's device, edges gateways, and, on rare occasions, a cloud network broker. As a result, one of the most difficult problems in edge computing is selecting where to schedule computational operations. We examine the present execution load and effectiveness in terms of average reaction time, hourly data centres responsiveness, and Virtual Machine (VM) cost, among other metrics with the help of pure edge simulator.

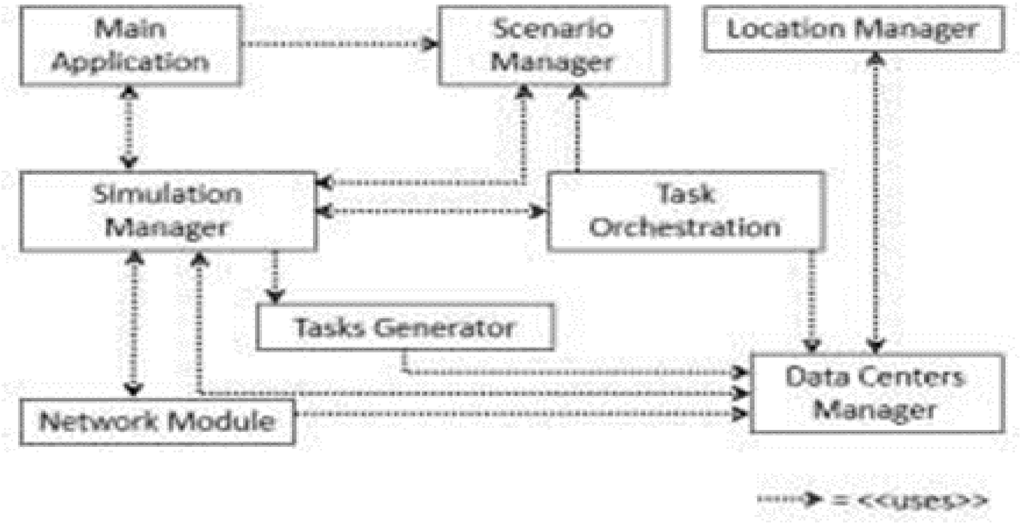

Figure 2: Modules in the proposed model

The proposed architecture contains totally seven modules which are discussed below

- Scenario Manager, which reads the data input (.xml and.prop files in the /settings/ folder) and imports the simulation parameters and user situation. It is made up of two classes: the File Parser, which verifies input files and loaded simulation parameters, and the Simulation Parameters, which is a placeholder for the various parameters.

- The Simulation Manager is the programme that starts the simulation, schedules all of the events, and generates the result. The Simulation Manager class, which administers the simulation, organizes the task creation, and so on, is one of the most significant classes. The Simulation Logger class records the simulation output in comma-separated value (CSV) format so that it can be easily used in any spreadsheet editors later (e.g., Microsoft Excel...).

- Data Centers Manager is a program that creates and maintains data centres and devices (i.e., Cloud, Edge or Mist). It comprises of two groups: the Data Center class, which contains the specific qualities of Edge devices such as position, movement, source of energy, and density energy if it is rechargeable batteries; and the Edge Device class, which contains the specific characteristics of Edge devices such as destination, flexibility, form of energy, and storage resources if it is rechargeable batteries. The Server Manager is the second class, which creates the required server and Edge devices, as well as their administrators and virtualization software.

- Tasks Generator, which is responsible for task generation, currently allocates an implementation to each Edge device, such as e-health, resourceful, or improved (as stated in the settings/applications.xml file). Then, according to the supplied type, it generates the required jobs, ensuring application heterogeneity.

- The Network Module, which is mostly comprised of the Network Model class.It is responsible for the task container request transfer.

- The Tasks Orchestrator, which serves as a decision maker and allows the user to specify the orchestration algorithm..

3.1. Working Mechanism of Round Robin in Edge

Client requests are distributed over a collection of servers via round robin load balancing. Each server receives a client request in turn. The load balancer is instructed to return to the top of the list and continue the process.

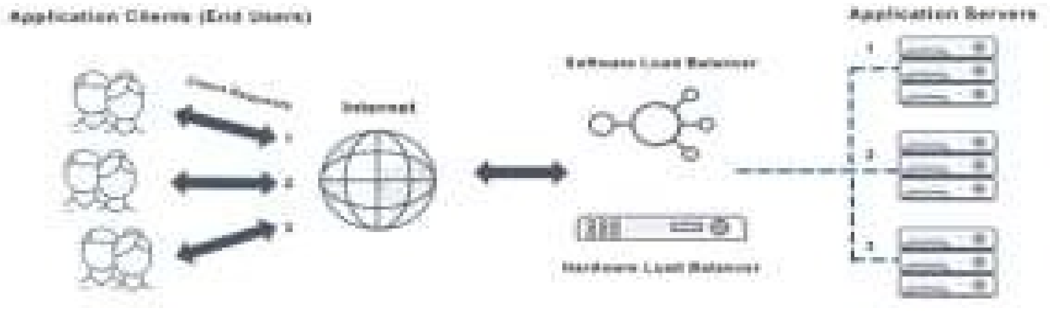

Figure 3: Round robin Architecture

A deliberate version of the first-come, first-serve CPU scheduling algorithm is round robin scheduling. Processes are queued in round robin scheduling in a first-in-first-out attempt, but each method is only allowed to operate for a certain amount of time. A time-slice or quantum is the term for this time span. It is then pushed to the bottom of the run queue, where it must wait for all of the other procedures in the queue to complete their cycle through the Processor. Round robin is a load balancing algorithm that distributes user request around a centralized server in a simplified way. Each server receives an user requests in turn. The load balancer is instructed to return to the top of the list and the process is repeated. Round robin is now the most extensively used load balancing technique because it is simple to create and understand. Client requests are cyclically forwarded to accessible services using this mechanism.When servers have about similar storage and processing capacities, round robin server load balancing works the best.

Advantages: Each model is described an equivalent amount of CPU time. This significantly reduces estimated arrival timings as compared to non-preemptive schedulers. This operational system is designed to ensure that it would workout throughout all illustrates the interaction by constraining a process to a specified period of time. Disadvantages: It's not always a great way to give each technique an equal portion of the Processor. Highly engaging processes, for example, will still not be programmed anything more regularly than Computation procedures. Client requests are distributed over a collection of machines via round robin load balancing. Every server receives an user requests in sequence. The load balancer is instructed to return to the number one spot and the process is repeated.
Trade -Off Algorithms

A tradeoff occurs when one item rises while the other lowers. It is a method of solving a problem with either less period and with greater space, or in very little time and with very little storage.This Method aids in the solution of a problem that consumes less storage and generates output in less time. However, meeting both of these requirements at the same period is sometimes not attainable. Some other option is to compute the solutions, which takes a long time but requires very minimum storage.

As a result, the much more space- efficient techniques you have, the less space-efficient they are. In a trade-off, several types of space-time exist.Compressed or Uncompressed data :This issue of information storage can indeed be solved via a space-time trade-off. Uncompressed information takes more energy but consumes less effort to store. However, if the data is compacted, the decompress procedure requires less maintenance but more time to perform. It is feasible to operate effectively with compressed data in a variety of situations. In the situation of compressed bitmap indices, where working with reduction is simpler than working without the other.

4. Performance comparison RR Vs Tradeoff: Experimental Evaluation

Edge and Mist (Extreme Edge) computing are two new computational paradigms that try to solve the limits of Cloud technology by putting operations to the network's edge. As a result, both latencies and Computing demand are reduced, resulting in a more scalable network.. Nonetheless, several difficulties, such as resource management methods, must be resolved in these distributed systems where multiple devices must offload their jobs to one another (either to lengthen their lifetime or to reduce job completion latency). Instead of testing them on a real distributed system, simulation allows for a reliable, controlled, and resource evaluation of suggested approaches and methodologies prior to their implementing.

4.1. Final Result difference between the Algorithms

In the edge, round robin performance is superior. Figures 4–8 demonstrate the performing findings for 100, 200, and 300 mobile devices, respectively. From all three settings, the task wait period (Figure 4a) of the suggested RR load balancing methodology is the least (24 ms), whereas the task wait period of the other methodology is 54 ms. The MIST has a long wait time since several jobs are attempting to reach the centralized cloud service simultaneously time..Thecpu Utilization use of virtualized on edge servers reveals how effectively edge servers are being used. When there are 300 mobile devices, Figure 4 depicts the cpu Utilization consumption of virtual servers on network edge. Because the Tradeoff chooses to use the central cloud server, the average CPU use of virtualization on edge servers is relatively low (0.13 percent), compared to 2.9 percent for the Round Robin and 2.14 percent for the Tradeoff. This demonstrates that the suggested load balancing strategy produces good use of edge servers than previous strategies. Figure 8 depicts the number of jobs successfully completed on edge servers. The suggested methodology, predictably, produces good outcomes, whereas the Tradeoff yields the worst. As a result, the Trade-off has the maximum number of jobs successfully performed on the central server (as shown in the figure 8). To put it another way, the Trade- off doesn't use edge servers and creates more internet traffic for transferring jobs.

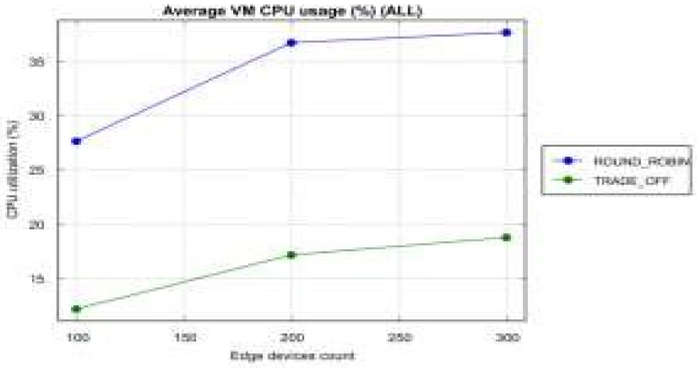

Figure 4: Average CPU Usage in All Env

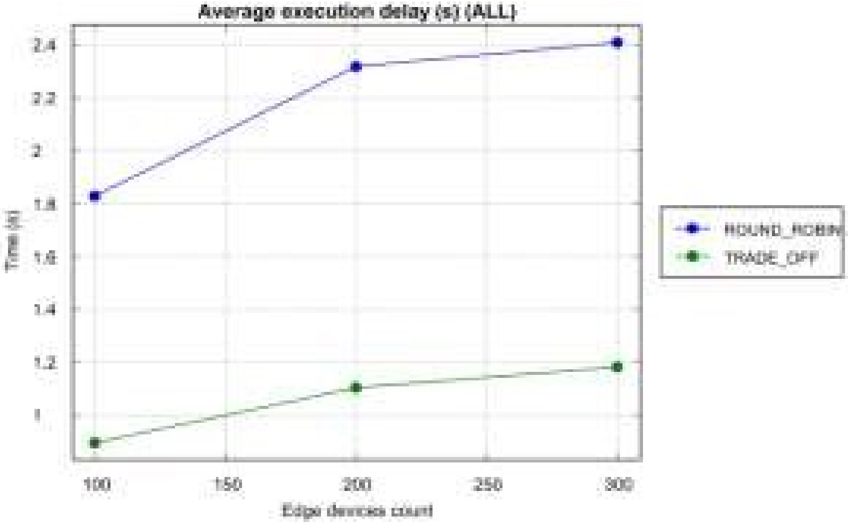

Figure 5: Average Execution delay in All Env

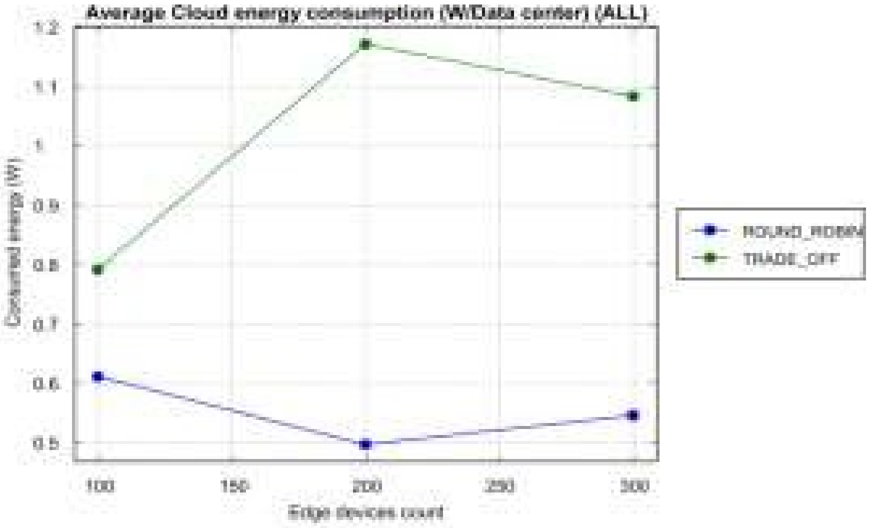

Figure 6: Average Energy in All Env

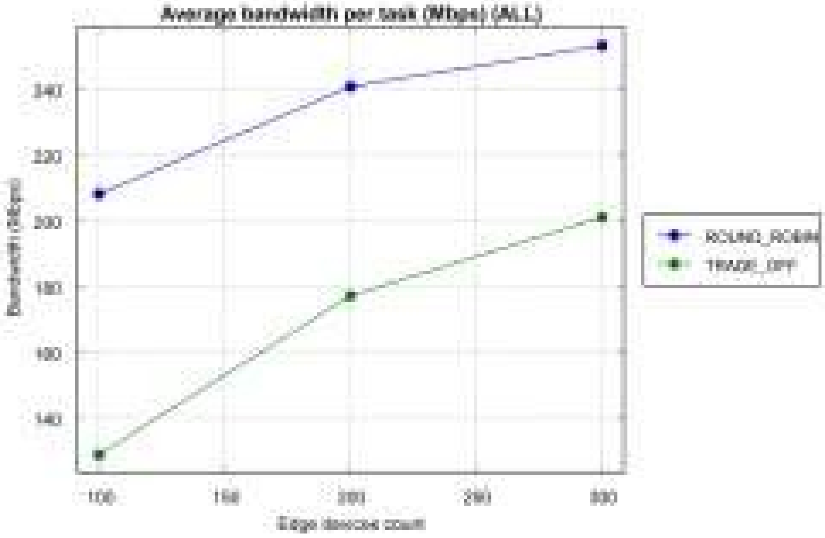

Figure 7: Average bandwidth per task in All Env

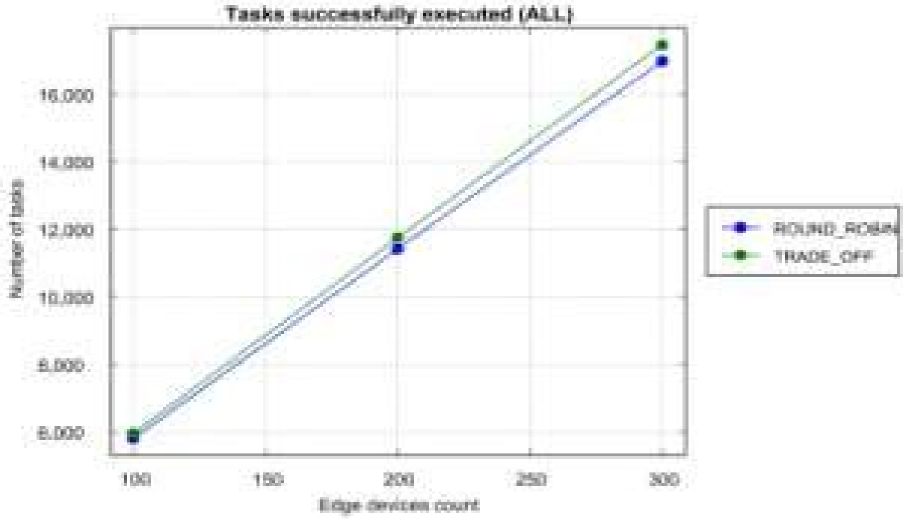

Figure 8: Task execution in All Env.

5. CONCLUSION

To provide scalability to data centres and servers is a major challenge, and past studies in edge cloud computing environments have suggested a few load balancing solutions. We investigated a load balancing algorithm that effectively distributes offloaded activities from a hot spot to neighbouring edge servers among round robin and tradeoff algorithms in order to implement a scalable load balancing strategy into edge cloud computing settings. The round robin approach outperforms tradeoff techniques and boosts the average CPU consumption of virtual machines, indicating a high utilization of edge servers, according to the data. When compared to other strategies used in prior studies, the round robin technique creates the least network traffic, resulting in a lower risk of network failure. Round robin makes good use of both edge servers and the central cloud server when it comes to offloading and executing tasks.

6. REFERENCES

[1] Violetta N. Volkova, Liudmila V. Chernenkaya, Elena N. Desyatirikova, MoussaHajali, AlmothanaKhodar, Alkaadi Osama, “Load Balancing incloud Computing ” 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) IEEE,2018.

[2] Muhammad SohaibShakir and Engr. Abdul Razzaque,” Performance Comparision of load balancing algorithm using cloud Analyst in Cloud Computing”, 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) IEEE,2017.

[3] Pawan Kumar and RakeshKumar ,”Issues and Challenges of Load Balancing techniques in cloud Computing : A Survey”, ACM Computingsurveys,Vol.51,no.6,2019.

[4] SandeepPatel,Ritesh Patel ,Hetal Patel and SeemaVahora,CloudAnalyst : A Survey of Load Balancing Policies, International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 21,May2015.

[5] Calheiros R.N. CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, Eprint: Australia, 2009,pp.9–17.

[6] Alakeel, A.M., A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security,2010. 10(6): p.153-160.

[7] Khiyaita, A., et al. Load balancing cloud computing: state of art. in Network Security and Systems (JNS2), IEEE,2012.

[8] Nuaimi, K.A., et al. A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. in Network Cloud Computing andApplications (NCCA), 2012 Second Symposium on, IEEE,2012.

[9] Alakeel, A.M., A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security, 2010.10(6): p.153-160.

[10] Simar P.S. , Anju S. and Rajesh K. Analysis of load balancing algorithms using cloud analyst, International Journal of Grid and Distributed Computing, vol. 9, No. 9, 2016, pp.11- 2.

[11] A. A. Jaiswal, Dr. Sanjeev Jain," An Approach towards the Dynamic Load Management Techniques in Cloud Computing Environment", International Conference on Power, Automation and Communication (INPAC),2015.

[12] Surbhi Kapoor, Dr. Chetna Dabas," Cluster Based Load Balancing in Cloud Computing", Eighth International Conference on Contemporary Computing (IC3),2015.

[13] Marco Dorigo, Christian Blum, "Ant colony optimization theory- A Survey", Theoretical Computer Science344,pp.243 – 278, Elsevier2005.