Implementation of Improved Multi-Queue Job Scheduling Algorithm (IMQJSA) for Load Balancing in Cloud Computing

Inderpal Singh
Department of Computer Science &Engineering
CKD Institute of Management and Technology, Taran

ABSTRACT
In this research paper, authors describe how to manage number of tasks simultaneously at a same time on virtual machines on cloud by utilizing enhanced MQS (Multi-Queue Scheduling). This proposed enhanced multi-queue scheduling method is works on Fuzzy Logic. The main significance to utilize this proposed method on fuzzy logic is to achieve maximum accuracy in results. This enhanced multi-queue scheduling method on fuzzy logic at first grouped the number of jobs according to burst time calculated by some formula. Membership Function plays an important role in enhanced multi-queue job scheduling. It gives more efficient results with decreased execution time as well as overhead. This new designed methodology helps organizations or enterprises to reduce the burden of load balancing when run their applications on cloud. Hence, the use of fuzzy logic in enhanced multi-queue job scheduling is an effective way to handle simultaneous jobs at a same time that also helpful to reduce overhead and provide live

Keywords
Cloud computing, Load balancing parameters and techniques, Membership function, Load, MQS scheduling, cloud server, throughput, and execution time.

1. INTRODUCTION
Cloud computing term is used for the delivery of hosted services over the internet. The main advantage to utilize this technology is to pay as you go. On the basis of rental charges user will use cloud service by demanding from the cloud service provider. The dependency of cloud computing is based on the main three parameters viz. abstraction, encapsulation and isolation of the resources on the demand basis. Several machines are in general connected with cloud data centers and the distributed nature approach is used by implementing any cloud strategy on cloud servers. Majorly, grid computing is work behind it. The main purpose to utilize this is it consumes less power of energy when especially operates on a large scale and can be easily managed by the utilization of centralized server approach. That ultimately results show the overall energy saving [3]. There are some important factors studied by the authors who are very important and more helpful to create an ambiance for cloud. For implementing ambiance MATLAB tools are used by the authors in this research paper. In this research paper, authors considered main two things first one is effective job scheduling[4] when come in multi-queue at a same time on cloud server and the second one is proper utilization of resources. This paper proposes an enhanced multi-queue job scheduling strategy that helps to provide a facility of effective job scheduling by load balancing [10] without any delay. The load balancing agreement with ambiance can be shown diagrammatically in figure 1 below:-

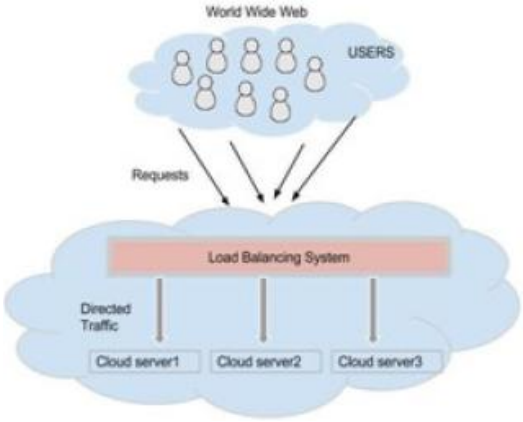

Fig 1: Load balancing arrangement in cloud ambiance [12].

In general there are various parameters in which load balancing depends that can be listed up below:-

- ✓ **Associated Overhead**: It is the overhead while implementing the load balancing scheme and contains the motion of jobs, communication between processes and between processors. If it has less value, then it is good for system.
- ✓ **Throughput**: It is measured by total jobs that finished their running process for some particular time. If it is high for machine, then it is good.
- ✓ **Fault tolerant**: It is the capability to carry out load balancing with help of particular technique in absence of proper path. All effective balancing schemes follow this property.
- ✓ **Migration time**: It is measure that check out the time for sending job from particular machine to another specific machine for executing that job. If it is low, then is it good for machine.
- ✓ **Response time**: It is duration that starts when job has submitted to first response generated. If it is low, then it is good.
- ✓ **Resource Utilization**: It is that which measures the level of utilization of resource. If it is optimal, then it is good.
- ✓ **Performance**: It measures the whole effectiveness of the machine. If every factor is enhanced, then automatically whole system is condition will be better.
- ✓ **Waiting Time**: It is the time which starts when job entered in ready queue, until the execution of job. The

system will be considered as good if its waiting time should be less [4].

On the behalf of some of the suitable parameters the load on the cloud can be easily balanced. These certain parameters may run on some load balancing techniques [9][11][12] that can be listed below:-

- ✓ **Throttled Load Balancing Algorithm:** This scheme is purely centered on virtual device. It involves two steps: Firstly, User calls load balancer for checking appropriate virtual device which handle that load straightforwardly. Secondly, carry out the action which is offered by customer [11].
- ✓ **Randomized:** It is a category of static load scheme in which job is commanded by specific node with some probability value p. The sequence for the allotment of the jobs is consistent for every processor which doesn't depend on allocation. This schemes work well if processors have equal load, otherwise it creates problem. It works well when RR scheme create overhead to run queue [13].
- ✓ **Equally Spread Current Execution Algorithm (ESCE):** In this scheme, load balancer maintains constant load on every virtual device which are interlinked to data center. It creates index table of virtual devices and also of no. of requests allotted to virtual device. The table is checked for minimum loaded virtual device, only if request is from the data center to allot the fresh Virtual device. The first detected virtual device will be considered to tackle the requests of customer, then data center controller will get id of virtual device from load balancer. Then, there is communication between data center and virtual device correspond to that id. After that data center checks that table by maximizing the number of allotment to that detected Virtual device. After that when Virtual device finishes the allotted job, then request is send to data center that afterwards told by load balancer. Now load balancer checks that table by minimizing the number of allotments for detected virtual device by 1 and there is also some overhead which is checked in repeated manner [10].

The choice of the cloud user depends on the type of cloud load balancing technique is used while performing job scheduling in cloud computing. If the choice of cloud technique is more efficient then naturally job will take minimum duration of time that ultimately reduces the response time as well as overhead. The main advantage to propose this method is to allocate the resources effectively at right time and reduce the overall overhead burden of cloud server. By implementing this proposed method the load on cloud can be easily managed and performance will be automatically improved on cloud server. In addition, this paper also provides a facility of live migration on cloud server. So that, if any user see or feels there is a huge load now on one cloud server then user will easily migrate job into other cloud server by utilizing this feature of live migration and hence the load can be easily managed. The working of cloud is pay as you go [2]. As studied in the literature by the authors there are various types of services provided by the cloud service providers as an example software, platform, infrastructure, network, storage etc [5][6]. And these services are broadly divided into different-2 sections that can be shown below in figure 2:-

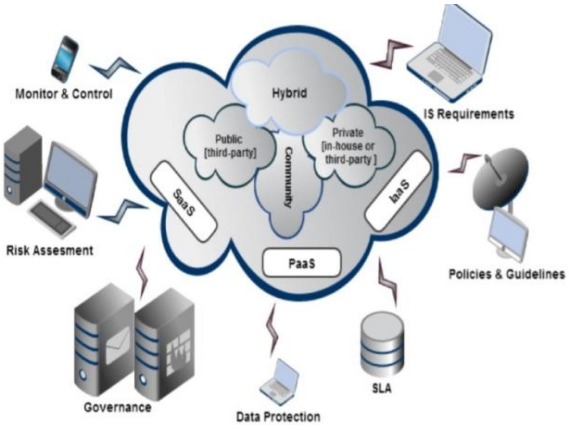

Fig 2: Different Service Models [7].

These different categories of cloud are further divided into different communities or types of cloud viz. public cloud, private cloud, hybrid cloud and green cloud etc [5] [8] [9] of cloud and that ca be diagrammatically shown in figure 3: -

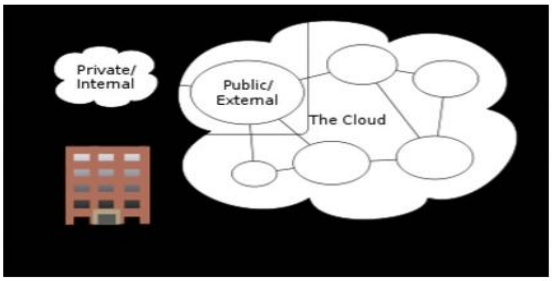

Fig 3: Different types of Cloud [8].

The structure of hybrid cloud can be separately diagrammatically represented as below given figure number 3:-

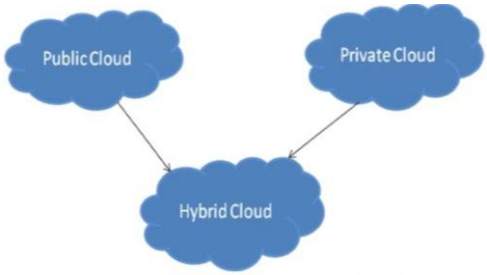

Fig 4: Structure of Hybrid Cloud [9]

2. RELATED WORK

MULTI QUEUE JOB SCHEDULING: In this scheme tasks are joined together on the behalf of their duration of execution. The user sends requests which are ordered on the basis of their increasing execution time. Every request is preferred in equal manner. The dynamically allotment of tasks is helpful to drop starvation. The queue manager is the backbone for handling and maintaining tasks time to time. [1] It is considered that there are 40% tasks which have small execution time and queued in one slot. Then 40% tasks those have medium execution time and queued in other different slot. At last, 20% tasks are those that have long execution time and are queued in another different slot [1]. Suppose that there are total 10 jobs and then jobs are selected in following manner [1]: There are two iterations i.e. first iteration takes two jobs from first queue in which short jobs present, two jobs

from second queue in which medium length jobs present and one job from third queue in which long jobs present and again second iteration is repeated as like first iteration. The results show the optimal improvement of MQS over traditional schemes that described in fig 6. It improves the utility of the resources and QoS (quality of service) in cloud ambiance. The Cloudsim is used to handle the jobs. The queue manger handles the jobs efficiently. The jobs are assigned with particular tag no. and when this tag no. is similar to scheduled task which indicated that job is completed. The processors are decided on the behalf of tasks. [1]

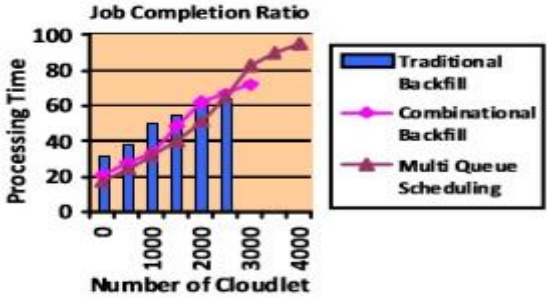

Fig 5: Evaluation of MQS with Traditional backfill and Combinational backfill Scheduling Algorithm [1]

3. FLOWCHART OF PROPOSED ALGORITHM (IMQJSA)

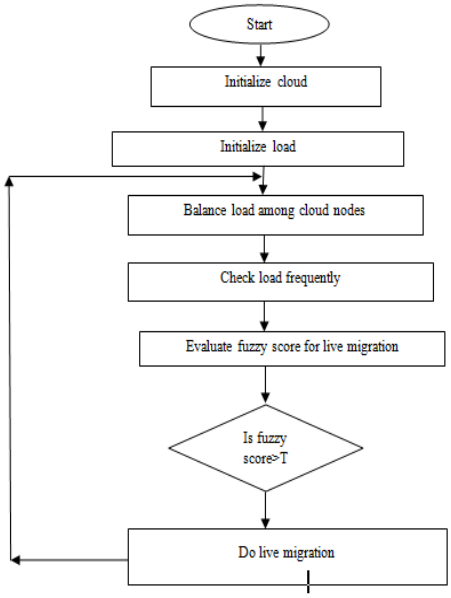

Figure.6: A Roadmap for Improved Multi-Queue Job Scheduling Algorithm (IMQJSA).

4. WORKING OF PROPOSED ALGORITHM

The first step of the proposed work is to initialize the cloud. When cloud is initialized, then after that, initialize the load. The cloud will then send request for server or processor which has highest capacity. The priorities are decided on the basis of speed and power consumption of the servers individually. That server will be considered as a too powerful active resource which consumes less power and has greatest speed. Then after that it is very important to equalize the load on various cloud nodes. After that examine load time to time and check whether which type of load is going to be matured or which load is not. The load which is matured which must be subtracted from the existing load of that particular cloud node or cloud server. Most importantly in next step, examine fuzzy value for transferring jobs from one machine to other. If this value is larger than particular unique value, then transfer jobs from one machine to other. But if this value is less in comparison of threshold value which lead to increase the efficiency, then equalize load on various cloud nodes or cloud servers and then repeat the process. The job waits up to 5 seconds otherwise job will be discarded.

5. IMPLEMENTATION AND RESULT

The proposed scheme is implemented in MATLAB. There is different no. of users and servers which interacts each other through Cloud service provider (CSP). The CSP examine that which resource or server is free and suitable for some particular job. If particular server is found, then CSP allot it to appropriate job. But if job is unable to find effective server, then it starts waiting until to find suitable or effective server. In that period, some other client starts requesting for suitable resource. The server with high precedence will accept request from client. At a time, one client will request for server to CSP.

6. PERFORMANCE EVALUATION

This can be measured on the basis of membership function which is a type of function that displays the values lie between 0to 1 with more accuracy. This can be represented by a straight line.

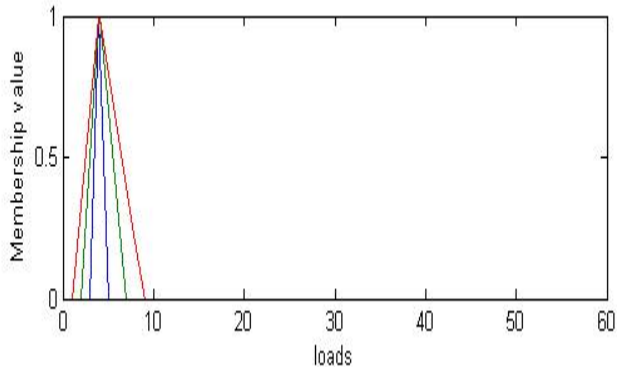

Fig. 7: Graphical Representation of Load versus Membership function.

The figure 7 depicts about the membership function. In this, blue, green and red line depicts three different processors or servers. It checks the load and membership value of different processors. So, red line has excellent degree of membership as compared to blue and green.

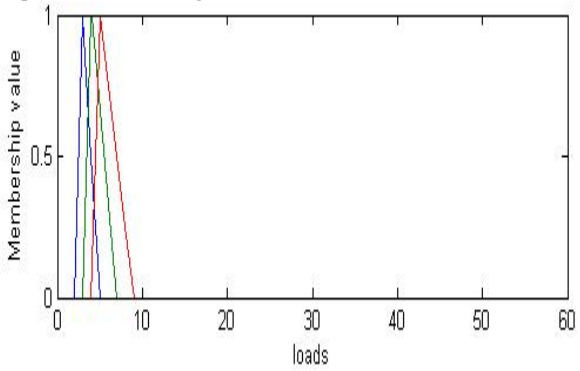

Fig. 8: Graphical Representation of Load versus Membership function.

The figure 7 depicts different processors named as red, green, blue line. It describes connection between different processors and shows that we also allot multiple processors to single job.

Red is in-between of two processors, so preference will be given to red processor or server.

The simulation results of execution time can be shown bin the below given paragraph.

Execution Time: It is the time involving the running of task. Here we use two timers, i.e. tic and toc. The tic initiates the timing of stop watch to check the performance. The tic command helpful to record the internal timingwhile execution. The elapsed time is displayed with help of the toc function.

The table 1 and table 2 shows the simulation time, execution time and throughput values of existing algorithm and proposed algorithm.

Table.1: Nomenclature for Existing and Proposed Scheme Execution Time parameters used in Load Balancing

Sr. No.	Simulation time	Exiting execution time	Proposed Execution time
1	10	2.175865	2.025148
2	20	2.207046	2.153125
3	30	2.171691	2.091139
4	40	2.186675	1.935427
5	50	2.029622	2.003603
6	60	2.114094	2.002846
7	70	2.176513	1.939489
8	80	2.008513	1.876676
9	90	2.293373	2.080493
10	100	2.048529	1.888318
11	110	2.043157	2.007913
12	120	2.187113	1.982048
13	130	2.072555	2.016074
14	140	2.097791	1.931285
15	150	2.112160	2.028744
16	160	2.146514	2.039794
17	170	2.194039	2.001716
18	180	2.130492	1.984401
19	190	2.189482	2.113547
20	200	2.081573	2.034895

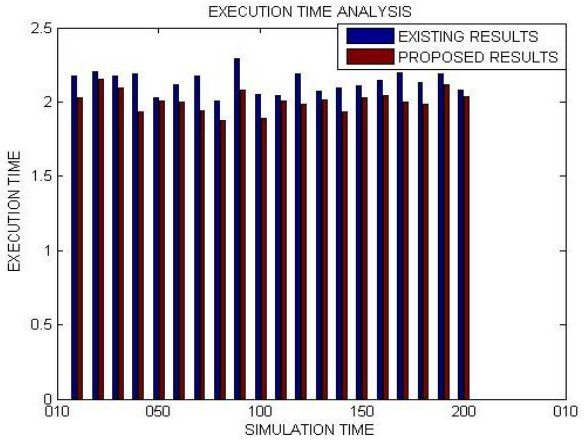

Fig 9: Shows the Execution Time Analysis

Fig 8. Indicates the Execution time of existing and proposed schemes when simulation time is changed. The x-axis depicts the simulation time and y-axis depicts the execution time. It shows that the execution of the proposed algorithm decreases than the existing algorithm. If execution is less means that system operates requests more quickly than the traditional and existing algorithms. The tic, toc timers are used to measure the execution time.

Throughput: It is that metrics which is helpful in tracing the overall tasks that completed their running procedure in some defined time constraints. It is evaluated by:

MEANTHP = sum (TOTALWRK) –

(total_waiting_time / simulation_time)

Table 2: Nomenclature for throughput of existing and proposed schemes with the variation of simulation time.

Sr.No.	Simulation time	Exiting Throughput	Proposed Throughput
1	10	10.822812	18.652653
2	20	11.924694	19.523333
3	30	13.444400	21.427812
4	40	14.126049	21.961700
5	50	15.312500	24.245278
6	60	15.441300	24.725000
7	70	16.626875	25.283200
8	80	17.304722	30.051111
9	90	22.467778	38.665000
10	100	30.880000	40.690000
11	110	61.47380	66.232810
12	120	60.825903	67.441111
13	130	73.544615	77.957633
14	140	77.553571	79.952347
15	150	79.059467	85.329156
16	160	85.605156	89.180937
17	170	94.678131	98.546401
18	180	97.448333	105.256852
19	190	104.556648	120.401828
20	200	117.637325	149.007350

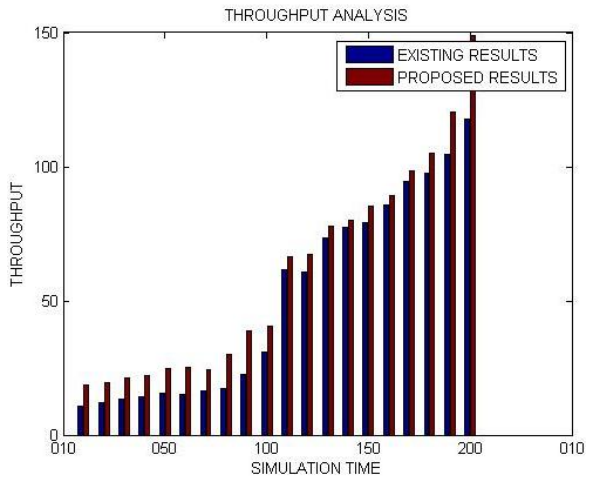

Fig 10: Throughput Analysis.

Fig.10: indicates the Throughput of existing and proposed schemes when simulation time is changed. The x-axis depicts the simulation time and y-axis depicts the Throughput. If it is high for machine, then it is more efficient.

7. CONCLUSION

It has been found that energy efficiency and scalability concepts are too much important for working with the virtual machines which are ignored by many researchers in cloud technology. This paper describes the membership function for enhanced MSS which easily and efficiently handles the management of resources on cloud servers while performing effective job scheduling in cloud computing. The bivalent theory is used for transferring the jobs from one machine to another machine. To handle such issues, an improved MSS scheduling using fuzzification that improved the load balancing based on fuzzy values has been proposed.The MATLAB is used for its design and implementation. In addition. Evaluations have been carried out on execution time analysis and throughput which leads to better management of power consumption and with high speed. This comparison results shows that the efficiency of the proposed algorithm is greater than the existing approaches.

8. REFERENCES

[1] AV.Karthick, E.Ramaraj, R.Ganapathy Subramanian, 2014. An Efficient Multi Queue Job Scheduling for Cloud Computing". World Congress on Computing and Communication Technologies.

[2] Vikas Kumar and Shiva Prakash, 2014. A Load Balancing Based Cloud Computing Techniques and Challenges. International Journal of scientific research and management.

[3] YatendraSahu,R.K. Pateriya, March 2013. Cloud Computing Overview with Load Balancing Techniques" International Journal of Computer Applications.

[4] Rajwinder Kaur and PawanLuthra "Load balancing in Cloud Computing". Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC.

[5] Soumya Ray and Ajanta DeSarkar, October 2012. Execution analysis of load balancing algorithms in cloud computing environment. International Journal on Cloud Computing: Services and Architecture.

[6] Geetha C. Megha raj, Dr. Mohan K.G., October 2013. Two Levels Hierarchical Model of Load Balancing in Cloud, International Journal of Emerging Technologyand Advanced Engineering.

[7] Monika Arya, security Challenges in different delivery model specifically SaaS, International Journal of Advanced Engineering Research andStudies.

[8] KalpanaParsi andM.Laharika, May 2013. A Comparative Study of Different Deployment Models in a Cloud, International Journal of Advanced Research in Computer Science and Software Engineering.

[9] MayankaKatyal and Atul Mishra, December 2013. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment". International Journal of Distributed and Cloud Computing.

[10] RatanMishra andAnantJaiswal , April 2012. Ant colony Optimization: A Solution of Load balancing in Cloud, International Journal of Web & Semantic Technology.

[11] Amandeep, VandanaYadav, Faz Mohammad, April 2014. Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study, International Journal of Scientific Research Engineering & Technology

[12] Nikita Haryani, DhanammaJagli, July-Aug 2014. Dynamic Method for Load Balancing in Cloud Computing, IOSR Journal of Computer Engineering.