Dynamic Resource Scheduling Cloud using Enhanced Queuing Model

R. Divya, VE. Jayanthi

Abstract: ***The important goal of cloud computing is to offer larger data center that satisfies the storage requirements of the customer. The entire data can't be saved in a single server. Cloud provider (CP) has cluster of servers to fulfill the cloud request from various real time applications. The data is fragmented in multiple servers to maintain availability. Since the data request of a customer needs data from various servers, there is a possibility of attaining dead lock. In this paper, an enhanced queuing model is proposed where the cloud request (CR) is received in queuing manner for allocation of resources. A session is created for the CR with the CP resource allocation from cloud severs. This enables to put constraint on the number of CR making a session with CP to avoid resource suppression. The Wait for Resource algorithm is used for allocating the server resources to a CR without deadlock in a session. This enables to forecast the resource requirements prior to resource allocation phase in a session. This makes the dynamic resource allocation efficient and free of deadlock. The results obtained evaluates the proposed model and helps the CP in dynamically choosing the number of server nodes necessary to achieve better performance for an real time application.***

Keywords: Cloud Computing, scheduling, deadlock, wait for resource algorithm, cloud providers.

I. INTRODUCTION

Cloud computing is a service provided over the internet on pay per use manner. Hardware and software facilities are outsourced by Cloud providers (CP) to the arriving requests and the requesting customers can use it anywhere in the world over the internet on rental basis. CP have a network of remote servers hosted on the internet to store, manipulate and maintain data instead of working on a local server or a desktop. For providing this service, the CP needs to establish a dynamic resource allocation for the services over the internet.

It is a rental model for providing on-demand access from a pool of resources that are been available on the CP side and virtually scalable. It is been provided to users on payment basis and less management effort. The main components of cloud computing are on-demand self-service, wide network access, scalability and other requirements made by the end user [1]. Cloud computing brings unlimited computational resources before the user allow them to pay only for what they actually use at any given time. CP provides the available resources to the arriving request using different pricing and allocation methodologies based on time, location, usage, dynamic instance or reserved instance.

Revised Manuscript Received on May 21, 2020.

R. Divya, Assistant Professor , Department of Computer Science & Engineering, NPR College of Engineering and Technology, Tamilnadu, India

VE.Jayanthi, Professor, Department of Electronics and Communication Engineering, PSNA college of Engineering and Technology, Dindigul, Tamilnadu, India.

Cloud users arrive with different requirements varying in terms of hardware, software, operating system, and database etc., CP has to maintain heterogeneous resources including servers and clusters. The management and allocation of such heterogeneous resources makes a challenging role in the research of cloud. This is been done manually for a long time that takes and fails in resource utilization. A dynamic provisioning technique is necessary for resource allocation in data centers and must be automated for checking the proper utilization of resources. This reduces the mean amount of resource and meets the performance of the service cloud as per the Service level agreement between the CP and the user [2].

Data are being generated in large over the internet and it needs a larger data center to store it. So CP has to maintain larger data centers that are used to host multiple servers serving many varying applications with diverse workloads to predict. The data centers must be capable of managing wide resources and should serve multiple users demanding many large applications [3]. This is a challenging issue in dynamic resource allocation and management. Many interactive applications over the net require good response time. The cloud service providers should be capable of serving any service request arriving at any random time by adopting techniques making the cloud flexible. Amazon cloud services have their predefined tariffs and allocation table for cost and servers making both static and dynamic allocation flexible [4].

Quality of service becomes the basic necessity in cloud. The cloud service providers should be concentrating on their quality of service for better service and efficiency. Thereby personal evaluation of their cloud system is necessary. They should consider the main factors of service like service time, throughput, response time, processor utilization etc., for their evaluation. In order to achieve it, the cloud service provider should use a model that considers the arriving number of users, the existing cloud resources and the type of services that are to be delivered. The innovative computing era requires the performance results for effective cloud design. The provider side and service side and the communication side performance have to be done. The cloud user requests follow Poisson distribution and the cloud users are considered to arrive in common probability distribution [5]. The cloud can be evaluated as a queuing model with the web service applications requested by the cloud users as the elements of the waiting queue and the virtual machines being the servers at the cloud. The implementation of queue helps in dynamic creation and removal of virtual machines and this helps in evaluating the scaling factor of cloud.

Virtual Live migration of cloud is not considered here to achieve simplicity. Queuing model helps in the evaluation of the cloud service implementation [6].

The cloud model should be providing not only the deadlock solution with least waiting time but also performance evaluation of the model. The results should be revealing realistic values for design and decision makings in cloud. The average processing time, Service time, Response time and waiting time are considered as the factors of evaluation. This type of Queuing model should help in the profit decision and effective resource allocation at the provider side [7]. Quality of service (QoS) should be transparent to both the cloud provider and the user. Even though the cloud provider aims at better profit, the cloud design and architecture provides the following challenges in the performance evaluation.

Cloud has

- Enormous servers to monitor where the conventional performance models could not evaluate large number of servers.
- The cloud response times and service times are not uniform and has to be considered as the general service probability distribution.
- The arriving cloud users and the requirements are not uniform because of the random and dynamic feature of the cloud. The cloud service must provide QoS in spite of diversifying loads[8],[9].
- While handling effective resource allocation, the investment and profit should also be considered in cloud.

As a solution, we have modeled the cloud as aQueuing model with wait for resource algorithm with Markovian arrival and Gaussian service for better response and avoiding deadlock. The performance comparison has been done between proposed algorithm, First Come First Serve (FCFS) algorithm, min-max algorithm and max-min algorithm. The performance results depict that the proposed model gives relatively better performance in terms of QoS compared to existing algorithms. This queuing model gives the best use of the data center and better response

This paper is organized as follows. The related work for resource allocation is discussed in Section 2. An enhanced queuing model as per the cloud requirements is presented in Section 3. Mathematical equations, performance comparisons and experimental results are shown in Section 4. Finally, Section 5 concludes the paper.

II. RELATED WORK

In 2003, Doyle et al. [10] presented a memory and storage based resource provisioning technique based on cache hit ratio and storage response time. In 2005, Bennani et al. [11] proposed queuing model using combinational search methodology for the best selection of the data center. In 2005, Tesauro et al. [12] suggested a queuing model based on parallel M/M/1 queues for increasing the response time and throughput. In 2005, Urgaonkar et al. [13] presented a queuing model for selecting servers for multi-tier application.

In 2006, Woodside et al. [14] presented a model based on queuing theory for decreasing the workload and improving the overall performance of the system and a scalable resource environment with good response time for the CR at the CP side is designed. A model is created for proper utilization of resources in the cloud. In 2007, Padala et al. [15] recommended a adaptive control based quality model for high resource utilization. In 2009, Ye et al. [16] suggested probability dependent priority scheduling for reducing the usage of servers. The work also presented shared and dedicated resource allocation algorithms to satisfy the user's need.

In 2011, Fayoumi [17] published a work based on discrete event model to access the performance for various workload conditions. The performance of the system was calculated based on the waiting time.

In 2012, X. Wang, et al et al. [18] presented a adaptive technique for allocation of resources and energy management for different workloads and varying applications. This was found to be better in efficiency in terms of workloads

In 2012, Wu et al. [19] presented a admission controlled scheduling algorithm for software applications to maximize profit. This model holds good for SAAS providers. In 2013, Sood [20] presented a prediction model for dynamic resource allocation based on the resource usage rate. The model depicts the resource usage rate and the system status prior to allocation. Thus the model saves the processing time instead of wasting the processing time for instant allocation and calculations regarding the same. The work transfers the load to resource manager instead of the server and thereby the server uses its processing capacity to serve the requests alone.

Many researchers have designed lot of monitoring models that helps in resource monitoring. The maintenance of resources is valuable for the cloud provider and the users. Many existing tools are commercial and some are open sourced. Using the monitoring models and tools, the cloud features namely processing time, memory, speed and throughput can be observed.Maintaining and Managing and organizing the hardware and software resources is the key feature of the cloud provider.

Mell et al., [21] have demonstrated the importance of monitoring tools and its role to thec cloud user and provider. Cloud service provider should serve the cloud users on demand and on time considering the utilization factor. The utilization factor needs to be effective without being over or under utilized. Most of the observing tools measure the performance of the cloud and give methodical results each under its own criteria.

Luo, Liang and Wu, Wenjun and Di, Dichen and Zhang, Fei and Yan, Yizhou and Mao, Yaokuan [22] proposed a cloud resource allocation algorithm for the cloud providers. The algorithm distributes the cloud resources considering energy as the priority index using pre allocation scheme. In the above model, they decided to achieve the Quality of Service through resource pre allocation among the virtual machines. The main drawback of the model is the resource pre allocation that is done statically. It does not monitor the time factor of the future arriving task, time allocation and the resource split from existing physical machines.

Zhong, Hai and Tao, Kun and Zhang, Xuejie [23] suggested the idea of modified GA that overcomes the basic drawbacks of GA for cloud maintenance. The modified GA outperforms the existing algorithms in speed and throughput. The utilization factor of resources, maximum usage, security and over utilization are not considered in the above suggested method.

Chen, Hanxiong and Fu, Xiong and Tang, Zhongrui and Zhu, Xinxin [24] suggested pre estimation and observation method for the cloud environment. They implemented vector auto regression by correlation between the resources for maintenance and observation of resources at each state. The suggested model produced good results but they have mentioned that it consumed more time and the time complexity factor has to be incorporated in future.

Shin, SaeMi and Lee, SuKyoung [25] implemented the conservative back fill priority algorithm that assigns priority based on the arriving job's weight and deadline. The jobs arriving at the cloud data center are being prioritized before execution and executed in the order of priority. The tasks arriving at cloud are dependent nowadays and thereby proposed algorithm could not produce better results for dependent tasks.

Han, Yaojun et al., [26] implemented modified min-006Din algorithm using least language first principle. The tasks with minimum deadline executed the limited capacity server being the basic principle with the task requesting resources and dependency checked here in addition. This again has the usual drawback of load balancing and QoS violation. Vivek Kumar Prasad [27] have developed a load balancer with task scheduling based on HMM prediction. The prediction helps in monitoring the load at the arrival of every request and the completion of request. The results holds good for parallel processing in distributed system.

III. PROPOSED MODEL

There are many resource allocation algorithms for static and dynamic requirements. The model presented in this paper uses Wait for Resource algorithm to avoid deadlock in serving CR at the CP side for dynamic resource allocation. The CP can provide better performance using queuing model and Wait for Resource algorithm. The work presented has two modules. Session creation for the CR is the first module and the second module is the resource allocation between the CR and the servers.The visual architecture of the cloud service is shown in Fig 3.

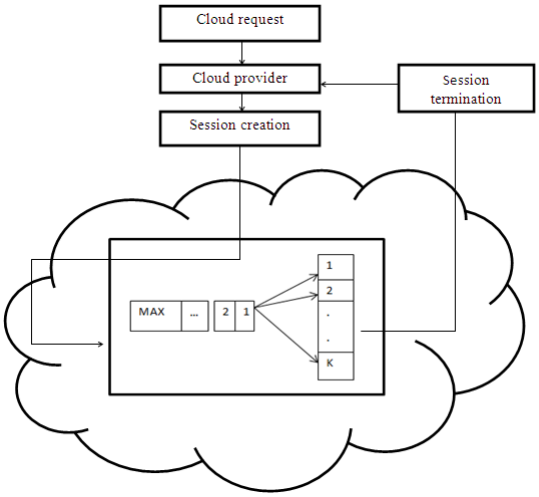

Fig 1: Proposed model architecture.

A. Module I: Session Creation

The Cloud provider allocates resources for CR in a session. The process of creating a session is shown in the below figure. The CR is submitted to the CP for resource allocation. If the number of CR exceeds the maximum value MAX, then the CR arriving thereafter will be added to the queue. The CR will be waiting in the queue until any of the CR's current session gets completed. The CR arrives and get depart as they complete their session. Thus the number of CR's in the session with the CP changes over time. The number of CR is increased when a new CR arrives the session and decreased when a CR gets completed. At any time, the number of CR's in the session is always less than the MAX value. The CP maintains the record of the CR's in the system. The process is shown in below Fig 2.

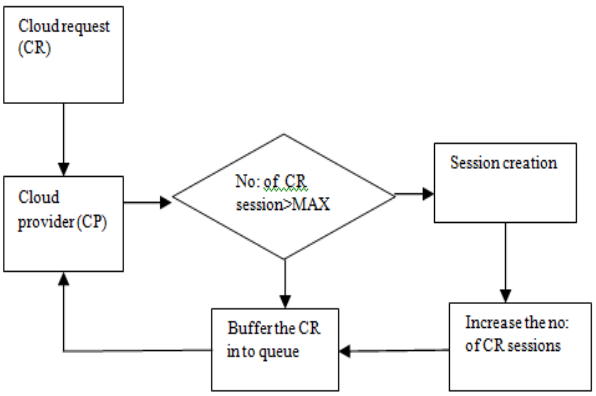

Fig 2: Session Creation

B. Model II: Resource allocation

Once the session has been created for CR's by CP, the requests has to be allocated for the CR. The allocation process is done by Wait for Resource algorithm. The resources are predetermined to avoid deadlock before the allocation starts. When a cloud request is received, the algorithm runs to check if the system will be in safe state if the allocation occurs. The algorithm determines the order of CR to be served so that the resources are utilized properly. This makes the server to be utilized without being idle.

When a CR requests a server which is been utilized currently, then the CR is made to wait till the CR utilizing the server gets free. Fig 3 depicts the above said process in flow chart.

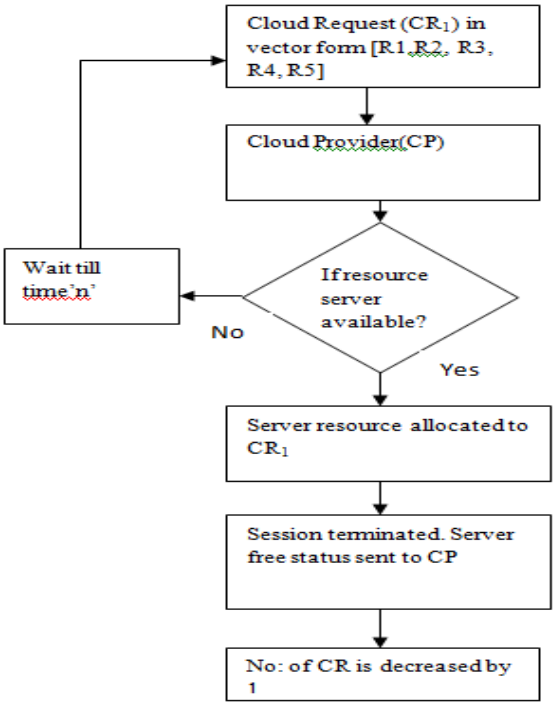

Fig3: Wait for Resource Algorithm

The cloud request comes in matrix form as [R1, R2, R3, R4, R5]. R1 stands for processor speed (in Teraflops), R2 denotes memory requirements(in Gigabytes), R3 denotes bandwidth (in GB/Sec), R4 denotes storage capacity (in Terabytes), R5 denotes the number of server nodes. For example, if the CR arrives as [34565], the customer requests for 3 teraflops processing speed, 4 GB memory, 5 Gb/Sec bandwidth, 6TB storage space and 5 server nodes. The proposed model develops a system with MAX as limit for users in the session with the CP. The MAX value always lies less than the number of CR's. The data center of CP contains MAX number of servers for serving the user request. The CR requesting a server being in use will wait for time n selected by the user and then sends its request again. This waiting for a resource is done to avoid deadlock. The resources are allocated in the order defined by the Wait for Resource algorithm so that the process are served in the way one after the other without deadlock. The CR exceeding MAX number of servers will be waiting in the queue for its time to enter. The session entered CR will be following Wait for Resource algorithm to get their resource from the servers. The allocation and queuing scenario is shown in the following figure.

When a CR arrives the system for the first time, a new session is created with the CP. The arrival is considered to follow Poisson distribution as per the queuing model with arrival rate λ. The arriving CR increments the CR count and if it is less than the servers at the provider side, then it enters the session. Else the CR waits in the queue. All the session entered CR's submit their request in terms of request vector with 5 parameters. The resources requested are allocated dynamically using Wait for Resource algorithm. Once the CR gets its job completed, it decreases the CR count in the session and gives way for the next CR in the queue to join its session. There are MAX identical servers with 5 instances of resources namely memory, bandwidth, storage capacity, Processing speed. The servers are used completely in the model without being idle and the CR makes use of them with less waiting time. Thus the proposed system gives better performance in terms of server utilization, waiting time and response time. The system model allows allocation of resources to be done dynamically. The number of server nodes for a CR changes according to the varying workload. The sequence of CR to be executed without deadlock is found using wait for resource algorithm and the queuing model incorporated gives more response time throughput.

IV. RESULTS AND DISCUSSION

Wait for resource algorithm has been implemented in Java netbeans and the sequence of cloud request to be executed in order to avoid deadlock is found. This section also shows the performance of the resource aware queuing system in terms of server utilization with respect to request rate. The system performance is mainly identified with the help of 3 values namely Monitor time (T), Processing Time (A), Finished (F).The Monitor time(T) is the amount of time the server is ready for request. Processing time (A) is the amount of time the server is active during the Monitoring time or the requested processing time. Finished (F)is the total number of requests finished during the monitoring period. From this, we can infer six significant values namely

Processor Utilization (U) : The percentage of processor capacity utilized during a specific period of time. U=A/T. Throughput of the system (X) is the average number of CR completed during given period of time. X=F/T. Average Service time (S) is the average time needed to complete the request. S=A/F. Capacity (C) is the number of requests that the system can handle. Queue length (Q) is the number of CR in the queue. Q=U/(1-U). Response time (R): The average time required to respond to a request (R). i.e. R = [(Q-1)*S) + S]/2 = (Q*S)/2

The implemented Wait for resource algorithm in Java Net beans was tested in Queuing simulator and results are tabulated below. Figure 4 shows the System response time versus the processing time. In iteration, the Monitor time of the cloud server is 120 sec, and 150 requests are completed in a case, and the Processing time (A) is estimated to be 100 seconds. The server utilization, throughput and response time was calculated for different user requested processing times.

The Processor Utilization (U) is 100/120=83.33 percent utilization. Throughput of the system (X) is 150/120=1.25 requests/sec, Average Service time (S) is100/150=0.66 seconds. Average Queue length (Q) is 0.8/(1-0.8)=4 requests. Average time taken by the server to respond(R)=(4*1.67)/2= 1.25

Table I: Cloud utilization

S.no.	Processing time (A)(In sec)	Processor utilization (U)	Finished requests (F)	Service time (s) (In sec)	Throughput (Requests/Sec)	Response time (R) (In sec)
1	85	60.71429	180	0.472222	1.285714	0.364899
2	82	58.57143	195	0.420513	1.392857	0.297259
3	83	59.28571	180	0.461111	1.285714	0.335721
4	84	60	185	0.454054	1.321429	0.340541
5	87	62.14286	200	0.435	1.428571	0.357028

The above table I shows the arriving request of processing time and the server utilization, throughput at the cloud provider side. This model enables the CP to have a clear picture of his environment and server better with appropriate number of servers.

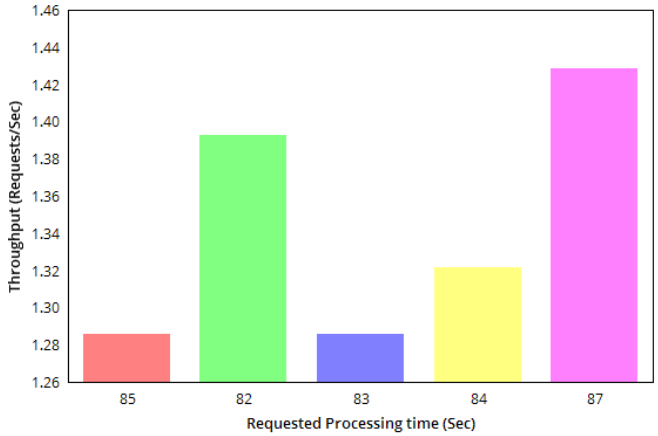

Fig 4: Throughput with respect to processing time of Request

The Fig 4 and Fig 5 depict the performance of the system in terms of throughput and server utilization for the arriving request of processing time.

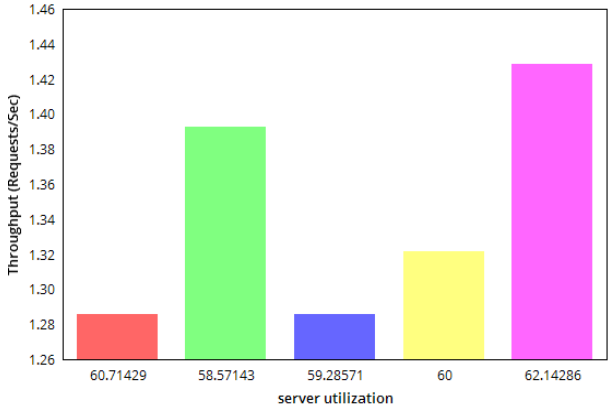

Fig 5:Throughput With respect to Server Utilization

The deadlock avoidance algorithms aim at freeing the server a little bit avoiding over utilization and better response time. The comparison between the existing and proposed deadlock avoidance algorithms is shown in table in Table II and the graphical form in Fig 6. The response time being the basic factor in small scale cloud services, the users prefer the best cloud service with least response time.

Table II: Comparison of Response time

S.No	Response time of Proposed Algorithm(sec)	Response time of Enhanced Load Balancing algorithm (ms)[28]
1	0.364899	0.25033
2	0.297259	0.48968
3	0.335721	0.25033
4	0.340541	0.25367
5	0.357028	0.48968
Average	0.33909	0.3468

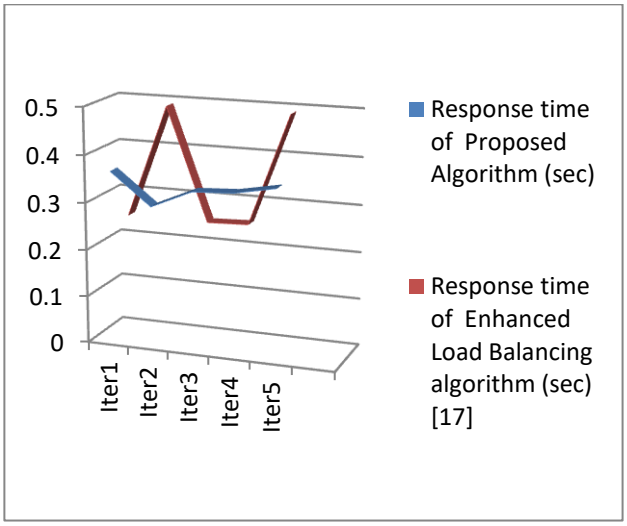

Fig 6: Comparison of response time for Iteration sequences.

It been studied from [29],[30] and [31] that the scheduling algorithms give better results for random inputs in cloud, based on a specific constraint and the results observed are tabulated in Table III.

Table III : Comparison of Scheduling Algorithms

S. No	Algorithm	Average Make span (Sec)	Throughput	Average Resource utilization	Application constraint
1	Min-Min algorithm	20	1.25	-	Incoming cloudlets are of shorter execution time
2	FCFS algorithm	46	1.10	68%	First Come First Serve irrespective of the order of their execution time.
3	Max-Min algorithm	14	1.15	-	Incoming cloudlets are of longer execution time.
4	Proposed algorithm	33	1.3	60%	Follows FCFS with queuing model for better response time and devoid of deadlock.

It is observed from the above table that cloud providers use the scheduling policy based on their requirement and type of service they provide. The Normal scheduling algorithms doesn't consider the deadlock parameter. The deadlock in cloud becomes a essential thing when comes to small scale providers when they share their resources among the cloud users. Thus a queuing model and wait for algorithm is necessary for providing better response time and deadlock free environment respectively. The below graph depicts the performance of the algorithms more clearly. The proposed algorithm gives better performance by balancing the throughput and the average make span. The below Fig7 depicts the comparison factors between the discussed algorithms.

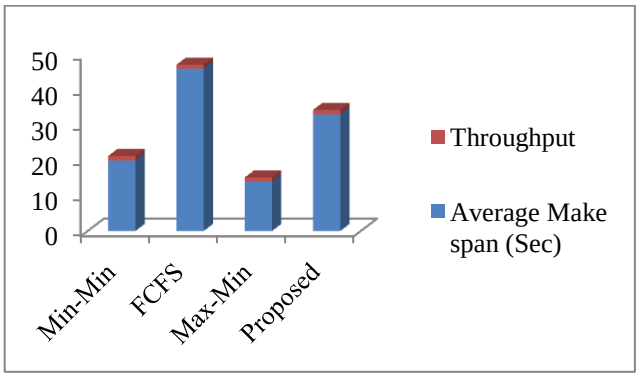

Fig7: Comparison of throughput and execution time between Algorithms.

The proposed model outperforms the existing priority model queuing, where queues of priority are maintained for better profit and the highly demanding user requests stay at the queue1 and the next level profitable requests stays at the queue 2 and so on. This makes starvation of the arriving requests with least demand. The comparison table IV is shown below for random iterations observed between the proposed and priority model suggested in [32].

Table IV: Comparison of waiting time

Iteration No	Priority Model (sec)	Proposed Model(sec)
1	0.38	0.47222
2	0.45	0.42051
3	0.5	0.46
4	0.32	0.45
5	0.44	0.43
6	0.6	0.41
7	0.35	0.46
8	0.4	0.4
9	0.7	0.49
10	0.2	0.5
11	0.5	0.44
12	0.9	0.46

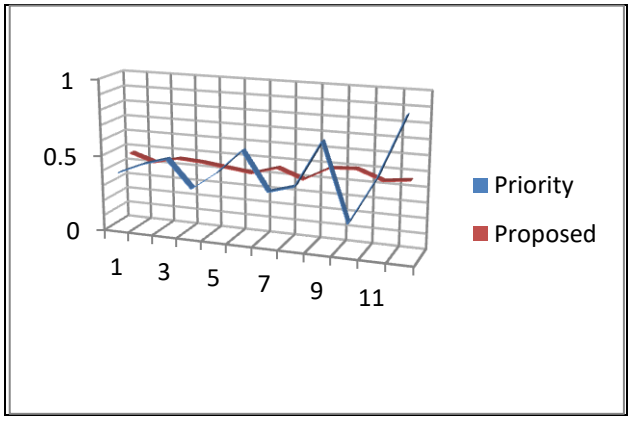

Fig 8: The waiting time Comparison graph

As seen from the Fig 8, it is clear that the average waiting time and the response time is good in the proposed model. The Quality of service is thereby achieved in the proposed model. The existing priority model based on queuing theory holds good for profit expecting cloud service providers without concentrating the system performance or QoS. The profit factor is better in the existing model provided that there is good demand of cloud service requests else degrades the system performance.

V. CONCLUSION AND FUTURE WORK

The work proposed gives a solution for resource allocation technique for cloud environment devoid of deadlock.

A new idea is suggested where the Cloud Provider (CP) creates session for allocation of resources.

The number of CR in a session changes over time. The number of CR's making a session with the CP is restricted using MAX value for proper utilization of resources. The proposed uses wait for resource algorithm thereby avoiding deadlock in real time and interactive applications. The excess CR are made to wait in the queue to avoid resource suppression. The CR provides resource vector to CP mentioning the resource requirements in advance. Thus the CP is able to plan the resources in advance. The proposed dynamic resource allocation model results in better performance in terms of server utilization and response time. The simulation results depict that the model is suitable for interactive applications. The further research work can be extended to adding security to the stored cloud data and supporting larger amount of cloud users incorporating virtual migration techniques.

REFERENCES

1. A. Rahimli, "Factors Influencing Organization Adoption Decision On Cloud Computing," in InternationalJournal of Cloud Computing and Services Science (IJ-CLOSER), vol. 2, no. 2, pp.140-146, 2013.
2. Sivadon Chaisiri, Bu-Sung LEE,"Optimization of Resource Provisioning Cost in Cloud Computing", IEEE Transactions on Services computing,2012.
3. S. K. Sood, "A Combined Approach to Ensure Data Security in Cloud Computing," in Journal of Network andComputer Applications, Elsevier Ltd,vol. 35, no. 6, pp. 1831-1838, 2012.
4. Amazon Elastic Compute Cloud, User Guide, API Version ed., Amazon Web ServiceLLC or Its Affiliate, http://aws.amazon.com/documentation/ec2, Aug. 2010.
5. K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing," Proc.IEEE World Conf. Services, pp. 693-700, 2009.
6. B. Yang, F. Tan, Y. Dai, and S. Guo, "Performance Evaluation of Cloud ServiceConsidering Fault Recovery," Proc. First Int'l Conf. Cloud Computing (CloudCom '09), pp. 571-576, Dec. 2009.
7. B.N.W. Ma and J.W. Mark, "Approximation of the Mean Queue Length of an M/G/C Queueing System," Operations Research, vol. 43, pp. 158-165, 1998.
8. M. Miyazawa, "Approximation of the Queue-Length Distribution of an M/G/s Queue by the Basic Equations," J. Applied Probability, vol. 23, pp. 443-458, 1986.
9. Hadi," Maximizing Profit in Cloud Computing System via Resource Allocation",University of Southern California, Los Angeles, CA 90089 .
10. R. P. Doyle, et al., "Model-based Resource Provisioning in a Web Service Utility," in USENIX Symposium onInternet Technologies and Systems, vol. 4, pp. 5-5, 2003.
11. M. N. Bennani and D. A. Menasce, "Resource Allocation for Autonomic Data Centers using Analytic Performance Models," in International Conference on Autonomic Computing. 2nd IEEE,2005, pp. 229-240.
12. G. Tesauro, et al., "Utility-function-driven Resource Allocation in Autonomic Systems," in 2nd InternationalConference on Automatic Computing, pp. 342-343, 2005.
13. B. Urgaonkar and A. Chandra, "Dynamic Provisioning of Multi-tier Internet Applications,"in 2nd InternationalConference on Autonomic Computing, pp. 217-228, 2005.
14. M. Woodside, et al., "Service System Resource Management based on a Tracked Layered Performance Model," in International Conference on Automatic Computing. IEEE, 2006, pp. 175-184.
15. P.Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem. Adaptive control of virtualized resources in utility computing environments. In Proc. of EuroSys, 2007.
16. H.Ye, et al., "Resource Provisioning for Cloud Computing," in Conference of Center for Advanced Studies onCollaborative Research. 2009, pp.101-111
17. A. G. Fayoumi, "Performance Evaluation of a Cloud based Load Balancer Severing Pareto Traffic," in Journal ofTheoretical and Applied Information Technology, vol. 32, no.1, pp. 28-34, 2011
18. X. Wang, et al., "An Adaptive Model-free Resource and Power Management Approach for Multi-tier Cloud Environments," in Journal of System and Software, vol. 85, no. 5, pp. 1135-1146, 2012.
19. L. Wu, et al., "SLA-based Admission Control for a Software-as-a-service Provider in Cloud Computing Environments," in Journal of Computer and System Sciences, vol. 78, no. 5, pp. 1280-1299, 2012.
20. S. K. Sood, "A Value based Model for Dynamic Resource Provisioning in CloudEnvironment," in International Journal of Cloud Applications and Computing, vol. 3, no. 1, pp1-12, 2013.
21. P. Mell, T. Grance, et al., "The nist definition of cloud computing," 2011.
22. L. Luo, W. Wu, D. Di, F. Zhang, Y. Yan, and Y. Mao, "A resource scheduling algorithm of cloud computing based on energy efficient optimization methods," in Green Computing Conference (IGCC), 2012 International, pp. 1– 6, IEEE, 2012.
23. H. Zhong, K. Tao, and X. Zhang, "An approach to optimized resource scheduling algorithm for open-source cloud systems," in 2010 Fifth Annual China Grid Conference, pp. 124–129, IEEE, 2010.
24. H. Chen, X. Fu, Z. Tang, and X. Zhu, "Resource monitoring and prediction in cloud computing environments," in Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on, pp. 288–292, IEEE, 2015.
25. S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in 2015, 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 814–819, IEEE, 2015.
26. Y.Han and X. Luo, "An effective algorithm and modeling for information resources scheduling in cloud computing," in Advanced Cloud and Big Data (CBD), 2013 International Conference on, pp. 14–19, IEEE, 2013. Harshil Mehta et al, International Journal of Advanced Research in Computer Science, 8 (3), 2013.
27. V. K. Prasad, "Load balancing and scheduling of tasks in parallel processing environment,"International Journal of Information & Computation Technology, ISSN 0974-2239 Volume 4, Number 16, pp. 1727-1732, 2014.
28. Dr. Amit Agarwal, Saloni Jain,"Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7– Mar 2014
29. Davinder Kaur and Sarpreet Singh,"An Efficient Job Scheduling Algorithm using Min-Min and Ant Colony Concept for Grid Computing", International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 03 Issue 07 July, 2014 Page No. 6943-6949
30. R.Jemina Priyadarsini and L.Arockiam," Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud", International Journal of Computer Applications (0975 – 8887) Volume 99– No.18, August 2014.
31. Vetha S and Vimala Devi K," Dynamic Resource Allocation In Cloud Using Queuing Model", Jr. of Industrial Pollution Control 33(2)(2017) pp 39-46, 2014.
32. M. Jaiganesh et al.,"Performance Evaluation of Cloud Services with Profit Optimization", Eleventh International Multi-Conference on Information Processing (IMCIP), Procedia Computer Science 54 ,24 – 30, 2015.

AUTHORS PROFILE

Mrs. R. Divya, is currently working as Assistant Professor in the department of Computer science and Engineering from NPR college of Engineering and Technology. She completed the B.Tech degree in Information Technology in PSG College of Engineering and Technology, Coimbatore. She received her masters in Computer and Communication from PSNA College of Engineering and Technology, Dindigul. She has 6 years of teaching experience. she is currently pursuing her doctorate in the field of cloud computing.

Dr.VE.Jayanthi, is Professor in the Department of Electronics and Communication Engineering from PSNA College of Engineering and Technology, Dindigul. She received her AMIE (ECE) in the Institution of Engineers (India) in 1993. She completed her Master's Degree in Applied Electronics, Anna University, Chennai in 2004. She was awarded a Doctoral degree in Information and Communication Engineering (ICE), Anna University, Chennai in 2013. She is the professional body member of ISTE, IAENG, IEEE (Signal Processing), Biomedical Engineering Society of India (BMSI). She has 20 years of teaching and programming experience and published more than 36 research paper in the reputed international journals. She submitted the proposals and received a sum of the amount of Rs 25 lakhs. Her research interests include Digital Image Processing, signal processing & VLSI Architecture design