iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** II **Month of publication:** February 2018

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor : 6.887
Volume 6 Issue II, February 2018- Available at www.ijraset.com

Queuing Theory Based Enhanced Resource Allocations over IAAS Cloud Architectures

Dr. Fernandes Jayasree Felix[1], Dr K B Priya Iyer[2]
[1]*Associate Professor, Department of Mathematics*
[2]*Associate Professor, Department of Computer Science*
M.O.P. Vaishnav College for Women (Autonomous), Chennai, India

Abstract: Cloud computing has been an emerging technology in recent years in providing computational and infrastructural resources as a service. Cloud computing greatly reduces the deployment and maintenance of web applications since it provides infrastructure as a service (IaaS) and platform as a service (PaaS) for web applications. This results in increase of number of web applications being deployed over cloud. Hence scheduling of resources and managing the performance of the cloud servers form the important research in cloud computing. This paper proposes an efficient resource allocation technique over cloud based on a (M/M/C):(∞/FIFO) queuing theory model. The proposed algorithm reduces the user waiting time and gives faster response rate. The performance of the proposed model is analyzed based on different parameters such as the arrival rate of customers and the number and service rate of processing servers etc. Numerical analysis and experimental simulation shows the effectiveness of the proposed model.

I. INTRODUCTION

Cloud Computing is a platform that offer services over internet both as application delivery as well as hardware and system software. These services are provided by data centers which are often referred as Software as a Service (SaaS). The Clouds are public when the services are made available in a Pay as you go manner and Utility Computing when it is sold. The term Private Cloud refers to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. Now-a-days a lot of web applications particularly from medium and small enterprises have been built into cloud environment. The leading IT companies also invested in establishing public commercial clouds. For example, Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Google App Engine enables enterprises to build and host web applications on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability. IBM also provides cloud options. Whether you choose to build private clouds, use the IBM cloud, or create a hybrid cloud that includes both, these secure workload solutions provide superior service management and new choices for deployment. We even can establish a private cloud with Ubuntu Enterprise Cloud to offer immediacy and elasticity in the infrastructure of web applications. As number of applications deployed over cloud increases gradually, allocating computing resources dynamically to the web applications on demand has a greater impact on performance of web applications as well as energy saving over cloud. The solution to this is to automatically scale quickly up and down the resources in response to load. The providers save money for web applications by optimizing the requested computing resource without violating service level agreements. Thus Resource allocation in a cloud computing environment can be modeled as allocating the requested amount of multiple types of computing resources simultaneously from a resource pool for a certain period of time. The main goal of the cloud service providers is to administer and optimize the computing resources, monitor the traffic to ensure maximum usage of the resources in minimum waiting time. The cloud servers service the users request and when more users enters the cloud for service then the cloud form queues or may lead to enter into reneging state which degrades the network performance. The goal of the analysis of a queuing system is finding analytical expressions for such performance measures as queue length, throughput and utilization. Queuing is the study of waiting lines or queues so that queue length and waiting time can be predicted. In the banks, people stand in a line in front of the counters and wait for the service. In the supermarkets, people wait in the queue to pay for the goods. Although there are no people stand any line in the barber shop, the customers will be served in the sequence they arrive. The performance measures are very important since they are often relevant with the work efficiency or economical losses of a real queuing system, such as an assembly line design. Therefore more accurate and deterministic results are expected in queuing system analysis.

This paper proposes an efficient resource allocation technique based on load of tasks over cloud using queuing theory model. The paper is organized as Section I introduces the concept of cloud computing and resource allocation in cloud. Section II gives the work done by various authors in resource allocations over cloud. Section III explains the proposed task based resource allocation technique using queuing theory. Section IV illustrates the mathematical model of proposed work using queuing theory. Section V gives the numerical analysis of the proposed model on various parameters.

II. LITERATURE REVIEW

In paper [5], the authors have modeled the cloud center as an M/G/m/m + r queueing system with single task arrivals and a task request buffer of finite capacity. A combination of a transform-based analytical model and an embedded Markov chain model is used, which obtained a complete probability distribution of response time and number of task in the system. Simulation results showed that their model and method provided accurate results for the mean number of tasks in the system. applications were modeled as queues and virtual machines were modeled as service centers. They applied the queueing theory models to dynamically create and remove virtual machines in order to implement scaling up and down. [1] aims at QoS modeling approaches suitable for cloud systems. In [3] an optimal resource provisioning algorithm (Bender decomposition approach to divide the problem into sub-problems) is derived to deal with the uncertainty of resource advance reservation. The algorithm reduces resources under- and over-provisioning by minimizing the total cost for a customer during a certain time horizon. [2] analysed the dynamic behaviour of the system with infinite servers by finding various effective measures like response time, average time spend in the system, utilization and throughput. Paper [11] focused on mathematical formulation using queuing system technique to show how throughput and time delay of a system may varies between a single server system and a multiple server system in a cloud-computing environment. [6] routes incoming requests to the queue with the smallest workload reduced workload, response time and the average length of the queue. These results indicate that the model increase utilization of global scheduler and decrease waiting time. The experimental results indicated that proposed model decrease waiting time at global scheduler in cloud architecture. In [9], a new effective and efficient task scheduling algorithm has been proposed. Multi Queue (MQ) scheduling algorithm overcomes the drawbacks of Round Robin and Weighted Round Robin scheduling algorithms and gives better makespan, average resource utilization rate and load balancing level as compared to existing algorithms. The requesting and service of the public cloud has been modelled using queueing theory [10] as a single server bulk service model. If the server is free, each user from the public cloud can enter into the M/M[Y]/1 with probability ϕ or leave the system with probability (1- ϕ) without accessing the cloud database. The results obtained calculates the performance measures of different class i of units in the cloud system and also compare the results of both the partial and full batch service model.

III. QUEUING SYSTEM

Queuing Theory is a collection of mathematical models of various systems of queues. It is widely used to analyze the arrival rate and service time. Formation of queues arises when demand for a service exceeds the limited capacity of the system. A queuing system composed of customers or units needing some kind of service who arrive at a service facility where such service is provided, join a queue. To analyse the arrival rate & service rate and to deliver the packet to the destination a Queuing model which is a Mathematical, Probalistic and Markovian model is applied at routing stages. Queuing system is characterized by the components namely:

Components	Description
Arrival rate	describes the way the population arrives either static or dynamically per some unit of time.
Service rate	describes how many customers can be served when the service is available .
No of service channels	Service channel contains single or multiple. Customers enter one of the parallel service channels and is served by the customer
Queue discipline	describes the manner in which customers choose for the service like First in Firstout(FIFO), Last in First Out(FIFO).
Capacity of the System	A system may have an infinite capacity. The queue in front of the server may grow to any length. This specifies the length of the queue.

Customer behavior generally be in following states. They are:

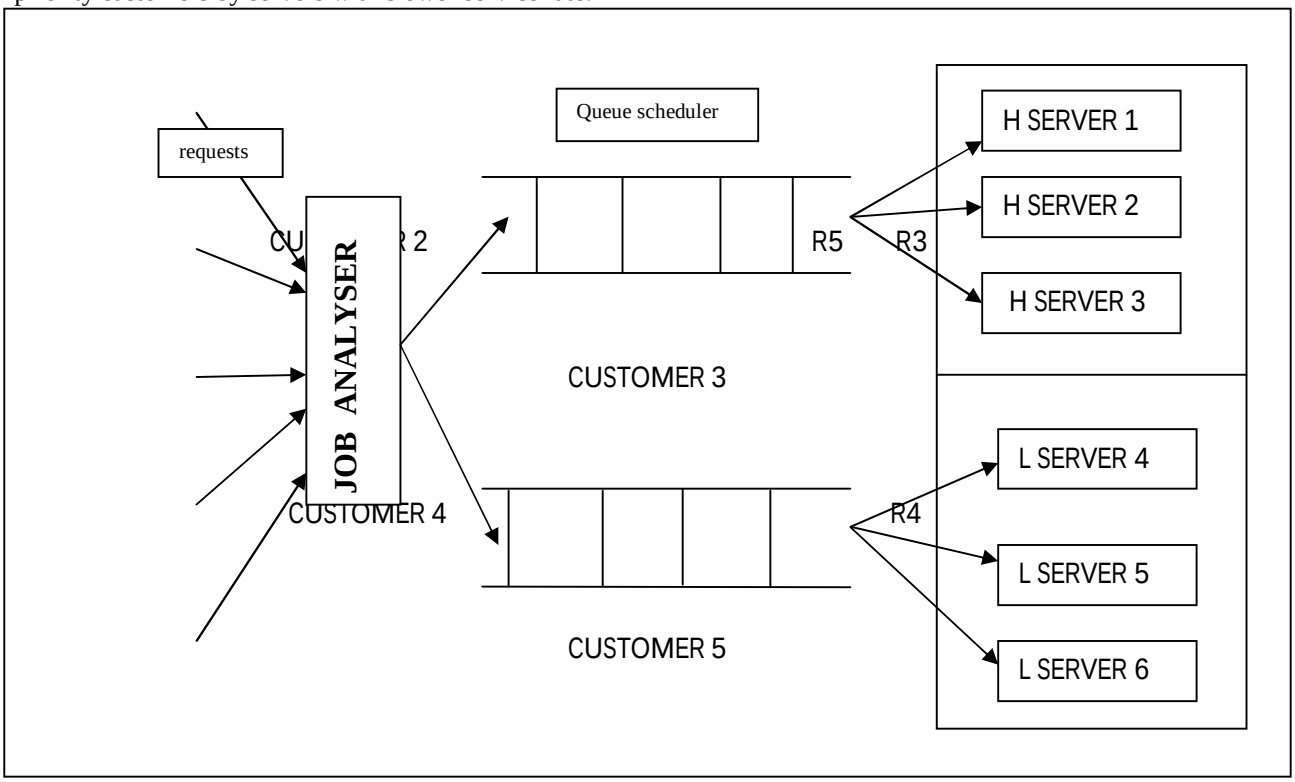
A. Kendell's notation

A Queuing system can be described based on their notations: A/B/C/D/E/F where

A	probability distribution of the arrival rate
B	service time distribution
C	number of servers
D	system capacity
E	population size
F	service discipline

B. Task based Resource Allocation Using Queuing Model

Consider a multi-server queuing system with Poisson input and exponential service time varied according to the categories of customers arriving in the system. The customers arriving in the system are categorized as high priority and low priority customers and they wait in separate queues to avail the service. Also the servers are categorized as specialized servers with faster service rate and ordinary servers with slower service rate. High priority customers are serviced by servers with faster service rate and low priority customers by servers with slower service rate.



Focusing on resources, the formulas of performance indicators such as customer waiting times and server idle times are derived by using the system parameters. A mathematical programming model is developed to determine how many servers should be allocated to each stage to minimize the total costs of customer waiting times and server idle times. Numerical experiments are conducted to analyze the discipline of server allocation and the impact of customer arrival rates and the probability of customer's feedback flow on the system costs.

C. Assumptions

1) The arrival and service processes are correlated and form a bivariate Poisson process.
2) Arrival and service processes are interdependent.
3) Customers are of two types, high priority and low priority.
4) types of servers namely specialized servers with faster service rate and ordinary servers with slower service rates are available.
5) High priority customers receive service from servers with faster service rate & low priority customers receive service from server with lower service rate.

D. Task based resource allocation (TRA) Algorithm

TRA(S[], Q[],C{}, job)

1) cat= Category classifier(job)
2) scat=allocserverqueue(cat)
3) flag=chkserver(scat)
4) if flag is true then
5) job is processed by the server
6) else
7) add the job to the corresponding server queue
8) stop

In the algorithm 1.1, the incoming task requests are sent to the categoryclassifier(). The function categoryclassifier() takes the job and analyses the workload of the task. The task is measured in terms of computation time taken, cost and storage. Basing on these factors, the work is classified as high task or low task. The assigned cat is then sent to allocserverqueue(). This function check the service capacity of the corresponding server. If the server is free then the request is processed. If the server is busy then the task is allotted to the corresponding high task queue or low task queue.

E. Mathematical Model

λ: Mean arrival rate
μ:Mean Service rate
= λ/ μ: server utilization
Steady state distribution: the system is in steady state when the behaviour of the system becomes independent of time.

λ_0 = arrival rate of high priority customers

λ_1 = arrival rate of low priority customers

μ_0 = service rate of specialized servers

μ_1 =service rate of ordinary servers

μ_n = mean service rate, when the system is in state n.

$$\mu_n = n\mu_i \quad 0 \le n < c \qquad i = 0, 1$$

$$= c\mu_i \quad n \ge c$$

N(t) = number of customers in the system at time t

$p_n(t) = P[\, N(t) = n \,]$

e = mean dependence rate (covariance between arrival and service processes)

$\lambda_i, \mu_i, u_n > 0$ and $0 < e < \min(\lambda_i, \mu_n)$, i= 0, 1

The arrival and service process follows a i = 0,1 bivariate Poisson process whose probability mass function is

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor : 6.887
Volume 6 Issue II, February 2018- Available at www.ijraset.com

$$P(X_1 = x_1, X_2 = x_2; t) = e^{-(\lambda_i + \mu_n - e)t} \sum_{j=0}^{\min(x_1, x_2)} \frac{(et)^j [(\lambda_i - e)t]^{(x_1 - j)} [(\mu_n - e)t]^{(x_2 - j)}}{j!(x_1 - j)!(x_2 - j)!}$$

The differential difference equations are

$$p_0'(t) = -(\lambda_0 + \lambda_1 - 2e)p_0(t) + (\mu_0 + \mu_1 - 2e)p_1(t) \quad \text{---------- 1}$$

$$p_n'(t) = -(\lambda_0 + \lambda_1 + n(\mu_0 + \mu_1) - 2(n+1)e)p_n(t) + (\lambda_0 + \lambda_1 - 2e)p_{n-1}(t) + [(n+1)(\mu_0 + \mu_1 - 2e)p_{n+1}(t) \quad \text{---------- 2}$$

$$[1 \le n \le c-1]$$

$$p_n'(t) = -(\lambda_0 + \lambda_1 + c(\mu_0 + \mu_1) - 2e(c+1))p_n(t) + (\lambda_0 + \lambda_1 - 2e)p_{n-1}(t) + c(\mu_0 + \mu_1 - 2e)p_{n+1}(t) \quad \text{---------- 3}$$

The steady state solution of the above equations are obtained by considering

$$p_n = \lim_{t \to \infty} p_n(t) \quad \text{and} \quad p_n'(t) \to 0 \text{ for all n}$$

$\therefore$ The above equations become

$$0 = -(\lambda_0 + \lambda_1 - 2e)p_0 + (\mu_0 + \mu_1 - 2e)p_1 \quad \text{---------- 4}$$

$$0 = -[(\lambda_0 + \lambda_1 + n(\mu_0 + \mu_1) - 2e(n+1)]p_n + (\lambda_0 + \lambda_1 - 2e)p_{n-1} + (n+1)(\mu_0 + \mu_1 - 2e)p_{n+1}$$

---------- 5

$$[1 \le n \le c-1]$$

$$0 = -[(\lambda_0 + \lambda_1 + c(\mu_0 + \mu_1) - 2e(c+1)]p_n + (\lambda_0 + \lambda_1 - 2e)p_{n-1} + c(\mu_0 + \mu_1 - 2e)p_n \quad \text{------- 6}$$

$$[n \ge c]$$

From 4

$$p_1 = \frac{(\lambda_0 + \lambda_1 - 2e)}{(\mu_0 + \mu_1 - 2e)} p_0$$

$$p_2 = \frac{(\lambda_0 + \lambda_1 - 2e)^2}{2!(\mu_0 + \mu_1 - 2e)^2} p_0$$

In general

$$p_n = \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n n!} p_0 \quad \text{for} \quad 1 \le n \le c-1$$

Similarly proceeding we get

$$p_n = \frac{1}{c^{n-c} c!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 , \quad n \ge c$$

Using the fact $\sum_{n=c}^{\infty} p_n = 1$ we get p_0

$$\sum_{n=0}^{c-1} p_n + \sum_{n=c}^{\infty} p_n = 1$$

$$\sum_{n=0}^{c-1} \frac{1}{n!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 + \sum_{n=c}^{\infty} \frac{1}{c^{n-c} c!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 = 1$$ On simplification

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{1}{n!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 + \sum_{n=c}^{\infty} \frac{1}{c^{n-c} c!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)} p_0 \right]^{-1}$$

$\therefore$ The steady state probability distribution of having n customers in the system is

$$p_n = \begin{cases} \frac{1}{n!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 & \text{if n = 0, 1,….., c-1} \\ \frac{1}{c^{n-c} c!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0 & \text{if n = c, c+1, ...} \end{cases}$$

E. Performance measures of the model

Length of the queue Lq = $\sum_{n=c}^{\infty} (n-c) p_n$

$$= \sum_{n=c}^{\infty} (n-c) \frac{1}{c^{n-c} c!} \frac{(\lambda_0 + \lambda_1 - 2e)^n}{(\mu_0 + \mu_1 - 2e)^n} p_0$$

$$= \frac{(\lambda_0 + \lambda_1 - 2e)}{c(\mu_0 + \mu_1 - 2e)} \text{ X } \frac{1}{\left(1 - \frac{\lambda_0 + \lambda_1 - 2e}{c(\mu_0 + \mu_1 - 2e}\right)^2} p_c$$

Where $p_c = \frac{1}{c!}\left(\frac{(\lambda_0 + \lambda_1 - 2e)}{(\mu_0 + \mu_1 - 2e)}\right)c$ X p_0

Average number of customers in the system Ls = Lq + $\left(\frac{(\lambda_0 + \lambda_1 - 2e)}{(\mu_0 + \mu_1 - 2e)}\right)$

Average waiting time of a customer in the queue Wq = $\frac{Lq}{\lambda}$

Average waiting time of the customer in the system Ws = $\frac{Ls}{\lambda}$

IV. NUMERICAL ILLUSTRATION

If the arrival rate of high priority customers λ_0= 9, arrival rate of low priority customers λ_1=5, service rate of specialized servers μ_0=5, service rate of specialized servers μ_1=3, e=0.7and c=4
P_0=0.5164S
Length of the queue L_q=0.15599
Average number of customers in the system L_s=2.0651
Faster rate of arrival and service:
Average waiting time of a customer in the queue W_q= 0.017332
Average waiting time of the customer in the system W_s=0.229455
Lower rate of arrival and service
Average waiting time of a customer in the queue W_q=0.031198
Average waiting time of the customer in the system W_s=0.41302
From the illustration it is informed that the probability of number of customers waiting in the queue to be served is very less. Also the waiting time of the high priority customers in the queue is less compared to that of low priority.

V. CONCLUSION

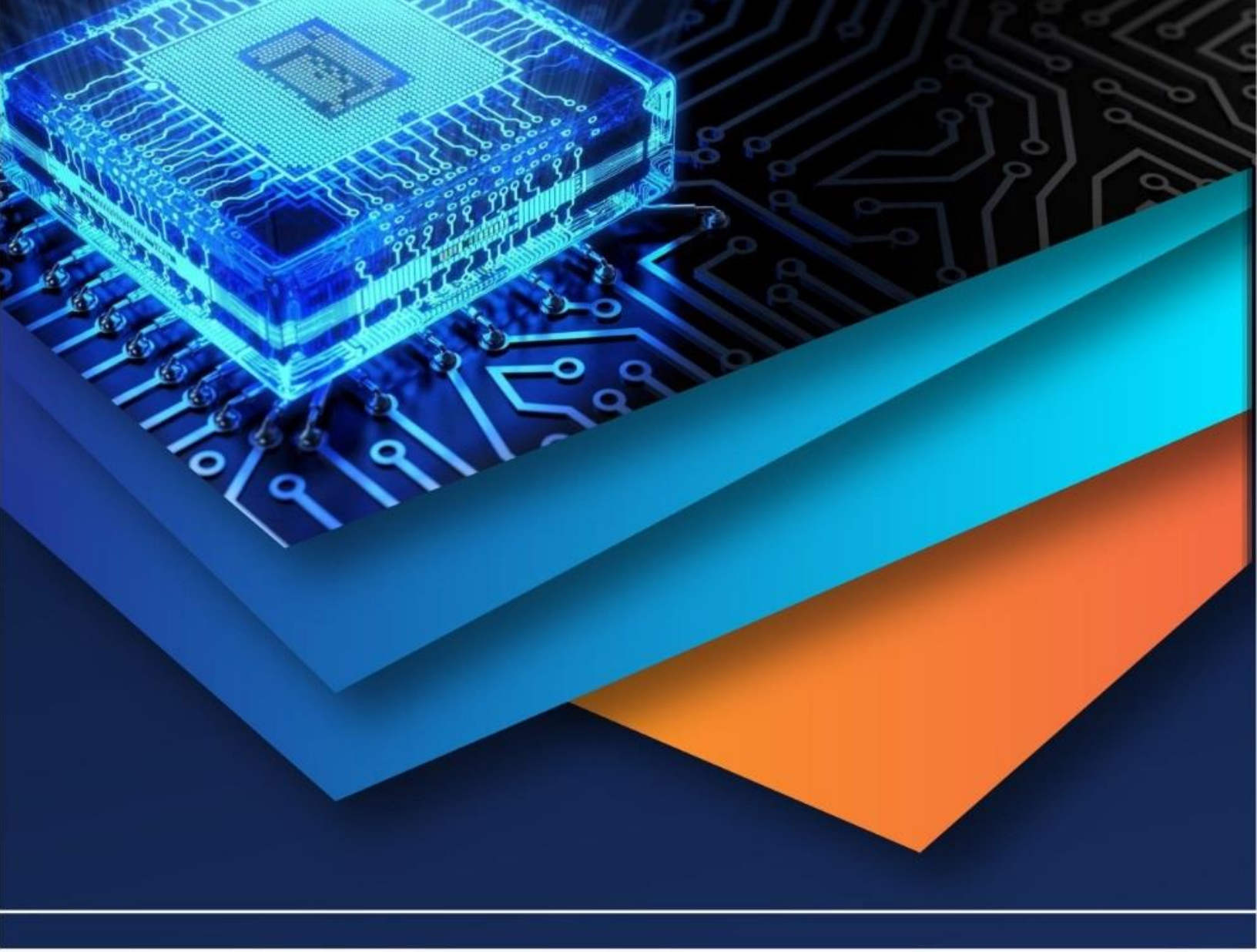
To minimize the customer waiting time and server idle time an efficient task based resource allocation technique is developed. TRA algorithm classifies the task as high or low based on the work nature of the job. The servers are then allotted based on the workload of the incoming requests. The job with lowest work is allotted low service rate server and with high job is allotted higher service rate server. If servers are busy then jobs are thrown to corresponding queue. Numerical experiments are conducted to analyze the discipline of server allocation and the impact of customer arrival rates and the probability of customer's feedback flow on the system costs. The results show the parameters like customer waiting and server idle time are significantly minimized.

REFERENCES

[1] Ardagna et al. Journal of Internet Services and Applications 2014, 5:11.
[2] A.Anupama et al. / International Journal of Computer Science & Engineering Technology (IJCSET). ISSN : 2229-3345 Vol. 5 No. 01 Jan 2014.
[3] Chaisiri S, Lee B-S, Niyato D (2012) Optimization of resource provisioning cost in cloud computing. IEEE Trans Serv Comput 5(2):164–177
[4] V. Goswami, S. S. Patra, and G. B.Mund, "Performance analysis of cloud with queue-dependent virtual machines," in Proceedings of the 1st IEEE International Conference on Recent Advances in Information Technology (RAIT '12), pp. 357–362, Dhanbad, India, 2012.
[5] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 5, pp. 936–943, 2012.
[6] Mohamed Eisa, E. I. Esedimy,M. Z. Rashad. Enhancing Cloud Computing Scheduling based on Queuing Models. International Journal of Computer Applications (0975 – 8887) Volume 85 – No 2, January 2014
[7] Pooja1, Dr Sanjay Tyagi. Scheduling of Heterogeneous tasks in cloud computing using Multi Queue (MQ) Approach. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

[8] K. Santhi 1,* and R. Saravanan. Performance Analysis of Cloud Computing Bulk Service Using Queueing Models. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6487-

[9] Yong-Hua1,2,*, Zhou Zhen2, Zeng Fan-Zi1 and Li Yuan. The Open Automation and Control Systems Journal, 2015, 7, 2280-2285. The Cloud Computing Center Performance Analysis Model Based on Queuing Theory.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY