Queue Control Model in a Clustered Computer Network using M/M/m Approach

(A case of Federal University of Technology Owerri Nigeria)

[1]Ejem A., [2]Njoku C. N., [3]Uzoh O. F., [4]Odii J. N.

[1234] *Department of Computer Science, School of Physical Sciences, Federal University of Technology, P.M.B. 1526, Owerri, Imo State, Nigeria*

ABSTRACT: *This paper presents an application of queueing model for controlling queues in clustered computer networks by incorporating multiple servers in the system using multi-server queueing algorithm for the simulation. It has been shown through theory and experiment that as arrival rate increases in clustered computer networks, single server becomes inefficient and prone to instability, irrespective of the queueing system involved leading to unnecessary delay for service. To address this problem, we deployed an M/M/m queueing model using queueing theory and Object Oriented System Analysis and Design Methodology to incorporate multiple servers to serve customers such that as arrival rate increases, the servers remain efficient, fair and stable which leads to a reduction in waiting time. The Model is simulated using Visual C# (C Sharp) and Microsoft.NET Framework programming tool. Though the application is modeled within the context of FUTO campus network but it is a plausible approach to estimating a reduction in waiting times, traffic intensity (queue length), increasing server utilization, and can be adapted to any other networks.*

Keywords: *Queue, Model, Cluster computing, Queueing Network, Computer Network*

1.Introduction

A cluster is any ensemble of independently operational elements integrated by some medium for coordinated and cooperative behaviour. This is true in biological systems, human organizations, and computer structures. Hence, computer clusters are ensembles of independently operational computers integrated by means of an interconnection network and supporting user-accessible software for organizing and controlling concurrent computing tasks that may cooperate on a common application program or workload (Thomas, 2002). Cluster computing is best described as the interconnection of independent computers and resources connected through hardware, networks, and software to behave as one. The terms cluster computing and high performance computing were initially viewed as one , however the technologies available presently have redefined the term cluster computing to extend above parallel computing to incorporate load-balancing and high availability clusters. An interconnected collection of stations in which customers move from one station to another asking for service are referred to as a queueing network, where each station consists of a queue where customers wait for service. Within the queueing system, the customers are organized according to some discipline which determines the order in which arriving customers get service (Ng et al, 2008). Computer simulation is the means through which a reallife or a hypothetical situation is modeling on a computer system, analyzing the output while studying its behavior. Queueing system is one of the many problems associated with discrete event systems, while computer simulation is an optimal means of solving queueing problems and analyzing its performance. So many people have attempted to solve this problem in different perspectives. Okoro (2013) applied a markovian queueing model in real life situation where the spread of disease is the arrival time and the rate of coming out from the disease is service time. In conclusion he found out that markovian queueing model as a birth-death process is very vital in computing paradigm. Another researcher, Bisnik et al (2009) modeled a random access multi-hops wireless networks as open G/G/1 queueing networks using diffusion approximation in order to evaluate closed form expressions for the average end-to-end delay. The paper was able to measure the delay that takes place between when a data packet travels from the sender-the source computer to the receiver-destination computer and so many other queueing models as revealed by literature. Despite all these attempts there are still records of delay-service time especiallyFUTO. It has therefore become imperative that a new approach be adopted to address this issue of delay-clustered computing. The M/M/m approach is an application which will reduce waiting time if not completely but at least to the barest minimum.

2. Review of Related Literature

According to Rajkumar (1999), a cluster is a type of parallel or distributed computer system, which consists of a collection of interconnected stand-alone computers working together as a single integrated computing resource. In recent years, researches have made some progress on analyzing and improving network performance in the application of finite capacity queueing network models. Kouvatsos et al (2003), described the priorities and blocking mechanisms with open-loop queueing network performance analysis, and queueing network parameters on the approximation and error estimates. In the same manner, Özdemira et al (2006), presented two Markov chain queueing models with M/G/1/K queues, which have been developed to obtain closed-form solutions for packets delay and packets throughput distributions in a real-time wireless communication environment using IEEE 802.11 DCF. Moreover, Mann et al (2008), developed a queueing model for analyzing resource replication strategies in wireless sensor network, which can be used to minimize either the total transmission rate of the network or to ensure that the proportion of query failures does not exceed a predetermined threshold. Even, Bisnik et al (2009), modeled random access multi-hops wireless networks as open G/G/1 queueing networks and used diffusion approximation in order to evaluate closed form expressions for the average end-to-end delay. However, Liehr et al (2010), introduced enhancements to the standard of extended queuing network models, which allow the modeling and the simulation of inter-process communication and highlight the benefits granted by their enhanced extended queueing networks approach. Furthermore, Odirichukwu et al (2013) were interested in banking queue system in Nigeria. The aim of their research was to minimize waiting time in queue by proper queue management and thereby maximizing throughput. They developed a web based application that assigns each customer queue number on arrival based on touching the screen and the queue number are stored electronically. Their research uncovered the applicability and extent of usage of queueing models in achieving customer satisfaction at the lowest cost. Finally, they recommended a multi-server queueing model to solve the problem of queues in Nigerian bank. In the same vein, Okoro (2013) applied markovian queueing model in real life situation. He reexamined; average number of customers and average number of time in the system, waiting in the queue and in service respectively. In conclusion he found out that markovian queueing model as a birth-death process is very vital in epidemiology study. Chandra et al (2013) introduced a markovian queueing system having a multi-task service counters and finite queue in front of each counter. They carried out sensitivity analysis to study the effect of variation of different parameters. Ajay et al (2013) have discussed the approach of queueing theory and queueing model. From their analysis the capability of the queueing system can have an important result on the quality of human and productivity of the process. Hence, Kumar et al (2014), studied a finite waiting space markovian single server queueing model with discouraged arrivals, reneging and retention of reneged customers. They derived the steady state solution of the model iteratively and the measures of effectiveness of the queueing model were obtained.

However, amidst the benefits of the above researches, they have not use our approach in addressing the problem of queue control in clustered computer network. Our approach of using an M/M/m queueing model was motivated by the need to optimize the performance of a clustered computer network with application to FUTO campus network.

3. The Existing Study Model

Clustered Computer networks are found in organization such as Universities, Government organizations, very large industrial layouts etc. The topology of our experimental clustered computer network model is based on the structure of a typical Nigerian University. In this regard Federal University of Technology Owerri (FUTO) serves as a study model. The University is organized into central administration block, Schools, Departments and service units. The University central administration block controls the entire activities of the entire School. The Vice Chancellor is the person in charge of the entire FUTO community, the Deans are in charge of Schools while the HODs are in charge of Departments. Presently FUTO as a typology of clustered network operates a single queue single server queue models. The central admin block and schools were considered as network stations with the departments as the nodes and the central admin block as the Head Station. The servers are classified as large server (which typifies the work of the Vice Chancellor), medium server (which typifies the work of the Deans) and small server (which typifies the work of the HODs), serving the central admin block, schools, departments and other external persons. The small server serves the department giving services to all the request that come to the department. Most often the services requested might need the medium server attention so it transfers the request to medium server else the small server service the request and the customer exit the system. When the medium server checks the request if it is within its capacity, services the request and the customer exit the system else it transfers the request to the large server if its attention is needed for that request. The large server services the request send it back to the medium server. The medium server then send it back to the small server and the customer exit the system. This scenario is vice versa.

3.1 Limitation of the Study Model

The problems of the existing system are outlined below:

(1) **Prolong Waiting:** it is obvious that when arrival process increases the server becomes inefficient, prone to instability leading to increase in waiting time. Usually the customer has to wait for days, weeks and sometimes months to get a feedback of his/her request; possibly because the server is not available or the server is doing so many things at a time.

(2) **Loss of Documents:** Usually in the process of moving the request from one station to another the documents bearing the request is lost demanding the customer to make fresh applications leading to waiting excessively.

(3) **Server Breakdown:** The server can breakdown due to excessive usage; the server does not have ample time for his/her own personal development and other recreational activities. In the case where the server is a human being, he or she can become ill which makes him or her to be unavailable for the period he or she is down with ill health thus leading to prolong waiting on

the side of the customers who is waiting for a response on their request.

3.2 The Proposed Model Architecture

The multiple server queuing model of a clustered computing network systems is illustrated using the Figure 1. Assuming that there are n requests and m services, each of them is independent. Since the consecutive arriving requests may be sent from two different users, the interarrival time is a random variable, which can be modeled as an exponential random variable in clustered computing network systems. Therefore, the arrivals of the requests follow an exponential distribution with arrival rate λ_i. Requests in the scheduler's queue are distributed to different computing servers and the scheduling rate depends on the scheduler.

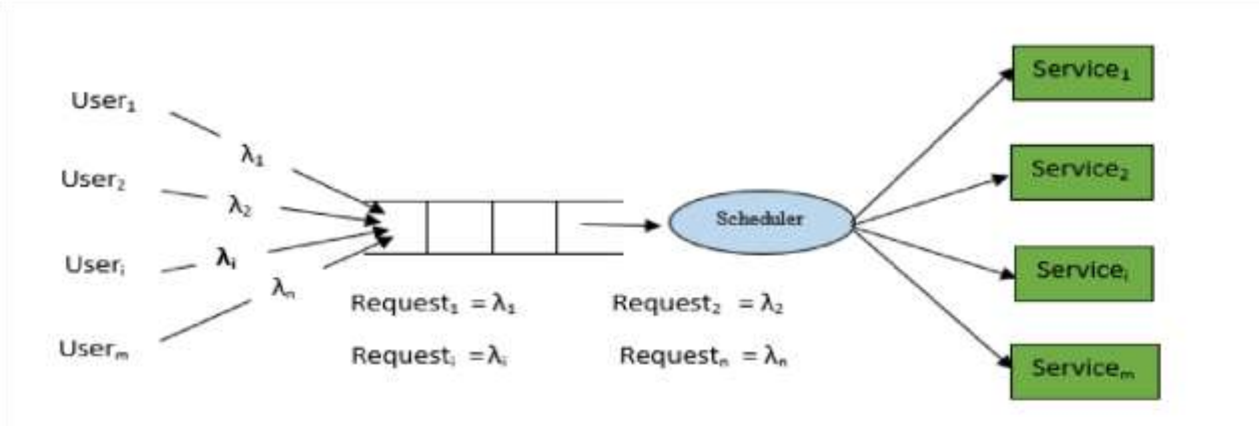


Figure 1: Multiple Server queueing Model in Clustered Computer Network

From figure 1, suppose that there are m computing servers, denoted as Service$_1$, Service$_2$, Service$_i$, and Service$_m$ in the clusters; the service rate is μ_i. So, the total arrival rate is

$$\lambda = \sum_{i=1}^{n} \lambda \mathbf{I} \qquad (1)$$

and the total service rate is

$$\mu = \sum_{j=1}^{m} \mu^{\mathbf{i}} \qquad (2)$$

Theory has proved that the system is stable, when $\lambda/\mu < 1$.The rate of service requirement follows the Poisson Process; it is the same as the customer arriving rate. μ_n and λ_n represent the mean service rate and the mean arrival rate for the overall queueing system when there are n customers in the system. The service rate per busy server is μ, the overall mean service rate for n busy servers must be $n\mu$. From session 2.7.4, the Model equation is given as: $\lambda_n = \lambda$ for all $n \geq 0$

$$\mu_n = \begin{cases} n\mu & n < m \quad (3) \\ m\mu & n \geq m \quad (4) \end{cases}$$

Hence,

$$P_n = \begin{cases} (\rho^n/n!)\,P_0 & ;\, n \leq m \quad (5) \\ (\rho^n/(m!m^{n-m})\,P_0 & ;\, n > m;\ \rho = \lambda / m\mu \quad (6) \end{cases}$$

Where P_n is the probability of n customers in the system. ρ is the traffic intensity, n is the number of customers and m is the number of servers.

3.3 Features of the New Model

(1) **High Performance, High Throughput:** increased performance processing is achievable with multiple interconnected systems.

(2) **High Availability –** When multiple systems are interconnected, the loss or failure of any one of them should have only a minor effects. When any system fails the other systems on network takes up its duties.

(3) **High Reliability:** The failure of any system does not mean the failure of the whole system.

(4) **Parallel processing:** In parallel processing the server will have a faster response time, as work can be done in parallel (concurrently) etc.

3.3.1 Justification of the Model for FUTO Request Services

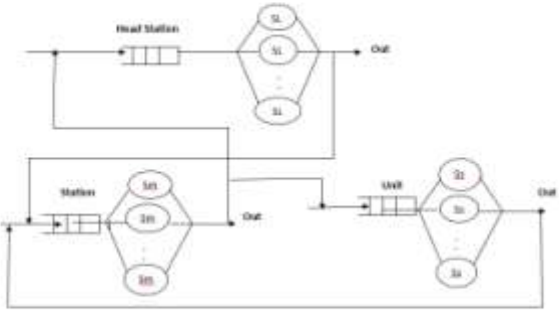


Figure 2: Model Diagram for FUTO Campus Network (Request Services)

Thus probability that the system is idle is given as:

$$P_0 = 1/\left[\, \textstyle\sum_{n=0}^{m-1} (1/n!)\,(\lambda / \mu)^n + (1/m!)(\lambda / \mu)^m (m\mu/ m\mu-\lambda)\right] \quad (7)$$

Usually, from figure 2, most FUTO service request arrives at the unit/department (Small server (s_s)). As request arrives at the unit, they are routed to the servers in parallel. Each of the server considers the request to ascertain that it has the capacity to attend to the request. While the request is within the server's capacity to process, it processes the request and the request departs the system else it sends the request to the next server with more capacity usually the medium server(s_m). The medium server processes the request and the request leaves the system and may be sent back to the small server for its exit. Sometimes, it sends the request to the large server which processes the request and send it back to the medium server, from where the request moves to the small server for its final departure. However, in the existing study model, one server attends to several service requests coming to the system which leads to increase waiting, long queues and server breakdown but in the proposed model more than one server running in parallel attend to the many service request coming to system which leads to a reduction in waiting time, reduction in queue length and guarantees customer's satisfaction.

4 Methodology

The methodology adopted in building the model was Object Oriented System Analysis and Design Methodology. In order to use this methodology effectively and efficiently, the following assumption was made to aid the simulation of the model:

(i) We assumed that the users are familiar with queueing theory and its analysis.

(ii) We assumed exponential distribution since it possesses the memoryless property that is customers exist independently of one another, and arrive at the service provider when they please.

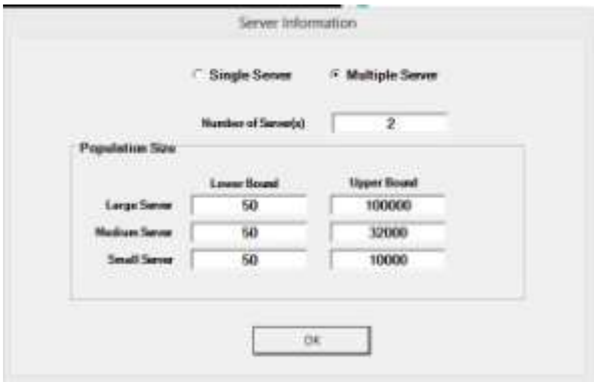
(iii) We assumed first come first serve for all communications in the Stations and Units/Nodes.

(iv) We assumed intuitively the value of **α** (alpha) and (beta) to aid in determining the arrival rate and service rate

4.1 Results and Discussion

Input Interface

In this research, there are categories of input data; these are the data supplied directly from the keyboard and those retrieved from the database/files such as the random number generated. Figure 3 is a typical input data interface of the system developed.



Server Information

○ Single Server ● Multiple Server

Number of Server(s): 2

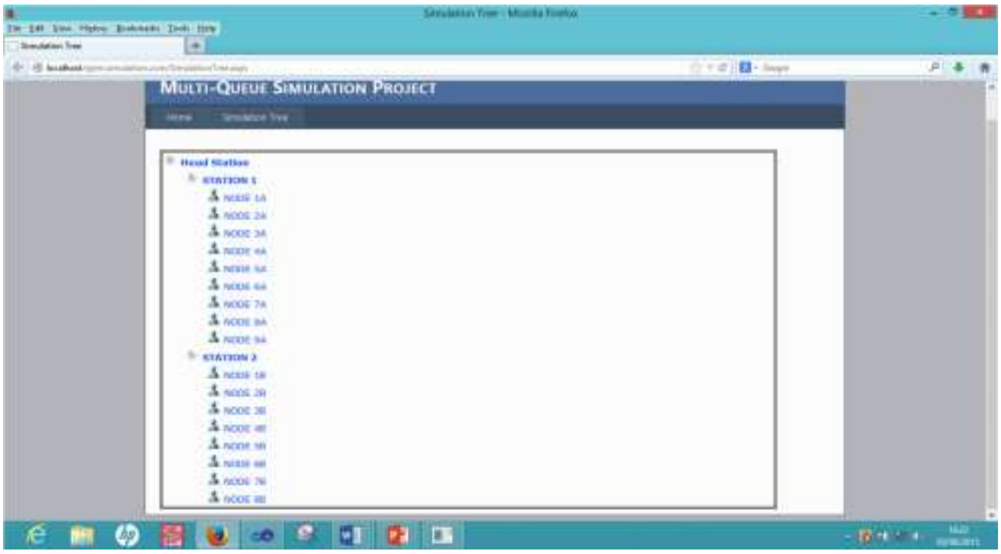
Population Size

	Lower Bound	Upper Bound
Large Server	50	100000
Medium Server	50	32000
Small Server	50	10000

Figure 3: Input Interface

Figure 3 gives a detail input information to the queueing model. It gives the user an option of choosing either the single server or the multiple server. It also provides an option for stating the lower bound and upper bound for generating the population size of each server.

4.2 Simulation Test Results /Output Interface



Simulation Tree - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Simulation Tree
localhost
Multi-Queue Simulation Project
Home
Simulation Tree

- Head Station
  - STATION 1
    - NODE 1A
    - NODE 2A
    - NODE 3A
    - NODE 4A
    - NODE 5A
    - NODE 6A
    - NODE 7A
    - NODE 8A
    - NODE 9A
  - STATION 2
    - NODE 1B
    - NODE 2B
    - NODE 3B
    - NODE 4B
    - NODE 5B
    - NODE 6B
    - NODE 7B
    - NODE 8B

Figure 4: Output Interface

Figure 4 gives the output form of the simulation model. The Head station, Stations and Nodes provide a link for accessing the queueing table information. When each of them is clicked it produces the queueing information and the performance measures for analyzing the system.

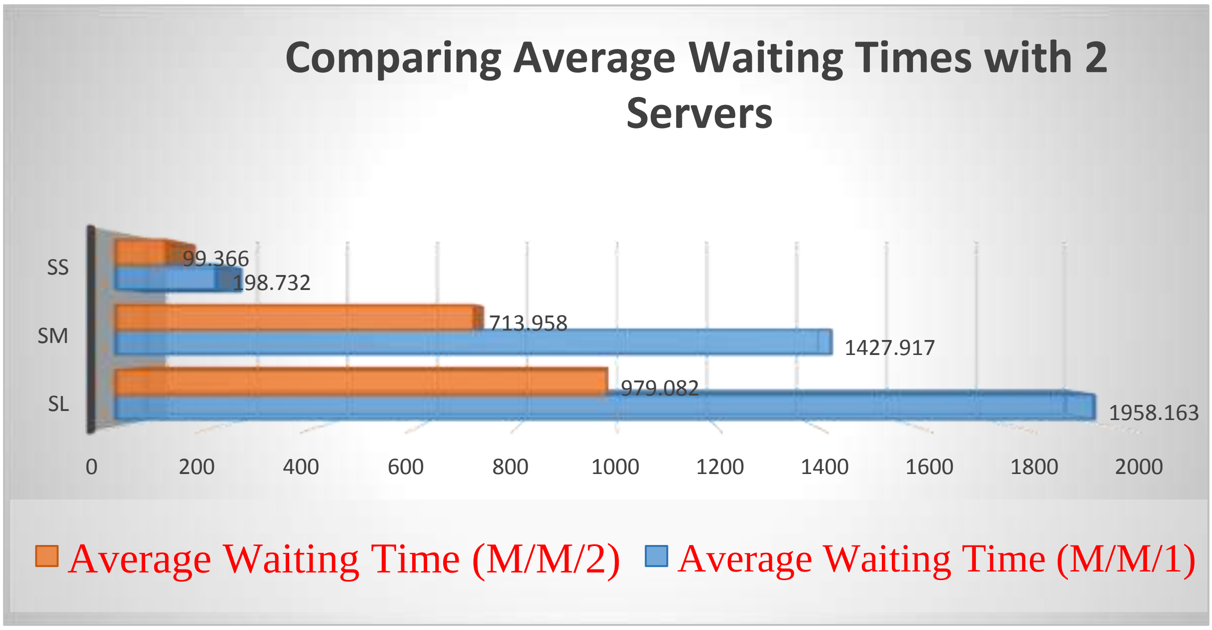
4.3 Comparative Evaluation/ Summary of Findings

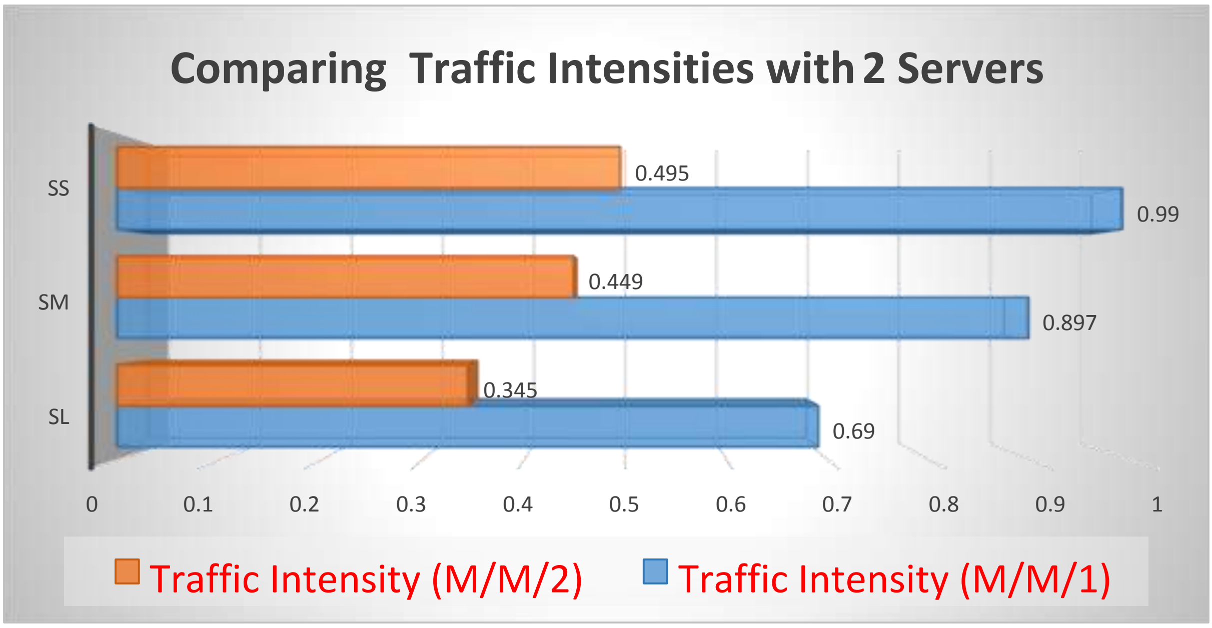
1(one) Server

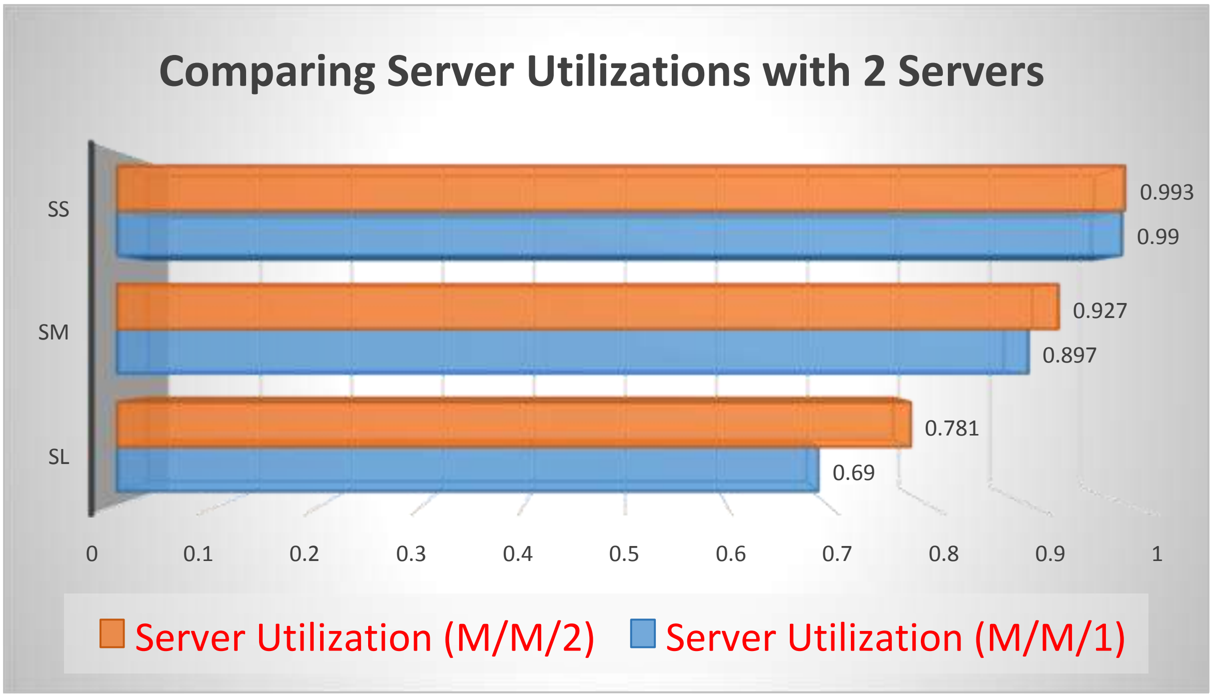
Server Type	Arrival Time	Service Time	Average Waiting Time	Traffic Intensity	Server Utilization
S_L	0.584	0.846	1958.163	0.690	0.690
SM	0.802	0.893	1427.917	0.897	0.897
S_S	0.879	0.887	198.732	0.990	0.990

2(two) Servers

Server Type	Arrival Time	Service Time	Average Waiting Time	Traffic Intensity	Server Utilization
S_L	0.584	0.846	979.082	0.345	0.781
SM	0.802	0.893	713.958	0.449	0.927
S_S	0.879	0.887	99.366	0.495	0.993







In this paper, we have shown that M/M/m queueing model increases performance over using one server- M/M/1. Analysis and simulation results vividly shows that application of M/M/m queueing model reduces waiting time, traffic intensity (queue length) and increases Server Utilization when compared to M/M/1 and also guarantees customer satisfaction.

4.4 Future Work

1) Propounding an optimized task assignment strategy whereby the utilization of clustered network resources as well as the performance indicators (e.g., waiting time, service time, blocking probability (if finite population is

considered), traffic intensity etc.) are improved simultaneously.

2) The research work can be extended by using homogenous servers in all the service stations and nodes since we considered a case of heterogeneous servers (large server, medium server, small server).

4.5 Conclusion

In this paper, a queueing control model in clustered computer network has been applied to analyze FUTO campus networks. The major issues considered were queueing and traffic management in clustered computer networks, framework for modelling and simulation of queues in clustered computer networks etc. Further a multi-server queueing model was deployed to aid control long queues associated with single server model in clustered computer networks. It was found out using multi-server instead of single server reduces waiting time and increases customer satisfaction.

REFERENCE

[1] Ajay K. S., Rajiv K., & Girish K. S. (2013),"Queueing Theory Approach with Queueing Model: A Study", International Journal of Engineering Science Invention, ISSN (online):2319-6734, ISSN Print: 2319-6726 qwww.ijesi.org volume 2 Issue 2,Pp1-11.

[2] Bisnik, N., & Abouzeid, A. A. (2009)," Queuing Network Models for Delay Analysis of Multihop Wireless Ad Hoc Networks", Ad Hoc Networks, Vol. 7, No. 1, Pp. 79-97.

[3] Chandra S., & Madhu J. (2013),"Finite Queueing Models With Multitask Servers and Blocking", American Journal of Operational Research,3(2A):Pp. 17-25, DOI:10.5923/s.ajor.201305.03

[4] Kouvatsos, D., & Awan, I.(2003)," Entropy maximisation and open queueing networks with priorities and blocking Performance Evaluation", Vol. 51, No. 2-4, Pp. 191-227.

[5] Kumar R., & Kumar S. S. (2014),"A Single Server Markovian Queueing System with Discouraged Arrivals and Retention of Reneged Customers", Yugoslav Journal of Operations Research 24 number 1, Pp. 119-126 DOI: 10.2298/YJOR120911019k

[6] Kumar R., & Kumar S. S. (2014),"Two Heterogeneous Server Markovian Queueing System with Discouraged Arrivals, Reneging and Retention of Reneged Customers", International Journal of Operations Research Vol. 11, No. 2, Pp. 064-068

[7] Liehr, A. W., & Buchenrieder, K. J. (2010)," Simulating inter-process communication with Extended Queueing Networks". Simulation Modelling Practice and Theory, Vol. 18, No. 8, Pp. 1162-1171.

[8] Mann, C. R., Baldwin, R. O., Kharoufeh, J. P., & Mullins, B. E (2008), "A queueing approach to optimal resource replication in wireless sensor networks. Performance Evaluation, Vol. 65, No. 10, Pp. 689-700.

[9] Ng C. H., & Song B. H. (2008), "*Queueing Modelling Fundamentals"* Second Edition, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England. Pp. 44-55

[10] Odirichukwu J. C., Tonye L., & Odii J.N. (2013)," Banking Queue System in Nigeria", Computing, Information Systems, Development Informatics and Allied Research Journal ISBN 9782257447(print) ISSN 21671710(online) Vol. 4 No.2, www.cisdijournal.net, Pp.98-105

[11] Okoro O. J. (2013),"On Markovian Queueing Model as Birth-Death Process", Global Journal of Science Frontier Research Mathematics and Decision Sciences Vol.13 Issue 11 version 1.0 USA, Pp. 21-33

[12] Özdemira M., & McDonald A. B (2006),"On the performance of ad hoc wireless LANs: A Practical queueing theoretic model. Performance Evaluation", Vol. 63, No. 11, Pp. 1127-1156.

[13] Rajkumar B. (1999), High Performance Cluster Computing: Architectures and Systems, vol. 1& 2, Prentice Hall.Pp.03-48

[14] Thomas S. (2002), Beowulf Cluster Computing with Linux, The MIT Press Cambridge, Massachusetts London, England. Pp. 01-29