Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Minimization of Waiting Time through Additional VMs in Fog Center

Bibhuti Bhusan Dash[1], Rabinarayan Satapathy[2], Sudhansu Shekhar Patra[1*], Utpal Chandra De[1]

[1]*School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India*
[2]*Faculty of Engineering and Technology, Sri Sri University, Cuttack, India*
Email: sudhanshupatra@gmail.com

Fog computing is a computing environment that brings the capabilities of the cloud computing paradigm to the network's edge, often referred to as fog networking or edge computing which is decentralized in nature. In this paradigm the computing resources and services are dispersed among several hardware and infrastructures that are placed in the nearby proximity to the network's edge, including edge servers, routers, switches, gateways, and other similar devices. Being forced to wait in a large queue is unpleasant, and in some delicate applications, the client gets irritated and quits the fog system. In certain situations, adding more servers can help shorten wait times. The multi-server queueing system with extra virtual machines (VMs) is the topic of this study. Following the Poisson process, client requests arrive and are handled by a pool of permanent and extra VMs using FCFS queue discipline. The precise method for calculating the client requests waiting in the queue has been discovered in terms of numbers. The stated analytical findings have been supported by several numerical instances.

Keywords: Fog computing, Multi-Server Queuing Model, FCFS, Discouraged client requests, Additional VMs.

1. Introduction

A computing paradigm, distributed in nature called "fog computing" brings cloud computing benefits to the network's outer edges. As a result, the data source is closer to computational resources and services, enabling faster processing, real-time analytics, and improved efficiency in various applications (Abou-El-Ata et al. ,1992)

Data is transported to distant data centres for processing and analysis under the cloud computing paradigm, which has historically been the predominant data processing and storage method. For the real time decision making as well as low latency and high bandwidth application cloud is not an appropriate choice. Fog computing is an alternative to the users where the computational resources are available at near by proximity to the clients to solve their latency sensitive applications for a faster results. This is done by decentralized the computational infrastructure (Behera et al., 2023).

The hierarchy of computing devices in the network architecture of fog computing is exists above the edge layer and below the cloud server i.e., in between the cloud and edge layer. Edge devices includes

IoT devices, sensors etc. closer to the data sources that are responsible for gathering data. Fog nodes act as an intermediaries between edge devices and cloud servers providing local data processing capabilities (De et al, 2023). The benefits of fog layer is that it enhances bandwidth efficiency by locally filtering and aggregating data, hence reducing the volume of information sent to the cloud. The scalability of fog computing is an additional significant advantage that effectively adjust its capacity according to demand. Fig. 1 shows the architectural model of the fog compting.

Fog computing has applications in various fields, including smart cities, industrial IoT, healthcare, transportation, and retail. Data processing as well as decision-making at the edge are efficient, resulting in increased operational efficiency, cost reductions, and better user experiences. Fog computing signifies a transformative change in the deployment and use of computer resources. It enhances responsiveness, agility, and intelligence in contemporary networked systems by situating processing nearer to the data source.

In many cases, the client requests to wait for a desirable service from a fog center and if all the VMs are busy providing service to other requests, there may be a chance of leaving the system by the sensitive applications or by the impatient client requests. There will be a need for additional VMs to be allocated to the fog system to avoid loss in the system and avoids inconveniences to the clients (Ghani et al., 2024).

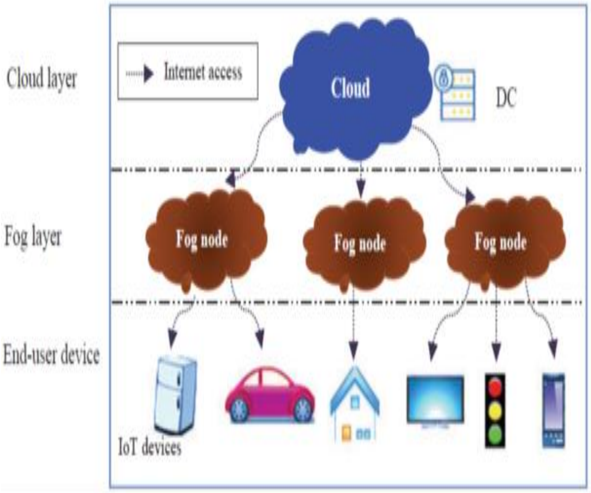

Fig. 1. Fog layer Architecture

Additional VMs can be implemented in a fog node in the fog layer to increase computing power and offer more resources for processing and storage. The normal location of these servers within the network architecture is close to the edge devices or at key intersections. The research that uses queuing theory and provides analytical modeling in the field of distributed computing is included in this area. Numerous research concentrates on the Markovian models with a single server, c number of servers, finite buffer capacity, and infinite buffer capacity where the arrival and service rates are distributed exponentially (Li et al., 2019). The authors tried to minimize the latency, meet the SLA, optimize the energy consumption, minimise response time, waiting time, and power consumption by modeling the distributed nodes by providing the additional server capacity in the distributed computing environment. These models provide precise mathematical illustrations. Some studies examined queue models that consider the possibility of a generic distribution for inter-arrival and service times. The authors in (Mukaddis et al., 1983) studied a queuing system featuring a small waiting area and an extra special channel for a large line. The authors in (Mukherjee et al.,2018) suggested a single-server infinite buffer queuing model with additional servers for an impatient customer. Authors in (Murari et al., 1968) extended this work to finite servers and finite buffer queuing systems with additional servers. This paper proposed a finite buffer queueing model with 'v' permanent VMs, 'a' additional VMs, and impatience of client requests.

In real applications, there are several instances of queuing when consumers may have a propensity to become discouraged by a lengthy line. As a result, clients either abandon the system prior to joining the waiting line (i.e., balk) or leave the waiting line because of impatience (i.e., renege) without receiving service. When all permanent servers are busy, it is beneficial to offer portable extra servers in order to meet the demand for the necessary level of service and to cut down on customers' resistance and reneging conduct. Authors in (Patra et al., 2018) steady-state probability for the multi-channel multi-server queue took into account the ideas of balking and reneging. In their work, the authors (Rezaee et al., 2024) covered a variety of topics related to reneging and balking. The analysis of the Markovian process with muti-servers and finite buffer with balking and reneging was done by the authors (Tran-Dang et al., 2023).

The authors in (Yousefpour et al., 2017) established a comprehensive framework for IoT-fog-based applications and presented a delay-minimizing strategy for fog-cloud apps, as well as for fog-capable devices, with the objective of reducing service delay in IoT applications. Subsequently, they created an analytical model to assess the policy and examined how the suggested framework mitigates IoT service latency. The authors in (Goswami et al., 2012) introduced a finite multiserver queueing paradigm with queue-dependent heterogeneous virtual machines, which are characterised as service providers. Cloud computing Service providers may use several virtual machines, with the quantity of active VMs varying based on queue length to mitigate queue duration and waiting time. This enables us to dynamically provision and decommission VMs for scalability purposes. The M/M/c/K queueing model is used by the authors (Dash et al., 2024) to characterise the fog system, determining the ideal number of VMs to activate in the fog layer for the efficient processing of offloaded jobs. This approach optimises the required number of VMs in the fog layer and minimises the waiting time for delay-sensitive task queues.

A. Motivation

The additional VM to a fog node can improve the processing power and resources at the network's edge in fog computing. Organizations may further disperse the burden and enable more localised data processing and analysis by putting an extra VM on a fog node. As a result, processing data-intensive operations may be done with greater responsiveness, decreased latency, and higher efficiency. The additional VM can be configured with certain software or apps that are specifically designed to fulfill the needs of the edge computing environment. It may perform a number of functions, including real-time analytics, hosting edge services, data filtering or aggregation, and enabling more IoT devices. A fog node's capabilities are increased by adding an additional VM, allowing for more localised processing and boosting the fog computing infrastructure's overall performance and responsiveness.

B. Contributions

The major contributions of the article are as follows:

- The impatience behaviour of the client requests due to long queues has been studied and the M/M/v/K queueing system is investigated with the inclusion of additional VMs and finite capacity FCFS queue.

Various analytical studies have been made to validate the proposed system with numerical results. To visualise the consequences of different parameters, the results are shown as graphs and tables that illustrate the anticipated amount of client requests and the likelihood that additional VMs will be active.

C. Organization

The remainder of the paper is organized as follows. Section 2 provides a description of the model. The results and discussion is depicted in section 3. The conclusion is presented in Section 4.

2. Model Description

The service rate follows the Poisson process with a mean rate of µ by a number of VMs ‘n’; a = n-v additional VMs are available in the fog center where the client arrivals follow the FCFS discipline. The arrival of the client requests follows a Poisson process, where λ is the mean arrival rate. The arrival process is shown in fig. 2. The following policies determine how many VMs are required based on the number of client requests that have been waiting in the waiting line of the fog system:

1. There will be v permanent VMs available in the fog system with client requests ≤ N.

2. An extra VM will be turned on in the system when there are more than N client requests but fewer than or equal to 2N in the fog system. When jN < client requests < (j+1)N, j €[1.. n – v – 1], the fog broker provides j number of additional VMs. When the number of client requests again goes to jN the jth where j € [1... n – v – 1] VM will be turned off.

All ‘a’ number of additional VMs will be available to the fog system when there are greater than ‘N’ client requests in the waiting line of the fog system.

The state-dependent arrival rates of the fog system are:

$$\lambda_i = \begin{cases} \lambda & ; i < v \\ \left(\frac{v}{i+1}\right)^{\gamma} \lambda; \ v \le i < N \\ \left(\frac{v+j}{i+1}\right)^{\gamma} \lambda; \ jN \le i < (j+1)N \\ \left(\frac{n}{i+1}\right)^{\gamma} \lambda; \ (i-v)N \le i < K \end{cases} \tag{1}$$

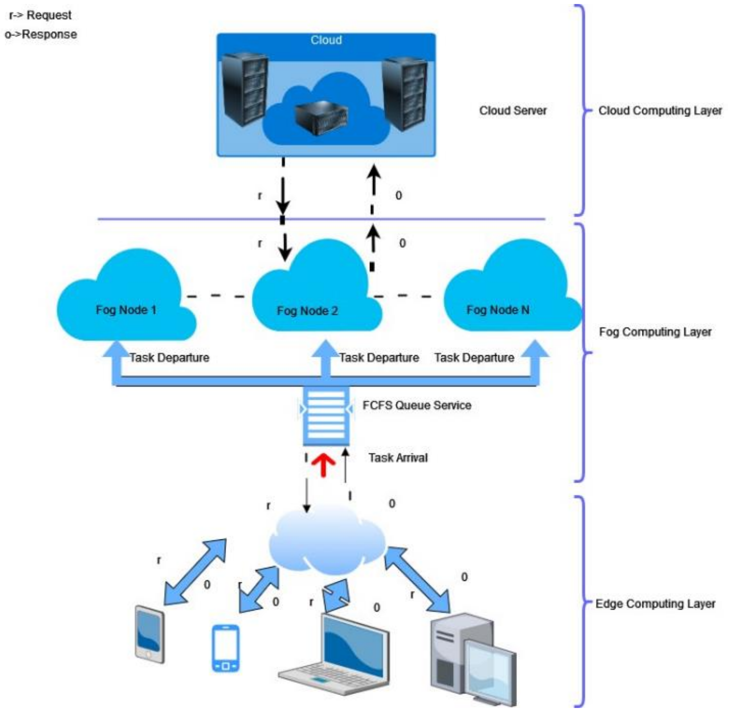

Fig.2. Queueing Model of client request to the fog layer (Patra et al., 2023)

Here γ is the degree of arrival rate affected by the no. of client requests per VM in the fog layer. The client requests balk the system having a probability of $1-\left(\frac{v+j}{i+1}\right)^{\gamma}$

where:

i is the no. of client requests in the fog layer; j € [1... n – v] is the additional VMs that may be scaled up in the system.

The state-dependent service rates of the fog system:

$$\mu_i = \begin{cases} i\mu & ; i < v \\ v\mu & ; v \le i < N \\ (v+j)\mu & ; jN \le i < (j+1)N, j = 1, 2....n-v-1 \\ a\mu & ; (a-v)N \le i < K \end{cases} \quad (2)$$

Let π_i is the steady-state prob. that i number of client requests. From equations (1), (2), & birth-death process:

$$\pi_i = \begin{cases} \pi_0 \dfrac{\rho^i}{i!};0<i\le v \\ \pi_0 \dfrac{\rho^i}{i!}\dfrac{1}{(v!)^{1-\gamma}v^{(i-v)}(1-\gamma)};v<i\le N \\ \pi_0 \dfrac{\rho^i}{i!}(\dfrac{v^{v-1}}{(v-1)!})^{1-\gamma}\dfrac{1}{\prod\limits_{a=0}^{n}(v+a)^{N(1-\gamma)(i-jN)}};\ jN\le i\le (j+1)N \\ \pi_0 \dfrac{\rho^i}{i!}(\dfrac{v^{v-1}}{(v-1)!})\dfrac{1}{\prod\limits_{a=0}^{n-v-1}(v+a)^{N(1-\gamma)}n^{(1-\gamma)(i-n-v)N}};\ (n-v)N\le i\le K \end{cases}$$

(3)

Here ρ is the traffic intensity defined as:

$$\rho = \frac{\lambda}{\mu(n-v)}$$

By the normalization condition when $\sum\limits_{i=0}^{K}\Pi_i = 1$ we get,

$$\pi_0 = [1+\sum_{i=1}^{v}\frac{\rho^i}{i!}+\frac{v^{v-1}}{(v!)^{1-\gamma}}\sum_{i=v+1}^{N}\frac{\rho^i}{i!v^{i(1-\gamma)}} + (\frac{v^{v-1}}{(v-1)!})^{1-\gamma}\sum_{j=1}^{n-v-1}\sum_{i=jN+1}^{(j+1)N}\frac{\rho^i}{i!}\frac{1}{\prod\limits_{a=0}^{n}(v+a)^{N(1-\gamma)}} + (\frac{v^{v-1}}{(v-1)!})^{1-\gamma}\frac{n^{N(n-v)(1-\gamma)}}{\prod\limits_{a=0}^{n-v-1}(v+a)^{N(1-\gamma)}}\sum_{i=(n-v)N+1}^{K}\frac{\rho^i}{i!}\frac{1}{n^{iN(1-\gamma)}}]^{-1} \tag{4}$$

3. Results & Discussion

The expected number of client requests with (n–v) additional VMs is:

$$E(Q)=\pi_0[\sum_{i=1}^{v}\frac{\rho^i}{(i-v)!}+\frac{v^{v(1-\gamma)-1}}{(v!)^{1-\gamma}}\sum_{i=v+1}^{N}\frac{\rho^i}{i!}\frac{1}{v^{i(1-\gamma)}}\ +(\frac{v^v}{v!})^{1-\gamma}\sum_{j=1}^{n-v-1}\sum_{i=jN+1}^{(j+1)N}(v+j)\frac{\rho^i}{i!}\frac{1}{\prod_{a=0}^{j}(v+a)^{N(1-\gamma)(i-jN)}}$$

$$+(\frac{v^v}{v!})^{1-\gamma}\frac{n^{N(n-v)(1-\gamma)+1}}{\prod_{a=0}^{n-v-1}(v+a)^{N(1-\gamma)}}\sum_{i=(n-v)N+1}^{K}\frac{\rho^i}{i!}\frac{1}{n^{iN(1-\gamma)}}]$$

(5)

The probability of the no. of client requests is in between (including the boundaries) (jN+1) and (j+1)N will be

$$\text{Prob}(Q\in[jN+1,(j+1)N]=\pi_0(\frac{v^v}{v!})^{1-\gamma}\sum_{j=1}^{n-v-1}\sum_{i=jN+1}^{(j+1)N}(v+j)\frac{\rho^i}{i!}\frac{1}{\prod_{a=0}^{j}(v+a)^{N(1-\gamma)}} \tag{6}$$

where j=0,1,2…,(n – v – 1).

In the fog layer, the probability of the number of clients requests more than (n – v)N is equal to:

$$\text{Prob}(Q\in[(n-v)N,K]=\text{Prob}(Q\in[aN,K]=\pi_0(\frac{v^v}{v!})^{1-\gamma}\frac{n^{N(n-v)(1-\gamma)-1}}{\prod_{a=0}^{n-v-1}(v+a)^{N(1-\gamma)}}\sum_{i=(n-v)N+1}^{K}\frac{\rho^i}{(i!)^{\gamma}}\frac{1}{n^{iN(1-\gamma)}}$$

(7)

Special Cases:

Case-I

When γ=0, π_i as shown in equation (3) reduced to the M/M/m/K queueing model with the extra VMs.

Case-II

When j = 0, v = n, the outcomes are equivalent to the model with client requests discouragement .

The issue of arrival rate on the average no. of client requests E(N) for constant γ=0.6, μ=1.0, K=50, N=2, and v=2 by varying the no. of VMs from n=3 to 5 (n = a + v, the no. of VMs including the additional VMs 'a') is shown in fig. 3. The graph shows that the average no. of client requests in the fog system increases by increasing the arrival rate for a fixed 'n'. One can find that as 'n' increases E(N) the average no. of client requests decreases.

Fig. 4 shows the Prob. [(aN <= Q < K] vs. λ by varying n from 3 to 5 for fixed values of γ=0.5, K=50, μ=1.0, N=2, and v=2. It shows that as the arrival rate of the client requests to the fog layer increases the probability also increases and as 'n' decreases the probability decreases.

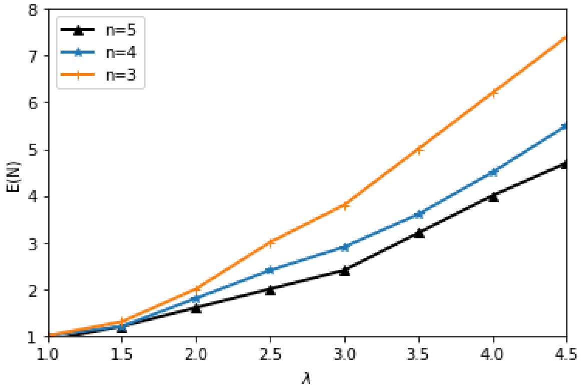

Fig. 3. Average number of client requests Vs Arrival rate

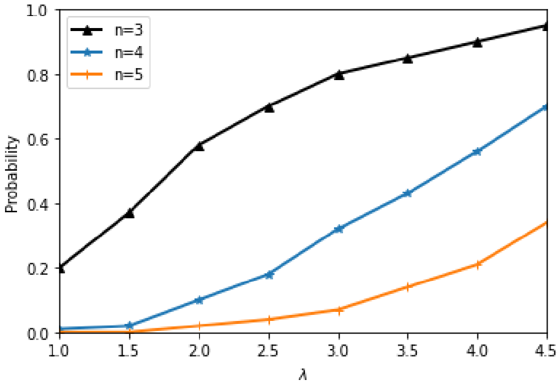

Fig. 4. The Probability of the number of clients requests Prob[aN <= Q <=K] Vs λ

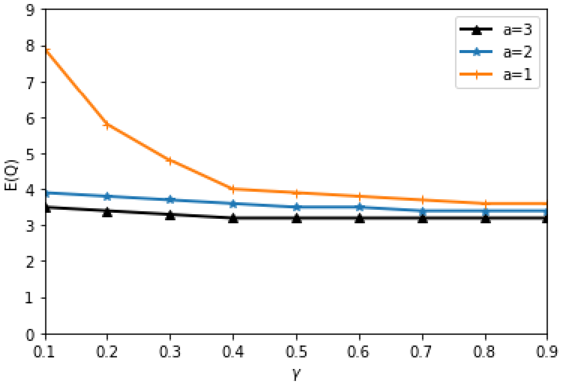

Fig 5. Expected number of client requests Vs γ

The effect of γ on the expected no. of client requests E(Q) for constant λ=3.0, μ=1.0, K=50, N=2, and v=2 by varying the no. of additional VMs from a=1 to 3 is shown in fig. 5. It can be observed that E(Q) in the fog system decreases by increasing γ and subsequently becomes constant. Also, it can be observed that, as the no. of additional VMs 'a' to the system increases E(Q) decreases.

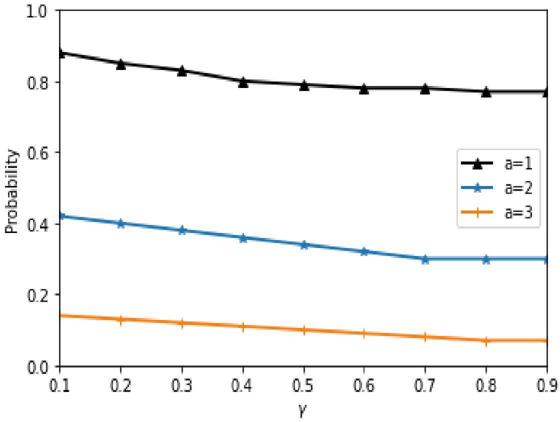

Fig. 6. The Probability of the no. of client requests Prob[aN <= Q <=K] Vs γ

Fig. 6 shows the Prob. [(aN <= Q < K] vs. γ by varying the additional number of VMs ‘a’ from 1 to 3 for fixed values of λ=3.0, μ=1, K=50, N=2, and v=2. It can be seen that as γ increases, the probability decreases and as ‘a’ decreases the probability increases.

TABLE I. PROBABILITY OF JTH VM BEING BUSY

a	ϒ	Prob(Q∈[(jN+1),(j+1) N])		
	j	1	2	3
1	0.1	0.515		
	0.5	0.512		
	0.9	0.510		
2	0.1	0.485	0.199	
	0.5	0.490	0.201	
	0.9	0.494	0.204	
3	0.1	0.382	0.242	0.093
	0.5	0.417	0.226	0.067
	0.9	0.444	0.208	0.048

Table I shows the probability that the jth VM is busy for different values of additional VM ‘a’ for fixed values of K=50, λ= 3.0, μ=1.0, N=2, and v=2. It is observed from the table that as the no. of additional VM increases the probability decreases for a fixed ϒ.

TABLE II. THE EXPECTED NUMBER OF CLIENT REQUESTS AND THE PROBABILITY THAT THE ADDITIONAL VMS ARE BUSY IN THE FOG SYSTEM

ϒ	E (Q)		Prob(Q €[aN,K])	
	n = 4	n = 6	n = 4	n = 6
0.10	7.77	3.18	0.89	0.14
0.20	5.37	3.07	0.85	0.13
0.30	4.43	2.97	0.82	0.11
0.40	3.91	2.90	0.80	0.10
0.50	3.58	2.83	0.79	0.09
0.60	3.34	2.74	0.77	0.08
0.70	3.16	2.71	0.76	0.07
0.80	3.02	2.66	0.75	0.06
0.9	2.90	2.70	0.74	0.06

Table II shows E(Q) for different n and the probability of all VMs being busy for different values of ϒ from 0.1 to 0.9 keeping the fixed values of K=50, λ=3.0, μ=1.0, N=2, and v=2.

4. Conclusion

The storage and processing components of fog computing are located at the cloud's edge and are currently a very effective computing paradigm for scientific computing and corporate IT applications. This work proposed a multi-server finite buffer queueing model for studying the performance of the fog system with additional VMs for reducing the waiting times of the client's request in case of a long waiting time in the buffer. Explicit formulas are obtained to immediately estimate the no. of client requests in the queue at any given time. An important factor in reducing the balking behaviour of requests for sensitive applications is the impact of adding more servers on the waiting time. To show how the system parameters affect the performance measurements, various numerical findings shown as tables and graphs are discussed. In the future, the cost and sensitivity analysis can be carried out in deploying the additional VMs to the system.

References

1. Abou-El-Ata, M. O., & Shawky, A. I. (1992). The single-server Markovian overflow queue with balking, reneging and an additional server for longer queues. Microelectronics Reliability, 32(10), 1389-1394.
2. Behera, S., Panda, N., De, U. C., Dash, B. B., Dash, B., & Patra, S. S. (2023). A task offloading scheme with Queue Dependent VM in fog Center. In 2023 6th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1-5). IEEE.
3. Dash, B. B., Satapathy, R., & Patra, S. S. (2023). Energy efficient SDN-assisted routing scheme in cloud data center. In 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN) (pp. 1-5). IEEE.
4. Dash, B. B., Satpathy, R., & Patra, S. S. (2024). Efficient SDN-based Task offloading in fog-assisted cloud environment. EAI Endorsed Transactions on Internet of Things, 10.
5. De, U. C., Satapathy, R., & Patra, S. S. (2023). Cost Analysis and Optimization of Virtual Machine Allocation in the Cloud Data Center. In 2023 International Conference on Inventive Computation Technologies (ICICT) (pp. 809-813). IEEE.
6. Ghani, N., Sajak, A. A. B., Qureshi, R., Zuhairi, M. F. A., & Baidowi, Z. M. P. A. (2024). A Review of Fog Computing Concept, Architecture, Application, Parameters and Challenges. JOIV: International Journal on Informatics Visualization, 8(2), 564-575.
7. Goswami, V., Patra, S. S., & Mund, G. B. (2012). Performance analysis of cloud with queue-dependent virtual machines. In 2012 1st international conference on recent advances in information technology (RAIT) (pp. 357-362). IEEE.
8. Li, Q., Zhao, J., Gong, Y., & Zhang, Q. (2019). Energy-efficient computation offloading and resource allocation in fog computing for internet of everything. China Communications, 16(3), 32-41.
9. Mukaddis, G. S., & Zaki, S. S. (1983). The Problem of Qucucing Systcm M/M/I with Additional Scrvers for a Longer Oueue. Ind. J. Pure and Appl. Maths, 14(37), 345354.
10. Mukherjee, M., Shu, L., & Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Communications Surveys & Tutorials, 20(3), 1826-1857.
11. Murari, K. "An Additional Special Channel, Limited Space Queuing Problem with Service in Batches of Variable Size." Operations Research 16, no. 1 (1968): 83-90.
12. Patra, S. S. (2018). Energy-efficient task consolidation for cloud data center. International Journal of Cloud Applications and Computing (IJCAC), 8(1), 117-142.
13. Patra, S. S., Mittal, M., Jude Hemantha, D., Ahmad, M. A., & Barik, R. K. (2021). Performance

evaluation and energy efficient VM placement for fog-assisted IoT environment. In Energy Conservation Solutions for Fog-Edge Computing Paradigms (pp. 129-146). Singapore: Springer Singapore.

14. Rezaee, M. R., Hamid, N. A. W. A., Hussin, M., & Zukarnain, Z. A. (2024). Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks and SDN Application. IEEE Access.
15. Tran-Dang, H., & Kim, D. S. (2023). Disco: Distributed computation offloading framework for fog computing networks. Journal of Communications and Networks, 25(1), 121-131.
16. Yousefpour, A., Ishigaki, G., & Jue, J. P. (2017, June). Fog computing: Towards minimizing delay in the internet of things. In 2017 IEEE international conference on edge computing (EDGE) (pp. 17-24). IEEE.