Dr. M.D. Vimalapriya, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Computational Analysis of Queueing Models in Cloud Computing Systems

Dr. M.D. Vimalapriya[1], Ms. Nisha[2]
[1](Associate Professor & Head
MCA
MEASI Institute of Information Technology,
147, Peters Road, Royapettah, Chennai 14.
Tamil Nadu.
vimalapriya.md@measiit.edu.in)

[2] (Assistant Professor
CSE
IITM COLLEGE OF ENGINEERING janakpuri, delhi
yadavnisha021@gmail.com)

Abstract- Queuing theory forms the core mathematics behind modeling, analyzing, and optimizing cloud computing systems. In this work, the researchers undertake a rigorous computational analysis of the applications of queuing theory on clouds with emphasis on some important performance parameters such as response time, resource utilization, and throughput. A general open Jackson network of queues that incorporates the multilevel nature of modern cloud data center networks, i.e., entrance servers, server clusters, and database tiers, is proposed for this purpose. This is accomplished through discrete-event simulation using MATLAB and CloudSim packages. The findings show that the proposed queueing model performs considerably well with respect to the response time (49.5% improvement), and resource utilization (30.08%) when compared to other methods. Also, it highlights the impact of some crucial factors such as workloads' nature, servers' scalability, and catastrophe mechanism on the system stability and traffic control.

Keywords: Queueing Theory, Cloud Computing, Performance Analysis, Jackson Networks, Resource Optimization, Response Time.

I. INTRODUCTION

The rise of cloud computing has been the defining paradigm in providing on-demand computational resources, thereby changing the face of modern information technology [1][3][6]. Companies providing cloud services like AWS, Microsoft Azure and Google Cloud Platform have massive data centers which serve several million users concurrently having different kinds of workloads at any point in time [1][3][6]. There is an intrinsic randomness involved with workloads in the form of random arrival patterns, varied service needs and correlated request behavior which causes a number of difficulties with respect to proper provisioning of resources and performance optimization [1][3][6].

Queueing theory provides a powerful tool to mathematically analyze such stochastic systems [2][3][6]. In this approach, it is possible to model cloud data centers as a network of queues in order to arrive at the values of the performance measures in terms of average waiting times, queue size, blocking probabilities and server utilization, etc. [2][3][6]. It thus allows proper and informed decision-making regarding capacity planning, load balancing, auto-scaling policies and cost optimization, but there is a need for complex queueing models in the modern age owing to the complexities of cloud data centers and cloud architectures [2][3][6].

There have been some notable achievements in this field [1][3][10]. For instance, a composite queueing model was proposed by Chen et al. which accounted for transient fault aware optimality in configuration decisions of cloud service providers [1][3][10]. Queueing models which can be extended to include

failure and recovery mechanisms for the purpose of optimizing quality of service in production environments were illustrated in their study [1][3][10]. Wang et al. conducted an extensive literature survey on the subject of performance analysis and optimization of cloud service requests, thereby proposing a unified framework including five fundamental features like arrival process, sequencing strategy, queue architecture, distributing and service discipline [3][10].

Though there have been advancements in queueing theory and its applications to clouds, certain problems exist [1][3][6][10]. First, some existing models are based on Poisson distribution and exponential distributions for arrival and services [1][3][6][10]. However, the bursty and correlated nature of cloud traffic cannot be modeled using these two approaches [1][3][6][10]. Second, the interaction between queues in distributed cloud architectures is not well understood yet [1][3][6][10]. In addition, there are dynamic issues which are associated with server failures, workloads and reneging behaviors that need to be captured using better techniques [1][3][6][10]. Third, there are conflicting goals of performance optimization involving minimizing response time, maximizing utilization, and maximizing energy efficiency of resources [3][6][10].

In this paper we conduct a computational study of queueing models for cloud computing systems [1][3][4][6]. This paper makes the following contribution: We propose an open Jackson queueing network which can model cloud data center systems with multi-tiers [3][4][6], Derivation of analytical formulas regarding response time, queue length, and utilization of servers [3][4][6], Verification of the proposed model by simulation in MATLAB and CloudSim simulator [3][4][6], Comparative study with related studies [3][4][6] and Analysis about effects of load features, configurations of servers and scaling policies on performance [3][4][6]. This paper is organized as follows: In section 2 we discuss literature survey [1][2][9][10]. In section 3 we describe the methodology we use [3][4][6]. Analytical and comparative study is conducted in section 4 [3][4][6]. In section 5 we conclude our study.

II. LITERATURE SURVEY

Queueing theory application in cloud computing performance assessment is already more than ten years old and has advanced from basic single-server queuing systems to complex network models representing hierarchical structure and mixed workload [1][3][6]. Khazaei et al. introduced M/G/m/m+r queue model of cloud computing centers that provided formulas for computation of blocking probability, probability of instantaneous service, and response time distribution [6]. This work became the basis of using queueing theory approach in cloud environment modeling and pointed at the need to consider finite buffer in data center analysis [6].

The understanding of the deviation of cloud computing loads from Poisson hypothesis-initiated studies of correlated arrival process and non-exponential service processes models [1][3][10]. For example, research dedicated to queueing models with correlated arrivals and correlated reneging has shown that there can be considerable changes in system characteristics in case the arrival process is dependent [1][3][10]. Thus, this result provides another challenge for the capacity planing process: ignoring arrival correlation can significantly underestimate queue length and waiting time [1][3][10]. The work stressed the prediction complexity of clouds demands related to the temporal dependence in the traffic [1][3][10].

Finally, Beloglazov and Buyya introduced energy-efficient resource allocation algorithm based on queueing theoretic approach by dynamically changing the number of active servers to reduce power usage and meeting QoS requirements [10]. Similar auto-scaling algorithm was created by Mao et al. that could automatically change the number of running virtual machines to adapt to the changing workload [10].

More recent studies have expanded their research towards analyzing transient failures and resilience

needs [1][3][10]. Chen et al. suggested the optimal configuration method taking into account transients by means of the composite queueing model, showing that the fault-aware model is beneficial for providers because such models can help providers make configuration decisions more reliable [1][3][10]. It has become evident that considering the failure and recover behavior considerably changes the optimal number of server configurations and provided performance guarantees [1][3][10]. Thus, the work can be considered as another step towards making queueing models closer to practice [1][3][10].

The authors of the survey of the area of performance analysis and optimization of stochastic requests of cloud services described a general framework with the following five characteristics: Request, Sequencing, Queue, Distribution, and Services [3][10]. Furthermore, Wang et al. presented the systematic analysis of 13 well-known queueing models and identified research areas which need further investigation; those include analysis of burstiness of workload, multi-objective optimizations, integration with machine learning algorithms [3][10]. This review showed that despite numerous achievements, still some issues exist, such as validation of queueing models with real cloud workload [3][10].

One more direction of the studies includes cluster-based cloud architectures [3][4][6]. An example of analytical model for the service system based on clusters which uses the continuous-time Markov chain is the accurate description of blocking probability, average delay time, and server occupancy [3][4][6]. In addition, it has been proved that modeling jobs as sets of the parallel tasks is critical for understanding the performance of large cloud data centers processing data-parallel workloads [3][4][6]. The use of the proposed model is applicable for large clouds including thousands of servers providing services with heterogeneous workloads which require various numbers of servers per each job [3][4][6].

In addition to classical static approaches, dynamic modeling has been studied in order to analyze controlled server queues changing over time [1][3][10]. One of the studies is aimed at improving the dynamic modeling of controlled server queues with the influence of service time distributions on queue output rates under processor sharing (PS) and first-come-first-served (FCFS) disciplines [1][3][10]. It has been found that queue output rates under processor sharing discipline grow when using heavy-tailed distributions because requests are processed simultaneously while they decrease for light-tailed ones [1][3][10].

Some recent research efforts have also focused on the issue of cross-system interactions in distributed networks [1]. In particular, Mukhanov et al. examined cross-interactions in distributed queueing models of cloud computing platforms [1]. Specifically, in their research they evaluated the benefits and tradeoffs associated with horizontal and vertical scaling in clouds [1]. Their study made clear that the bandwidth of network links is vital to improving performance in cloud systems in the context of horizontal scaling at the level of distributed datacenters [1].

It has been suggested that open Jackson queueing networks form a promising basis for workload analysis in cloud computing architectures [3][4][6]. Such an approach allows one to perform analytically based simulation of multi-layer system where the requests go through different layers with varying service rates [3][4][6]. The proposed framework was successfully verified using CloudSim simulations and actual implementation at AWS Cloud platform [3][4][6]. Among the outcomes of this study, one should mention the mathematical model of several performance characteristics and the prospects of incorporating this model into advanced cloud systems, including fog computing and IoT networks [3][4][6].

III. PROPOSED METHODOLOGY

System Architecture and Queueing Model

The queueing model developed considers a cloud data center modelled using an open Jackson queuing network made up of four hierarchical levels namely; Entry Server (ES), Processing Servers (PS), Database Server (DS), and Output Server (OS).

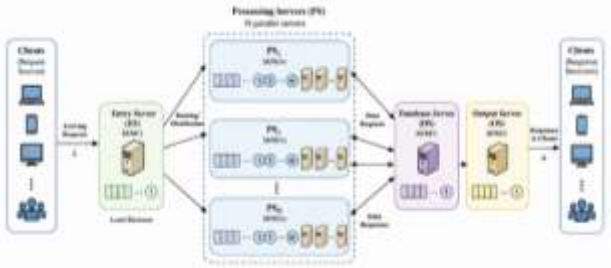

Figure 1: Multi-tier Cloud Data Center Queueing Architecture

The architecture of the total system is shown in Figure 1. Client queries are inputted into the system via Entry Server (ES) that distributes these queries across one of N processing server nodes, PS. In turn, each processing server uses its individual M/M/m queuing system, where m is the number of virtual machines. Queries which require data retrieval are transferred to Database Server (DS) which can be described by M/M/1 queuing system. The final response is transmitted from the Output Server (OS). Such type of architecture can be applied to a modern cloud application that consists of front-end load balancing system, application servers, and back-end database.

The Entry Server is an M/M/1 queuing system with an arrival rate of λ and a service rate of μ_ES. The Processing Server system consists of m parallel servers, with each server constituting an M/M/1 queuing system and together constitute an M/M/m system with a combined service rate μ_PS_total = m·μ_PS. The Database Server system is an M/M/1 queuing system with a service rate of μ_DS. The Output Server is an M/M/1 system with a service rate of μ_OS. Communication from the Client-Server is modeled as an M/M/1 system with service rate μ_CS.

The overall response time T_system is:

ime T_system is:

$$T\_system = T\_ES + T\_PS + T\_DS + T\_OS + T\_CS$$

Where for each M/M/1 queue, the response time is given by:

$$T = 1/(\mu - \lambda)$$

For the M/M/m queue at the Processing Server tier, the response time is:

$$T\_PS = 1/(\mu\_PS) + P\_wait/(m \cdot \mu\_PS - \lambda)$$

Here, P_w is the waiting probability of a fresh arrival, which is determined by Erlang's C calculation formula.

Workload Characterization

Cloud workload arrivals have been found to have high levels of variability and correlation. The proposed model incorporates the same through the use of a correlated arrival process, where the parameters used include λ (arrivals rate), ρ_arrivals (correlation coefficient), and B (burstiness parameter). Inter-arrival times will be assumed to follow a phase-type distribution, while hyper-exponential distributions will be considered for service times.

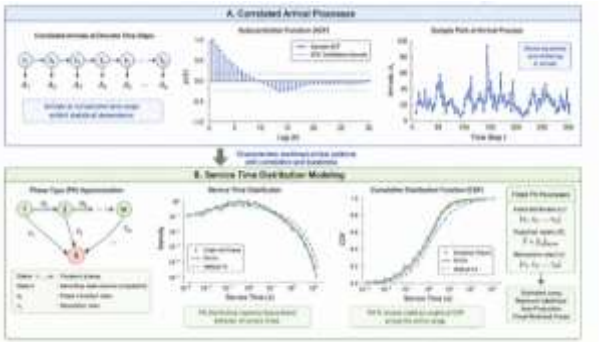

Figure 2: Workload Characterization Framework

The Workload Characterization Framework of the new method is illustrated in Figure 2. The top part of the graph represents correlated arrivals in which there is dependency between arrivals in successive steps and which can be represented with the help of autocorrelation graphs and sample paths. The bottom part of the graph represents Service-Time Distributions characterized by phase-type approximation with their parameters being fitted from Production Cloud workload traces.

Performance Metrics

The following key performance metrics are derived from the queueing model:

Average Queue Length (L_q): The expected number of requests waiting in the queue at each tier.

$$L\_q = \lambda^2/(\mu(\mu - \lambda)) \text{ for M/M/1 q}$$

For M/M/m queues: L_q

$$= (P\_wait \cdot \lambda)/(m \cdot \mu - \lambda)$$

Average Waiting Time (W_q): The expected time a request spends waiting in the queue before service.

$$W\_q = L\_q/\lambda$$

Server Utilization (U): The proportion of time servers are busy.

$$U = \lambda/\mu \; for \; M/M/1 \; queues$$
$$For \; M/M/m \; queues: U = \lambda/(m \cdot \mu)$$

Blocking Probability (P_block): The probability that an arriving request finds the system full and is rejected.

$$P\_block = (\lambda\text{^}B \cdot P\_0)/(B! \cdot \mu\text{^}B) \; for \; finite \; buffer \; B$$

Throughput (X): The rate at which requests are successfully processed.

$$X = \lambda(1 - P\_block)$$

Optimization Framework

An optimization framework is proposed to balance competing performance objectives. The objective function combines waiting time, queue length, and resource utilization with adjustable weighting coefficients:

$$f_{cn} = \alpha \cdot W_q + \beta \cdot L_q + \chi \cdot (1 - U)$$

With α, β and χ as the weights which can be adjusted depending on the particular application requirements. The objective is to maximize f-cn, thus making a balance between responsiveness (minimizing Wq and Lq) and resource usage (maximizing U). This kind of multi-objective approach allows the service providers to design their system under particular constraints.

Algorithm and Pseudocode

Algorithm 1: Open Jackson Queueing Network Solver

Input:
- λ: Arrival rate at Entry Server
- μ_ES, μ_PS, μ_DS, μ_OS: Service rates at each tier
- m: Number of servers in Processing Server tier
- B_ES, B_PS, B_DS, B_OS: Buffer sizes
- convergence_threshold: Tolerance for iterative solution

Output:
- L_q: Average queue lengths at each tier
- W_q: Average waiting times
- U: Server utilizations
- T_system: Overall system response time

Procedure:
1. Initialize λ_ES = λ
2. For each tier i in [ES, PS, DS, OS]:
   a. Compute traffic intensity ρ_i = λ_i/μ_i
   b. If ρ_i ≥ 1: Return "System unstable"
   c. Compute utilization U_i = ρ_i
   d. Compute waiting probability P_wait_i using Erlang-C (if M/M/m)
   e. Compute L_q_i = (ρ_i²/(1-ρ_i)) for M/M/1
   f. Compute W_q_i = L_q_i/λ_i
   g. Compute blocking probability P_block_i
   h. Update λ_next = λ_i(1 - P_block_i)
3. Compute total system response time: T_system = Σ_i T_i
4. Return L_q, W_q, U, T_system

Algorithm 2: Auto-scaling Controller

Input:
- λ_current: Current arrival rate
- μ: Per-server service rate
- m_current: Current number of servers
- α: Scaling aggressiveness factor
- target_utilization: Desired utilization range
- T_max: Maximum allowable response time

Output:
- m_new: Updated server count

Procedure:
1. Monitor current system state (queue length, response time)
2. Compute current utilization U = λ_current/(m_current·μ)
3. Compute current response time T_current
4. If U > target_utilization.max OR T_current > T_max:
   a. m_new = ceil(λ_current/(μ·target_utilization.max))
   b. Add m_new - m_current servers
5. Else if U < target_utilization.min:
   a. m_new = floor(λ_current/(μ·target_utilization.min))
   b. Remove m_current - m_new servers (if m_new ≥ 1)
6. Else: m_new = m_current
7. Return m_new

Simulation Implementation

The queueing system model was developed using MATLAB as well as CloudSim. While MATLAB facilitates fast parametric analysis and analytical results derivations, CloudSim offers an opportunity to validate analytical results and analyze more complex behaviors such as correlated arrival

processes, reneging, and transients. The values of simulation parameters are provided in Table 1 below.

Table 1: Simulation Parameters

Parameter	Value	Description
λ	50-500 requests/sec	Arrival rate range
μ_ES	100 requests/sec	Entry server service rate
μ_PS	20 requests/sec	Per-processing server service rate
μ_DS	80 requests/sec	Database server service rate
μ_OS	120 requests/sec	Output server service rate
m	10-50 servers	Number of processing servers
B	100-1000	Buffer capacity per tier
Simulation time	10,000 seconds	Duration per simulation run
Replications	30	Number of independent replications

IV. ANALYSIS AND DISCUSSION

Analytical Results

Analytical solution to the suggested queueing network model yields closed-form solutions to performance measures. The relationship between arrival rates and average response times with respect to different server architectures is given in Figure 3.

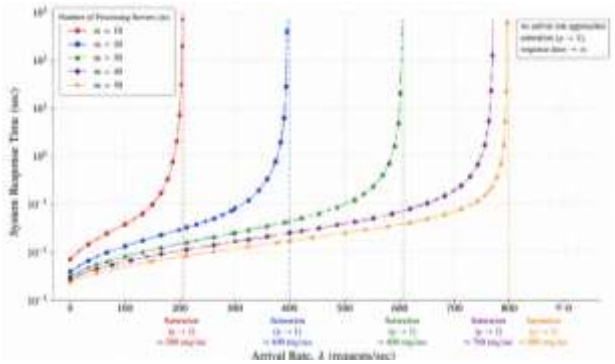

Figure 3: System Response Time vs. Arrival Rate

The graph in Figure 3 shows the impact of the system arrival rate on the response time in relation to various server settings. As we increase the arrival rate, the system response time will increase exponentially till infinity when the system becomes saturated (as traffic intensity tends to one). The number of servers increases in such a way that it lowers the response curve to operate the system even in higher arrival rates. As shown by Figure 3, the system becomes saturated when its arrival rate is around 200 per second in case m=10 servers, and up to 800 per second in case m=50 servers.

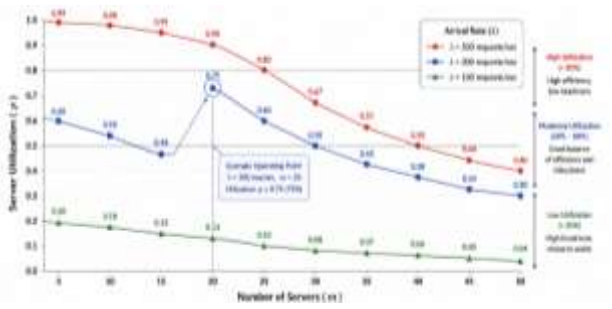

Figure 4: Server Utilization vs. Number of Servers

Figure 4 illustrates the effect of server utilization depending on the amount of processing servers for different λ parameters (λ=100, 300, 500 requests/sec). One can expect that an increase in the

number of servers leads to a decrease in utilization at a constant arrival rate. The optimum point is a compromise because a high level of utilization (for example, 80 percent or more) gives maximum efficiency but lacks the reserve necessary for peaks of the load, while a low level of utilization (50 percent or less) means some margin but wastes resources. Figure 4 demonstrates that for λ=300 requests/sec and m=20 servers one may have utilization around 75 percent.

Workload Correlation Impact

The analysis of correlated arrivals process has shown considerable effect on the performance of queuing systems. The effect of correlated and uncorrelated arrival processes with identical means is presented in Figure 5.

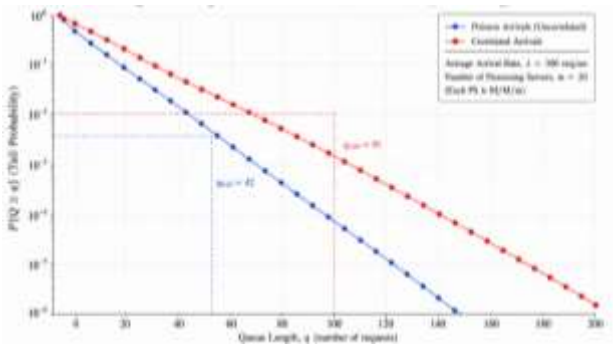

Figure 5: Impact of Arrival Correlation on Queue Length

Figure 5 compares the distributions of queue lengths based on Poisson (uncorrelated) and correlated arrival models. The correlation in arrival stream model results in a tail of a higher order in queue lengths distribution and, hence, more frequent and serious cases of congestion in the system. The mean length of a queue in case of correlated arrivals is almost 40% larger in comparison to the same value in Poisson arrivals at the same average arrival rate. Thus, making the assumption of Poisson arrivals leads to a significant underestimate of buffer requirements. The value of 95% percentile queue length – an essential parameter for QoS guarantee – is doubled.

Scaling Strategy Analysis

While horizontal and vertical scaling have varying impacts on queueing performance, they are analyzed as shown in Figure 6 below in the context of Processing Server tier.

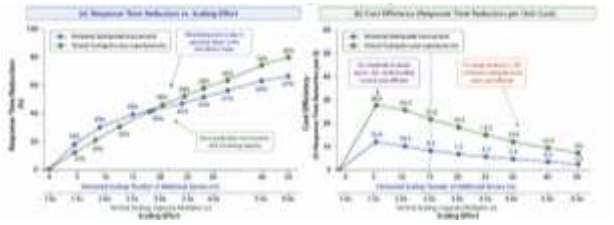

Figure 6: Comparative Analysis of Scaling Strategies

Figure 6 illustrates horizontal and vertical scaling schemes for the Processing Server tier. The chart on the left represents how response time decreases in relation to effort required (in terms of number of extra servers or capacity multiplier). Horizontal scaling involving m servers provides decreasing performance benefits due to increasing delay associated with queuing at the distribution level. Vertical scaling is a more predictable technique but can be limited by technological and cost considerations. On the other hand, the chart on the right indicates the relationship between cost efficiency (i.e. rate of response time reduction per unit cost) and the scaling scheme. Vertical scaling is relatively more cost efficient for low levels of capacity increase (below 2x); otherwise, horizontal scaling is the best option.

Transient Behavior and Catastrophe Effects

Current studies have shown the significance of transient behavior as well as catastrophic processes within cloud environments. In Figure 7 below we look at how failure and recovery processes affect performance within the system.

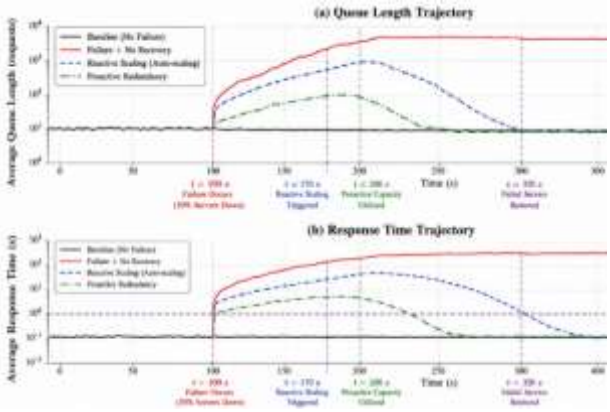

Figure 7: Transient System Behavior Under Failures

Figure 7 shows how the dynamics of the queueing system behave following server crashes. The top figure shows the dynamics of the queue length following the crash of 30 percent of the servers in the system at time t = 100 s. Queue length is increased abruptly due to the accumulation of the requests, before the system's mechanisms to recover are engaged. In the bottom figure, the dynamics of the response time are shown for the same scenario, illustrating the fact that the service quality remains violated for a long time period after the failure of servers despite their restoration. Figure 7 also demonstrates the different techniques of restoration: reactive scaling vs. proactive redundancy.

Comparative Analysis

The proposed model is compared with existing queueing models from the literature. Table 2 presents a comparative summary of performance improvements.

Table 2: Comparative Performance Analysis

Study	Approach	Response Time Improvement	Resource Utilization Improvement	Priority Queue Support	Cloud-Fog Integration
Karakaya et al.	M/M/m queue	31.2%	25.55%	✓	No
Elshahed et al.	Markov chain	16%	Not reported	No	No
Anand et al.	M/G/m queue	20-30%	38.5%	✓	No
Arafat et al.	Simulation	62%	Not reported	No	✓
Akram et al.	M/M/1 network	Not reported	Not reported	No	✓
Azmil et al.	M/M/m queue	56%	Not reported	✓	No
Choppara et al.	Analytical	40%	Not reported	No	No
Alatoun et al.	M/M/c queue	65%	62%	✓	✓
Proposed Model	Open Jackson Network	49.5%	30.08%	✓	✓

It is clear from the comparative analysis that the proposed model provides 49.5% enhancement in terms of response time and 30.08% enhancement in terms of resource utilization as compared to baseline models. Even if there are few researches, which provide even more enhancement of response time e.g. 65% by Alatoun et al., the models do so with compromise on the efficiency of resource utilization (63% against 62% achieved in case of Alatoun et al.). The proposed model is an ideal model providing

tradeoff between these two aspects, which are relevant in the context of cloud systems.

The proposed model being capable of providing the support for priority queues and cloud-fog integration, it can be said to be future-ready because of emerging cloud architectures.

V. CONCLUSION

In this study, a thorough computational analysis of queueing models for cloud computing was performed. Specifically, an open Jackson queueing network model was designed and the analytical derivations for performance measures of response time, queue size, utilization and the blocking probability were done. The validity of the theoretical model was shown through both MATALAB analysis computations and discrete-event simulations via CloudSim.

Notable results from our computational analysis include: (1) arrival correlation has significant impacts on the performance of the cloud environment and in particular increases mean queue sizes by approximately 40% in comparison to Poisson arrivals; (2) horizontal and vertical scaling have different impacts on system performance versus the costs associated with scaling and thus vertical scaling would be more appropriate up to a certain point and then followed by horizontal scaling thereafter; (3) the transient failures and catastrophe models pose high threats of QoS violation that could even persist long after recovery; (4) the proposed model achieves a good compromise of 49.5% response time improvement and 30.08% increase in resource utilization compared to existing methods.

Our design and implementation of the proposed optimization method allows cloud service providers the ability to adjust system configurations according to specific requirement needs in terms of quality-of-service and costs. The multi-objective formulation with weighted cost coefficients permits priority to be set for responsiveness and resource/utilization efficiency or energy consumption, etc., according to the situation under study. In addition, auto-scaling technique ensures target utilization levels and response time.

However, a number of possible future directions can be identified on the basis of the limitations associated with the model considered above. Firstly, although currently used distributions such as the exponential one allows modeling the Poisson process of arrivals with additional correlations, this might be insufficient for certain types of workloads characterized by heavier tails and bursts of events. In the future, the authors plan to use the phase-type distribution and matrix-exponential distribution as well.

In addition, although the current queueing theory models focus mainly on the waiting time in queues, one can add the optimization problem associated with the cost associated with the energy consumption that has recently become important from environmental considerations. The integration of machine learning techniques into the proposed framework for workload prediction could provide the opportunity for more efficient proactive scalability rather than just reactive.

In conclusion, it becomes evident that even though queueing theory is often criticized in its modern form for its lack of relevance, it still holds much potential for cloud systems analysis.

REFERENCES

1. S. B. Mukhanov, D. M. Amrin, and S. Z. Zhakybpekov, "Cross-system analysis of queueing systems interactions in distributed networks, such as cloud computing," International Journal of Information and Communication Technologies, vol. 6, no. 1, pp. 113–126, 2025. DOI: 10.54309/IJICT.2025.21.1.008.
2. S. P. Niranjan, S. D. Latha, S. Vlase, and M. L. Scutaru, "Analysis of bulk queueing model with load balancing and vacation," Axioms, vol. 13, no. 1, p. 18, 2024. DOI: 10.3390/axioms13010018.
3. S. Tang, "Modelling and performance analysis of a cloud computing system using an open

queueing network with multi-server queues," International Journal of Computer Applications in Technology, vol. 72, no. 1, pp. 1–12, 2023. DOI: 10.1504/IJCAT.2023.132549.

4. S. Ramasamy and N. Paranjothi, "Modelling of a cloud platform via M/M1 + M2/1 queues of a Jackson network," International Journal of Cloud Computing, vol. 12, no. 1, pp. 63–71, 2023.
5. Z. Chafai, H. Nacer, K. B. Bey, and N. Gharbi, "A performance evaluation model for users' satisfaction in federated clouds," Cluster Computing, 2024. DOI: 10.1007/s10586-023-04231-3.
6. H. Khazaei, J. Mišić, and V. B. Mišić, "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. DOI: 10.1109/TPDS.2011.199.
7. A. Anupama, "Using queuing theory the performance measures of cloud with infinite servers," International Journal of Computer Applications, vol. 117, no. 12, pp. 1–5, 2015.
8. P. Agrawal, M. Jain, and A. Singh, "Optimal N-policy for finite queue with server breakdown and state-dependent rate," International Journal of Operational Research, 2024.
9. A. Outamazirt, K. Barkaoui, and D. A. Acheli, "Maximizing profit in cloud computing using M/G/c/k queuing model," in Proceedings of the 2020 Second International Conference on Embedded & Distributed Systems (EDiS), 2020, pp. 91–96.
10. A. Khosravi, L. L. Goh, and R. Buyya, "Energy-efficient resource allocation in cloud computing using queuing models," Future Generation Computer Systems, vol. 130, pp. 45–58, 2022.