Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November, 2022, Maldives

A Queueing-theoretic Analysis of the Performance of a Cloud Computing Infrastructure: Accounting for Task Reneging or Dropping

1st Godlove Suila Kuaban
Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
Baltycka 5, 44–100 Gliwice, Poland
gskuaban@iitis.pl

2nd Bhavneet Singh Soodan
Department of Mathematics
Chandigarh University
Punjab, India, 140413
bhavneet.e11868@cumail.in

3rd Rakesh Kumar
Department of Mathematics and Statistics
Namibia University of Science and Technology
Namibia
rkumar@nust.na

4th Piotr Czekalski
Department of Computer Graphics, Vision and Digital Systems,
Silesian University of Technology
Akademicka 16, 44–100 Gliwice, Poland
piotr.czekalski@polsl.pl

***Abstract*—Cloud computing has revolutionized the information technology era. It offers high-speed computing, storage, and ICT resources on demand. One of the significant challenges in cloud computing is the impact of impatient users or request (task) reneging. When a request has been compromised, missed its execution deadline, or depends on other rejected ones, it must be removed from the queue without being processed. The reneging or removal of tasks from queues may trigger the reneging or removal of other tasks that depend of them. We refer to this reneging of requests from the load balancing or computing queues as correlated reneging. We presented the performance analysis of a network of queues that constitute a queueing model of a simplified cloud computing infrastructure. We show the relationship between the load, delay, and probability of buffer saturation (blocking probability). It is seen that at about 80% utilization, a small increase in utilization (due to a small increase in the arrival rate $\Delta\lambda$ for a fixed service rate μ) results in a sharp increase in the delay experienced by the tasks, and on the probability of task rejection or blocking.**

***Index Terms*—Performance analysis, cloud computing, data centre infrastructure, tasks scheduling, and tasks reneging or dropping**

I. Introduction

With the rapid advancement of cloud computing together with other emerging technologies such as Software Defined Networking (SDN), Network Function Virtualization (NFV), Artificial Intelligence (AI), big data analytics, fog/edge computing, and the Internet of Things (IoT), many organizations have adopted cloud computing as an indispensable component of their infrastructure and services. Therefore, the evaluation of the performance of cloud computing systems is of significant importance, and the cloud service providers (CSP) must have deep insights into the relationship between the performance metrics of the cloud computing system and the available resources or design parameters in order to utilize or exploit their infrastructures fully while satisfying the performance requirements of the users [1]. Therefore, Quality of Service (QoS), security, and energy consumption are key constraints in the design and provisioning of cloud computing services and Infrastructure [2].

In the cloud computing paradigm, computing resources, software applications, platforms, and infrastructure are offered to users as a service. Instead of setting up or purchasing ICT resources and infrastructure, the users access existing ones offered by cloud service providers over the Internet. The users pay for the resources or services based on the specified Service Level Agreement (SLA) and the billing model of the service provider. When users are no longer interested in the services, they can terminate them. Therefore, cloud computing enables businesses to quickly access reliable and quality ICT resources and services on demand.

Cloud service providers often desire to size their systems to determine the resources required to meet the service level agreement (SLA) with their customers while minimizing resource and energy costs [3]. They are also interested to understand the relationship between the design parameters and the performance metrics. Queueing theory has been used as an effective tool for the evaluation of the performance of clouding computing infrastructure. Given the characteristics of the interarrival times of tasks into the cloud computing buffers and their service or processing times, queueing theory models are used to estimate performance metrics of interest (delay or blocking probability). In cloud computing, the task arrival rate is dynamically challenging over time, and a feasible model that captures the dynamic arrival of tasks is studied in [4].

Datacentre energy-related costs and environmental impacts have become a subject of interest, and much research is ongoing to find energy-aware resource management strate-

978-1-6654-7095-7/22/$31.00

gies [5]. In order to save or reduce energy consumption, it is desirable to drop requests that are likely to miss their respective deadlines so that the precious resources can be saved for requests that are likely to complete in time [6], [7], and also to create space in the buffers to admit more requests. The dropping of requests before they are serviced is called the request or task reneging [8]. One of the major challenges in cloud computing is the development of strategies to deal with the negative effects of impatient users or request reneging, such as poor throughput, wastage of resources, and unpredictable workloads [9], unpredictable waiting time, and energy wastage. The problem of tasks or request reneging in cloud computing queues has been studied in [2], [6], [9]–[13], but none of these studies considered the analysis of the entire cloud infrastructure. Yu He et al. [14] investigated the problem of maximizing the revenue of cloud service providers in online task offloading in mobile cloud computing. Authors in [15] investigated energy efficiency for data centers. They accomplished improved savings than the legacy algorithm.

In IoT-based cloud computing data centers, Distributed Denial-of-Service (DDoS) attacks constitute a severe threat as a huge number of Internet of Things (IoT) devices can be used to create an army of botnets to overwhelm the cloud computing servers. This kind of cybersecurity attack is more likely as IoT devices can easily be compromised (due to weak security mechanisms designed to minimised energy consumption) and used to lauch large-scale DoS attacks on cloud or fog computing servers. One of the aims of these attacks could be to saturate the buffers so that incoming packets or tasks from legitimate sources will be dropped. One mitigation approach is to use security monitoring tools to detect and drop compromised requests or requests that are likely to fail without being scheduled for execution. The load-balancer can be used for attack detection by analyzing the performance measures, and its effect is mitigated by isolating the victim machine [16] or moving sensitive tasks and VMs to more secure machines.

When the size of the queue of requests grows beyond a certain threshold, queue management mechanisms may be employed to randomly drop low-priority requests (reneging) depending on the Service Level Agreement (SLA) between the Cloud Service Provider (CSP) and users in order to avoid buffer saturation which will result in consecutive dropping of tasks or Denial of Service (DoS). Thus, when a request has been compromised, missed its execution deadline, or depends on other rejected ones, it must be removed from the queue without being processed. The reneging or removal of tasks from queues may trigger the reneging or removal of other tasks that depend of them. We refer to this reneging of requests from the load balancing or computing queues as correlated reneging.

In this paper, We presented the performance analysis of a network of queues that constitute a queueing model of a simplified cloud computing infrastructure. We show the relationship between the load, delay, and probability of buffer saturation (blocking probability). It is seen that at about 80% utilization, a small increase in utilization (due to a small increase in the arrival rate $\Delta\lambda$ for a fixed service rate μ) results in a sharp increase in the delay experienced by the tasks, and on the probability of task rejection or blocking. We present a simplified architectural model of a cloud computing data centre infrastructure in section II. We present a queueing-theoretic analysis of a cloud computing infrastructure: accounting for task reneging or dropping in section III. Section IV contains a discussion of the performance metrics that determine the QoS of Cloud computing services, section V contains modelling of queues of cloud servers, and conclusion in section VI.

II. The architecture of a cloud computing network infrastructure

A simple cloud computing infrastructure consists of a load balancer, cloud computing physical machines, each of which contains virtual machines running in parallel, and storage computer systems, as shown in Figure 1. In private cloud computing infrastructure, the data centre is usually located closer to the users or task sources, while in the case of public cloud infrastructure (like those owned by the prominent cloud service providers, e.g. Amazon, Google, IBM), the data centre is usually located a significant distance away from the users or task sources. The tasks or requests or tasks submitted by the users (both human users or cyber-physical systems such as the IoT devices) at the various access networks (e.g., IoT and wireless sensor networks, the cellular networks (3G/4G/5G), the Internet Service Provider (ISP) access networks, enterprise access networks) are usually transported through the internet core network to the data centres.

The tasks created consists of packets of very small sizes. To ensure bandwidth efficiency, efficient use of network resources, and slight reduction in energy consumption due to traffic overhead, the small electronic packets from the access networks are aggregated to larger packets at the ingress edge node, the larger packets are converted into optical packets and then transmitted through the internet core network to the data centre. At egress edge node of the internet core network the optical packets are dissagregated and then delivered to the data centre network [17].

The packets that constitute the cloud computing are sent to the load balancing server. The tasks may wait at the load balancer if there are queues of tasks at the load balancer as the traffic at the load balancer is random. The load balabcer then schdedule the requests to one of the processing servers. The results from the cloud computations can be sent to the storage devices or sent back to the users.

A Cloud computing queuing model consists of sources of requests, queues of requests waiting to be served and servers that perform the processing of requests as shown in Figure 1. The input sources consist of cloud computing clients that generate requests or tasks. A cloud computing data centre can be abstracted into a number of queueing systems which consist of buffers and servers (scheduling server or processing servers). Tasks that arrive when the servers are busy are stored in the buffers and then processed later on or may be dropped or renege. The task or task can either be rejected, wait in the

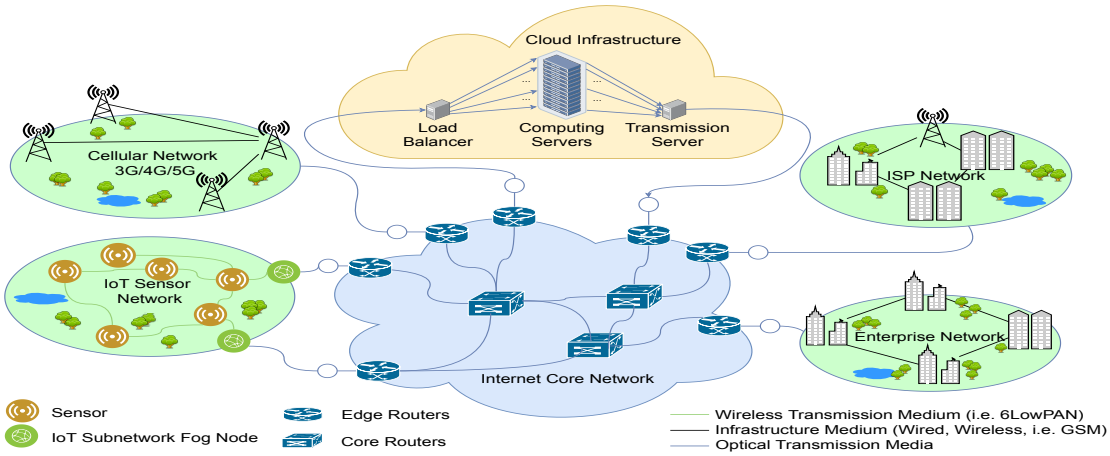

Fig. 1. Cloud computing queuing system model

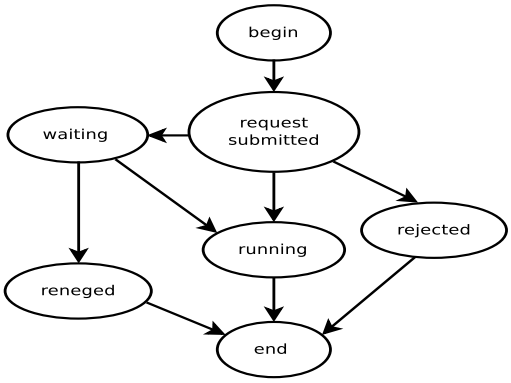

Fig. 2. Request processing state diagram

queue, renege or scheduled for execution (running) as shown in the request processing state diagram in Figure 2.

III. Queueing-theoretic analysis of a cloud computing infrastructure: accounting for task reneging or dropping

It is assumed that the tasks arrive into the load balancer queue according to a Poisson process with the mean rate λ and that the scheduling times (time required to remove a task from the load balancer queue and schedule it to the appropriate physical machine) are exponentially distributed with the mean rate μ_s ($\lambda < \mu_s$ for a stable scheduling queue). When the load balancer buffers are full (that is, the number of tasks in the queue is equal to the buffer size, N), tasks that arrive after that epoch will be rejected. The effective mean arrival rate of tasks into the buffer of the load balancer is

$$\lambda_{el} = \lambda(1 - P_{Nl}) \tag{1}$$

Where P_{Nl} is the probability of task rejection. Any of the tasks admitted into the buffer of the load balancer can be dropped from the queue at any random time, for security reasons, as an active queue management mechanism, or because its execution deadline has expired, when it is still waiting in the buffer. Suppose that the average reneging rate at the load balancer is λ_{rl}, then the mean departure rate of tasks from the load balancer is

$$\lambda_s = \lambda - \lambda_{el} - \lambda_{rl} \tag{2}$$

The load balancer assigns every task to the appropriate physical machine in such a way that some physical machines should not be overloaded while others are idle (load balancing). Task scheduling should also guarantee the QoS (minimise the delay of tasks in buffers and probability of task rejection or dropping). Suppose that the probability that a given task is scheduled to the i^{th} processing machine is r_i, then, the mean rate at which tasks are scheduled to the i^{th} processing machine is $r_i\lambda_s$. The probabilistic task scheduling schemes considered have been discussed in [18]. If the tasks are distributed with equal probabilities among the various M processing machines, then, the scheduling probability, r_i is $1/M$. The authors in [18] proposed a "sensible decision" algorithm which uses a weighted goal function, which consists of the QoS parameters such as the response time or the probability of task losses (due to task rejection or due to buffer overflow or task dropping) at each processing machine. The QoS parameters are continuously measured, and each time the QoS parameter of interest changes, the goal function is updated. Suppose that for an i^{th} processing machine, the recently measured value of the goal function is G_i^*, then, the old value of the goal function G_i is updated as follows:

$$G_i \leftarrow (1 - \gamma)G_i + \gamma G_i^* \tag{3}$$

where the parameter $0 \leq \gamma \leq 1$ is used to adjust the weight given to the most recent measurement of the goal function as compared to its previous value. Therefore, the probability of scheduling a task to the i^{th} processing machine is

$$r_i = \frac{\frac{1}{G_i}}{\sum_{j=1}^{M} \frac{1}{G_i}}, \quad 1 \leq \gamma \leq M \tag{4}$$

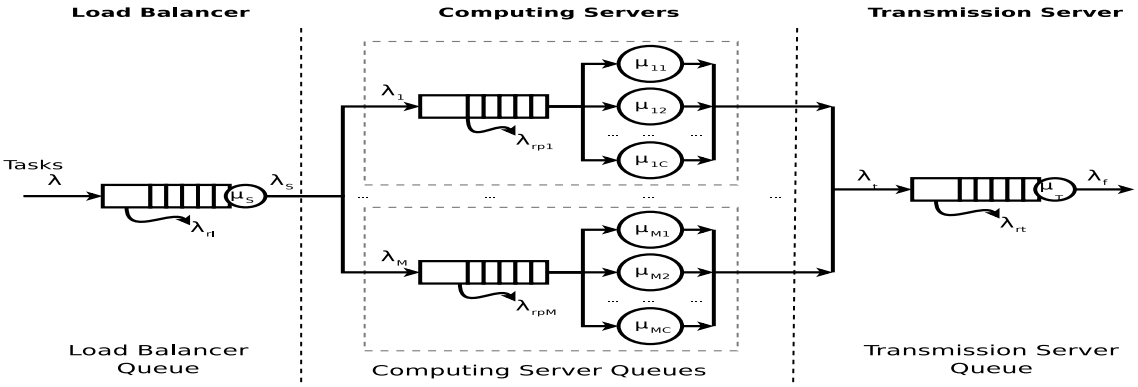

Fig. 3. The cloud computing queuing model

In this scheduling approach, the scheduler may tend to favour those processing machines that provide a better QoS, and could eventually overload them [19]. To avoid this kind of scenario, a multi-variable goal function that takes into consideration number of tasks in the queue, the response time and probability of task losses due to task rejection and task dropping, is

$$G_i = \alpha L_{qi} + \beta R_i + \theta p_i, \quad \alpha + \beta + \theta = 1 \tag{5}$$

where L_{qi}, R_i and p_i are the mean queue size, the mean response time, and the probability of task losses at the i^{th} processing machine respectively. The parameters α, β and θ are the relative importants of L_{qi}, R_i and p_i respectively.

The authors in [18] demonstrated the use of an intelligent scheduling approach, that is based on the random neural network (RNN) with reinforcement learning (RL) concept. For a given task, a RNN is used to determine to which processing machine the task should be assigned in order to ensure a better quality of service while balancing the load among the M processing machines. In this approach, each neuron is assigned to a unique processing machine and all the neurons are fully connected (there are m neurons). The "excitation probability", q_i $(i = 1, 2, \cdots, M)$ for each neuron is calculated as discussed in [18], [19]. At each scheduling moment, the scheduler assigns the task to the processing machine whose neuron has the highest value of the "excitation probability" (q_i). The QoS and the load of the processing machines are continuously being measured, and the goal function is updated. The decision threshold is determined, weights of the neurons are updated, and the "excitation probabilities" for the neurons are recalculated using the new weights. The authors in [18] showed that this scheduling could be reduced to a sensible algorithm (a probabilistic scheduling scheme), where the scheduling probability is

$$r_i = \frac{q_i}{\sum_{j=1}^{M} q_j} \tag{6}$$

The component flows resulting from stochastically splitting Poisson flows still retain the properties of Poisson flows (are still Poisson flows). Therefore, it is assumed that the tasks arrive into the i^{th} processing machine according to a Poisson process with mean rate λ_i, and is given by

$$\lambda_i = r_i \lambda_s + \sum_{j=1}^{M-1} \lambda_j r_{ji}, \quad i = 1, 2, \cdots, N \tag{7}$$

where r_{ji} is the probability of moving tasks from the j^{th} processing server to the i^{th} processing server with a mean rate of λ_j, provided task migration is supported. If the buffer of the i^{th} processing machine is full, incoming tasks will not be admitted, and the effective arrival rate of tasks is

$$\lambda_{ei} = \lambda_i (1 - P_{Ni}) \tag{8}$$

where, P_{Ni} is the probability of task rejection at the i^{th} processing machine.

Each physical machine consists of a set of independent virtual machines (VMs), $\{VM_1, VM_2, VM_3, \cdots, VM_c\}$, that are running in parallel. It is often assumed that the processing times are exponentially distributed, each with mean rate μ_i [9], [13]. However, both the Poisson arrival time and exponentially distributed processing times assumptions are simplifications of the reality [18], but are acceptable for the modelling of cloud computing and web servers [20]. If the average dropping rate of tasks from the queue at the i^{th} processing machine is λ_{rpi}, then, the actual mean rate of task at the i^{th} processing machine is

$$\lambda_{pi} = \lambda_i - \lambda_{ei} - \lambda_{rpi} \tag{9}$$

When the tasks are processed a feedback response may be sent to the user, the results are stored into a database or none of these. The feedback is sent to the user through the transmission server or system. When tasks are removed from the buffers with being processed it degrades the quality of service and could even results in financial loses depending on the SLA between the service priver and users. suppose that the mean arrival rate of tasks to the transmission server is λ_t, with a mean reneging or dropping rate of λ_{rt}, the effective arrival rate is

$$\lambda_f = \lambda_t - \lambda_t (1 - P_{Nt}) - \lambda_{rt} \tag{10}$$

where P_{Nt} is the probability of dropping tasks when a transmission queueing is full.

IV. Performance metrics that determines the QoS of a Cloud computing services

The QoS experienced by cloud computing users is largely influenced by performance parameters such as delays experienced by tasks when they wait in queues, the loss probability due to task dropping reneging, and jitter experienced by tasks that belong to multimedia services such as audio or video streaming.

The probability that a task is lost due to reneging or task dropping can be determined tasking into consideration the probability of losing a task at the queue in the load balancer, any of the processing queues, or transmission server queue.

$$p_l = \left(\frac{\lambda - \lambda_s}{\lambda}\right)\left(\frac{\lambda_s - \sum_{i=1}^{M}\lambda_{pi}}{\lambda_s}\right)\left(\frac{\lambda_t - \lambda_f}{\lambda_f}\right) \quad (11)$$

The total mean delay experienced by a task that is created by a user and it is transported through the access network, core network, load balancer to one of the cloud processing servers where it is processed and no response is required to be send back to the user after processing is

$$R_{Ti} = R_{au} + R_{cu} + R_l + \sum_{i=1}^{M} r_i R_{pi} \quad (12)$$

Where R_au and R_cu are the uplink response time at the access and core networks respectively. They consist of the delays due to packet queueing in the buffers in network equipments and the time required to process and transmit the packets. For high speed optical core networks, the response time R_c is relatively small. R_l is the response time due to queueing and scheduling of tasks at the load balancer and R_{pi} is the response time due to queueing and processing of tasks at the i^{th} processing machine. The response experienced by a user that submits a request or tasks and then expect to receive the results consist of the sum of the up link and down link delays as

$$R_{Ti} = R_{au} + R_{cu} + R_l + \sum_{i=1}^{M} r_i R_{pi} + R_t + R_{cd} + R_{ad} \quad (13)$$

Where R_{ad} and R_{cd} are the down link response time at the access and core networks respectively. R_t is the response time at the trasmission server. The variation of delay causes jitter which degrade the quality of service of multimedia traffic.

V. Modelling of the cloud server queues with the possibility of task reneging or dropping

An important approach in improving the performance and reducing energy consumption in cloud computing is the use of load balancing techniques. Load balancing in cloud computing is a mechanism that detects the overloaded servers and those that are under utilized and then striving to balance the load among them [21]. The authors in [22], introduced an energy aware load balancing approach in which an energy-optimal operation regime is defined and they tried to optimize the number of servers operating within this regime. The authors in [23], used M/M/1/K and M/M/m/K queuing models to estimate the QoS parameters and energy consumption of a cloud data center that is composed of a load balancer and a set of physical machines.

The the steady-state and transient-state analysis of the load balancer with correlated reneging was presented in [12]. Correlated reneging implies that when a task renege or is dropped from the queue, other tasks that depend on it are also dropped. The steady-state and transient-state queueing models with simple reneging were studied in [13], and were later extended to consider correlated reneging of tasks in [2]. The focus of this paper is to present a queueing network analysis of a cloud computing infrastructure.

The results presented in this paper are based on the solution of the queueing models presented in [12] and [2]. The Chapman-Kolmogorov equations for continuous-time Markov chains in [12], and [2] are solved using the Runge-Kutta method to obtain the state probabilities. From the state probabilities of the number of customers present in the queue, the mean queue size and the mean delay are determined and plotted against the load or utilisation $\frac{\lambda}{\mu}$. The probability of rejection or blocking is the state probability $P_{N,r}(t)$, where N is the buffer size ($r = 1$ reneging occurred in the previous at previous transition mark; otherwise $r = 0$). $P_{N,r}(t)$ is obtained from the solution of Chapman-Kolmogorov equations for continuous-time Markov chains in [12] and [2].

We use the queueing models discused in [12] to study the relationship between the load and the performance metrics at the load balancer as shown in Figs. 4 and 5. The values of parameters used are: $P_{80,0}(0) = 1, \mu = 1100, \xi = 0.5, p_{00} = 0.8, p_{01} = 0.2, p_{10} = 0.7$, $p_{11} = 0.3, t = 3$, and $N = 100$. For a fixed $\mu = 1100$ task per second, the variation of the delay and probability of dropping tasking because the buffers are full increases with the load or utilization, $\varrho = \frac{\lambda}{\mu}$. At $\varrho = 80\%$ utilization, a small increase in utilization by a small increase in the arrival rate $\Delta\lambda$ will results in a large increase in the delay and and probability of tasks rejection or blocking. At 80% utilization, arriving tasks could be redirected to an alternative or backup load balancer to reduce delays and task rejection or blocking.

We use the queueing models discused in [2] to study the relationship between the load and the performance metrics at the cloud computing processing queue as shown in Figs. 6 and 7. The values of parameters used are: $\mu = 60, c = 10, \xi = 0.3, p_{00} = 0.8, p_{01} = 0.2, p_{10} = 0.7, p_{11} = 0.3, P_{1,0}(0) = 1, t = 3$, and $K = 50$.. The relationship between the load at a process queue is not linear like the case in the load balancer discussed above.

VI. Conclusion

As a result of the stochastic nature of the arrival times of tasks into cloud computing servers and the processing time for each task, some tasks have to wait in buffers. Tasks could be removed or dropped from the buffer due to the user's impatience, missing execution deadline (for deadline constraint tasks), security reasons, or an active queue management strategy. We have presented the performance analysis of a network

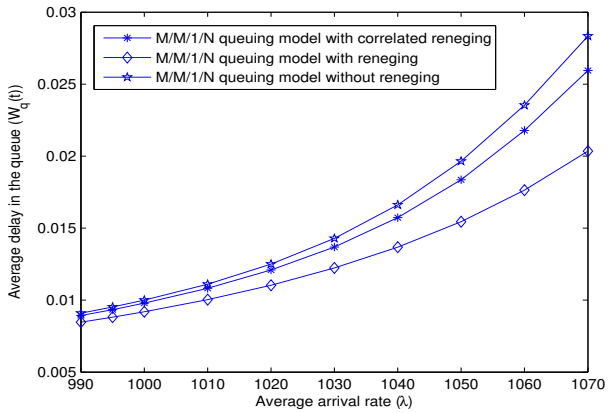

Fig. 4. Variation of the average delay in the load balancer queue, $(W_q(t))$ with the average arrival rate (λ)

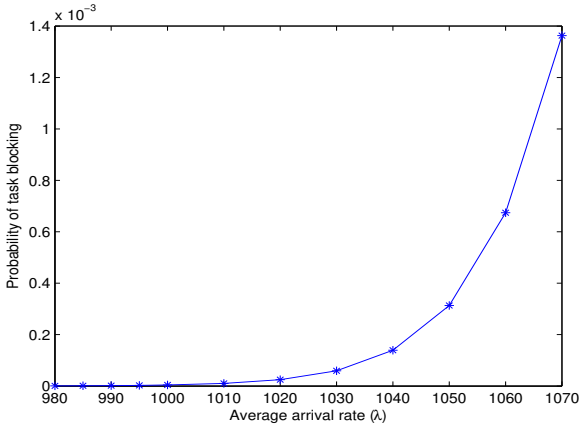

Fig. 5. Variation of the probability of task blocking at the load balancer queue with the average arrival rate (λ)

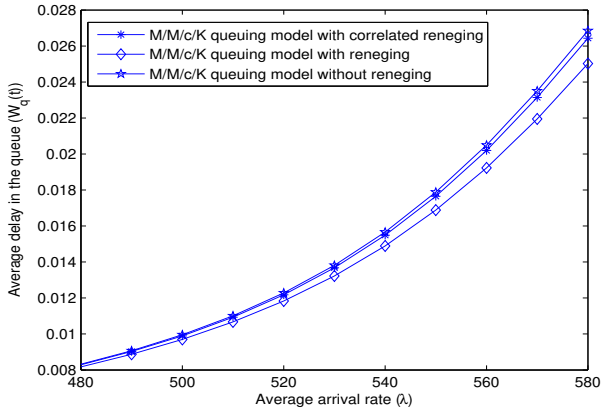

Fig. 6. Variation of the average delay at the i^{th} processing server queue, $(W_q(t))$ with the average arrival rate (λ_i)

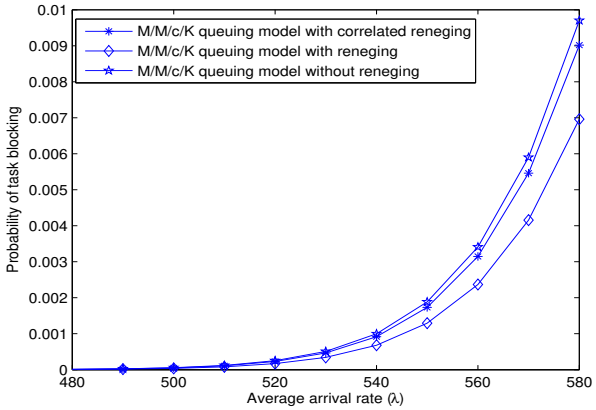

Fig. 7. Variation of the probability of task blocking at the i^{th} processing server queue with the average arrival rate (λ_i)

of queues that constitute a queueing model of a simplified cloud computing infrastructure. We have demonstrated the relationship between the load, delay, and probability of buffer saturation (blocking probability). It is seen that at about 80% utilization, a small increase in utilization (due to a small increase in the arrival rate $\Delta\lambda$ for a fixed service rate μ) results in a sharp increase in the delay experienced by the tasks, and on the probability of task rejection or blocking. We intend to use queueing models with general arrival and service time distributions, such as the diffusion approximation queueing model models presented in [24], [25]. Diffusion approximation enable the use of any distribution of in the interarrival time and service times (including the use of real or measured distributions). That is, it remove the Poison assumption that we have considered in this paper because they differ from reality.

REFERENCES

[1] Q. Duan, "Cloud service performance evaluation: status, challenges, and opportunities a survey from the system modeling perspective," *Digital Communications and Networks*, vol. 3, pp. 101–111, 2017.

[2] G. S. Kuaban, B. S. Soodan, R. Kumar, and P. Czekalski, "Analysis of the performance of a cloud computing processing queue with correlated reneging of tasks and resubmission," in *Proceedings of the 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2021, pp. 1–8.

[3] T. Atmaca, T. Begin, A. Brandwajn, and H. Castel-Taleb, "Performance evaluation of cloud computing centers with general arrivals and service," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 8, pp. 2341–2348, 2016.

[4] M. Liu, W. Dou, S. Yu, and Z. Zhang, "A decentralized cloud firewall framework with resources provisioning cost optimization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 621–631, 2014.

[5] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, "Energy-aware vm consolidation in cloud data centers using utilization prediction model," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 524–536, 2019.

[6] S. Homsi, G. A. Chaparro-Baquero, O. Bai, S. Ren, and G. Quan, "Workload consolidation for cloud data centers with guaranteed qos using request reneging," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 2103–2116, 2017.

[7] Y. Sharma, BahmanJavadi, WeishengSi, and D. Sun, "Reliability and energy efficiency in cloud computing systems: Survey and taxonomy," *Journal of Network and Computer Applications*, vol. 74, pp. 66–85, 2016.

[8] R.O.Al-Seedy, A.A.El-Sherbiny, S.A.El-Shehawy, and S.I.Ammar, "Transient solution of the m/m/c queue with balking and reneging: a survey," *Computers and Mathematics with Applications*, vol. 57, no. 8, pp. 1280–1285, 2009.

[9] Y.-J. Chiang, Y.-C. Ouyang, and C.-H. Hsu, "Performance and cost-effectiveness analyses for cloud services based on rejected," *IEEE Transaction on Services Computing*, vol. 9, no. 3, pp. 446–455, 2016.

[10] V. K. Q. Rodriguez and F. Guillemin, "Performance analysis of resource pooling for network function virtualization," in *Proceedings of the 2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks)*. IEEE, 2016, pp. 158–163.

[11] Y.-J. Chiang and Y.-C. Ouyang, "Profit optimization in sla-aware cloud services with a finite capacity queuing model," *Mathematical Problems in Engineering*, vol. 2014, no. 8, pp. 1–12, 2014.

[12] R. Kumar, B. S. Soodan, G. S. Kuaban, P. Czekalski, and S. Sharma, "Performance analysis of a cloud computing system using queuing model with correlated task reneging," *Journal of Physics:Conference Series, presented in 5th International Scientific Conference on Information, Control, and Communication Technologies (ICCT-2021)*, vol. 2091, no. 012003, pp. 4–7, 2021.

[13] G. S. Kuaban, B. S. Soodan, R. Kumar, and P. Czekalski, "Performance evaluations of a cloud computing physical machine with task reneging and task resubmission (feedback)," in *Proceedings of the International Conference on Computer Networks (CN2020)*, P. G. et al., Ed. Gdansk, Poland: Springer, 2020, pp. 185–198.

[14] Y. He, L. Ma, R. Zhou, C. Huang, and Zongpeng, "Online task allocation in mobile cloud computing with budget constraints," *Computer Networks*, vol. 151, pp. 42–51, 2019.

[15] A. Carrega and MatteoRepetto, "Coupling energy efficiency and quality of service for consolidation of cloud workloads," *Computer Networks*, vol. 174, pp. 42–51, 2020.

[16] N. Agrawal and S. Tapaswi, "Defense mechanisms against ddos attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.

[17] G. S. Kuaban, T. Atmaca, and T. C. and, "Performance analysis of packet aggregation mechanisms and their applications in access (e.g., iot, 4g/5g), core, and data centre networks," *Sensors*, vol. 21, no. 3898, pp. 1–33, 2021.

[18] L. Wang and E. Gelenbe, "Adaptive dispatching of tasks in the cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 33–45, 2018.

[19] E. Gelenbe, J. Domanska, P. Frhlich, and S. N. Mateusz Nowak, "Self-aware networks that optimize security, qos and energy," *Proceedings of the IEEE*, vol. 108, pp. 1150–1167, 2020.

[20] X. Nan, Y. He, and L. Guan, "Queueing model based resource optimization for multimedia cloud," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 928–942, 2014.

[21] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture, advances in big data and cloud computin," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 149–158, 2020.

[22] A. Paya and D. C. Marinescu, "Energy-aware load balancing and application scaling for the cloud ecosystem," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 15–27, 2017.

[23] S. E. Kafhali and K. Salah, "Modelling and analysis of performance and consumption in cloud data centers," *Arabian Journal of Science and Enginering*, vol. 43, no. 2, pp. 7789–7802, 2018.

[24] T. Czachórski, E. Gelenbe, G. S. Kuaban, and D. Marek, "Time-dependent performance of a multi-hop software defined network," *Applied Sciences*, vol. 11, no. 2469, pp. 1–21, 2021.

[25] T. Czachórski, G. S. Kuaban, and T. Nycz, "Multichannel diffusion approximation models for the evaluation of multichannel communication networks," in *Distributed Computer and Communication Networks. DCCN 2019. Lecture Notes in Computer Science*, K. D. e. Vishnevskiy V., Samouylov K., Ed., vol. 11965. Moscow, Russia: Springer, 2019, pp. 43–57.