Queueing Theory based Performance Analysis of Cloud based BlockChain as a Service (BaaS)

Sheikh Muhammad Arsalan Jamil Arsalan (arsalanjamil24@gmail.com)
Karachi Institute of Economics and Technology

Fahad Bin Nasir Fahad
Karachi Institute of Economics and Technology

Sameer Hashmat Qazi Qazi
Karachi Institute of Economics and Technology

Bilal Muhammad Khan Khan
National University of Sciences and Technology

Research Article

Keywords: Cloud Computing, BlockChain, BaaS, SLA, Queuing, Markov Chain, Distributed Ledger Technology

Posted Date: July 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1832922/v1

Abstract

Third party cloud service providers are becoming common for offering BlockChain-as-a-Service (BaaS). In this framework, incoming requests are processed by a number of stages sequentially with each service stage having different processing times. Each stage will manifest a particular component of BlockChain primitives such as digital transactions undertaken by businesses within permissioned users. This will require sophisticated steps of digital fingerprint/certificate generation, secure transaction management based on permission levels, federated consensus and transaction validation among others using distributed cryptographic techniques. Such architectures are becoming common as third-party Cloud Services will begin to offer BlockChain-as-a-Service (BaaS) for improved security for clients. BaaS will typically incorporate the M/Hypo/1/L queueing model in which incoming requests are Poisson distributed, processing times are hypo-exponential and having finite buffers. Our main contributions in this paper are: (i) To develop a realistic Queueing based performance model of BaaS over third party cloud service provider; (ii) to develop rigorous mathematical analysis to estimate the model parameters such as calculating system idleness, system utilization, queue waiting times, and the number of occupants in the system and queue; (iii) Furthermore, our model helps to estimate real-time Service Level Agreement (SLA) criteria such as through-put, response time, and request drop as a function of system security implemented through mining stage.

I. Introduction

Cloud based service provision is a ubiquitous paradigm now offering a range of services, like Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a service (PaaS). These typically mean that the cloud will take responsibility of hardware solutions in the form of availability of high-end servers and high-end software computing services to bring secure, reliable and cost-effective solutions for enterprise needs.

With the advent of distributed computing applications such as BlockChain based applications, it is not possible to execute the client service in a single go, on a single server. This is because such applications require federated computing resources in a distributed manner to achieve the goals of reliability, security, trustworthiness to implement Distributed Ledger Technology (DLT) based secure application. Typical use cases include execution of financial transactions like digital asset trading, supply chain provenance tracking, secure management of health records, claims settlement and identity verification by large businesses within permissioned users. The use of word ‘transaction’ thereafter in this paper could include any of the above use-cases as mentioned. These distributed processing stages would typically include several of fundamental BlockChain primitives such as digital fingerprint/certificate generation, transaction management based on permission levels, development of federated consensus and transaction validation among others through complex cryptographic algorithms and puzzle solving techniques. Hence, such BlockChain based services would typically require distributed processing in intermediate stages before the final output can be generated. On a physical level, such BlockChain based service requests will require asynchronous service-to-service communication over federated machines

over cloud architectures such as OpenStack based RabbitMQ, Qpid, and ZeroMQ. Intermediate BlockChain messages are stored on the queue until they are processed in serial manner based on availability of federated Cloud Nodes.

As described above cloud networks providing BaaS will process the arriving requests in more than one service stage with different processing rate for each stage depending on the exact primitive of BlockChain application being actioned. The incoming requests are first placed in a global queue. In this type of system, a request is processed sequentially by single stage at a time so that only one stage is active and all the other stages are inactive. This is typical of Block Chain based application relying on distributed digital verification of the transactions using cryptographic procedures over a federated architecture that develop the trustworthiness to move to the next stage.

To study the intrinsic performance of Cloud based systems offering BaaS, they can be emulated as a multi- stage queuing system in which the processing time (or service time) possesses a hypo-exponential distribution of type M where M is number of stages in the block chain process (typically 7) as shown in Figure 1. These types of systems are denoted as M/Hypo/1/L in Kendall's notation, having Poisson arrival of requests, Figure 1. *Blockchain as a service shown as a sequence of 7 steps*

finite queue length L; L representing number of block requests that can be queued at a time with one block being processed after the other. Note that each block contains multiple transactions considered for one block generation as a system parameter and service time for each stage that is hypo-exponentially distributed (Figure 2). There is not much work done in performance modelling of cloud-based queuing systems with hypo-exponential service times with finite buffer for BaaS. The purpose of this research is to demonstrate an analytical queuing model for studying and estimating the performance metrics of networked servers deployed on cloud with hypo-exponential service time distribution. Our model makes use of Markov Chain method to derive mathematical equations for some key performance measures which include average through-put, system inactivity, total utilization, probability of request drop and average queuing delay. Engineers can optimize the system parameters like the number of stages, processing time, request arrival time and queue length to achieve the required SLA.

New emerging areas of research are now focusing on parallel blockchain processing techniques such as the work by Fitzi et al. [1]. Similarly, Lee et al. [2] have investigated a hierarchical MultiBlock Chain architecture for parallel execution of blocks to increase block chain throughput for cryptocurrency based transactions. Hazari and Mahmoud have also investigated a parallel blockchain processing system based on parallel mining strategy [3]. However, such systems are still in infancy, and it will be still some time before their widespread adoption. In this paper we don't focus on emerging architectures rather on the performance modelling of conventional blockchain based systems that are already in use.

Our main contributions in this paper are: (i) To develop a realistic Queueing based performance model of BaaS over 3rd party cloud service provider; (ii) to develop rigorous mathematical analysis to estimate the model parameters such as calculating system idleness, system utilization, queue waiting times, and the number of occupants in the system and queue; (iii) Furthermore, our model helps to estimate real-time

Service Level Agreement (SLA) criteria such as through-put, response time, and request drop, our model can estimate the required values for mean request arrival rate, buffer size (queue length), and average service rate of each stage.

Ii. Related Work

Mathematical analytical models have been developed since a long time to understand network availability, achievable QoS and other parameters over various network traffic models and limited shared network resources [4],[5]. Xiong and Perros [6] developed mathematical analytical models to investigate the performance of cloud based networking services. Similarly, Vaquero et al. [7] are one of the early contributors to study cloud networks operation and its performance.

Queueing systems have been used for a long time to model the performance of systems where several requestors come to avail shared services. Faris et al implemented the queueing system abstraction of a physical hospital scenario to monitor the outpatient waiting times at various stages in the hospital processes using an open Jackson Network implementation [8]. Goto et al. [9] analyzed the queueing system performance of a software defined network using popular OpenFlow based switching model.

As cloud networks became popular, researchers' interest has been greatly increased for its capabilities through its performance modelling [10]. Maiyama et al. [11] have studied the queueing network modelling (QNM) for OpenStack Infrastructure as a Service (IaaS) for cloud computing platforms. Vilpana et al. [12] presented a queueing model based abstraction of cloud network services comprising of an open Jackson network with several single server or parallel server stages to simulate the various facets of the cloud computing network, such as single and parallel operating Processing Servers, Database servers etc. The model is used for services departing after limited lifetimes in the system, as some user requests would require recurrent services from the cloud network. Interestingly, network signal propagation over the telecommunication network links joining the different components of the cloud network together and with the user as another single server form a queueing network. This is already well known by telecom researchers as network request arrivals and departures follow Poisson distribution and exponential processing times respectively over telecommunication links. Thus, the performance of an entire cloud computing network processing user requests can be easily modelled using well developed concepts of the queueing theory. Vilpana et al. [13] in another contribution, study the performance of a cloud based e-health monitoring system using queueing theory perspective.

Li et al. [14] studied Markov modelling of cloud systems where jobs were parallelizable so that they could be broken down and served in parallel over multiple parallel server banks.

Khazei et al. [15],[16] investigated more tunable queueing theory model to approximate the performance of the cloud data center in terms of task blocking probability and number of server requirements to meet the performance benchmarks.

Khomonenko et al. [17] studied a non-Markov system to study cloud system performance using multichannel queues which are decomposed into micro states.

Mas et al. [18] have investigated a queueing theory based performance monitoring of a fog computing network similar to a cloud computing network using open Jackson network. The studied architecture is hybrid which involves both partial processing in the fog as well as cloud network.

Newer cloud based applications such as BaaS are rapidly becoming popular on cloud network. Such services digress from the traditional computing services (IaaS, SaaS, PaaS etc) as they require complex cryptography based federated responses from multiple machines before the next step in the sequence can be executed instead of centralized server resources in direct control of the cloud network. Kim and Park [19] investigated a Hybrid decentralized Byzantine Fault tolerant BlockChain framework for cloud based application over OpenStack message queue. Another related work by Zhang et al. [20] investigated the BlockChain consensus algorithm performance for message queue with Byzantine fault tolerance. Similarly, Salah and Shiltami [21] have investigated Message-Queueing-as-a-Service (MaaS) cloud application which goes through several processing stages in the cloud network before an output can be generated. This has been modelled as a hypoexponential processing times queue network investigated in more detail by Salah and El Kafhali [22].

More recently, Memon et al. [23] have simulated two popular blockchain systems, BitCoin and Ethereum using queueing theory modelling however, they use simple M/M/1 system for simulation of memory pool management and a M/M/c system for simulation of mining. This model lacks few of the intricacies of the queuing system model when generalized over a P2P based cloud network; these are explained later in next section. A better queue based modelling of block chain is presented by Li et al. [24] which models the block chain as a combination of Markovian batch service, queueing process and game model, an abstraction of two sequential queueing stages one that of block generation and the second one of block mining (verification). The model developed in this paper is more inspired with this second model.

Message Queueing services such as that for BlockChain based services will typically incorporate the M/Hypo/1/K queueing model in which incoming requests are Poisson distributed, processing times are hypo-exponential and having finite buffers [21]. Hypoexponential and Hyperexponential random variables are studied in detail by Yanev recently [25]. Marin and Bulo [26],[27] developed a queueing model using hypo-exponential distribution of service times but considering queues with infinite buffer. Practical cloud systems have finite buffer spaces in their systems.

Iii. Analytical Model

As shown in Fig. 1, the lifecycle of blocks on a block chain system to be modelled typically consists of the following 7 steps:

(i) Unicast: *Nodes initiating transactions and forwarding to cloud entity*

(ii) Broadcast: *A 3rd party Cloud entity broadcasting transactions to peers; Peers accept selected transactions to form their own 'blocks;*

(iii) Mining: *Peers set about to solve the cryptographic puzzle through 'nonce' generation satisfying hash requirements*

(iv) Unicast: *Winner reports back calculated 'nonce' value to cloud entity*

(v) Broadcast: *Cloud entity further broadcasts 'nonce' to other peers for validation.*

(vi) Unicast: *Other peers validate the transaction and report back to cloud entity.*

(vii) Broadcast: *Block Published and added to block chain network through majority vote algorithms.*

Queueing Theory forms the basis of services encountered over shared, networked devices over computer networks in general [28]. The same model is extended for cloud services which dominate the computing requirements of enterprises today. Popularly known Kendell's notation A/B/C/D is often used to describe the type of queueing system. The first alphabet denotes request arrival distribution which could be one of 'M' for memory less used to denote Poisson distribution, 'D' for Deterministic or 'G' for General distribution. The second alphabet denotes distribution of the processing times; this could be 'M' to denote memoryless or exponentially distribute processing time, 'D' for Deterministic or 'G' for General. The third alphabet denotes the number of processing servers which could take any finite numerical value where any value greater than one denotes parallel service possible through multiple servers. Finally, the last alphabet denotes the size of the finite queue, some theoretical calculations could involve assuming queue size of infinity.

More recently, Memon et al. [23] have simulated two popular blockchain systems, BitCoin and Ethereum using queueing theory modelling. However, they use simple M/M/1 system for simulation of memory pool management, accepting transactions in real time and a M/M/c system for simulation of mining process. This model lacks few of the intricacies of the queuing system model. For example, in M/M/c systems it is assumed that *identical* servers process *heterogenous jobs* (which have exponentially distributed processing times) in a FCFS (First Come First Served) manner from a queue as a server becomes idle. In real block chain scenario (i) similar mining jobs go to *non-identical* servers during a mining process for the discovery of nonce. (ii) once nonce calculation is completed by the 'winner', processing by others becomes *irrelevant* even if the process was assumed to be conducted in parallel before, thus M/M/c model does not fit this scenario. (iii) miners are assigned jobs but may choose to accept or decline a job for both permissioned or non-permissioned block chain networks; (iv) There are practical limitations on number of blocks being mined in a particular interval of time which violates the work conservation principle in typical queueing system analysis. Hence due to these weaknesses, we digress from this model and formulate another model which we believe gives a better model for queueing system-based blockchain.

In our model we assume that: (i) we only have trusted peers as in a permissioned block chain network, i.e. none of the peers would decline a job based on financial incentives to keep the model only dependent on technical aspects. (ii) cloud entity monitors network conditions of the peer to peer overlay network through established network probing techniques [29] or statistical techniques [30] so that network outages or inactive peers are catered for beforehand.

Similar to model developed by Li et al. [24], we design blockchain queue, in which the block-generation and blockchain building processes are expressed as a several stages of batch services. Block generation and Block chain building are modelled as several M/M/1 stages with different process times (as independently and identically distributed (I.I.D) random variables). Block generation is modelled as one M/M/1 system while the other stages of the block chain generation such as federated consensus generation through several P2P network broadcast requests are also modelled as separate M/M/1 queueing stages (Fig. 2). Li models all network information broadcast stages as one M/M/1 system (termed as the block generation stage) cascaded after the M/M/1 block mining stage; we argue that this is not an accurate model as several information propagation stages also precede the mining stage such as broadcasting of new transactions over the network as well as follow it such as handshaking protocols for federated consensus and finally broadcasting the *approved* block over the network. In all we use 7 stages in our queueing model of the block chain network as outlined in Fig. 1.

Figure 2

Queuing system with a finite buffer and hypo-exponential service stages

This section explains our queuing system's analytical model with hypo-exponential service distribution and a finite buffer length(L) i.e. M/Hypo/1/L is presented. As shown in Fig. 2, first of all incoming requests are stored in a centralized queue with storage of L-1 requests and then processed in M stages having different mean service rate i.e., $\mu_1, \mu_2, \ldots, \mu_M$. Note that each request here is at the abstraction level of one block. And each block request can have multiple transactions (with a limit on maximum number of transactions) summed up through superposition principle to determine new block request arrival rate. Ties between simultaneously occurring transactions are broken through assignment of higher priority to higher transaction fee or any other appropriate criteria. Thus, transactions can be logically separated to be part of one block or another. The processing of each individual block (comprising multiple transactions) requests is done in a mutually exclusive and sequential manner that is: a new block request will pass into first stage only when the previous block request leaves the final stage M becoming part of the block chain. Mutually exclusive means that at any instant of time only one of the M stages is active. This is typically the case in permissioned block chain networks like BaaS where one block of transactions is completed before executing a second one as each block involves broadcasting transactions (belonging to that block) to all Blockchain peers who must aggregate the information and signature verify each block through cryptographic algorithms.

Figure 4: State

Figure 3

Transition diagram of -the Markov Process

For this analysis, incoming block requests follow a Poisson distribution having arrival rate 'λ'. Also, the service time of each stage is independent and follows an exponential distribution. Another important assumption is that the requests in queue are serviced in FCFS basis.

For our analytical model, we assume that the arrival of requests follows a Poisson distribution, and service times of network steps (unicast and broadcast) and mining step are exponentially distributed. It has been seen in different literature that the arrival of request may not always follow a Poisson process [31] in which case an accurate analytical solution is very hard to model. Secondly, in our model an assumption is made that all the requests pass through multiple stages sequentially as shown in Fig. 1; one might argue that this may not be true for all systems for example at a stage, processing may be done in parallel by using multiple peers such as mining. But as explained earlier, mining process is dictated by the 'winner' node. So, parallel processing in such cases become meaningless as far as queueing system analysis is considered. Although newer systems are investigating parallel processing but it is beyond the scope of this work. For federated block chain systems, this assumption is valid as next sequential steps can only be executed based on results of previous steps from distributed peers such as majority vote to determine execution of next steps. An approximate solution for this case can be computed using out model by taking sequential servers with average service rate.

Our model utilizes the Markov Chain with finite buffer capacity. The model describes the operation of a multi-stage queuing service. The state space can be generically defined using a 2-tuple value as:

$$S = \{(l, m), 0 \leq l \leq L, 0 \leq m \leq M\}$$

1

Where m represents the active service processing node at the moment and l represents the number of current block requests in the system, with each block request having multiple transactions considered for block formation. The system has a queue length of L-1 for pending block requests as one block request is currently in service at all times through work conservation principle. State $(0,0)$ is a special state representing no block formation requests in the system i.e., the system is in idle state.

Probabilities of state (l, m) can be represented as $p_{l,m}$. A system of difference equations to compute the state probabilities can be written as below, with one of the universal Global Balance Equation relating all state probabilities through the universal law of probability.

The Global Balance Equation

$$p_{0,0} + \sum_{l=1}^{L} \sum_{m=1}^{M} p_{l,m} = 1$$

2

Few other state probabilities can be calculated through the difference equations which can be verified from the Markov Diagram in Fig. 3.

State(0,0)

$$-\lambda p_{0,0} + \mu_M . p_{1,M} = 0$$

3

State(1,M)

$$-\left(\lambda + \mu_M\right) p_{1,M} + \mu_{M-1} . p_{1,M-1} = 0$$

4

State(1,m)

$$-\left(\lambda + \mu_M\right) p_{1,m} + \mu_{m-1} . p_{m-1} = 0$$

5

$$(2 \le m \le M-1)$$

State(1,1)

$$-\left(\lambda + \mu_1\right) p_{1,1} + \lambda p_{0,0} + \mu_M . p_{2,M} = 0$$

6

State(l, M)

$$-\left(\lambda + \mu_M\right) p_{l,M} + \lambda p_{l-1,M} + \mu_{M-1} . p_{l,M-1} = 0$$

(7)

$$(2 \le l \le L-1)$$

State(l, m)

$$-\left(\lambda + \mu_m\right) p_{l,m} + \lambda p_{l-1,m} + \mu_{m-1} p_{l,m-1} = 0$$

8

$$(2 \leq l \leq L-1; 2 \leq m \leq M-1)$$

State(l, 1)

$$-\left(\lambda + \mu_1\right) p_{l,1} + \lambda p_{l-1,1} + \mu_M p_{l+1,M} = 0$$

9

$$(2 \leq l \leq L-1)$$

State(L, M)

$$-\mu_M p_{L,M} + \lambda p_{L-1,M} + \mu_{M-1} p_{L,M-1} = 0$$

10

State(L, m)

$$-\mu_m p_{L,m} + \lambda p_{L-1,m} + \mu_{m-1} p_{L,m-1} = 0$$

11

$(2 \leq m \leq M-1)$

State(L, 1)

$$-\mu_1 p_{L,1} + \lambda p_{L-1,1} = 0$$

12

From the above equations we can calculate the state probabilities $p_{l,m}$ where $(1 \leq l \leq L, 1 \leq m \leq M)$. Furthermore, other important performance metrics like throughput, utilization factor, waiting time etc. can be computed.

The departure rate of the system is basically the average throughput, the rate of departure of requests from *stage M.*

$$\gamma = \lambda \sum_{l=1}^{L} p_{l,M}$$

13

Throughput can also be written as

$$\gamma = \frac{(1 - p_0)}{\bar{X}}$$

14

Where $\bar{X}$ is mean service time of the multi-stage system.

$$\bar{X} = \sum_{m=1}^{M} \frac{1}{\mu_m}$$

15

Throughput (departure rate) γ can also be represented as

$$\gamma = \lambda(1 - P_{loss})$$

16

Where P_{loss} is the blocking probability when incoming request sees that the queue is completely occupied.

$$P_{loss} = \frac{p_0 + \rho - 1}{\rho}$$

17

Where $\rho$is expressed as $\rho = \lambda \bar{X}$ often referred as offered load

P_{loss} can also be expressed in terms of state probabilities as

$$P_{loss} = \sum_{m=1}^{M} p_{L,m}$$

18

The mean number of requests in the system at a time can be termed as *System occupancy* and expressed as

$$E[L] = \sum_{l=1}^{L} \sum_{m=1}^{M} l p_{l,m}$$

19

The mean number of requests in the queue are defined as Queue Occupancy and expressed as

$$E\left[L_q\right] = \sum_{l=1}^{L} \sum_{m=1}^{M} (l-1) p_{l,m}$$

20

$$E\left[L_q\right] = E[L] - (1 - p_0)$$

21

The average amount of time spent by a request in the system after entering the queue can be computed by using Little's law

$$W = \frac{E[L]}{\gamma}$$

22

The time spent by a request waiting to be served in the queue is termed as Queuing delay and it can be computed as

$$W_q = \frac{E\left[L_q\right]}{\gamma} = W - \bar{X}$$

23

Total system utilization can be expressed as

$$U = \gamma \bar{X}$$

24

For various values of L and M, the stage transition matrices can be written as a system of linear equations. For example, we present the example for L = M = 2 and L = M = 3 Eq. 25 and Eq. 26 respectively.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \lambda & 0 & -\mu_2 & 0 & 0 \\ 0 & \mu_1 & -(\mu_2+\lambda) & 0 & 0 \\ 0 & \lambda & 0 & -\mu_1 & 0 \\ 0 & 0 & \lambda & \mu_1 & -\mu_2 \end{bmatrix} \begin{bmatrix} p_{00} \\ p_{11} \\ p_{12} \\ p_{21} \\ p_{22} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

25

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \lambda & 0 & 0 & -\mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu_1 & -(\lambda+\mu_2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu_2 & -(\lambda+\mu_3) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & -(\lambda+\mu_1) & 0 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & \lambda & 0 & \mu_1 & -(\lambda+\mu_2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & \mu_2 & -(\lambda+\mu_3) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 & 0 & -\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 & \mu_1 & -\mu_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda & 0 & \mu_2 & -\mu_3 \end{bmatrix} \begin{bmatrix} p_{00} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (26)$$

Iv. Numerical Results

In this section, we list down the results acquired from analytical model after its implementation in MATLAB. We also report key performance metrics of our queuing system.

For our analysis, the number of processing stages are set to 7, buffer length is 10 block requests. The block arrival requests, and service rates are shown in Table 1. The effect of time complexity of mining stage is considered through variation of service rate of stage 3 while keeping the service rate of other stages as fixed.

Table 1
Values of Parameters used in Simulation

Parameters used in simulation	Value
Block Arrival Request Rate (λ)	*Varied from 0 to 2 req/sec*
Service Rate stage 1 (μ_1)	*100 req/ sec*
Service Rate stage 2 (μ_2)	*20 req/ sec*
Service Rate stage 3 (μ_3)	*Varied from 0.001 to 10 req/ sec*
Service Rate stage 4 (μ_4)	*6.67 req/ sec*
Service Rate stage 5 (μ_5)	*5 req/ sec*
Service Rate stage 6 (μ_6)	*4 req/ sec*
Service Rate stage 7 (μ_7)	*3.33 req/ sec*
Buffer Size (L)	*10 block requests*

These values are based on practical blockchain system stages where the initial stages which are involved with initiation of blockchain based transactions involve simple (signed) transaction documents of single peers with latter stages involving cryptographic based verification from multiple peers for authentication (after the mining step) and verification of the blockchain based transactions. Thus, the processing speed of latter stages become slower as compared to the initial stages. The most expensive stage is the mining stage which is depicted as the 3rd stage as shown in Fig. 1. We study the overall effect on the blockchain processing by variation of the time duration (complexity) of the mining step.

Figure 4 shows the performance results for Utilization factor U in relation to blockchain request rate λ. We see that the system utilization factor approaches 0.65 as λ = 0.6 req/sec when the mining step complexity is comparable to other stages (10 req/sec) but when the mining step complexity is increased by factor of 10; (i.e., service rate reduced 1 req/sec), the system utilization approaches 0.98. System utilization increases rapidly as the time complexity of mining step increases.

Figure 5 shows the performance results for throughput γ in relation to offered blockchain request rate λ. Throughput is roughly halved as the mining step complexity increases by a factor of 10; throughput reduced from a value of 0.95 to just 0.5 as mining service rate is reduced from 10 req/sec to 1 req/sec on further increasing the time complexity of mining stage the throughput reduces drastically.

Figures 6 and 7 shows the performance results for system occupancy E[L] and queue occupancy E[Lq] in relation to offered block chain requests λ. When offered load value λ = 1, the queue occupancy increases drastically by about 2-fold, from 6 to 9 (5 to 8 respectively) if only the mining stage complexity of the 7 stages increases by tenfold (mining stage service rate decreasing from 10 to 1).

Figure 8 shows the performance results for System idle probability p_o in relation to offered traffic λ. At λ = 0.5, Probability that system is idle is also about 0.5 when the mining stage service rate is 10. When the mining stage complexity is increased by factor of ten, thereby reducing its service rate from 10 to one, the corresponding figure for idle probability reduces to just 0.08.

Figure 9 shows the performance results system blocking probability (or probability of loss) P_L in relation to offered arriving blockchain requests λ. As mining stage complexity increases (service rate reduces from 10 to 0.1), system blocking probability increases from 0.5 to about 0.94 when λ = 2.

Figures 10 and 11 show the performance results for System waiting time W and Queuing delay Wq in relation to offered incoming requests λ. Both the system waiting time and queueing delay value increases very gradually as mining stage service rate is decreased from 10 to 0.1, but then increases sharply jumping by a factor as mining stage service rate is reduced from 1e-3 to 1e-4 even when λ = 0.1.

Overall, as summary, we note that the mining stage complexity serves both in ensuring system security and system integrity but has a deep impact on system performance parameters.

By examining these performance curves engineers can optimize the systems as per the required SLAs. Mining stages could perhaps be well optimized through parallel processing stages in order to optimize overall system performance without any caveat in the form of reduced system security.

Figure 4

System Utilization Vs Arrival Requests as Mining Stage Complexity is varied

Figure 5

System Throughput Vs Arrival Requests as Mining Stage Complexity is varied

Figure 6

Expected Requests in System Vs Arrival Requests as Mining Stage Complexity is varied

Figure 7

Expected Queue Occupancy Vs Arrival Requests as Mining Stage Complexity is varied

Figure 8

System “Idle” Probability Vs Arrival Requests as Mining Stage Complexity is varied

Figure 9

Blocking Probability Vs Arrival Requests as Mining Stage Complexity is varied

Figure 10

System Waiting Time Vs Arrival Requests as Mining Stage Complexity is varied

Figure 11

Queue Waiting Time Vs Arrival Requests as Mining Stage Complexity is varied

V. Conclusion

The purpose of this paper is to present an analytical model for analyzing and estimating the working of 3rd party based BlockChain as a Service (BaaS). We model the system M/Hypo/1/L queuing system and develop its Markov model for analyzing it. Mining stage places a crucial role in maintaining system integrity and system security. However, this comes at the cost of performance. From our analytical model we derived mathematical equations for essential performance metrics which include total system utilization and system idleness, request loss, through-put, queuing delay and queue and system occupancies and see the impact on their performance as a function of mining complexity in the system.

Declarations

Ethics approval and consent to participate: Not Applicable.

Consent for publication: All authors gave their consent to publish this research paper.

Availability of data and materials: Not Applicable.

Competing interests: The authors declare no competing interests.

Funding: This research didn’t receive any funding.

Authors' contributions: Proof of concept and Implementation S.M.A.J.; Project Administration and Supervision, S.Q.; Analysis F.B.N, B.M.K.; Writing Original draft, S.Q., F.B.N, B.M.K.; proof reading and reviewing S.M.A.J., S.Q.

Acknowledgements: We are grateful to the higher authorities of the Karachi Institute of Economics and Technology in Karachi, Pakistan, for providing an exceptional research environment as well as motivating support for our effort.

References

1. M. Fitzi, P. Gaži, A. Kiayias, and A. Russell, “Parallel Chains: Improving Throughput and Latency of Blockchain Protocols via Parallel Composition.” 2018. [Online]. Available: https://eprint.iacr.org/2018/1119

2. N.-Y. Lee, “Hierarchical Multi-Blockchain System for Parallel Computation in Cryptocurrency Transfers and Smart Contracts,” Applied Sciences, vol. 11, no. 21, p. 10173, Oct. 2021, doi: 10.3390/app112110173.
3. S. S. Hazari and Q. H. Mahmoud, “A Parallel Proof of Work to Improve Transaction Speed and Scalability in Blockchain Systems,” in 2019 *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2019, pp. 0916–0921. doi: 10.1109/CCWC.2019.8666535.
4. M. Martinello, M. Kaâniche, and K. Kanoun, “Web service availability—impact of error recovery and traffic model,” Reliability Engineering & System Safety, vol. 89, no. 1, pp. 6–16, Jul. 2005, doi: 10.1016/j.ress.2004.08.003.
5. J. Martin and A. Nilsson, “On service level agreements for IP networks,” in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, NY, USA, 2002, vol. 2, pp. 855–863. doi: 10.1109/INFCOM.2002.1019332.
6. K. Xiong and H. Perros, “Service Performance and Analysis in Cloud Computing,” in *2009 Congress on Services - I*, Los Angeles, CA, USA, Jul. 2009, pp. 693–700. doi: 10.1109/SERVICES-I.2009.121.
7. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: towards a cloud definition,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008, doi: 10.1145/1496091.1496100.
8. Monike Febriyani Faris, Y. Farida, D. C. R. Novitasari, N. Ulinnuha, and Moh. Hafiyusholeh, “Implementation of The Open Jackson Queuing Network to Reduce Waiting Time,” *J. Mat. Mantik*, vol. 6, no. 2, pp. 83–92, Oct. 2020, doi: 10.15642/mantik.2020.6.2.83-92.
9. Y. Goto, H. Masuyama, B. Ng, W. K. G. Seah, and Y. Takahashi, “Queueing Analysis of Software Defined Network with Realistic OpenFlow–Based Switch Model,” in 2016 *IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, London, United Kingdom, Sep. 2016, pp. 301–306. doi: 10.1109/MASCOTS.2016.30.
10. M. Armbrust *et al.*, “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010, doi: 10.1145/1721654.1721672.
11. K. M. Maiyama, D. Kouvatsos, B. Mohammed, M. Kiran, and M. A. Kamala, “Performance Modelling and Analysis of an OpenStack IaaS Cloud Computing Platform,” in 2017 *IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, Prague, Aug. 2017, pp. 198–205. doi: 10.1109/FiCloud.2017.54.
12. J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella, and J. Rius, “A queuing theory model for cloud computing,” J Supercomput, vol. 69, no. 1, pp. 492–507, Jul. 2014, doi: 10.1007/s11227-014-1177-y.
13. J. Vilaplana, F. Solsona, F. Abella, R. Filgueira, and J. Rius, “The cloud paradigm applied to e-Health,” BMC Med Inform Decis Mak, vol. 13, no. 1, p. 35, Dec. 2013, doi: 10.1186/1472-6947-13-35.
14. X. Li *et al.*, “Performance Analysis of Cloud Computing Centers Serving Parallelizable Rendering Jobs Using M/M/c/r Queuing Systems,” in 2017 *IEEE 37th International Conference on Distributed*

Computing Systems (ICDCS), Atlanta, GA, USA, Jun. 2017, pp. 1378–1388. doi: 10.1109/ICDCS.2017.132.

15. H. Khazaei, J. Misic, and V. B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 5, pp. 936–943, May 2012, doi: 10.1109/TPDS.2011.199.
16. H. Khazaei, J. Misic, and V. B. Misic, "A Fine-Grained Performance Model of Cloud Computing Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2138–2147, Nov. 2013, doi: 10.1109/TPDS.2012.280.
17. A. D. Khomonenko, S. I. Gindin, and K. M. Modher, "A cloud computing model using multi-channel queuing system with cooling," in 2016 *XIX IEEE International Conference on Soft Computing and Measurements (SCM)*, St. Petersburg, Russia, May 2016, pp. 103–106. doi: 10.1109/SCM.2016.7519697.
18. L. Mas, J. Vilaplana, J. Mateo, and F. Solsona, "A queuing theory model for fog computing," *J Supercomput*, vol. 78, no. 8, pp. 11138–11155, May 2022, doi: 10.1007/s11227-022-04328-3.
19. Y. Kim and J. Park, "Hybrid decentralized PBFT Blockchain Framework for OpenStack message queue," Hum. Cent. Comput. Inf. Sci., vol. 10, no. 1, p. 31, Dec. 2020, doi: 10.1186/s13673-020-00238-6.
20. J. Zhang, J. Zhao, X. Song, Y. Liu, and Q. Wu, "A Blockchain Consensus for Message Queue Based on Byzantine Fault Tolerance," in *Economics of Grids, Clouds, Systems, and Services*, vol. 12441, K. Djemame, J. Altmann, J. Á. Bañares, O. Agmon Ben-Yehuda, V. Stankovski, and B. Tuffin, Eds. Cham: Springer International Publishing, 2020, pp. 3–14. doi: 10.1007/978-3-030-63058-4_1.
21. K. Salah and T. R. Sheltami, "Performance modeling of cloud apps using message queueing as a service (MaaS)," in 2017 *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, Paris, Mar. 2017, pp. 65–71. doi: 10.1109/ICIN.2017.7899251.
22. K. Salah and S. El Kafhali, "Performance modeling and analysis of hypoexponential network servers," Telecommun Syst, vol. 65, no. 4, pp. 717–728, Aug. 2017, doi: 10.1007/s11235-016-0262-3.
23. R. Memon, J. Li, and J. Ahmed, "Simulation Model for Blockchain Systems Using Queuing Theory," Electronics, vol. 8, no. 2, p. 234, Feb. 2019, doi: 10.3390/electronics8020234.
24. Q.-L. Li, J.-Y. Ma, and Y.-X. Chang, "Blockchain Queue Theory," in *Computational Data and Social Networks*, vol. 11280, X. Chen, A. Sen, W. W. Li, and M. T. Thai, Eds. Cham: Springer International Publishing, 2018, pp. 25–40. doi: 10.1007/978-3-030-04648-4_3.
25. G. P. Yanev, "Exponential and Hypoexponential Distributions: Some Characterizations," Mathematics, vol. 8, no. 12, p. 2207, Dec. 2020, doi: 10.3390/math8122207.
26. A. Marin and S. Rota Bulò, "Explicit solutions for queues with Hypo-exponential service time and applications to product-form analysis," presented at the 5th International ICST Conference on Performance Evaluation Methodologies and Tools, Paris, France, 2011. doi: 10.4108/icst.valuetools.2011.245722.

27. A. Marin and S. Rota Bulò, “Explicit solutions for queues with Hypo- or Hyper-exponential service time distribution and application to product-form approximations,” Performance Evaluation, vol. 81, pp. 1–19, Nov. 2014, doi: 10.1016/j.peva.2014.07.021.
28. D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of queueing theory*, 4. ed. Hoboken, NJ: Wiley, 2008.
29. D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, Banff Alberta Canada, Oct. 2001, pp. 131–145. doi: 10.1145/502034.502048.
30. S. Qazi and T. Moors, “On the impact of routing matrix inconsistencies on statistical path monitoring in overlay networks,” *Computer Networks*, vol. 54, no. 10, pp. 1554–1572, Jul. 2010, doi: 10.1016/j.comnet.2009.11.006.
31. M. Andersson, A. Bengtsson, M. Höst, C. Nyberg, and J. Holst, “Web server traffic in crisis conditions.,” presented at the 3:rd Swedish National Computer Networking Workshop (SNCN)., 2005.

Figures

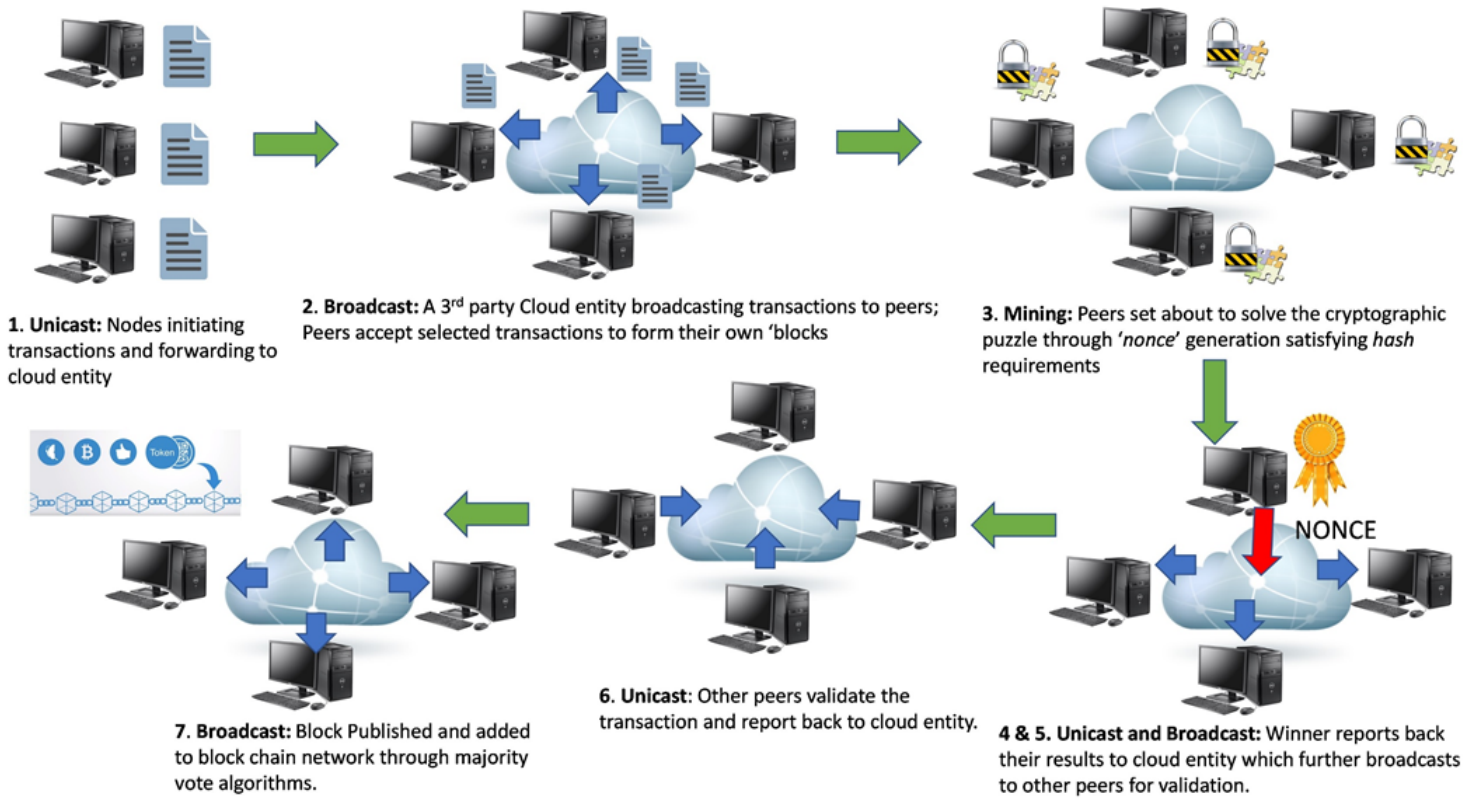

Figure 1

Blockchain as a service shown as a sequence of 7 steps

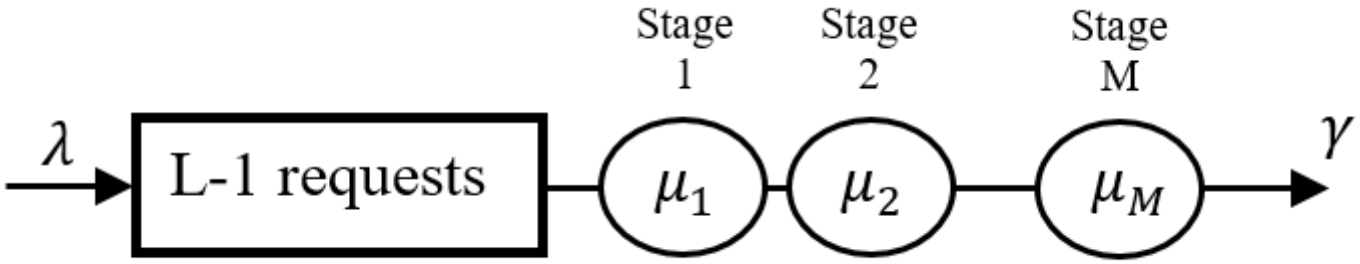

Figure 2

Queuing system with a finite buffer and hypo-exponential service stages

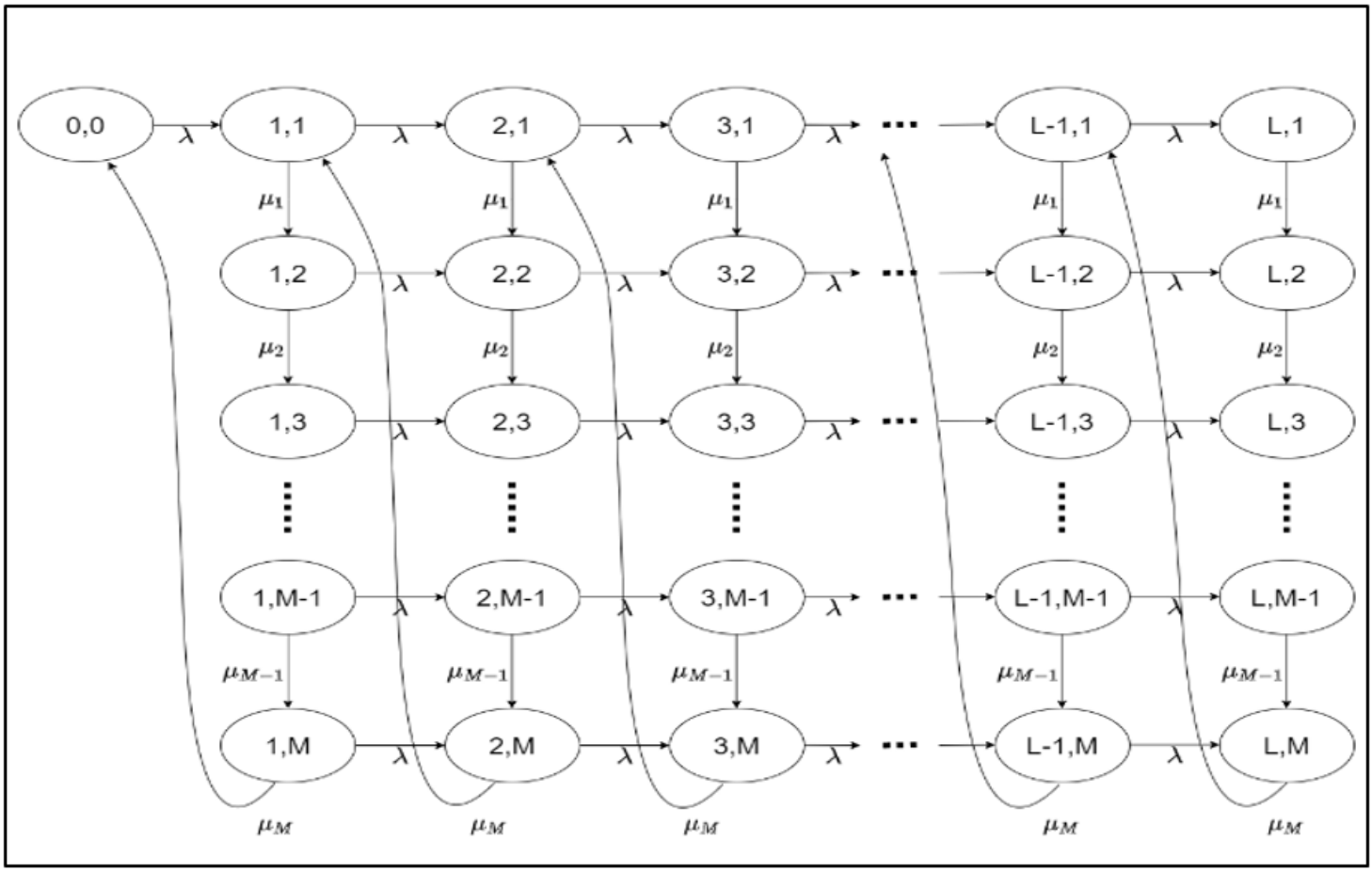

Figure 3

Transition diagram of -the Markov Process

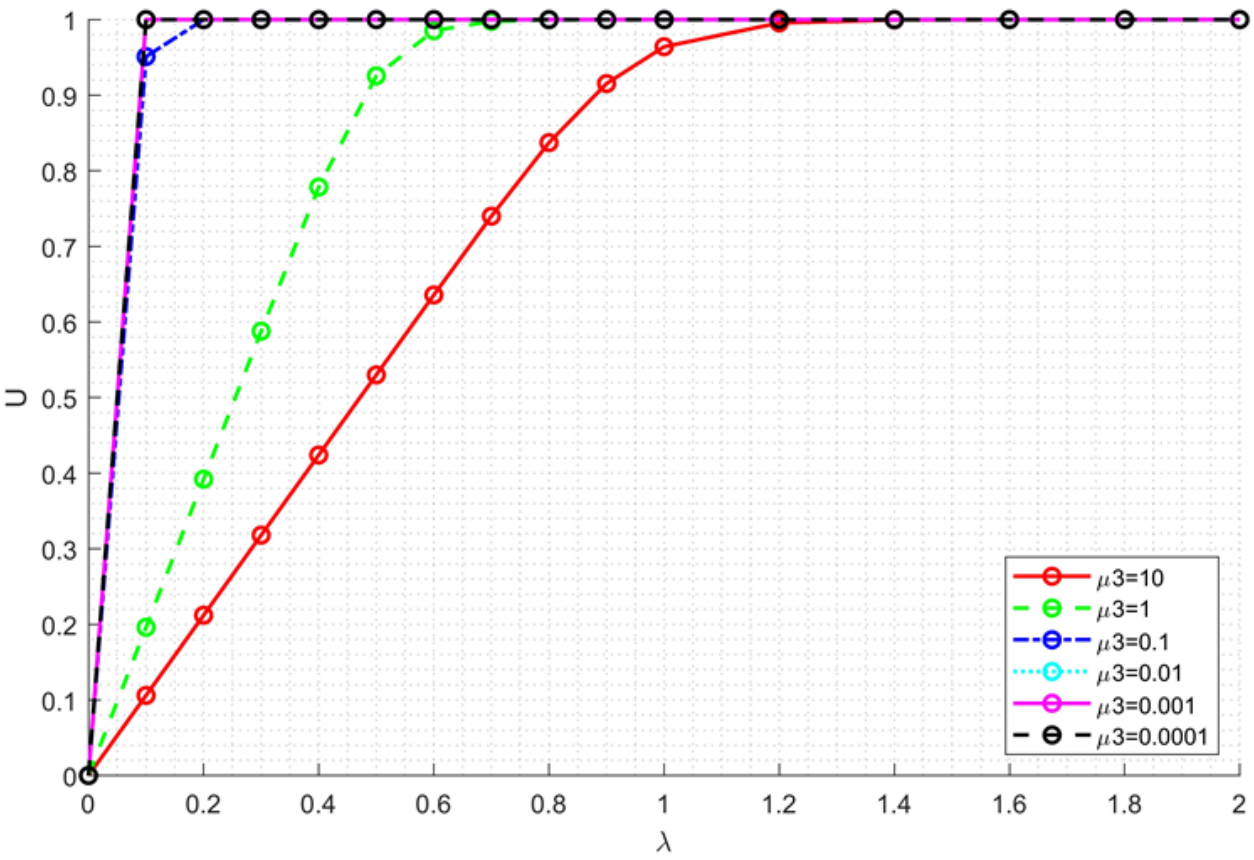

Figure 4

System Utilization Vs Arrival Requests as Mining Stage Complexity is varied

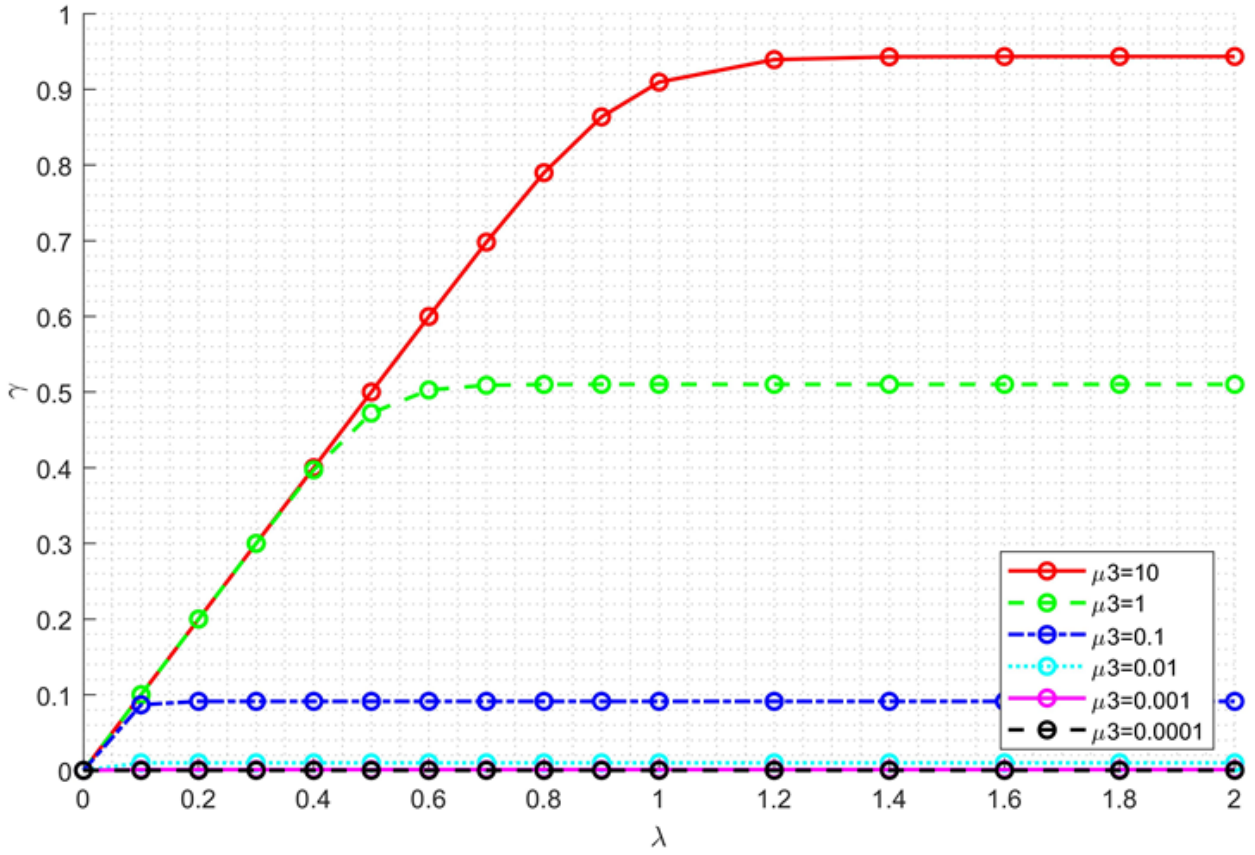

Figure 5

System Throughput Vs Arrival Requests as Mining Stage Complexity is varied

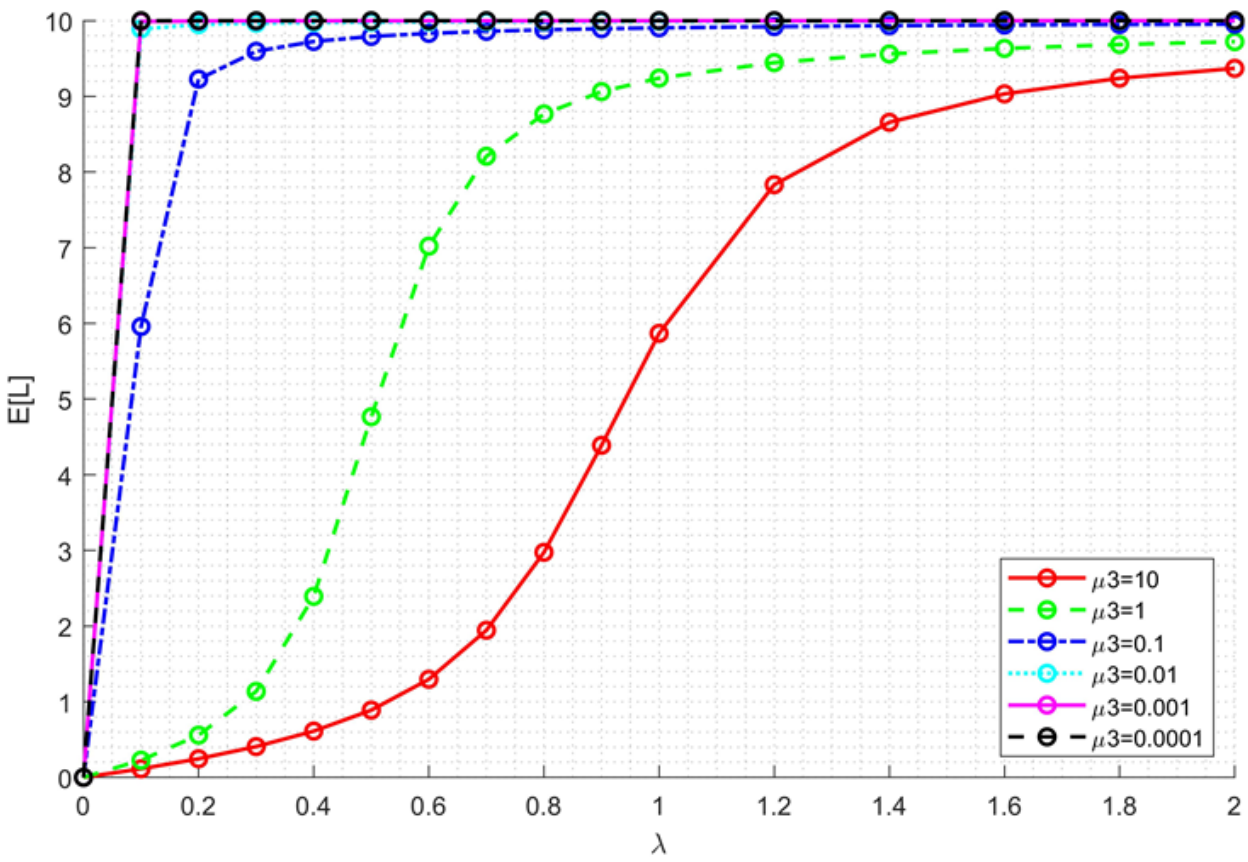

Figure 6

Expected Requests in System Vs Arrival Requests as Mining Stage Complexity is varied

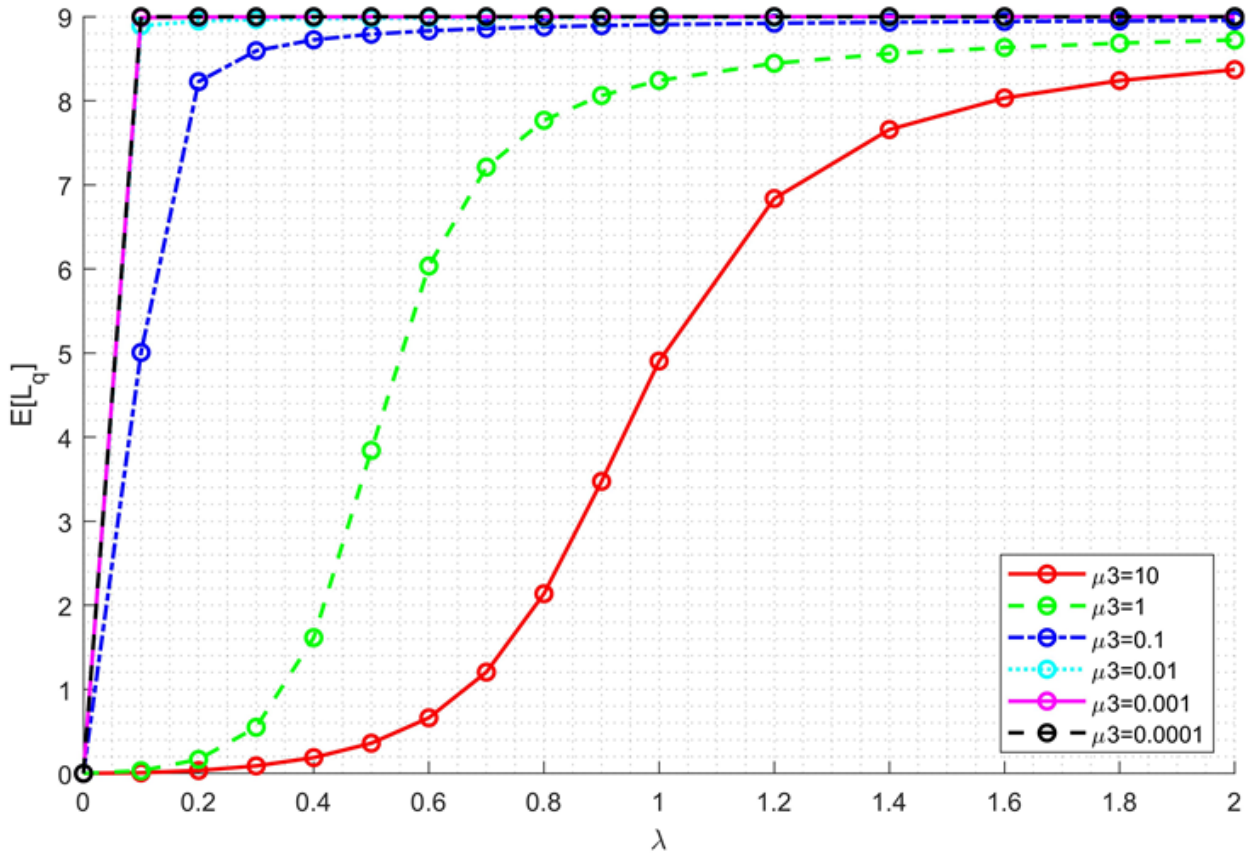

Figure 7

Expected Queue Occupancy Vs Arrival Requests as Mining Stage Complexity is varied

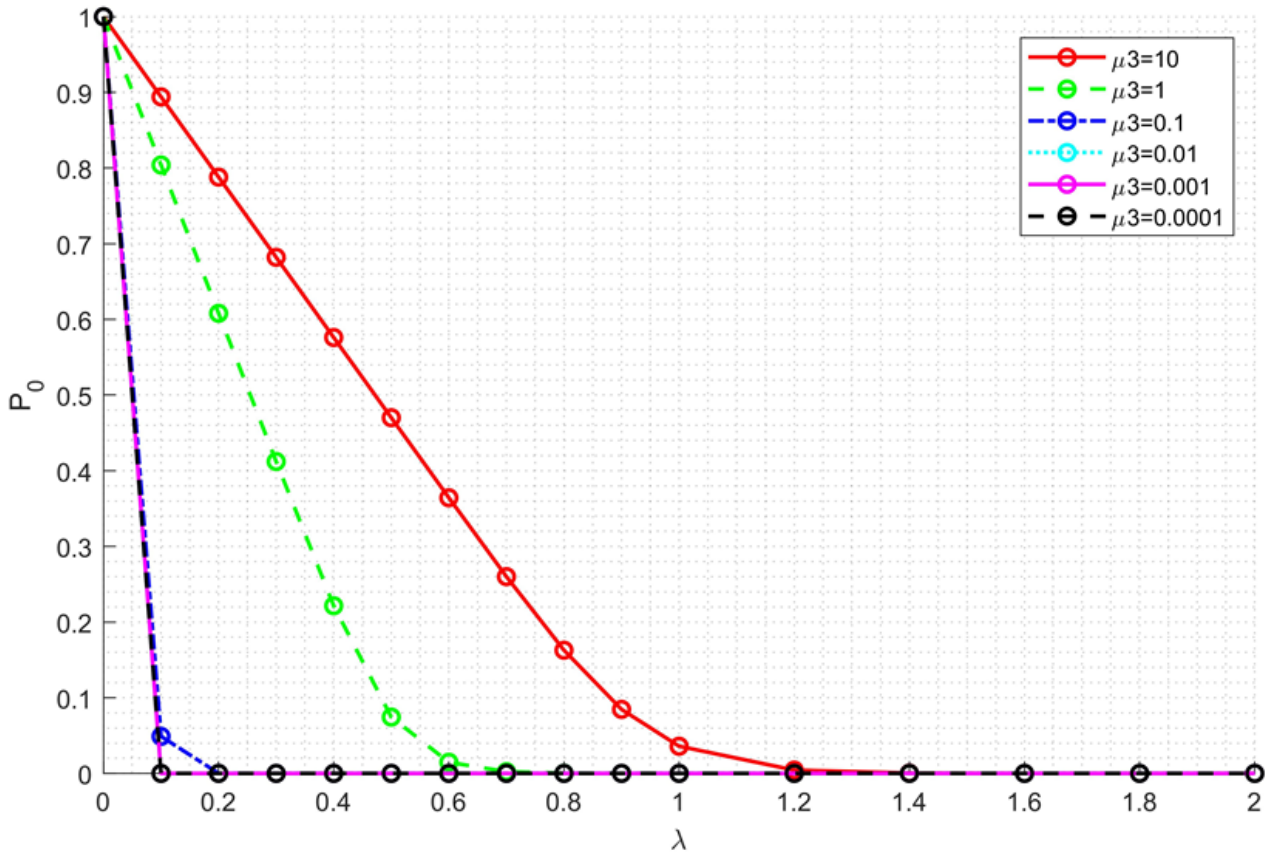

Figure 8

System “Idle” Probability Vs Arrival Requests as Mining Stage Complexity is varied

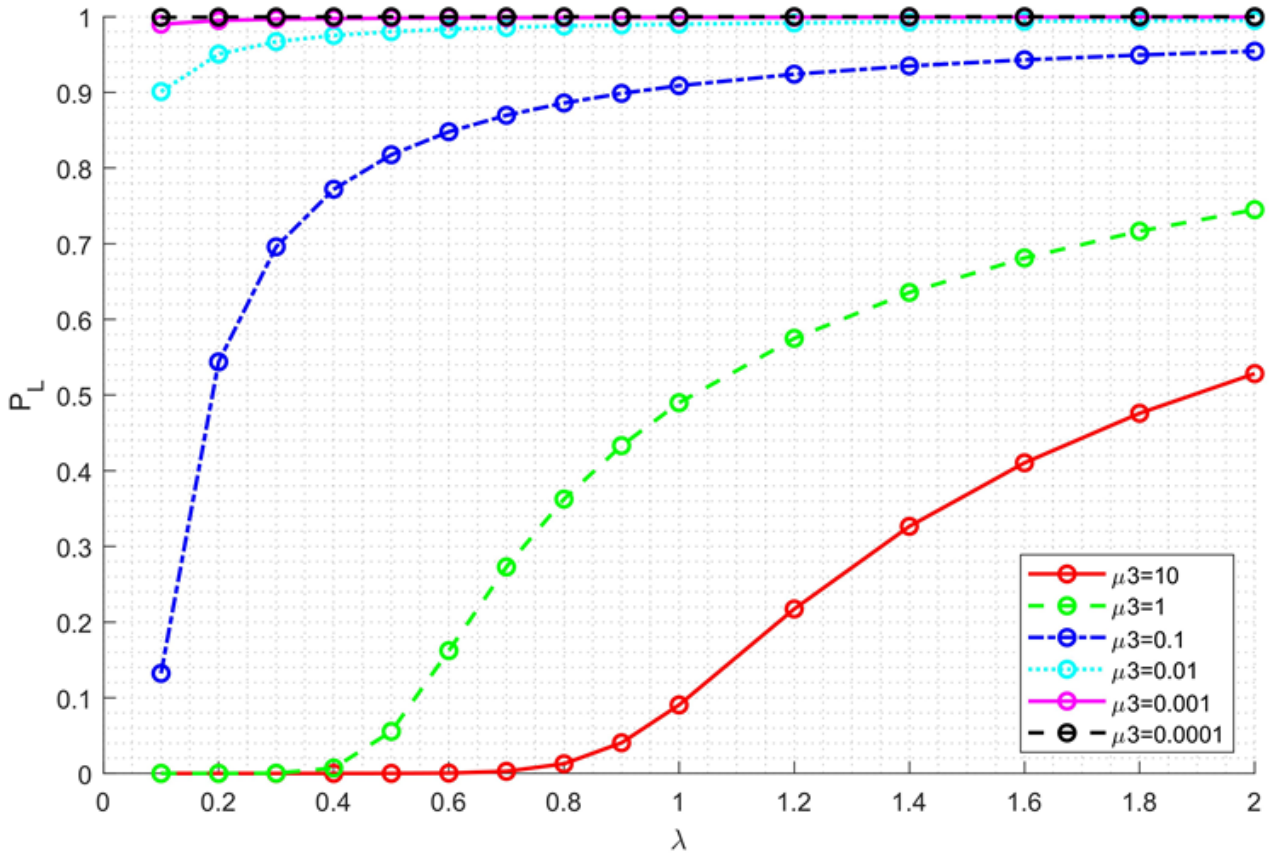

Figure 9

Blocking Probability Vs Arrival Requests as Mining Stage Complexity is varied

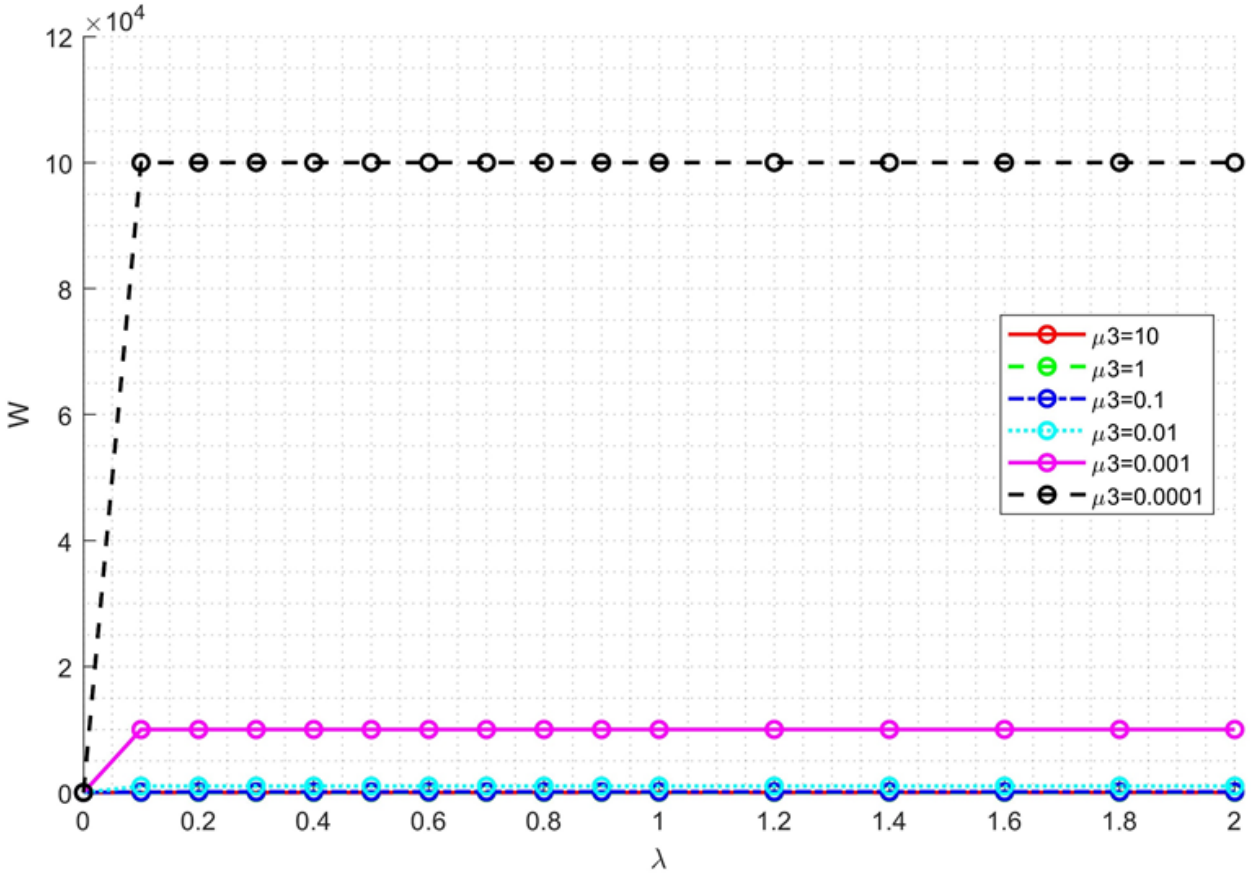

Figure 10

System Waiting Time Vs Arrival Requests as Mining Stage Complexity is varied

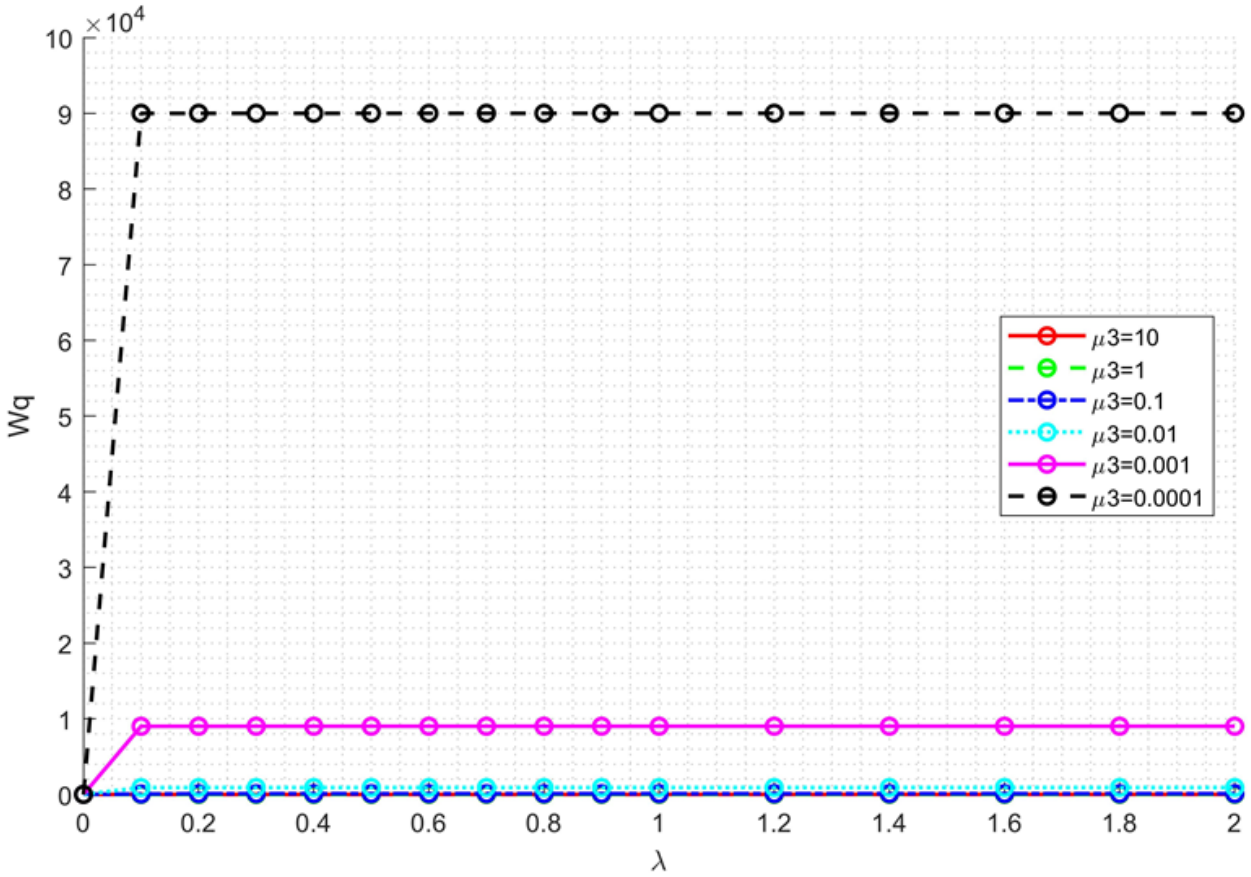

Figure 11

Queue Waiting Time Vs Arrival Requests as Mining Stage Complexity is varied