Performance Evaluation of M|M|1 Queueing Models in Cloud Computing Environments

Xinzichen Li *

Khoury College of Computer Sciences, Mills College at Northeastern University, Oakland, CA, 94613, United states

* Corresponding Author Email: li.xinzi@northeastern.edu

Abstract. Queues are ubiquitous, occurring at any given moment. Traffic congestion, healthcare facilities, dining establishments, and even the digital realm. With the continuous advancement of global society, there has been a significant surge in the utilization of the internet, leading to a substantial increase in the volume of incoming service requests occurring on a constant basis. The emergence of cloud computing can be attributed to its inherent advantages in terms of user comfort and cost-effectiveness. The utilization of cloud computing enables users to do computational tasks without the need for physical server ownership. However, it is important to note that the presence of queues is not eliminated as a result. Queueing theory was established as a means to address the issue of lengthy and sluggish lines. This research focuses on the analysis and optimization of queueing systems in order to develop efficient and expedient queues, while also considering cost-effectiveness. In the context of cloud computing, particular attention is given to the goal of energy conservation. This study aims to examine the performance of M|M| queues within the context of cloud computing. It will include both numerical and textual analyses, as well as proofs, while primarily focusing on comparing the similarities and differences of two M|M| queues.

Keywords: Performance Assessment, M|M| Queueing Models, Cloud Computing Environments.

1. Introduction

Cloud computing is rapidly growing; therefore, service delivery efficiency and responsiveness are crucial. Cloud computing, or the "cloud," has become a popular method for configuring computer resources and providing web application infrastructure [1]. Premier IT corporations have also provided public, commercial clouds as an investment opportunity. Amazon Elastic Compute Cloud (EC2) provides scalable cloud processing. It helps developers use web-scale computing [2]. Cloud computing requires Infrastructure as a Service (IaaS) to remotely access computer resources. Cloud computing infrastructures remotely provide network access, routing, storage, and application services. Microsoft Azure's Virtual Machines, Google Compute Engine [3], and Amazon Web Services' Elastic Compute Cloud are prominent IaaS providers.

Predicting, measuring, and optimizing performance under different workloads and situations is crucial as enterprises move their computing needs to the cloud. The queueing model is a popular performance assessment method. In computer networks, telecommunications, and transportation systems, queuing models have been used to study service performance. The M/M/1 queue is one of the most commonly researched models due to its tractable mathematical qualities and ability to give valuable insights into system performance. M/M/c models, where c is the number of servers, provide for a more realistic portrayal of real-world systems, especially in cloud computing where resources may be dynamically assigned.

This study evaluates cloud computing-specific M/M/c queueing models. Due to the cloud's scalability, flexibility, and fluctuating demand patterns, knowing how numerous servers (or virtual machines) handle workloads, communicate, and impact system performance is critical. This dissertation will thoroughly examine M/M/c model parameters, assumptions, and ramifications in such circumstances. To connect queueing theory to cloud computing's practicalities.

2. Background Concepts

2.1. Understanding Cloud Computing

Cloud computing, in its most basic form, refers to the provision and consumption of computer resources as a service through the internet. It marks a paradigm change from traditional on-premises infrastructure, in which enterprises own, control, and maintain their physical hardware, to a model in which computing capabilities are dynamically distributed, scaled, and maintained by third-party providers in faraway data centers.

The essence of cloud computing is its capacity to give on-demand access to a shared pool of customizable resources (e.g., networks, servers, storage, applications, and services) that can be swiftly supplied and released with no administration effort or contact from service providers. This game-changing strategy leverages virtualization technology to allow several users to share the same physical resources while creating the illusion of exclusive access. The essential feature that differentiates cloud computing from previous computing models is On-Demand Self-Service, which allows users to access computing capabilities. Users can also access available materials from any device, including those with internet connection.

2.1.1. Service Models

IaaS (Infrastructure-as-a-Service): IaaS gives pay-as-you-go access to essential computing resources—physical and virtual servers, networking, and storage—over the Internet. IaaS allows end users to grow and reduce resources as needed, eliminating the need for large, upfront capital expenditures or needless on-premises or 'owned' infrastructure, as well as overbuying capacity to accommodate periodic spikes in consumption. In contrast to SaaS and PaaS (and even newer PaaS computing paradigms like containers and serverless), IaaS gives consumers the most control over cloud computing resources. When it first appeared in the early 2010s, IaaS was the most popular cloud computing paradigm. While it remains the cloud model of choice for many applications, the adoption of SaaS and PaaS is expanding far faster [4].

PaaS (Platform as a Service): PaaS offers software developers an on-demand platform (hardware, entire software stack, infrastructure, and even development tools) for running, developing, and managing applications without the expense, complexity, and cost of maintaining the platform on-premises. Inflexibility. PaaS allows cloud providers to host everything in their data centers (servers, network, storage, operating system software, middleware, and databases). Developers may "spin up" the servers and environments they need to operate, create, test, deploy, manage, update, and grow their applications by simply selecting from a menu. PaaS is now often developed using containers, a virtualized computing approach that eliminates virtual servers. Containers virtualize operating systems, allowing developers to bundle programs with only the operating system functions required to operate on any platform, without the need for changes or middleware [5].

SaaS (Software-as-a-Service): SaaS, also known as cloud-based software or cloud applications, is application software that is hosted in the cloud and is accessed by users using a web browser, a specialized desktop client, or an API that connects with a desktop or mobile operating system. SaaS customers typically pay a monthly or annual membership fee; however, some may provide 'pay-as-you-go' pricing depending on the user's actual usage. Users do not lose data if their device crashes or malfunctions since SaaS keeps application data in the cloud with the application. Today, SaaS is the dominant distribution paradigm for the majority of commercial software. There are hundreds of SaaS solutions available, ranging from industry-specific and departmental apps to strong enterprise databases and AI (artificial intelligence) software [6].

Cloud computing combines decades of technology progress into a unified and scalable platform, giving people and companies unprecedented access to processing power and storage capacities. The deep ramifications of cross-sections of cloud computing and its performances will be further examined and studied as this research proceeds.

2.2. Fundamentals of Queueing Theory

Queuing theory is the study of lines waiting to be served. It has its origins in mathematics and operations research. This theory has its roots in the study of real-world events involving queueing, whether it involves individual people or data packets. The Danish engineer, statistician, and mathematician Agner Krarup Erlang studied the Copenhagen telephone exchange in the early 1900s [6], which is considered the beginning of modern queueing theory. In order to anticipate queue lengths, waiting times, service times, and other relevant performance indicators, queueing theory employs mathematical models to decipher the dynamics of queues as a theoretical framework [7].

When supplies are low, lines form. Since the lack of a line indicates an expensive overcapacity, some waiting time is acceptable even in the best of circumstances. With the help of queueing theory, we can create systems that service clients fast and effectively without breaking the bank [8].

Queuing theory is the study of how customers arrive to a business and how those customers are now being served, including evaluations of computer performance. The final product is a series of findings that attempts to locate problems and propose solutions [9].

The next sections will address the general research topics posed by this study by delving more deeply into the individual queuing models and their applicability to cloud computing.

3. Methodology

3.1. Criteria for Performance Assessment

Performance evaluation requires a well-defined set of criteria to provide a thorough, detailed, and practical study, especially when it comes to cloud computing systems that employ queueing theory.

Mean Response Time (E[T]): This is the entire amount of time that an entity (such as a request or data packet) spends in the system. It covers both the time spent waiting in the line and the time spent processing. Minimizing reaction time is critical in cloud systems for guaranteeing user satisfaction and service level agreement (SLA) compliance.

Throughput (X): Often expressed in requests per unit time, throughput shows the cloud system's ability to handle and execute activities. An ideal cloud configuration should increase throughput without sacrificing other performance measures. The following equation connects throughput and mean reaction time: N = E[T] X, where N is the number of occupations

Utilization: The percentage of time a resource (such as a server or virtual machine) is actively processing requests. High utilization indicates effective resource use, however extremely high numbers indicate resource congestion or possible bottlenecks.

Mean Waiting Time (E[TQ]): Mean waiting time denotes how long a job will spend in the Box in Figure 1, i.e. how long it will wait to be served. This is connected to the number of tasks in the queue (E[NQ]) using the following equation: E(NQ) = E(TQ).

Service Rate Variability: Given the variety of cloud environments, analysing the variability in service rates across different resources can provide insights into potential performance differences and bottlenecks.

When these criteria are combined, it is clear that performance evaluation in cloud computing, as supported by queueing theory, is complex. However, due to the inherent unpredictability, volatility of cloud computing environments, and mathematical complexity of queueing models, the performance evaluation would have limitations that will be discussed later.

3.2. Limitation due to External Factors

External variables have an impact on queueing theory in cloud computing. Other factors, in addition to the core model, can contribute to volatility and bias in performance assessments. To provide relevant and effective queueing-based investigations, these external variables must be recognized, assessed, and controlled. The sections that follow discuss key approaches and issues in this project.

Latency and bandwidth: Network performance is critical in distributed cloud systems. Network latency and capacity constraints can have an impact on queue dynamics, particularly request arrival and servicing. These problems may be avoided by continuously monitoring and standardizing network conditions throughout the investigation.

Workload Characterization: The complexity, kind, and interdependence of cloud workloads vary. A consistent and representative workload will eliminate task heterogeneity biases throughout the study.

Differences in Hardware: Cloud infrastructures may span several data centers with varying hardware. CPU speed, memory access, and storage I/O all have an impact on queue performance. To account for these changes, the model must preserve hardware homogeneity during analysis or parameterize hardware variability.

Queuing can be slowed by security protocols such as encryption and intrusion detection systems. To account for their influence, the performance overheads of these methods are considered into evaluations.

Human factors: Cloud management, maintenance, and manual interventions can all lead to uncontrollable fluctuations. Standardized operating procedures can help to decrease human-induced variability.

Overall, queueing theory provides a solid mathematical framework for analyzing cloud computing efficiency, but its practical application necessitates a more thorough approach. By recognizing and correcting for external variables, scientists and practitioners may ensure accurate and generalizable conclusions.

4. Analysis of M/M/ Queue Models in Cloud Contexts

4.1. The M/M/1 Model: An Illustration

In the context of queuing models or queuing theory, it is necessary to regard the arrival times as random variables and service times, which also exhibit random characteristics. However, to not overload, the income job frequency must be lower than the serves speed. The queuing system seen in Figure 1 may be described as a process in which consumers consciously decide to present themselves for a designated examination, patiently wait if the examination cannot start immediately, and depart after they have been attended to. The server in this case is the CPU, which would provide service to 1 job at a time. Once the server fully serves the current job, it will move on to the next one. The phrase "customer" encompasses individuals, products, machines, and other entities.

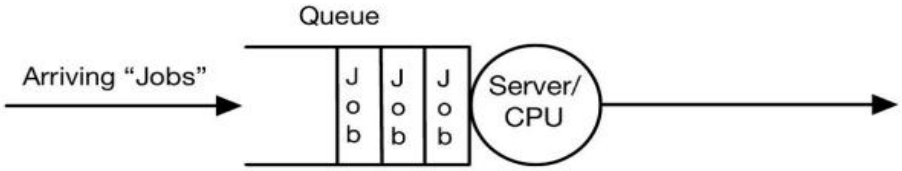

Figure 1. A queueing system (Photo/Picture credit: Original).

This model is constructed based on specific assumptions regarding the queueing mechanism. The system is characterized by either an exponential inter-arrival time distribution or a Poisson distribution of arrivals with a mean rate denoted as "λ" (Lambda). The inter-arrival times exhibit independent, identical, and exponential distribution with parameter λ. As seen in Figure 2, the symbol λ corresponds to the category labeled "Arriving Jobes" in Figure 1, thereby serving as the designated unit for this particular category. The queuing system depicted in the illustration has a single service unit, where the service times follow an independent, identically distributed exponential distribution with the parameter "μ" (Mu). Therefore, the processing speed of a task is contingent upon the server, denoted as "μ", as well as the size of the job being served, measured in units of $1 \div \mu$.

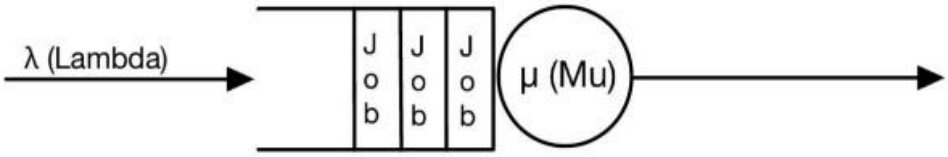

Figure 2. Units for a queuing system (Photo/Picture credit: Original).

The theoretical capacity of the proposed system is deemed to be infinite; but, in practical terms, such a scenario is unattainable. The presence of infinite lines inside the system will inevitably lead to a catastrophic system failure. The queue of requests is processed in a First Come First Serve (FCFS) manner, which is often regarded as an equitable approach. The M/M/1 queue model is widely employed in cloud computing due to its simplicity and cost efficiency since it enables the achievement of efficiency in systems for delivering improved quality of service while simultaneously decreasing cost, complexity, and energy consumption. However, the request/response process is time-consuming due to the utilization of a single service unit and the presence of long waiting lines. This aspect will be further elaborated upon in subsequent papers.

4.1.1. Related Analysis

Figure 3 shows the notation of the queuing system on the example of the M|M|1 queuing systems this paper is going to investigate about. As it is shown, the first letter "M" stands for how interarrival times of the jobes are being distributed, while the second "M" shows the distribution of Job sizes, which relates to the service time (S). In this situation, the service time S follows an Exponential distribution, which should be written as: S ~ Exp(μ) [10]. The number in the notation simply stands for the number of servers used in the queueing models, hence, M|M|1 uses 1 server.

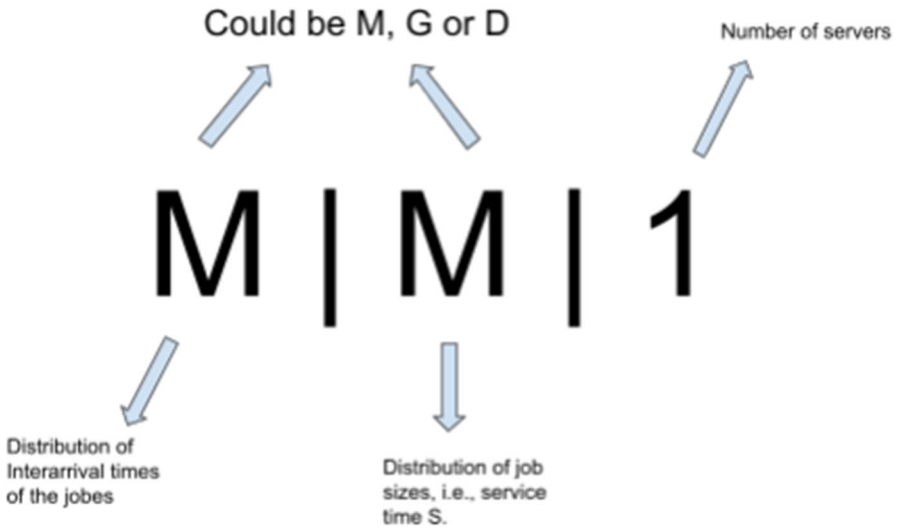

Figure 3. Queueing system notation (Photo/Picture credit: Original).

Please note that other possible letters can be used for the first two parts, replacing M|M. Firstly, the M here represents memoryless or Markovian, signifying that both the interarrival rate and service time follow an Exponential distribution. The letter G stands for general, indicating that both the interarrival rate and service time follow an Arbitrary distribution. Another possible letter is D, which represents Deterministic.

The analysis of the mean waiting time, denoted as E[TQ], involves a specific equation. This equation can be expressed in another form involving the expectation of service time, denoted as E[S]. It is important to note that in the M|M|1 queue, an increase in certain parameters directly augments the delay time, leading to reduced efficiency in the queues.

4.1.2. Exploration

When a job enters the system, its delay is dependent on two primary factors:
It must wait for all preceding jobs currently in the queue.

It must wait for the job that is currently being served upon its arrival.

Hence, the term represents the average remaining service time for the job in service when a new job arrives. In the M|M|1 queueing model, if the coefficient of variation is high, then the average response time will dramatically increase as well.

4.2. Delving into the M/M/C Model

This model is built on the same assumptions as the M/M/1 model, except that it includes several servers running in parallel, equal to C. Customers/jobs are expected to come at an average rate of "customers per unit time" according to a Poisson process, this is the same as M|M|1 which the unit is λ. At any service unit, requests are served on an FCFS basis, which is still the same as M|M1. Customers are served at an average rate of "customers per unit time" by identical servers, which is $\frac{1}{\mu}$. This model may be used to measure the performance of a large number of service units, whereas M/M/1 or M/M/2 can be used for one or two service units. According to Figure 4 below, M|M|C queues would look almost identical, depending on the number of servers for C.

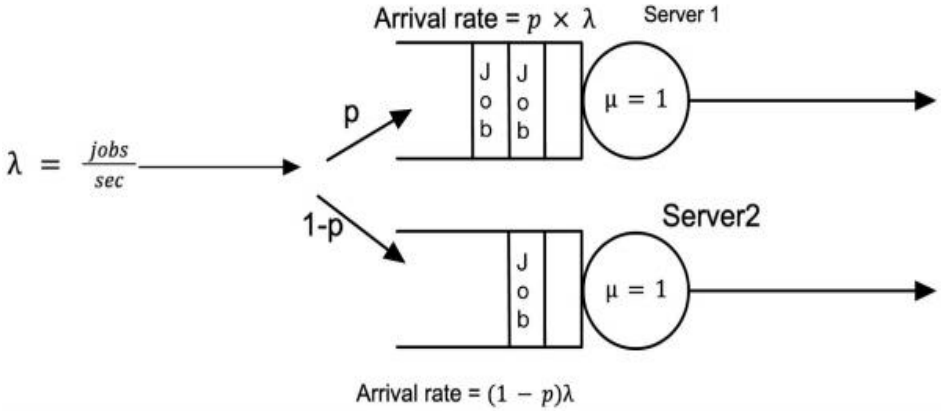

Figure 4. Multi servers queue model (Photo/Picture credit: Original).

Based on Figure 4, it is evident that the M|M|C system maintains a central queue. Each time a server completes a job, the next queued job proceeds to the server. In this scenario, both S1 and E[T] values are defined, as are S2 values.

Implementing the M/M/C paradigm has notably enhanced throughput and reduced reaction time. However, having essentially unlimited buffer capacity may not be practical in cloud server applications. The M/M/C model indeed outperforms the M/M/1 and M/M/2 models in terms of performance. Nevertheless, the performance dynamics change when different μ values are considered. M|M|1 could potentially outperform M|M|C when M|M|C's μ is less than M|M|1's μ, a situation analogous to many slower servers competing with a supercomputer.

In these conditions, the expected time a job spends in the system, E [T M|M|C], is defined with PQ representing the probability that an arriving job has to wait in the queue before being served. After combining M|M|C and M|M|1, the expected time a job spends in the M|M|K system, E [T M|M|K], is also defined, leading to two potential cases:

When δ is high (approximately 1), most jobs will likely spend a defined time in the system. In this case, M|M|C and M|M|1 have about the same mean response time.

When δ is low (approximately 0), PQ is also approximately 0. In this case, M|M|K is dramatically slower than M|M|1.

Nonetheless, significant job size fluctuations in the M|M|1 system tend to cause delays for smaller tasks queued behind bigger jobs, resulting in poorer performance compared to the M|M|K system, which allows smaller-sized jobs to pass through more expeditiously.

4.3. Insights into the M/M/C/N Model

In the present model, all remaining parameters remain constant except the newly included parameter "N". The symbol "N" represents the numerical value denoting the buffer's capacity, indicating the maximum number of requests it can accommodate. Moreover, in the queuing system, two scenarios may occur when there are N requests at any one moment. The first instance is when N is less than C in the

M|M|C|N notation, indicating no queue. On the other hand, if N is greater than or equal to C, all the servers will be occupied and a queue will be established. Several potential application areas for this paradigm include using counters in libraries to manage the process of issuing and returning books, as well as border checkpoints established to verify the customers' passports. Thus far, this model demonstrates notable attributes such as energy efficiency, dynamicity, and enhanced performance efficiency. This model fulfills a significant role in upholding the standard of service quality.

5. Challenges

In addition to the aforementioned limitations on the methodology, the study encounters some other challenges. To commence, it is noteworthy that there exists a lack of both a simulation and an implementation of any entity. The essay thus far has not considered the diversity present in modern data centers, encompassing both real computers and virtual machines. This study aims to enhance the precision of estimating the number of virtual machines necessary to provide more accurate responses. The identification of bottlenecks and the determination of optimal parameter values. The significance of this notion is paramount in the context of meta-analysis. Furthermore, the issue of redundancy is not taken into consideration.

6. Conclusion

In conclusion, the objective of this study was to investigate the utilization of queueing theory with cloud computing. Extensive descriptions, explanations, and justifications were employed to showcase the operational efficiency of the M|M| system inside a cloud computing environment. The M|M|C queueing models have a notable benefit compared to the M|M|1 models in scenarios when tasks of varying sizes and greater income levels are being handled. The performance of M|M|1 queues shown notable improvement when the server in the M|M|1 system effectively leveraged its superior service rate, while receiving a lower influx of tasks. Furthermore, the article offers a comprehensive elucidation of the entire queueing procedure, commencing from the moment a task is appended to the queue and culminating at the juncture when a job is completed. The M|M queue models have identified the key characteristics that significantly contribute to the computation of the mean response time, hence reflecting the quality of the served items. These factors have been analyzed and deconstructed into their constituent elements. M&M queues are widely used in cloud computing as a predominant queueing technology, and they also hold a prominent position in physical queueing scenarios. Instances such as grocery stores, fast food establishments, or the queue at an airport terminal for the purpose of embarking on an airplane serve as illustrative examples. Further research might explore further aspects of queueing theory in the context of cloud computing. Potential factors to consider may encompass scalability, cost-effectiveness, variability in service rates, fault tolerance, and consistency. Comparisons may also be drawn between M|M| queue models and other types of queuing models, such as M|G|1 or M|G|S queue models.

References

[1] Goswami, V., Patra, S. S., & Mund, G. B. (2012, March). Performance analysis of cloud with queue-dependent virtual machines. In *2012 1st international conference on recent advances in information technology (RAIT)* (pp. 357 - 362). IEEE.

[2] Gahlawat, M., & Sharma, P. (2013). Analysis and performance assessment of cpu scheduling algorithms in cloud using cloud sim. *Int. J. Appl. Inf. Syst*, *5* (9), 5 - 8.

[3] Bai, W. H., Xi, J. Q., Zhu, J. X., & Huang, S. W. (2015). Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. *Mathematical Problems in Engineering*, *2015*.

[4] Calheiros, R. N., Ranjan, R., & Buyya, R. (2011, September). Virtual machine provisioning based on analytical performance and qos in cloud computing environments. In *2011 International Conference on Parallel Processing* (pp. 295 - 304). IEEE.

[5] Li, L. (2009, June). An optimistic differentiated service job scheduling system for cloud computing service users and providers. In *2009 Third international conference on Multimedia and Ubiquitous Engineering* (pp. 295 - 299). IEEE.

[6] Zhu, X., Zhao, Z., Wei, X., & others. (2021). Action recognition method based on wavelet transform and neural network in wireless network. In 2021 5th International Conference on Digital Signal Processing (pp. 60 - 65).

[7] Xiong, K., & Perros, H. (2009, July). Service performance and analysis in cloud computing. In *2009 Congress on Services-I* (pp. 693 - 700). IEEE.

[8] Rajput, R. S., & Pant, A. (2018). Optimal resource management in the cloud environment-a review. *International Journal of Converging Technologies and Management (IJCTM)*, *4* (1), 12 - 24.

[9] Osman, Y. (July 2023). Analytical Design and Performance Evaluation of Computing Systems [Lecture 3]. Carnegie Mellon University.

[10] Xia, Y., Zhou, M., Luo, X., Zhu, Q., Li, J., & Huang, Y. (2013). Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. *IEEE Transactions on Automation Science and Engineering*, *12* (1), 162 - 170.