UNIVERSITEIT VAN AMSTERDAM

Discrete Events Simulation - Queuing Theory

December 4, 2023

Priyank Venkatesh 14133830
Guoda Paulauskaitė 12255076
Koen Weverink 14711982

Abstract

This paper aims to investigate different aspects of queuing theory. The aspects investigated are changing the number of servers ($n = 1, 2, 4$), modifying the load capacity ($\rho = 0.8, 0.95, 0.99$), implementing different scheduling policies (First in First out (FIFO) & Shortest Job First (SJF)) and modifying the service rate distributions. These modifications are implemented to investigate their impact on the average waiting time for customers in a queue. The average waiting time in the M/M/1 system appears to be higher compared to M/M/n systems, particularly as ρ approaches 1. No significant difference was found between the average waiting times between FIFO and SJF. The deterministic service rate distribution outperformed its stochastic counterparts (exponential and hyper-exponential) in a single-server system, yet no statistically significant results were found for multiple-server systems. Hyper-exponential and exponential distributions yield statistically different results, but which of the two distributions performs better remains undetermined and requires further investigation.

1 Introduction

Queuing theory emerged as a branch of applied mathematics in the early 20th century with the work of Danish engineer A.K. Erlang [1]. In 1909, Erlang developed the foundations of queueing theory for the Copenhagen Telephone Exchange. He aimed to address the challenge of providing efficient telephone service by understanding the patterns of calls arriving at the center and the optimal number of telephone lines required to meet demands [2]. As queuing theory evolved, it has been applied to various sectors such as computer science, transportation, healthcare, and more [3].

There are several aspects of querying theory that are worthwhile to study. This report examines the impact of modifying a few of those aspects. They involve changing the number of servers (n = 1, 2, 4), load capacity ($\rho = 0.8, 0.95, 0.99$), First in First out (FIFO) & Shortest Job First (SJF) scheduling policies, and modifying the distribution of the service rates. These changes are made to examine their impact on the average waiting time of the customers in a queue.

In the Theoretical Framework section, the report examines the elements of queuing theory, covering notation, Little's Law, and the Pasta Theorem. The methodology section details the implementation of various queuing schemes, outlining the software tools utilized for simulations and statistical analysis. The results section presents and comments on various plots and Welch's tests obtained during the experiments. The paper concludes with a discussion section, providing an interpretation of the findings within the context of queuing theory and a proposition for future work.

2 Theoretical framework

2.1 Queuing Theory

Queuing theory constitutes a discipline within applied probability theory and operates as a discrete event simulation (DES) methodology. It has versatile applications in domains such as computer systems, telecommunication networks, and customer service in retail establishments. At its core, queuing theory revolves around the concept of a *request* entering a designated *queue*, awaiting subsequent *servicing* [4]. Using simulation techniques, queuing theory aims to derive valuable insights into statistics, such as the mean time spent in the queue, mean response time, mean utilization of services, and the distribution of requests within the queue.

2.2 Notation

Before describing the different model elements, first it is necessary to understand the notation most commonly used in Queuing Theory, namely the Kendall Notation. Introduced by Kendall, this notation serves as a convenient method for characterizing queuing methodologies [5]. The general format of the Kendall Notation is $A/B/m/N-S$. Here, A signifies the inter-arrival time distribution and B represents the service time distribution. Commonly used abbreviations for these components include M (Memoryless), D (Deterministic), H (Hyper-exponential), and G (General). Following these, the notation includes the parameter m, denoting the number of servers. Subsequently, N denotes the maximum queue size (omitted if the queue is infinite); finally, S indicates the queuing discipline. The subsequent subsection will discuss a more in-depth exploration of these components.

2.2.1 Model Elements

Within the queuing model, various elements are delineated, each contributing distinct characteristics. The Memoryless component relies on the fact that the occurrence of the next event does not depend on any past or future events. In this context, the report exclusively employs the exponential distribution, represented as $A(t) = 1 - e^{-\lambda t}$. The Deterministic component does not include randomness, as it draws a constant value. The Hyper-exponential distribution is the summation of exponential distributions, each weighted with a certain probability. Expressed with k phases, it takes the form $A(t) = \sum_{j=1}^{k} q_j(1 - e^{-\mu_j t})$ [4]. An unspecified distribution characterizes the General case, although typically, at least the mean and variance are known. Finally, the system load, represented by ρ, can be calculated in a single server system: $\rho = \frac{\lambda}{\mu}$. For a system with n servers, it is $\rho = \frac{\lambda}{(n\mu)}$, where parameters λ is the request arrival time, and μ is the capacity of each of n equal servers [6]. Both are independently and identically distributed random variables sampled from the exponential distribution.

Beyond these probabilistic components, various queuing types are identified. The default type is FIFO (First In, First Out), where requests are serviced according to their arrival. Alternatively, Random queuing involves handling requests in a random order. Priority queuing assigns a predefined priority level to each request, with the highest priority being addressed first. Lastly, SJF (Shortest Job First) prioritizes jobs based on their service time, with the shortest job taking precedence [6].

2.3 Little's Law and the PASTA Theorem

Little's Law, a fundamental principle, establishes a relationship among the mean number of customers $\mathbb{E}(L)$, the mean arrival rate (λ), and the mean sojourn time $\mathbb{E}(S)$. Assuming a non-infinite queue, this relationship is expressed as $\mathbb{E}(L) = \lambda \cdot \mathbb{E}(S)$ [7].

The Poisson Arrivals See Time Averages (PASTA) Theorem is a notable property stemming from the memorylessness inherent in the exponential distribution (M/-/-). It is crucial to note that this theorem is exclusively valid for Poisson arrivals. The theorem states that the fraction of requests encountering the system in a specific state is identical to the time the system spends in that state [4].

2.4 Average Waiting Times: M/M/1 vs. M/M/n

This section compares the average times in a single-server system with a multiple-server system. We consider interarrival times and distribution of service times to be Markovian with a FIFO service discipline. Intuitively, the average waiting times are expected to be lower in a multiple-server system due to parallel processing and task distribution across servers. The system load is distributed, preventing a bottleneck that otherwise can occur in a single-server system. Said observation can also be proven mathematically [7].

The PASTA property states that an arriving individual perceives the average number of customers to be $\mathbb{E}(L)$. Every customer's service time is expressed as $\frac{1}{\mu}$. Then, we can conclude that an arriving customer has to wait for every customer to be served, including themselves, resulting in the following expression for the sojourn time (time spent in the system):

$$\mathbb{E}(S) = \mathbb{E}(L)\frac{1}{\mu} + \frac{1}{\mu} \quad (PASTA) \tag{1}$$

Furthermore, by Little's Law, we know that the following is true:

$$\mathbb{E}(L) = \lambda \mathbb{E}(S) \tag{2}$$

Solve the system of equations (1 and 2) by substituting 2 into 1:

$$\mathbb{E}(S) = \lambda \mathbb{E}(S)\frac{1}{\mu} + \frac{1}{\mu}$$

$$\Rightarrow \mathbb{E}(S) = \frac{1}{\mu - \lambda}$$

For one queue, $n = 1$, the arrival rate is λ, and for multiple queues, the arrival rate for each queue is $\frac{\lambda}{n}$ as the customers are distributed. Hence, the sojourn times are as follows:

$$\text{For } n = 1, \quad \mathbb{E}(S) = \frac{1}{\mu - \lambda}$$

$$\text{For } n > 1, \quad \mathbb{E}(S) = \frac{1}{\mu - \lambda/n}$$

The waiting times W can be obtained by subtracting the service time of the customer from the total time spent in the system:

$$\mathbb{E}(W_{n=1}) = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{\rho}{\mu - \lambda}$$

$$\mathbb{E}(W_{n>1}) = \frac{1}{\mu - \lambda/n} - \frac{1}{\mu} = \frac{\lambda/n}{\mu(\mu - \lambda/n)} = \frac{\rho}{n\mu - \lambda}$$

For $\mathbb{E}(W_{n>1})$, the denominator is larger when compared to $\mathbb{E}(W_{n=1})$ so $\mathbb{E}(W_{n>1}) < \mathbb{E}(W_{n=1})$. Then, we can conclude that the waiting time in a system with more than one queue is smaller than in a system with only one. Also, the waiting time decreases as n increases.

3 Methodology

Section 3.1 details the common packages used in the experiments. Section 3.2 describes the used pseudo-random numbers software. Implementing the FIFO and SJF policies for various changes in server number $n = 1, 2, 4$ is discussed in section 3.3. Section 3.4 discusses the implementation of different service rate distributions for M/M/N, M/D/N, and M/H2/N. Section 3.5 is for statistical testing.

There are some control variables in place for the FIFO and SFJ implementation for fair comparisons: number of simulations = 30, $\mu = \frac{1}{2}$, number of customers = 50,000 in increments of 5000, and change in ρ = 0.8, 0.95 and 0.99.

3.1 Common Software Packages

For both experiments, there is overlap with the software packages used in subsections 3.3, 3.4 & 3.5: Python 3.9 software [8], SimPy 4.1.1 [9], Numpy 1.24.0 [10], Pandas [11], Matplotlib 3.5.2 [12] and Scipy [13]. SimPy is the main tool used to implement the different discrete event simulations. Numpy selects the ten random seeds (with the chosen seed being the run count); Pandas is used to open, save, read, and manipulate CSV files and data frames; Scipy is used to run various statistical tests; and Matplotlib is used to visualize the simulations.

3.2 Pseudo-Random Numbers

To obtain statistically significant results, these experiments require running several simulations; a suitable method to achieve this is using the Numpy random number generator [10]. These numbers are generated using the Mersenne twister algorithm [10]. The produced random numbers allow for reproducible results obtained in the experiments.

3.3 FIFO & SJF implementation

The experiments use a discrete-event simulation (DES) program implemented in Python with the SimPy library [9]. The primary class for modeling the queuing system is MMNQueue, which allows for the exploration of different scenarios involving multiple servers (n), traffic intensity (ρ), and scheduling policies (FIFO and SJF).

The MMNQueue class initializes the simulation environment, server resources, and parameters such as arrival rate (λ), service rate (μ), and scheduling policy. The class includes two primary scheduling policies: First-In-First-Out (FIFO) and Shortest Job First (SJF). The former serves customers in the order of arrival, while the latter prioritizes the shortest service time.

Multiple simulations are conducted with varying values of n and ρ to investigate the impact of system parameters. Specifically, the experiments cover different numbers of servers (1, 2, 4) and traffic intensities ranging from $\rho = 0.8$, 0.95, 0.99. The methodology ensures statistical significance by executing multiple simulations for each configuration. The results are stored in a dataframe, and plots are generated and saved as PNG files.

3.4 Distributions of Service rate implementation

To explore the influence of various service time distributions on the average waiting time, the program considers M/M/N, M/D/N, and M/H2/N scenarios. The latter scenario describes a 2-phase hyperexponential distribution where 75% of jobs have an exponential service time with a mean of 1.0, and the remaining 25% have a mean of 5.0. To ensure comparability with the hyperexponential distribution, the exponential distribution in scenario M/M/N is set to have a mean service time of 2. This decision was made due to the weighted mean service time from the hyperexponential distribution:

$$p_1 * (1/\mu_1) + p_2 * (1/\mu_2) = 0.75 * 1.0 + 0.25 * 5.0 = 2$$

The hyperexponential distribution is implemented through the hyperexponential_2_phases function when service_distr='H2' in the MMNQueue class. Said function decides, with some preset probabilities (here 0.75 and 0.25), which mean (here 1.0 and 5.0 respectively) is used in the exponential distribution sampling. To reduce the variance, the simulations are executed multiple times, and confidence intervals are calculated to quantify the uncertainty around the average waiting time. The experimental design systematically explores different service rate distributions ('M,' 'D,' 'H2') and varying numbers of servers (n = 1, 2, 4). The results are captured in a dataframe, and plot visualizations are generated and saved as PNG files.

3.5 Statistical Testing

The Scipy library implements the Welch statistical test [14]. The report uses the function ttest_ind and sets the equal variance to False to run a Welch test [14]. This test compares the means of two independent groups of scores. We use this test since we compare samples from different experimental setups. Additionally, no information about the population variances is available which is fitting for a Welch test.

3.5.1 FIFO: $n = 1$ vs. $n = 2$ vs. $n = 4$

The H_0 states there is no difference in the expected value of the average waiting times when comparing FIFO implementations with different numbers of servers ($n = 1, 2, 4$). The alternative hypothesis H_A states that the expected value of the average waiting time from the FIFO implementation with fewer servers is expected to be greater than that with a larger number of servers.

3.5.2 SJF vs. FIFO

The H_0 states there is no difference in the expected value of the average waiting times when comparing SJF & FIFO implementations. The alternative hypothesis H_A states that the average waiting time expected value from the FIFO implementation is larger than that from the SJF implementation.

3.5.3 Distributions of different Service rates

The H_0 states there is no difference in the average waiting time expected values between the different service rate distributions. The H_A states that the expected values between the implemented service rates differ. However, note that we consider a one-sided alternative hypothesis when comparing with the deterministic service rate distribution where the expected value is expected to be smaller than its stochastic counterparts [15].

4 Results and Analysis

This section details the results obtained using the different number of servers, changes in scheduling policy, and service rate distributions discussed in Section 2. It displays the average wait time for the different ρ and examines its impact when different numbers of servers are available. It has the confidence interval in tabular form and is visible as a shaded area in Figures 1 and 2. The maximum number of customs used in all experiments is 50,000, increasing in step size of 5000. The simulations are run 30 times to produce significant results, and the average results over those runs are plotted.

4.1 Plotting Convention

Figures 1 & 2 each depicts six plots in total. The first row of plots for 1 & 2 is the average waiting time vs. the change in the number of customers with confidence intervals. The bottom row of plots represents the change in standard deviation over the change in the number of customers.

4.2 First in First Out (FIFO) Experiments

ρ	$\mathbb{E}(M/M/1) > \mathbb{E}(M/M/2)$	$\mathbb{E}(M/M/2) > \mathbb{E}(M/M/4)$	$\mathbb{E}(M/M/1) > \mathbb{E}(M/M/4)$
0.80	5.86e-5	1.04e-4	4.04e-6
0.95	1.56e-4	2.25e-4	6.37e-6
0.99	5.73e-4	5.15e-4	1.62e-5

Table 1: FIFO Experiments: P-values for (one-sided) Two-Sample Welch Test

Figure 1 shows that as ρ approaches 1, the average waiting time and associated confidence interval increase. This is visible in all three changes in n where the shaded area, representing the confidence interval, increases as ρ approaches 1. As the number of available servers $\boldsymbol{n}$ increases, we also see that the average waiting time decreases, visible in all three plots of Figure 1.

Table 3 compares the average waiting time for changes in ρ and n at 50,000 customers. The average waiting time for $n = 4$ & $\rho = 0.99$ peaks at just shy of 40. Compare that to when $n = 1$ & $\rho = 0.99$, the peak average waiting time is 165. This makes sense as more available servers result in a lower average queue time for customers as there are more people to help serve (refer back to proof in Section 2.4).

Some observations of the bottom row of plots from Figure 1 reveal that when $n = 1$, the standard deviation for $\rho = 0.8$ seems to level out at approximately 20000 customers. For $\rho = 0.95$, the standard deviation looks to be decreasing and still hasn't plateaued, as for $\rho = 0.99$, the standard deviation is still

Figure 1: First in First Out Experiments

increasing. In general, as the number of servers increases, the average wait time and standard deviation decreases and will stabilize sooner. The increase in the standard deviation for when ρ approaches one can be due to the queue being close to its capacity, leading to congestion and increased wait times for new arrivals (see theory in Section 2.4). Thus resulting in a larger variability in the waiting times. As the system becomes heavily loaded, small variations in customers' arrival times or service times can significantly impact the system's behavior. This increased sensitivity to variations can contribute to larger standard deviations and confidence intervals.

The observations from Figure 1 are supported by statistically significant results presented in Table 1. The p-values of the one-sided Welch test are sufficiently small (less than $\alpha = 0.05$) to conclude that the average waiting times are smaller for systems with a greater number of servers (for all service load values, ρ, considered in the experiment) using a FIFO service discipline. The statistical test considers and compares $n = 1, 2, 4$ servers.

4.3 Shortest Job First (SJF) Experiments

	$\mathbb{E}(M/M/n \text{ FIFO}) > \mathbb{E}(M/M/n \text{ SJF})$		
ρ	$n = 1$	$n = 2$	$n = 4$
0.80	4.80e-1	4.16e-1	4.16e-1
0.95	4.76e-1	4.38e-1	4.67e-1
0.99	2.51e-1	6.30e-1	4.57e-1

Table 2: SJF vs. FIFO Experiments: P-values for (one-sided) Two-Sample Welch Test

Comparable observations can be noted with an SJF scheduling policy akin to FIFO (see Section 4.2). As the load capacity ρ nears 1, there is an increase in the average waiting times and an increase in the standard deviation and corresponding confidence intervals. In table 3, an overview of all the different experiments can be seen when the number of customers is 50,000.

Compared to the FIFO experiments, the SJF scheduling policy consistently exhibits a lower peak in average waiting time across all combinations of ρ and n. This trend holds for each specific pairing of these parameters. Additionally, it is noteworthy that the standard deviation is generally reduced for SJF compared to FIFO. Analyzing the Confidence Intervals, it becomes apparent that, in most instances, they are slightly narrower for SJF than for FIFO. As anticipated, both FIFO and SJF demonstrate an increase in average

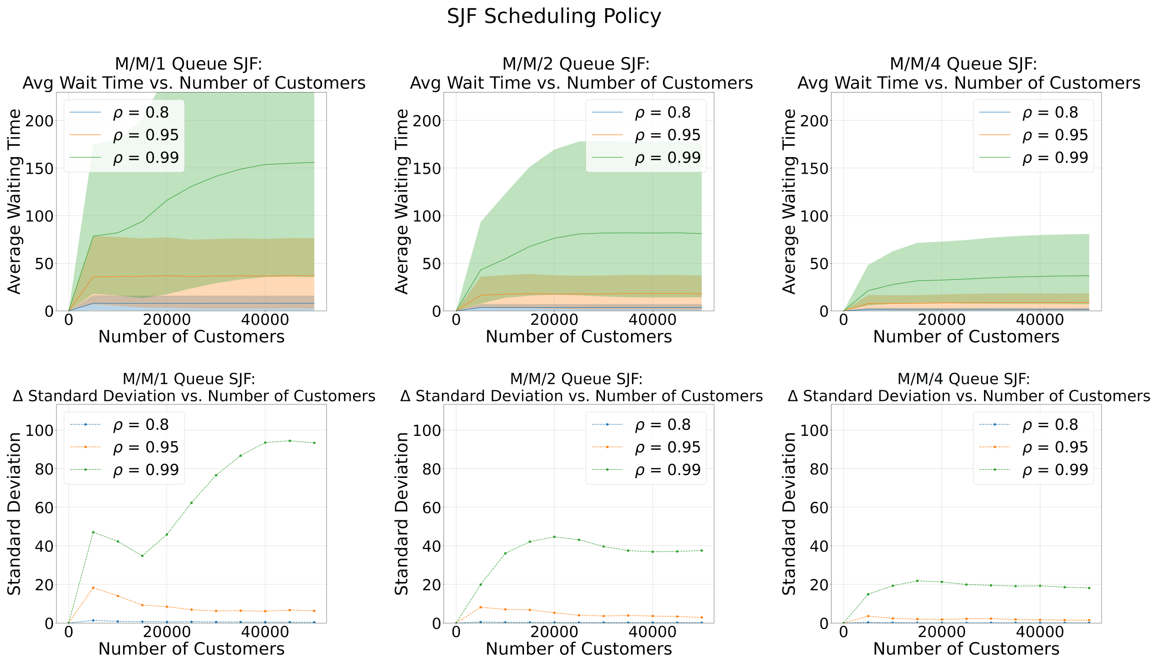

Figure 2: Shortest Job First Experiments

Scheduling Policy	n	ρ	Avg Waiting Time	Std	Confidence Interval
FIFO	1	0.80	7.99427	0.41307	(7.83739, 8.15115)
FIFO	1	0.95	38.58299	6.23116	(36.21646, 40.94952)
FIFO	1	0.99	165.89613	79.93547	(135.53747, 196.25479)
FIFO	2	0.80	3.59280	0.17838	(3.52505, 3.66055)
FIFO	2	0.95	19.05012	3.47593	(17.73, 20.37025)
FIFO	2	0.99	83.59537	55.65020	(62.46, 104.73074)
FIFO	4	0.80	1.48667	0.11683	(1.4423, 1.53104)
FIFO	4	0.95	8.50240	1.28768	(8.01335, 8.99145)
FIFO	4	0.99	40.12146	18.48111	(33.10252, 47.14039)
SJF	1	0.80	7.88388	0.39197	(7.73501, 8.03274)
SJF	1	0.95	36.92042	6.29717	(34.52882, 39.31202)
SJF	1	0.99	156.03455	93.39046	(120.56583, 191.50328)
SJF	2	0.80	3.50925	0.17473	(3.44288, 3.57561)
SJF	2	0.95	17.96471	2.90896	(16.85992, 19.0695)
SJF	2	0.99	81.10642	37.56273	(66.84049, 95.37236)
SJF	4	0.80	1.47672	0.08999	(1.44254, 1.5109)
SJF	4	0.95	8.85484	1.48774	(8.28981, 9.41987)
SJF	4	0.99	36.89474	18.17042	(29.99381, 43.79568)

Table 3: Comparing SJF and FIFO

waiting time with higher values of n and ρ. However, SJF consistently shows a comparatively smaller increase in waiting time than FIFO.

Although the general trend line suggests an improvement in the waiting times using the SJF scheme as compared to FIFO, nothing can be concluded since a two-sample one-sided Welch test did not yield sufficiently small p-values (see Table 2). Large confidence intervals and non-significant results suggest a need for further testing.

4.4 Variation in Service Rate Distribution Experiments

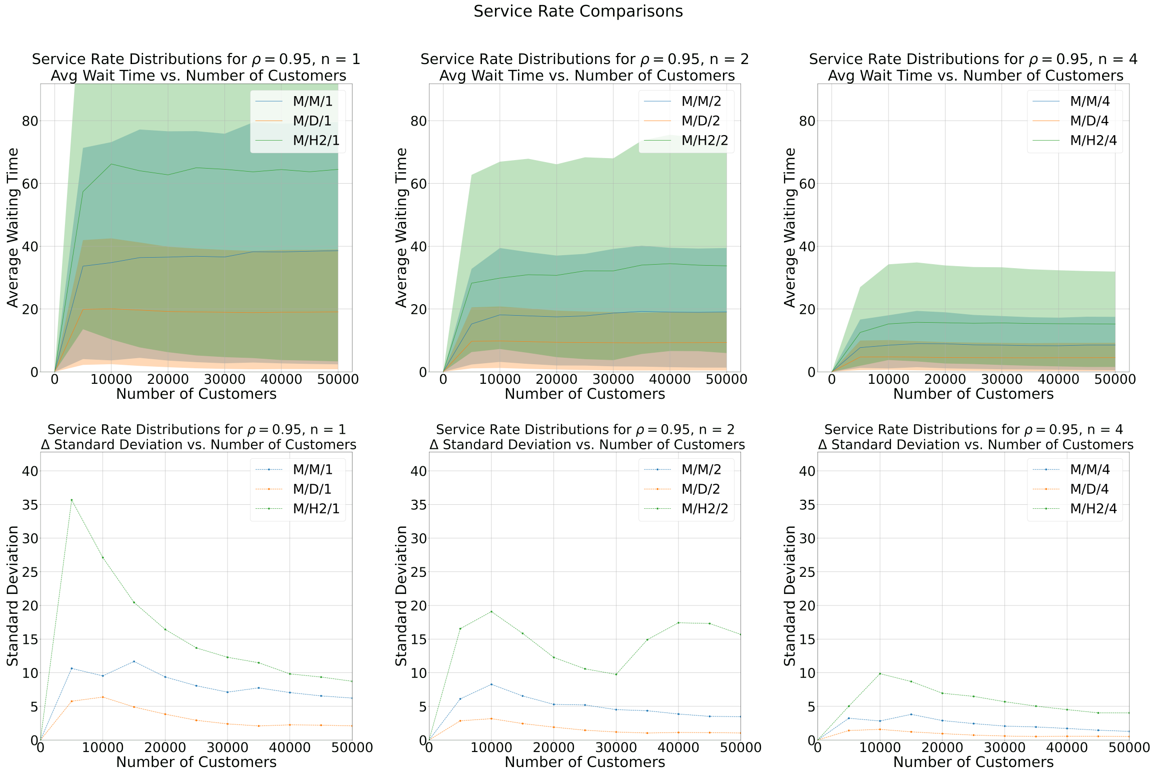

Figure 3: Average Waiting Times for Different Service Rate Distributions

Service Distribution	n	Avg Waiting Time	Std	Confidence Interval
M	1	38.58299	6.23116	(36.21646, 40.94952)
D	1	19.08362	2.10043	(18.2859, 19.88134)
H2	1	64.47779	8.72813	(61.16294, 67.79265)
M	2	19.05012	3.47593	(17.73, 20.37025)
D	2	9.33424	1.04677	(8.93669, 9.73179)
H2	2	33.74750	15.68499	(27.7905, 39.70449)
M	4	8.50240	1.28768	(8.01335, 8.99145)
D	4	4.51363	0.52229	(4.31527, 4.71199)
H2	4	15.19093	4.03211	(13.65957, 16.72228)

Table 4: Comparing different Service Rates

n	$\mathbb{E}(M/M/n) > \mathbb{E}(M/D/n)$	$\mathbb{E}(M/D/n) \neq \mathbb{E}(M/H2/n)$	$\mathbb{E}(M/M/n) \neq \mathbb{E}(M/H2/n)$
1	3.95e-4	2.64e-5	2.29e-3
2	6.60e-1	2.64e-5	2.14e-5
4	1.00e0	2.64e-5	5.28e-6

Table 5: Different Service Rate Distributions Experiments: P-values for (one-sided) Two-Sample Welch Test

Figure 3 displays a similar pattern to previously described results in comparing performance with different server numbers. We observe that the average waiting times decrease as the number of servers increases for all service rate distributions. In all three service rate distribution trends, we observe a sharp increase in the waiting times as the number of customers increases. This trend plateaus at around 5000 customers and remains steady as the customers increase past this value.

In Table 4, we observe that the deterministic service time distribution outperformed its stochastic counterparts across various servers. Deterministic service times have fewer delays than stochastic distributions where delays (and hence increased waiting times) are more likely [15]. Furthermore, the average waiting times were lower for the exponential service rate distribution than for the hyper-exponential distribution. Nonetheless, we encountered large confidence intervals and standard deviation values for stochastic distributions, as displayed in the lower plots in Figure 3. According to a one-sided two-sample Welch test (see Table 5), the deterministic service rate distribution is only better (yields shorter average waiting times) than its stochastic counterparts in a one-server system. We did not obtain significant values for many servers and cannot draw well-founded conclusions. When comparing the exponential distribution with the hyper-exponential distribution, we analyze the outcome of a two-sided Welch test (see Table 5). The p-values presented are sufficiently low to conclude that the two distributions produce different average waiting time outcomes. However, a two-sided test alone is insufficient to conclude which distribution is superior regarding average waiting times.

5 Discussion

The findings show that the Welch test did not identify a significant difference between FIFO and SJF, contrary to our anticipated outcome. Various factors could contribute to this result, but it is likely influenced by our decision to assess the scheduling policies with a customer count of 50.000. This choice necessitated a reduction in the number of simulations to 30. We hypothesize that obtaining more significant results may involve decreasing the customer count and concurrently increasing the number of simulations. This is especially necessary when the ρ gets closer to 1. Therefore, we recommend this for future work.

This study's chosen service time μ was established at 0.5. This decision was motivated by the selected hyper-exponential distribution parameters, wherein 75% of the requests adhered to an exponential distribution with an average service time of 1.0. At the same time, the remaining 25% followed an exponential distribution with an average service time of 5.0. The weighted average of these parameters resulted in the service above time of 0.5. Nevertheless, altering these parameters may introduce variations in experimental outcomes, warranting further exploration in future research.

Further recommendations for future research include a more thorough investigation of parameter settings. In the present study, values of ρ were set to 0.8, 0.95, and 0.99, with a preference for proximity to 1 due to the expected more interesting results. There is potential for further exploration by increasing the value of ρ to see even more interesting outcomes.

6 Conclusion

The average waiting time in the M/M/1 system appears to be higher compared to M/M/n systems, particularly as the ρ approaches 1. The waiting times in M/M/1 systems exhibit significant fluctuations in standard deviation, suggesting a potential for variability in system performance, again even more so when ρ is closer to 1. Conversely, M/M/n systems tend to have lower standard deviations, implying a more consistent and predictable system performance.

Performing statistical Welch tests on various FIFO and SJF scheduling comparisons indicated no significant difference. However, when observing the results, it is likely that under different circumstances, there might have been a significant difference. The reason for this has been highlighted in the Discussion (5). Furthermore, the SJF scheduling policy consistently demonstrates advantages over FIFO in various aspects. SJF's prioritization of shorter jobs contributes to a reduction in average waiting times. The lower standard deviation associated with SJF implies higher consistency and predictability in waiting times compared to FIFO. Moreover, the narrower confidence intervals observed for SJF indicate a heightened confidence level in the estimated average waiting time, underlining its reliability.

Distinct service rate distributions impact average waiting times, particularly in single-server systems. The influence persists in multi-server systems, with notable exceptions found in the comparisons between M/M/2 and M/D/2, as well as between M/M/4 and M/D/4. So, the deterministic service rate distribution outperformed its stochastic counterparts in a single-server system, yet no statistically significant results were found for multiple-server systems. Hyper-exponential and exponential distributions yield statistically different results, but which of the two distributions performs better remains undetermined and requires further investigation.

Appendix

Author	Time (hours)	Surviving code lines	Commit count	Files	Distribution
Priyank Venkatesh	2	333395	4	2	99.0/30.8/18.2
Guoda Paulauskaitė	2	2080	4	6	0.6/30.8/54.5
Koen Weverink	3	1388	5	3	0.4/38.5/27.3

Table 6: Work distribution

The work distribution on this report is shown in table 6. For the methodology, please refer to our GitHub repository `https://github.com/koenweverink/stochastic_simulation_assignment_2.git`.

References

[1] Ryan Berry. Queuing theory - whitman college. *Whitman*.

[2] K J E L L S T O R D A H L. *The history behind the probability theory and the queuing theory*, 2007.

[3] Sunday A. Afolalu, Kunle O. Babaremu, Samson O. Ongbali, Abiodun A. Abioye, Ademola Abdulkareem, and Samuel B. Adejuyigbe. Overview impact of application of queuing theory model on productivity performance in a banking sector. *Journal of Physics: Conference Series*, 1378(3):032033, dec 2019.

[4] Andreas Willig. A short introduction to queueing theory. Technical report, Technical University Berlin, Telecommunication Networks Group, Sekr. FT 5-2, Einsteinufer 25, 10587 Berlin, July 1999.

[5] D. G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 24(3):338–354, 1953.

[6] Gabor Zavodszky. Discrete event simulations queuing theory. Lecture, 2023. Course "Stochastic Simulations", University of Amsterdam.

[7] Ivo Adan and Jacques Resing. Queueing theory. Technical report, Eindhoven University of Technology, Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, February 2002.

[8] Python Software Foundation. Python 3.9.0.

[9] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[11] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[12] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[14] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy: Open source scientific tools for python. *Nature Methods*, 17:261–272, 2020. Accessed on 18-11-2023.

[15] Andreas Willig. *A Short Introduction to Queueing Theory*. Technical University Berlin, Telecommunication Networks Group, Sekr. FT 5-2, Einsteinufer 25, 10587 Berlin, July 21 1999.