Discrete Event Simulation

Stochastic Simulation – Assignment 2

Daan van Ingen (10345078)
Walter Vianen (11811293)

1 Introduction

Discrete event simulation (DES) models the operations of a system as a sequence of discrete events. Each event happens at a certain point in time and changes the system state. Between two events it is assumed that the system state does not change. In this report, a queuing system where customers arrive into the system and wait in a single line before they are served is investigated.

In general this behaviour is seen as customers arriving at some service center in random fashion. The service center can have one or more servers capable of serving one customer at the time. The time that is needed for serving a customer is usually modelled as a random variable. Furthermore, there are lots of details that can be added like the size of the waiting line (finite or infinite) and in which order incoming customers are handled.

In particular, two systems are compared here. The first is a system only one server is available to serve all customers. In the second system n servers are employed and the arrival rate is also n times larger than for the single-server system.

The outline of the report is as follows. In section 2 the notation conventions in queuing theory are introduced and some theoretical results are derived for the systems that are investigated. Next, in section 3 the methods used to design and evaluate simulations to test the theoretical stipulations are explained. Results of these simulations are provided in section 4 along with discussion. Finally, the work is summarised and conclusions are drawn from all previous work in section 5.

2 Theory

2.1 Notation

To indicate what kind of queuing system is considered, the extended Kendall notation [1] is employed, which is as follows:

$$A/B/m/N - S$$

Where

- **A** denotes the distribution of the inter-arrival times:
  - M: Markov
  - D: Deterministic
  - E_k: Erlang-k
  - H_k: Hyper-k
  - G: General
- **B** denotes the distribution of the service times.
- **m** denotes the number of servers.
- **N** denotes the maximum size of the waiting line. If N is not specified the queue is assumed to be infinite.
- **S** denotes the used service discipline (default is FIFO):
  - FIFO: First in, First out
  - LIFO: Last in, First out
  - Random Service: customers are served in random order
  - Round Robin: each customer gets a time slice. If the service is not completed the customer re-enters the queue.
  - Priority: order of service depends on some priority parameter

The most simple queuing system can then be described as M/M/1. This is a single infinite waiting line where the arrival times are i.i.d. and exponentially distributed with some parameter λ. The customers are then served by a single server according to the FIFO principle where the customer service times are again i.i.d. and exponentially distributed with parameter μ. This is the first system that is considered in this report. The second system is a M/M/c system, where we have c servers, and the arrival rate is c times higher than in the first system.

2.2 System Analysis

The M/M/1 queue has i.i.d. inter-arrival and service times which are exponentially distributed with parameters λ and μ respectively. From this it is trivial to find the underlying Markov chain, because at each state only two state changes can happen: either a new customer arrives in the system or a a customer leaves the system after being served. This is also shown in figure 1.

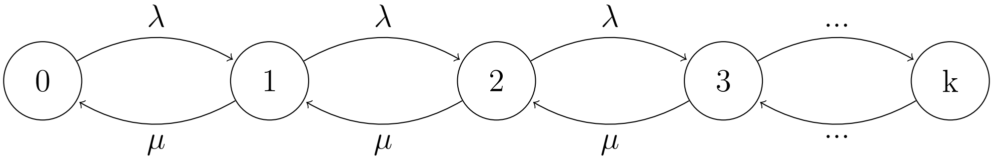

Fig. 1: Illustration of the Markov chain M/M/1 queue.

An effective mathematical description of queuing systems was introduced by John Little [2]. He derived a relation between the mean number of customers in the queue $E(L_q)$, the rate of arrival of customers λ, and the mean waiting time $E(W)$:

$$E(L_q) = \lambda E(W). \tag{1}$$

Little's work has been greatly extended and other meaningful relations exist. Derivations for the following statements are found in [3]. Firstly, define the system load ρ as:

$$\rho = \frac{\lambda}{c\mu}. \tag{2}$$

An important quantity is the probability that a customer has to wait. This is also known as the delay probability Π_W. It can be shown that for a M/M/c system:

$$\Pi_W = \frac{(c\rho)^c}{c!}\left((1-\rho)\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!}\right)^{-1}. \tag{3}$$

Then the mean queue length:

$$E(L_q) = \Pi_W \cdot \frac{\rho}{1-\rho} \tag{4}$$

and the mean waiting time:

$$E(W) = \Pi_W \cdot \frac{1}{1-\rho} \cdot \frac{1}{c\mu}. \tag{5}$$

The delay probability Π_W decreases as the number of servers c increases for constant ρ. Knowing this, and noting a factor c in the denominator in equation 5, the mean waiting time $E(W)$ decreases as the number of servers c increases. This important result will be the subject of analysis in the simulations.

3 Methods

3.1 Implementation

To simulate the queuing system and service station we implemented a python program using SymPy [4]. SymPy creates an environment where a resource (the service station) can be used by some source (the customer). Each source can hold a resource for a specific time and thereby simulate some queue of processes waiting to be served. The outline of this code can be found below.

Unless stated otherwise we will be looking at M/M/n queues where we are particularly interested in the difference between a M/M/1 queue with arrival times drawn from an exponential distribution with mean λ and a M/M/n queue with mean λ/n. Theory tells us that the mean arrival times for the M/M/n queue with an n-fold lower inter-arrival time should be lower than for an M/M/1 queue with a n-times longer time between customers.

Next to the FIFO implementation, we will also take a look at shortest job first scheduling, where the order of jobs executed in the queue depends on their job duration. This theoretically should have some advantages as it focuses on minimising the average waiting time as there is a high throughput for small jobs. However it can be that jobs that take a long time have to wait a long time before being served, which is referred to as process starvation.

Lastly we will experiment with different customer service times in the form of M/D/1 or M/D/n queues and a more fat-tailed distribution for the service times. It will be interesting to see how the dynamics change and if the results of the previous experiments still apply.

Pseudocode 1 Pseudocode for the SymPy queuing system.

```
function SOURCE(amount of customers):
    for amount of customers do
        create customer and add to queue
        generate inter-arrival time
        wait this time before creating the next customer

function CUSTOMER:
    arrival = now
    wait for an available resource
    waiting time = now - arrival
    generate service time
    hold resource for this time
    release resource

environment = simpy environment
resource = simpy resource(environment, capacity)
source = simpy source(Source)
run environment
```

3.2 Parameter settings

The main dynamics of a queue are determined by the system load ρ. To make sure the system queue does not keep growing ρ has to be smaller than one. But to keep things interesting ρ will not be much smaller than one in the experiments. Unless mentioned otherwise we assume that for a M/M/1 queue a customer arrives every 10 seconds and is served in 9 seconds on average, giving $\rho = 0.9$. Inter-arrival times for M/M/n queues are changed where applicable such that $\lambda_n = \lambda_1/n = 10/n$. Keeping μ constant, the system load ρ is now equal for all systems.

In order to discuss other simulation parameters and result analysis consider figure 2 which shows the average waiting time as a function of customer number (customer one is the first to arrive to an empty server, then customer two follows, and so on). In this case, the mean waiting time is the average waiting time of all customers with the same number. Results have been gathered from 1000 runs.

We find that the mean waiting time approaches the theoretical mean waiting times provided by the relations in section 2.2. These values are 81, 38 and 18 seconds for the systems with 1, 2 and 4 servers respectively.

Clearly depicted in this figure is the start-up time of the system; the system starts with an empty queue and needs some time to reach its steady-state. Including the start-up period in analysis of the results will greatly influence the derived mean waiting times [5]. To avoid this we consider the system to be in steady-state after the first 500 customers and, hence, remove the first 500 when computing system characteristics.

Furthermore, where appropriate, 95% confidence intervals are given according to:

$$\left[\bar{x} - 1.96\frac{s}{\sqrt{N}}, \bar{x} + 1.96\frac{s}{\sqrt{N}}\right] \tag{6}$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation and N is the number of independent runs.

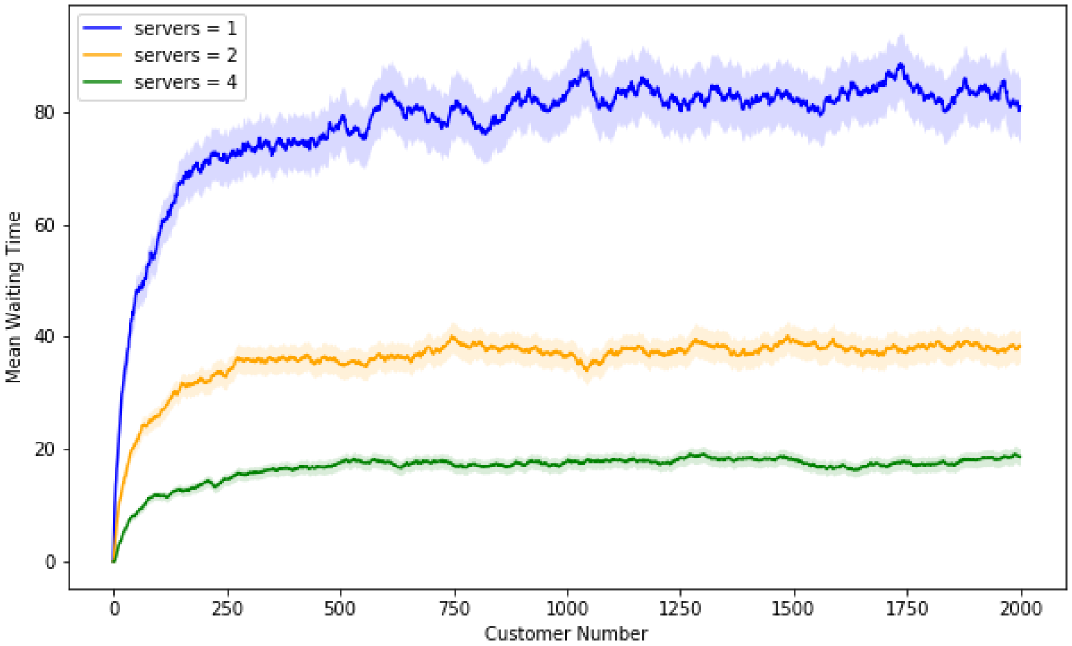

Fig. 2: Mean waiting time as function of customer number. The shaded area is the 95% confidence interval. The system load $\rho = 0.9$.

4 Results & Discussion

4.1 M/M/n Queue

Firstly, we consider the mean waiting time as a function of the system load ρ depicted in figure 3. As the system load increases, the mean waiting time also increases; the servers need to serve more people on average. From this figure it can also be concluded that the variance in mean waiting time increases as a function of the system load ρ. Hence, as ρ increases the amount of measurements needed to achieve sufficient statistical significance also increases.

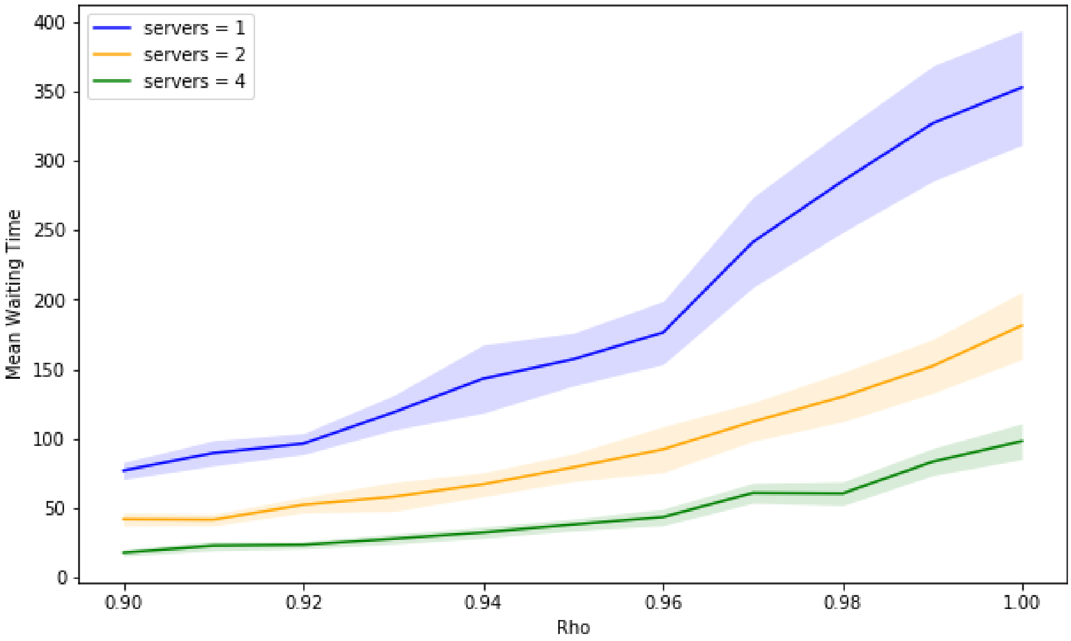

Fig. 3: Mean waiting time as a function of the system load ρ. The total number of served customers is 2000 and the number of runs is 100. The maximum value is $\rho = 0.999$.

Moreover, we see a clear distinction between the waiting times for different number of servers. When the number of servers increases, the mean waiting time decreases. This intuitively makes sense, because if in a one server system the service center is occupied by a large job this effects everyone in the queue. But if the service station is large then we expect that the time we have to wait is close to the difference between the mean arrival time and mean service time (so in this case we expect to be served almost immediately). This corresponds with the law of large numbers: as the number of service stations increases, the chance of a customer arriving at the front of the queue and only seeing occupied service stations (so all customers having an above average service time) should go to zero. This is also displayed in figure 4, where the capacity of the server is outlined against the mean waiting time. We can see a decline towards zero as the capacity of the server increases.

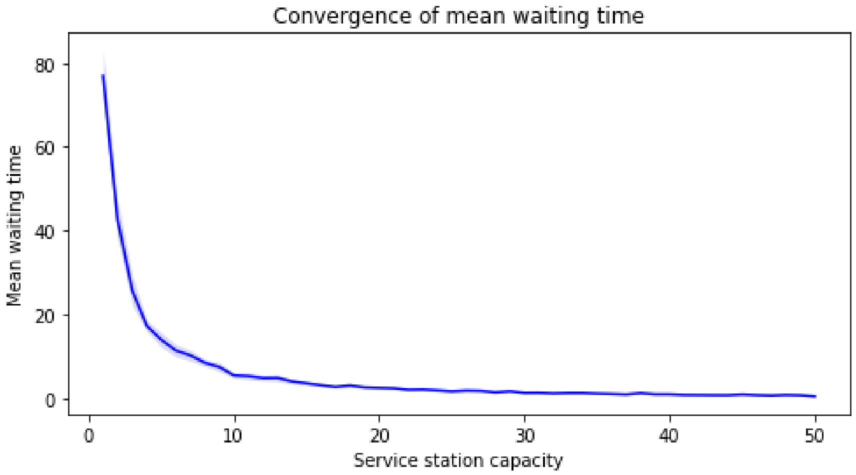

Fig. 4: Mean waiting time as a function of the service station capacity. Parameter settings are: $\rho = 0.9$, $N = 100$ and number of customers is 2000.

A final insightful result for this system is the distributions for the mean waiting times over the runs, depicted in figure 5. The distributions are approximating the normal distribution, as is expected by the central limit theorem. The distributions still have a fatter tail on the right, characteristic of exponential distributions. Increasing the number of customers will decrease this effect. This figure also confirms that the variance in the mean waiting time decreases as the number of servers decreases, as is indicated by the narrower distributions for higher number of servers.

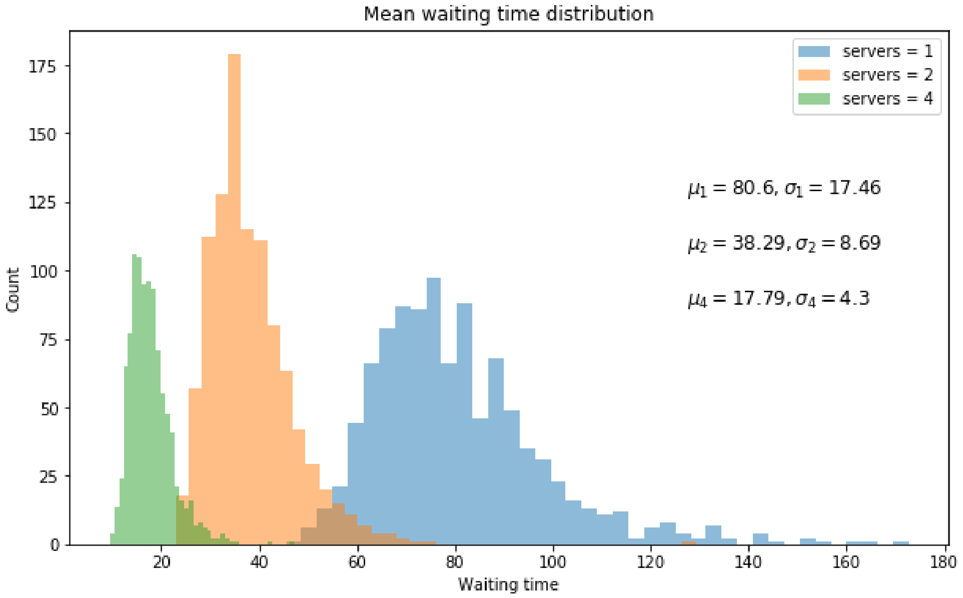

Fig. 5: Distribution of mean waiting times over 1000 runs. The number of customers is 10000 and $\rho = 0.9$. Distribution characteristics are also given in the plot.

4.2 Shortest job first scheduling

Next, we will take a look at the results of shortest job first scheduling and compare them to the results of FIFO scheduling. The results of this are displayed in figure 6. As we can see the mean waiting times for shortest job first scheduling are on average lower than for the FIFO queue, compared to figure 5. Additionally, the distributions are closer together and narrower than for FIFO scheduling.

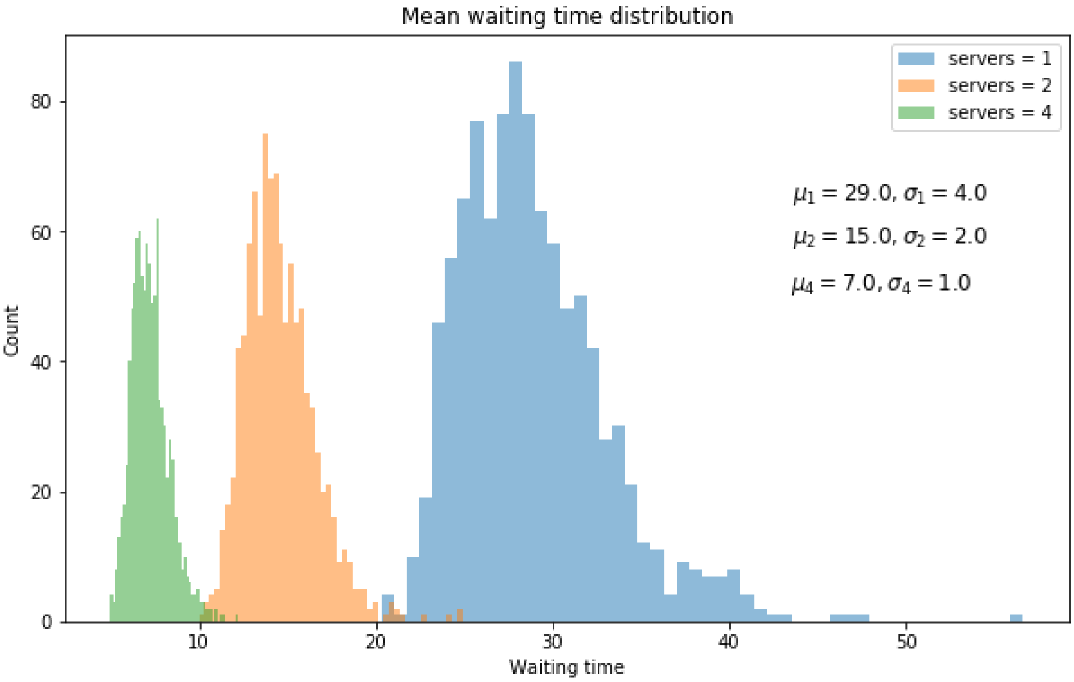

Fig. 6: Mean waiting times for M/M/n-P systems. The number of customers is 10000, $\rho = 0.9$ and $N = 1000$.

We know that short job first scheduling can suffer from job starvation. To examine this we took a look at the maximum waiting time of a customer in a FIFO or shortest job first queue. The results of different service station sizes are taken together to be able to compare the different queues as a whole. The results are displayed in figure 7.

We can see that for shortest job first some customers have to wait an extremely long time compared to the FIFO queue before they are served. This is an excellent example of starvation. For some systems these long waiting times can be disastrous, for other systems the mean waiting times are more important and some customers having to wait a long time is acceptable.

A final noteworthy remark about the distribution for the FIFO method is that the influence of the different underlying distributions for different number of servers is visible in the distribution for the maximum waiting time; the distribution seems to be a combination of three others, because three peaks are distinguishable. This feature is not visible when the shortest job first method is employed. This method mitigates that effect.

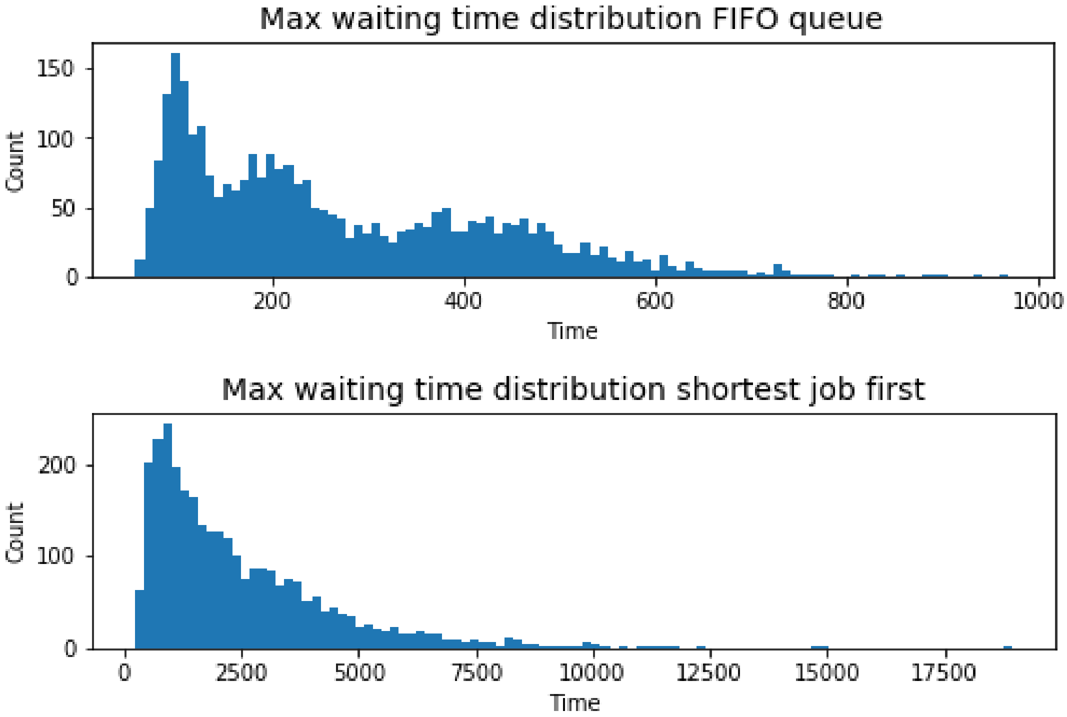

Fig. 7: Maximum waiting time for a customer in a FIFO and shortest job first queue. Data from systems with different amount of servers are combined per scheduling method. Thus 30000 customers for the FIFO and shortest job method, $\rho = 0.9$ and $N = 3000$ per scheduling method.

4.3 M/D/n Queue

In an M/D/n queue the service times are deterministic and constant. Here we take a constant serving time of 9 seconds. The mean waiting time distributions for this system are given in figure 8. Comparing these results to figure 5 with the M/M/n system, we find that the M/D/n mean waiting times are approximately half of that of the M/M/n system. This is a known result, for which a derivation is given in [6]. The other characteristics of these results are similar to that of the M/M/n queue.

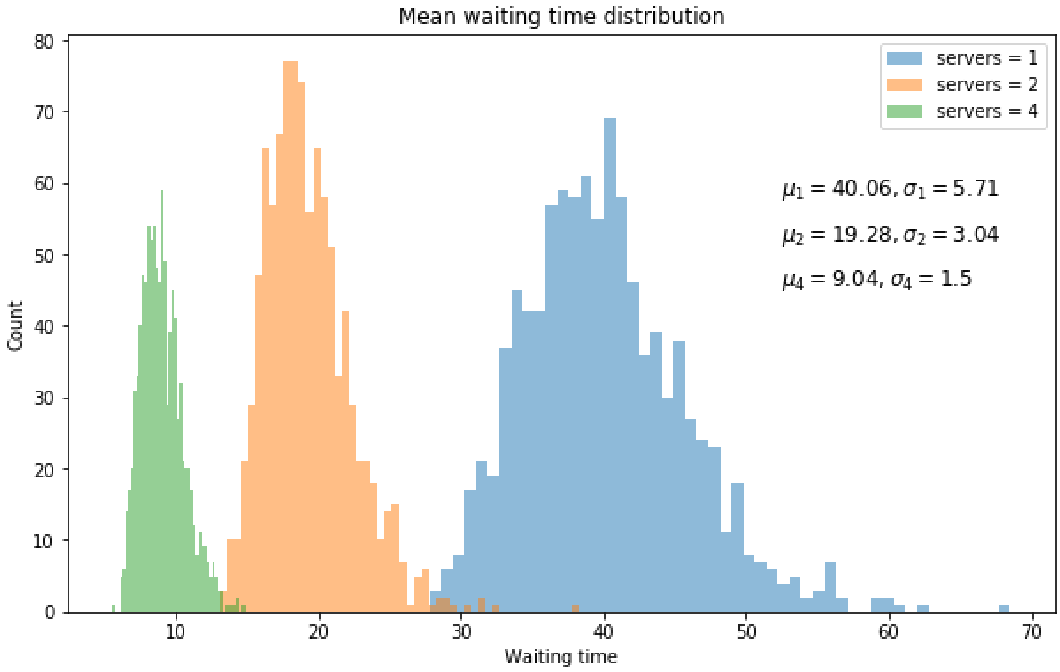

Fig. 8: Distribution of mean waiting times for the M/D/n model. The number of customers is 10000, $\rho = 0.9$ and $N = 1000$.

4.4 Long-Tail Distribution

The last system that is investigated at present is a system with exponentially distributed inter-arrival times, a long-tailed service time distribution and n servers. The long-tailed service time distribution is a hyperexponential distribution where 75% of the customers have an exponential distribution with an average service time of 1.0, and the remaining 25% an average distribution with an average service time of 5.0. The inter-arrival time distribution is characterised by $\lambda = 5/n$. To effectively analyse this system also a M/M/n simulation is done where the service time is modelled by an exponential distribution with mean 1.0 for all customers. All corresponding results are given in figure 9.

From the figures it is clear that with these parameter settings the relative difference between the distributions for different number of servers is larger. Moreover, when using a long-tailed distribution, the overall average waiting times are longer. Due to a small number of customers with a relatively very large service time, all customers in the queue have to wait longer on average. Using a long-tailed distribution magnifies this effect of customers with short waiting times having to wait for customers with very long waiting times.

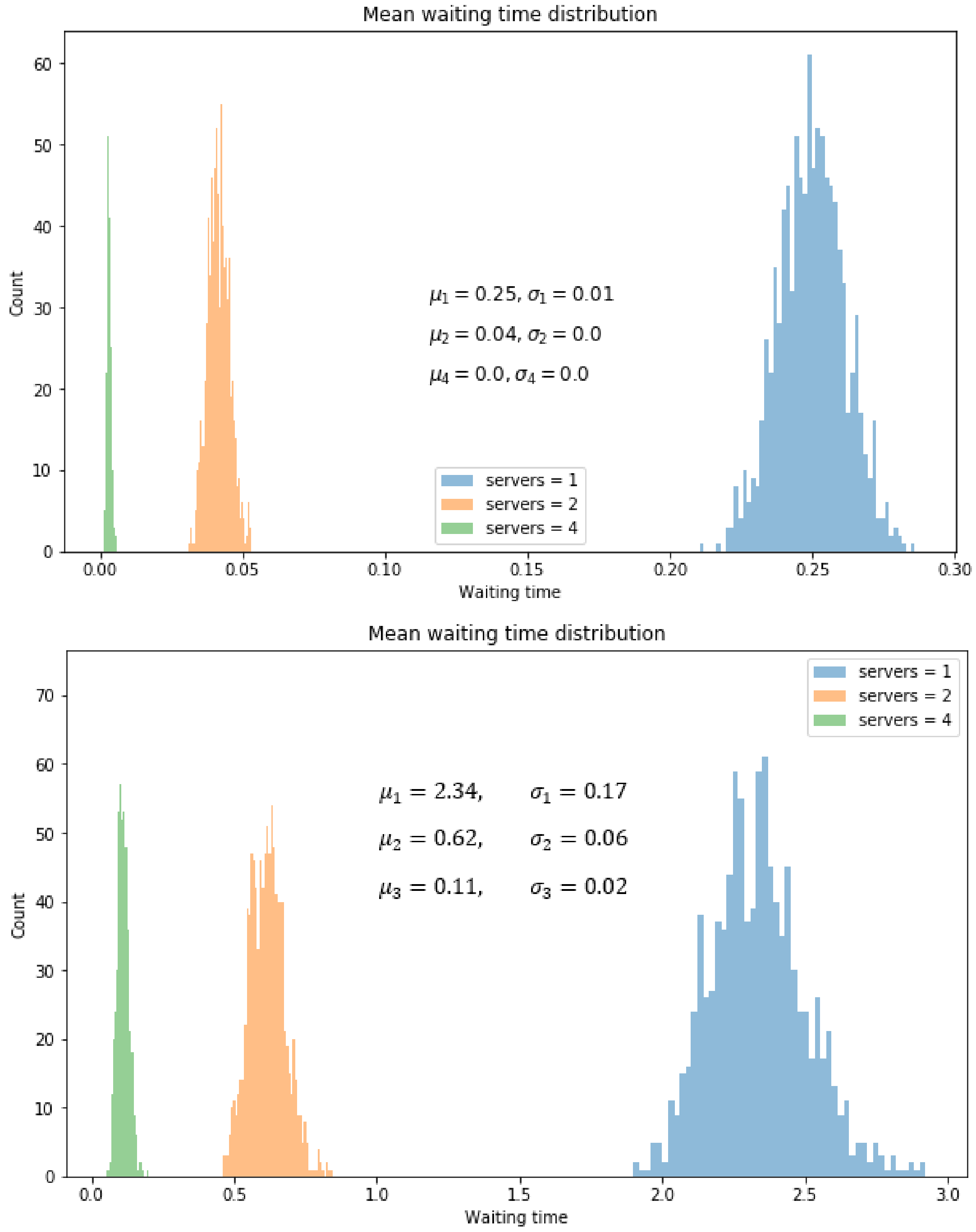

Fig. 9: A base distribution (top) M/M/n, with $\lambda = 5/n$ and $\mu = 1$, and a long-tailed distribution (bottom) where 75% of the customers have an exponential distribution with an average service time of 1.0, and the remaining 25% an average distribution with an average service time of 5.0. Furthermore, the number of runs $N = 1000$ and the number of customers is 10000.

5 Conclusion

The general claim that systems with more servers in a queuing model have lower average waiting times has been investigated for several systems. For all presented systems this statement has been successfully verified through discrete event simulations done with SimPy.

Moreover, we have shown the influence of the start-up phase of the simulation and by mitigating this effect we were quite able to match simulation results with predictions from the theoretical mean waiting time (equation 5).

When comparing the FIFO scheduling method with a shortest job first scheduling method, it was found that the second method decreases the mean waiting time for the customers. However, due to some customers with high service time having to wait longer, the phenomenon of starvation was also observed.

When using constant deterministic service times the mean waiting time for customers is approximately halved when compared to exponentially distributed service times. This is a known result from theory, and was verified through simulation.

Lastly, a long-tailed service time distribution vastly increases mean waiting time for customers due to some customers with high service time holding up the queue and thereby causing a delay.

These systems are only a small selection from all possible queuing systems. However, discrete event simulation has proven to be a useful tool in analysing these systems by verifying results from theory. This provides confidence that this method is also useful when investigating systems where theoretical relations are not readily available or cannot be derived.

References

[1] 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*, pages 338–354, 1953.

[2] J.D. Little. A proof of the queueing formula: $l = \lambda w$. *Operations Research*, 9(3):383–387, 1961.

[3] I. Adan and J. Resing. Queueing theory. *Department of Mathematics and Computing Science Eindhoven University of Technology*, pages 43–44, 2001.

[4] N. Matloff. Introduction to discrete-event simulatino and the simpy language. 2008.

[5] K. Pawlikowski. Steady state simulation of queueing processes: A survey of problems and solutions. *Department of Computer Science University of Canterbury Christchurch*, 1988.

[6] T.L. Saaty. Elements of queueing theory: with applications. *McGraw-Hill New York*, 34203:153–170, 1961.