CM2605 - CW

Vinuji Hewapathirana

2025-03-12

Introduction

This model has been implemented using an **M/M/c** queuing system where:
- M: Markovian (Poisson) arrival process
- M: Markovian (Exponential) service time distribution
- c: Number of servers

1. Model

```r
# Function to calculate inter-arrival times (Poisson arrivals)
calculate_arrivals <- function(rate, total_time) {
  # Convert rate from customers per hour to customers per minute
  rate_minute <- rate / 60

  # Calculate inter-arrival times using exponential distribution
  inter_arrival_times <- rexp(1000, rate_minute)

  # Convert to actual arrival times
  arrival_times <- cumsum(inter_arrival_times)

  # Keep only arrivals within the simulation time
  arrival_times <- arrival_times[arrival_times <= total_time]

  return(arrival_times)
}

# Function to calculate service times (Exponential distribution)
calculate_service_times <- function(mean_service_time, customers) {
  # Caoculate service times
  service_times <- rexp(customers, 1/mean_service_time)

  return(service_times)
}

# Function to simulate the queuing system
simulate_queue <- function(arrival_times, service_times, servers) {
  customers <- length(arrival_times)

  # Initialize tracking variables
```

```r
  waiting_times <- numeric(customers)
  service_start_times <- numeric(customers)
  service_end_times <- numeric(customers)

  # Track availability time of each server
  server_available_times <- rep(0, servers)

  # Track which server serves which customer
  server_assignment <- numeric(customers)

  # For each customer
  for (i in 1:customers) {
    # Earliest available server
    earliest_server <- which.min(server_available_times)

    # Record which server is serving this customer
    server_assignment[i] <- earliest_server

    # Calculate when service can start
    service_start_times[i] <- max(arrival_times[i],
server_available_times[earliest_server])

    # Calculate waiting time
    waiting_times[i] <- service_start_times[i] - arrival_times[i]

    # Calculate service end time
    service_end_times[i] <- service_start_times[i] + service_times[i]

    # Update server availability
    server_available_times[earliest_server] <- service_end_times[i]
  }

  # Calculate system exit times
  exit_times <- service_end_times

  # Calculate queue lengths at different points in time
  time_points <- sort(unique(c(arrival_times, service_start_times)))
  queue_lengths <- numeric(length(time_points))

  for (t in 1:length(time_points)) {
    current_time <- time_points[t]
    in_queue <- sum(arrival_times <= current_time & service_start_times >
current_time)
    queue_lengths[t] <- in_queue
  }

  # Calculate server utilization
  total_busy_time <- numeric(servers)
  for (s in 1:servers) {
```

```
    # Find all customers served by this server
    server_indices <- which(server_assignment == s)
    if (length(server_indices) > 0) {
      total_busy_time[s] <- sum(service_end_times[server_indices] -
service_start_times[server_indices])
    } else {
      total_busy_time[s] <- 0
    }
  }

  # Calculate server utilization using max(exit_times) as total simulation
time
  total_simulation_time <- max(exit_times)
  server_utilization <- sum(total_busy_time) / (servers *
total_simulation_time)

  # Results
  results <- list(
    waiting_times = waiting_times,
    queue_lengths = queue_lengths,
    time_points = time_points,
    server_utilization = server_utilization,
    average_waiting_time = mean(waiting_times),
    average_queue_length = mean(queue_lengths),
    max_queue_length = max(queue_lengths),
    total_customers = customers
  )

  return(results)
}

# Run the simulation
run_simulation <- function(arrival_rate, service_time, simulation_time,
servers) {
  # Calculate arrivals
  arrival_times <- calculate_arrivals(arrival_rate, simulation_time)

  # Calculate service times
  service_times <- calculate_service_times(service_time,
length(arrival_times))

  # Run simulation
  results <- simulate_queue(arrival_times, service_times, servers)

  return(results)
}
```

Run the model

```r
# Set parameters
arrival_rate <- 10  # customers per hour
service_time <- 5   # minutes per customer
simulation_time <- 480  # minutes (8-hour workday)

# Run simulation with 2 servers
results_2_servers <- run_simulation(arrival_rate, service_time,
simulation_time, 2)

# Run simulation with 3 servers
results_3_servers <- run_simulation(arrival_rate, service_time,
simulation_time, 3)
```

2. Results and evaluation

System with 2 servers

```r
# Key metrics for system with 2 servers
cat("Performance with 2 servers:\n")

## Performance with 2 servers:

cat("Average waiting time:", round(results_2_servers$average_waiting_time,
2), "minutes\n")

## Average waiting time: 0.81 minutes

cat("Average queue length:", round(results_2_servers$average_queue_length,
2), "customers\n")

## Average queue length: 0.39 customers

cat("Maximum queue length:", results_2_servers$max_queue_length,
"customers\n")

## Maximum queue length: 3 customers

cat("Server utilization:", round(results_2_servers$server_utilization * 100,
2), "%\n")

## Server utilization: 36.61 %

cat("Total customers served:", results_2_servers$total_customers, "\n")

## Total customers served: 76
```

System with 3 servers (Additional server)

```r
# Key metrics for system with 3 servers
cat("Performance with 3 servers:\n")
```

```
## Performance with 3 servers:

cat("Average waiting time:", round(results_3_servers$average_waiting_time,
2), "minutes\n")

## Average waiting time: 0.08 minutes

cat("Average queue length:", round(results_3_servers$average_queue_length,
2), "customers\n")

## Average queue length: 0.12 customers

cat("Maximum queue length:", results_3_servers$max_queue_length,
"customers\n")

## Maximum queue length: 2 customers

cat("Server utilization:", round(results_3_servers$server_utilization * 100,
2), "%\n")

## Server utilization: 27.81 %

cat("Total customers served:", results_3_servers$total_customers, "\n")

## Total customers served: 84
```

For this model, following formulas have been used:

The arrival rate λ:

$$\lambda = 10 \text{ customers per hour}$$

The service rate μ:

$$\mu = \frac{1}{\text{mean service time}} = \frac{1}{5 \text{ minutes}} = 12 \text{ customers per hour}$$

The traffic intensity ρ (we need to find this as we are using multiple servers, where c is the number of servers):

$$\rho = \frac{\lambda}{c\mu}$$

The average waiting time in the queue W_q:

$$W_q = \frac{\rho}{\mu(1-\rho)}$$

The average number of customers in the queue L_q:

$$L_q = \frac{\rho^2}{1-\rho}$$

The server utilization U (although the theoretical traffic intensity is also similar to this, here actual observed utilization is measured by tracking server busy time):

$$U = \frac{\lambda}{c\mu}$$

Given below are the final key metrics received after running the model.

With 2 servers:
- Average queue length – 0.39 customers
- Average waiting time – 0.81 minutes
- Server utilization – 36.61%

With 3 servers:
- Average queue length – 0.12 customers
- Average waiting time – 0.08 minutes
- Server utilization – 27.81%

According to these results, with two servers the model is running quite well. The customers have to wait an average time of 0.81 minutes (approximately 46 seconds) to get served and the queue length is 0.39 customers. At the peak times, a maximum of 3 customers can be observed in the queue. The server utilization is 36.61% which is quite low, showing that the current resources may be underutilized. Furthermore, the total number of customers that can be served within an 8 hour work day is 76.
With the given specifications, adding another server should be considered if the average waiting time gets above 15 minutes. As the waiting time is around 46 seconds in the model with 2 servers, the current model is running really well and the customers do not need to wait for a long time to get served and the system is handling the customer flow effectively.

However, according to the results we got after running the model with an additional server (total of 3 servers), the average waiting time is 0.08 minutes which has reduced the waiting time for 2 servers by over 90%. The average queue length is 0.12, number of customers that can be observed in the peak time is 2 and total number of customers that can be served within the day is 84. Therefore, average waiting time, queue length, number of maximum customers handled at the peak time and the total number of customers served is better than the model with 2 servers. The server utilization has reduced to 27.81% which suggests potential resource efficiency.
Although customer experience gets increased with three servers, given that the waiting times are already minimal with two servers, this improvement will not be greatly effective. Furthermore, because of the low server utilization, there is a possibility of overstaffing with 3 servers and this would increase the operational cost without much effective benefits.

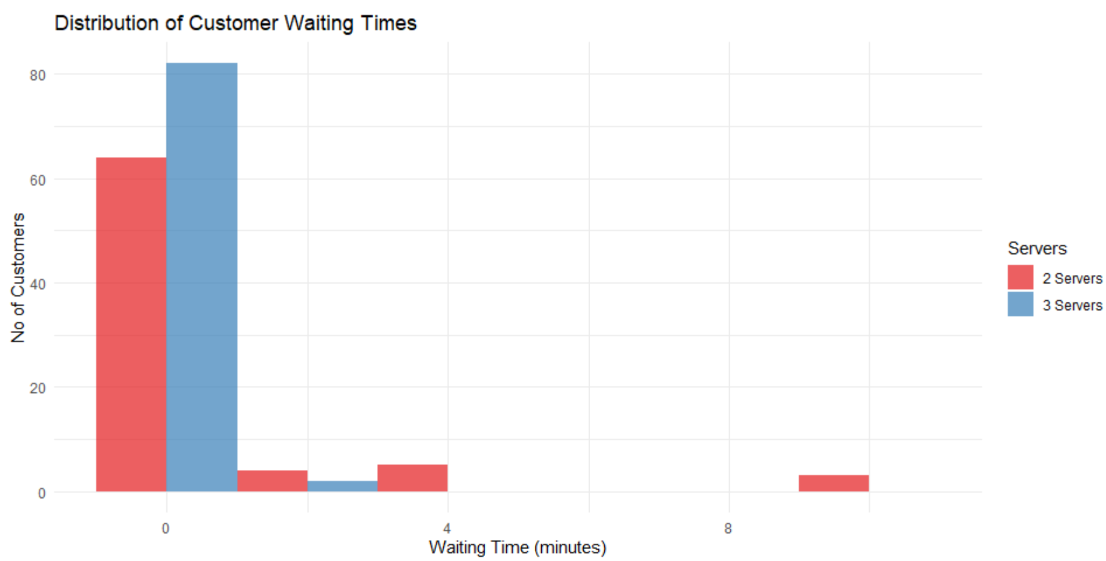
3. Visualization & Report

Visualization

```
# Dataframe for waiting time distribution
waiting_times_df <- data.frame(
```

```
  Waiting_Time = c(results_2_servers$waiting_times,
results_3_servers$waiting_times),
  Servers = factor(c(rep("2 Servers",
length(results_2_servers$waiting_times)),
                   rep("3 Servers",
length(results_3_servers$waiting_times))))
)

# Histogram of waiting times
ggplot(waiting_times_df, aes(x = Waiting_Time, fill = Servers)) +
  geom_histogram(position = "dodge", binwidth = 2, alpha = 0.7) +
  labs(title = "Distribution of Customer Waiting Times",
       x = "Waiting Time (minutes)",
       y = "No of Customers") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")
```



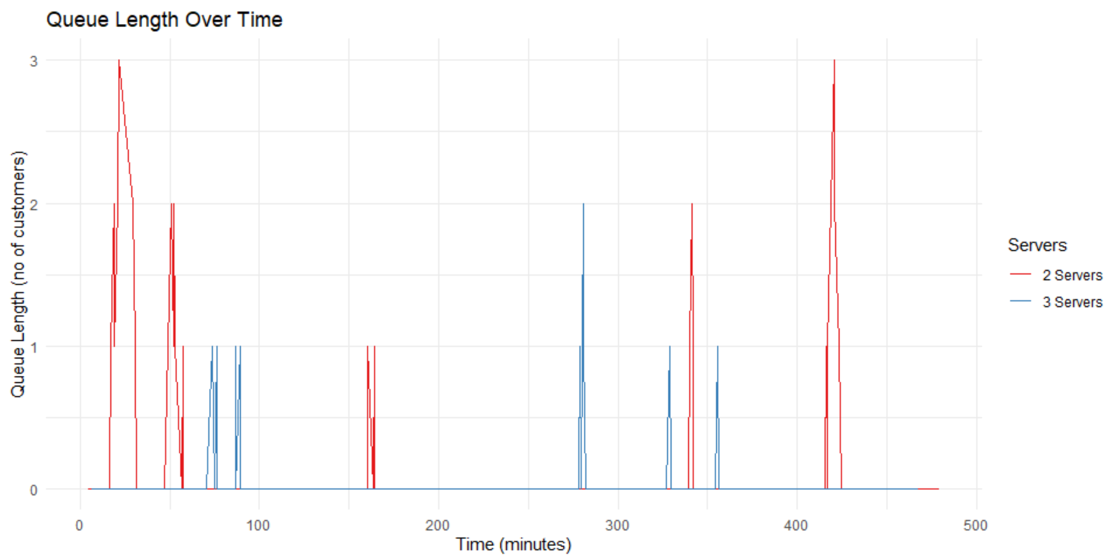

```
# Dataframe for queue lengths
queue_lengths_2 <- data.frame(
  Time = results_2_servers$time_points,
  Queue_Length = results_2_servers$queue_lengths,
  Servers = "2 Servers"
)

queue_lengths_3 <- data.frame(
  Time = results_3_servers$time_points,
  Queue_Length = results_3_servers$queue_lengths,
  Servers = "3 Servers"
)

queue_lengths_df <- rbind(queue_lengths_2, queue_lengths_3)

# Line plot of queue lengths
```

```r
ggplot(queue_lengths_df, aes(x = Time, y = Queue_Length, color = Servers)) +
  geom_line() +
  labs(title = "Queue Length Over Time",
       x = "Time (minutes)",
       y = "Queue Length (no of customers)") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")
```



Performance Comparison Table

```r
# Dataframe for comparison
comparison_df <- data.frame(
  Metric = c("Average Waiting Time (min)", "Average Queue Length", "Server
Utilization (%)"),
  Two_Servers = c(round(results_2_servers$average_waiting_time, 2),
                  round(results_2_servers$average_queue_length, 2),
                  round(results_2_servers$server_utilization * 100, 2)),
  Three_Servers = c(round(results_3_servers$average_waiting_time, 2),
                    round(results_3_servers$average_queue_length, 2),
                    round(results_3_servers$server_utilization * 100, 2)),
  Improvement = c(round((results_2_servers$average_waiting_time -
results_3_servers$average_waiting_time) /
                      results_2_servers$average_waiting_time * 100, 2),
                round((results_2_servers$average_queue_length -
results_3_servers$average_queue_length) /
                      results_2_servers$average_queue_length * 100, 2),
                round((results_2_servers$server_utilization -
results_3_servers$server_utilization) /
                      results_2_servers$server_utilization * 100, 2))
)

knitr::kable(comparison_df, caption = "Performance Comparison Between 2 and 3
Servers")
```

Performance Comparison Between 2 and 3 Servers

Metric	Two_Servers	Three_Servers	Improvement
Average Waiting Time (min)	0.81	0.08	90.63
Average Queue Length	0.39	0.12	69.77
Server Utilization (%)	36.61	27.81	24.03

For visualization, following two charts have been created through the model to indicate the final results.

1. Distribution of Customer Waiting Times
- This histogram displays the frequency distribution of waiting time of the systems with both 2 servers and 3 servers. Two-server system is shown with red bars and three-server system with blue bars.
- According to this, in the two-server system, the majority of customers (around 65) go through minimal waiting time of nearly 0 minutes. A small group of customers wait around 4 minutes and even smaller group waits for 10 minutes.
- In the three-server system, around 82 customers get a waiting time of 0 minutes and negligible number of customers wait longer than 2 minutes.
- This demonstration shows that with 3 servers customers can experience a better service.

2. Queue Length Over Time
- The queue length of the two systems throughout the 8 hour working day is demonstrated by the above line chart.
- According to this chart, the two-server system has 0 waiting customers for the majority of the time period and 3 customers at the peak times (30, 420 minutes). This shows an efficient service.
- The three-server system also has 0 customers waiting for the majority of the time and a maximum of 2 customers at peak times, which is a little better than the two-server system.
- According to these two visualizations, both systems provide excellent service with the three-service system providing marginal improvements resulting in much shorter waiting times and queue lengths.

Final Report

This model has used the parameters customer arrivals, service times and 8 hour simulation time period and implemented through custom functions using to output the arrival times and service times. It has tracked the metrics waiting times, queue lengths and server utilization.

According to the outputs gained through the model ran for the two systems, the first system with two servers perform well with waiting time which is less than the maximum of 15 minutes. However, the server utilization of 36.61% shows the system is operating well below capacity.
On the other hand, the system with 3 servers has reduced the waiting time by 90.63% and decreased both queue level and server utilization. The low level of server utilization shows

a issue related to resource efficiency and there is a possibility of overstaffing which will result in unnecessary expenses.

Conclusion and Recommendations

Although the system with three server shows much lower waiting time for the customers, it will result in financial losses because of overstaffing and resource under-utilization issues due to low server utilization.
Furthermore, as you can see in the performance comparison table, the two server system already results in a good customer experience with low waiting time and better server utilization, making deployment of an additional server inefficient for barely improved measurements.

As the conclusion, because of the financial concerns and maximum operational efficiency, I recommend that it is **better to maintain the bank with the two server system** instead of implementing a system with an additional server.