DES Various Queuing Systems

Alireza Sheikh-Zadeh, Ph.D.

Several service counters

When we introduce a system including several counters (servers), we must decide what kind of queuing discipline to use. Are customers going to make one queue, or are they going to form separate lines in front of each counter? We first consider a single queue with several counters and later consider different isolated queues. We will not look at jockeying (entities are not allowed to switch lines).

Several counters but a single queue

Here we model a bank whose customers arrive randomly and are to be served at a group of counters, taking a random time for service, where we assume that waiting customers form a single first-in-first-out queue. (This is not new. We had done this when we modeled the clinic example.)

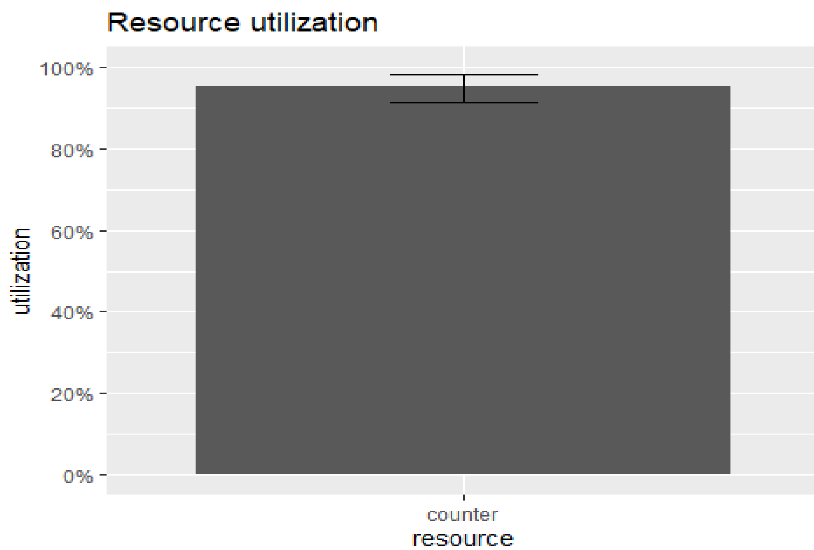
Here is the model information:

- There are three counters, and the system has only a single queue.
- The customer arrivals are following an exponential distribution with a mean of 4 minutes.
- The service time at every counter takes 10 to 15 minutes (uniformly distributed).

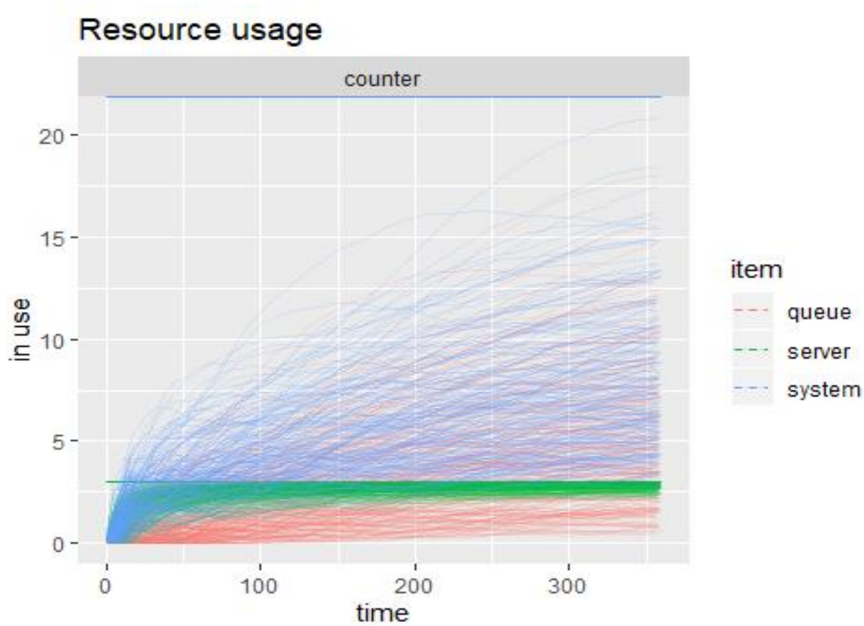
We run this model for 6 hrs (360 minutes), with 200 replications.

```
library(simmer)

set.seed(123)

customer <-
  trajectory("Customer's path") %>%

  seize("counter", 1) %>%
  timeout(function() runif(1, 10, 15)) %>%
  release("counter", 1)

library(simmer.plot)

# Very simple structure:
#plot(customer, verbose = T)
```

```r
set.seed(123)
envs <- lapply(1:200, function(i) {
simmer("Bank") %>%
  add_resource("counter", 3) %>%
  add_generator("customer", customer, function() rexp(1, 1/4)) %>%
  run(360)
})

resources <- get_mon_resources(envs)
plot(resources, metric = "utilization")
```



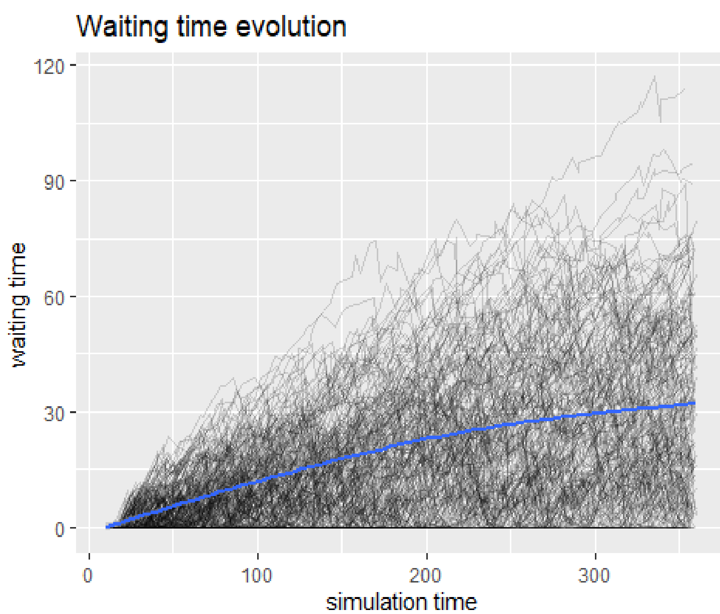

```r
plot(resources, metric = "usage")
```



Question:

- Visualize the waiting time.
- What is the average waiting time of customers?
- Find a 95% confidence interval for the waiting time?

```r
library(simmer.plot)
arrivals <- get_mon_arrivals(envs)
plot(arrivals, metric = "waiting_time")
```




```r
# Average waiting time
waitingTime = (arrivals$end_time - arrivals$start_time) - arrivals$activity_t
ime
mean(waitingTime)

## [1] 19.93408

# Find 95% Confidence Interval
arrivals2 = cbind(arrivals, waitingTime)
xbar = aggregate(arrivals2$waitingTime, by = list(arrivals2$replication), FUN
=mean)
quantile(xbar$x, c(0.025 , 0.975))

##       2.5%     97.5%
##   3.291684 46.338105
```

Several counters with individual queues

Each counter is now assumed to have its own queue. The programming is more complicated because the customer has to decide which line to join. The obvious technique is to make each counter a separate resource. In practice, a customer might join the shortest queue. We implement this behavior by first selecting the shortest line using the select function. We use seize_selected to enter the chosen line and later release_selected. The rest of the program is the same as before.

Our goal is to compare the system's waiting time when we use a single queue versus using individual queues.

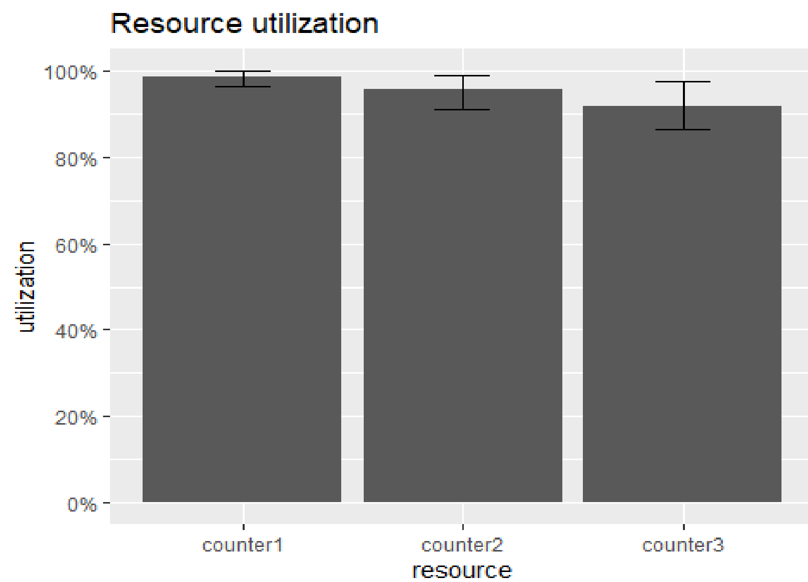
Here is the model information:

- There are three counters, and each counter has its own queue.
- The customer arrivals are following an exponential distribution with a mean of 4 minutes.
- The service time at every counter takes 10 to 15 minutes (uniformly distributed).

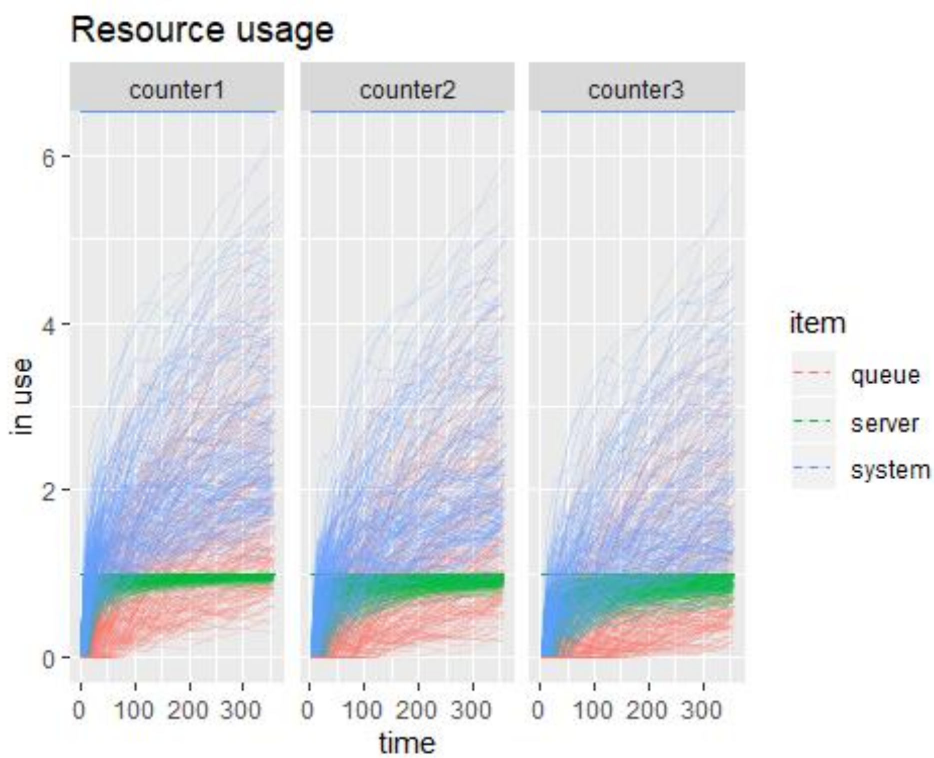
We run this model for 6 hrs (360 minutes), with 200 replications.

```
library(simmer)

set.seed(123)

customer <-
  trajectory("Customer's path") %>%

  select(c("counter1", "counter2", "counter3"), policy = "shortest-queue") %>
% # try policy = "random" and see the difference on the outputs.
  seize_selected() %>%
  timeout(function() runif(1, 10, 15)) %>%
  release_selected()


#library(simmer.plot)
# Very simple structure:
#plot(customer, verbose = T)

set.seed(123)
envs2 <- lapply(1:200, function(i) {
simmer("Bank") %>%
  add_resource("counter1", 1) %>%
  add_resource("counter2", 1) %>%
  add_resource("counter3", 1) %>%
  add_generator("customer", customer, function() rexp(1, 1/4)) %>%
  run(360)
})
```

```
resources <- get_mon_resources(envs2)
plot(resources, metric = "utilization")
```



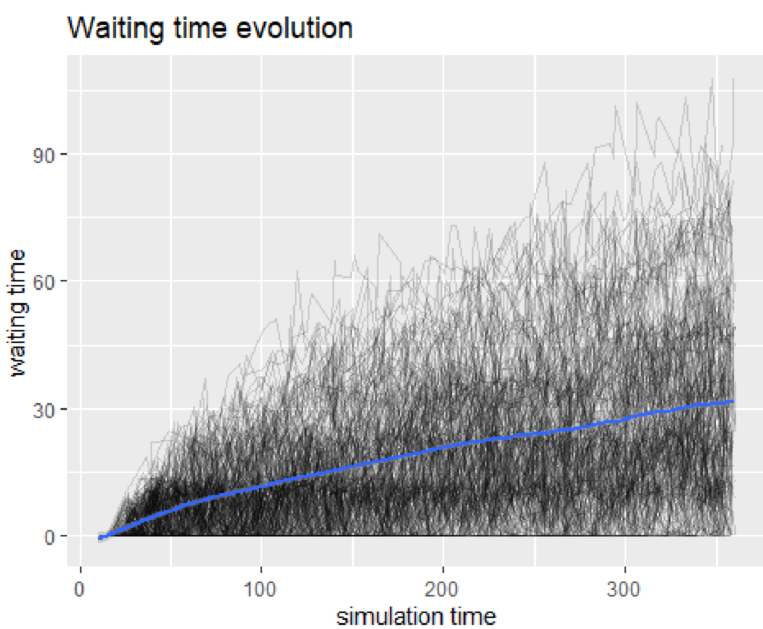

```
plot(resources, metric = "usage")
```



Question:

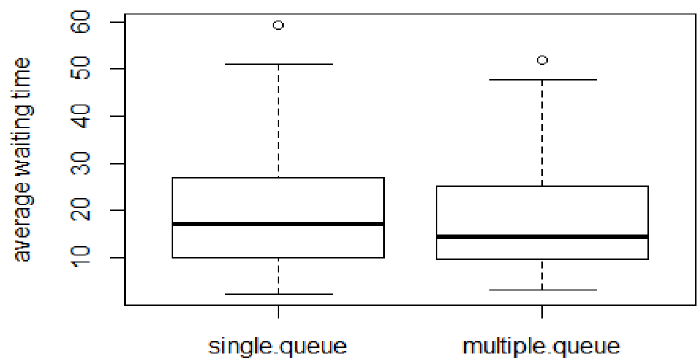
- Visualize the waiting time.
- What is the average waiting time for customers?
- Find a 95% confidence interval for the waiting time?
- Compare the waiting time of the system when we use a single queue versus using individual queues.

```
library(simmer.plot)
arrivals <- get_mon_arrivals(envs2)
plot(arrivals, metric = "waiting_time")
```




```
# Average waiting time
waitingTime = (arrivals$end_time - arrivals$start_time) - arrivals$activity_t
ime
mean(waitingTime) # Interpreted in the video.

## [1] 18.6902

# Find 95% Confidence Interval
arrivals2 = cbind(arrivals, waitingTime)
xbar2 = aggregate(arrivals2$waitingTime, by = list(arrivals2$replication), FU
N=mean)
quantile(xbar2$x, c(0.025 , 0.975))

##     2.5%    97.5%
##  5.03504 43.78023
```

```r
boxplot(xbar$x, xbar2$x, names = c("single.queue", "multiple.queue"), ylab =
"average waiting time")
```



Several counters with individual queues, each counter with different service time

Here is the model information:

- There are two counters, and each counter has its own queue. Customers select queues randomly.
- The customer arrivals are following an exponential distribution with a mean of 4 minutes.
- The service time at every counter is different:
  - Counter 1: Uniform(10, 15)
  - Counter 2: Normal(mean = 10, sd = 1)

We run this model for 6 hrs (360 minutes), with 200 replications.

Our goal is to compute:

- **the average queue size per counter.**
- **the average waiting time of customers per counter.**

```r
library(simmer)
customer <-
  trajectory("Customer's path") %>%

  branch(option = function() sample(1:2, 1, replace=T), continue = c(T,T),
```

```
  trajectory("A") %>%
  seize("counter1", 1) %>%
  timeout(function() runif(1, 10, 15)) %>%
  release("counter1", 1),

  trajectory("B") %>%
  seize("counter2", 1) %>%
  timeout(function() rnorm(1, 10, 1)) %>%
  release("counter2", 1)
)

#library(simmer.plot)
#plot(customer, verbose = T)

set.seed(123)
envs <- lapply(1:200, function(i) {
simmer("Bank") %>%
  add_resource("counter1", 1) %>%
  add_resource("counter2", 1) %>%
  add_generator("customer", customer, function() rexp(1, 1/4)) %>%
  run(360)
})

# The Overall Average Queue Size
resources = get_mon_resources(envs)

mean(resources$queue)

## [1] 7.432434

# The Average Queue Size per Counter
aggregate(resources$queue, by = list(resources$resource), FUN = mean)

##    Group.1        x
## 1 counter1 8.868399
## 2 counter2 6.124220

# The Overall Average Waiting Time
arrivals <- get_mon_arrivals(envs, per_resource = T)
waitingTime = (arrivals$end_time - arrivals$start_time) - arrivals$activity_t
ime
arrivals2 = cbind(arrivals, waitingTime)
aggregate(arrivals2$waitingTime, by = list(arrivals2$resource), FUN=mean)

##    Group.1        x
## 1 counter1 62.33137
## 2 counter2 42.15522
```