A system with multiple types of services and customers

Alireza Sheikh-Zadeh, PhD

A system with two types of customers

Suppose we have two different types of customers. Customers type 1 get service from counter 1, then leave. Customers type 2 get service from counter 2 and 3, sequentially. Here are the key parameters of the system.

- 80% of customers are type 1 and 20% type 2
- The arrival rate is Poisson with $\lambda = 12$ per hour. Therefore, the time between arrival is exponential with a mean of 5 minutes (why!).

- Serice Time on counter 1 is Normal(mean = 15, sd = 2)
- Serice Time on counter 2 is Normal(mean = 30, sd = 3)
- Serice Time on counter 3 is Normal(mean = 5, sd = 0.5)
- We have two staff for counter 1 and they can serve two customers at the same time.
- We have one staff for counter 2 and one staff for counter 3.

Run this model for 6 hours in 30 replications. (please watch the associated video and add notes to this document about the details of implementation and decision conclusion).

```
library(simmer)
library(simmer.plot)
```

```
## Loading required package: ggplot2
```

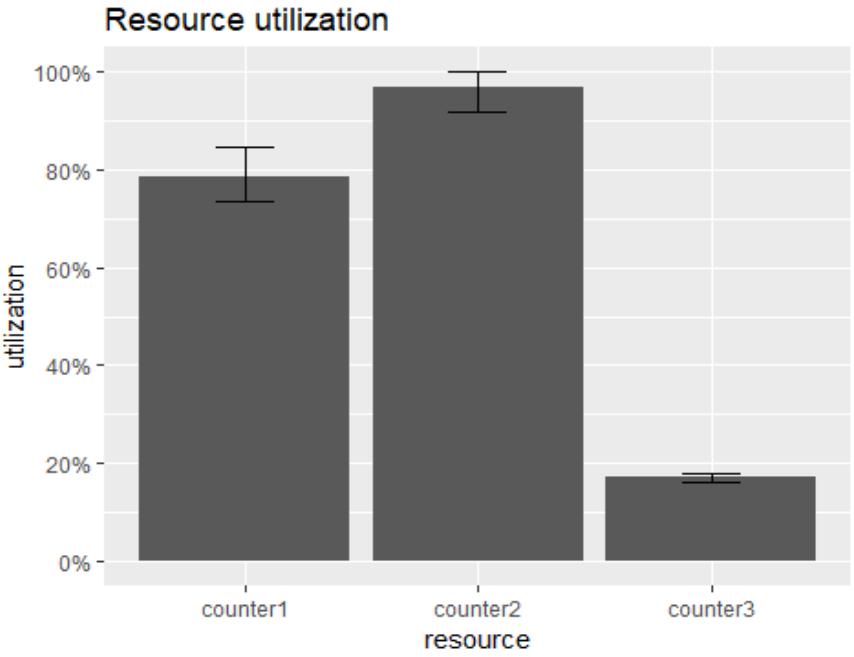
```
##
## Attaching package: 'simmer.plot'
```

```
## The following objects are masked from 'package:simmer':
##
##     get_mon_arrivals, get_mon_attributes, get_mon_resources
```

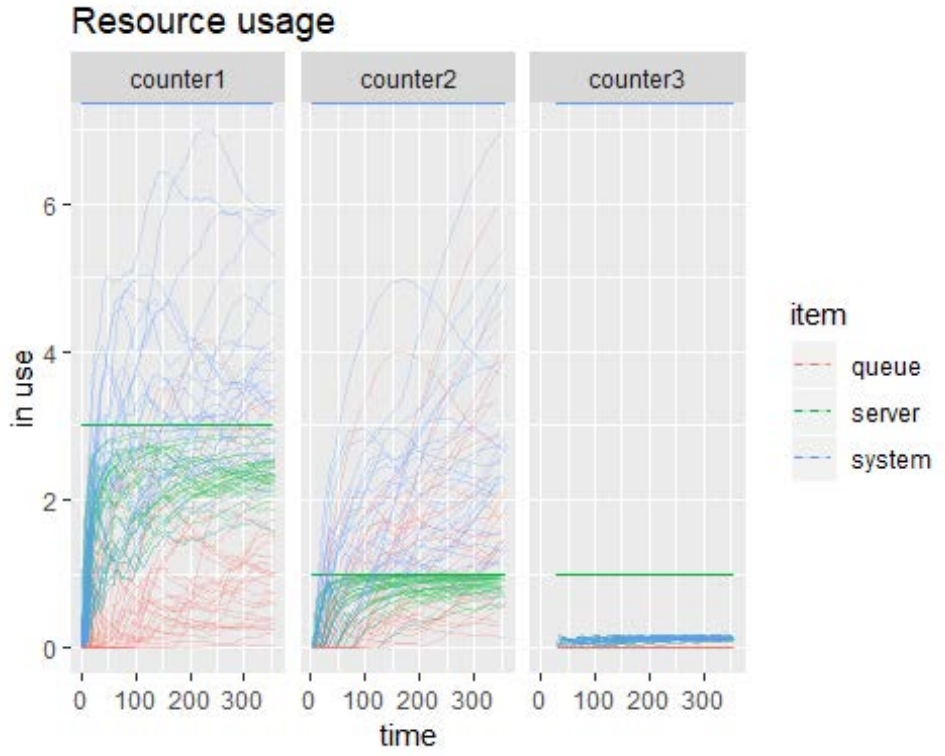
```
set.seed(123)
```

```
customer <- trajectory() %>%
  branch(option = function() sample(1:2, 1, prob = c(.8,.2), replace = T), co
ntinue = c(T, T),

         trajectory("Service A") %>%
           seize("counter1", 1) %>%
           timeout(function() rnorm(1, 15, 2)) %>%
           release("counter1", 1),
```

```
        trajectory("Service B") %>%
          seize("counter2", 1) %>%
          timeout(function() rnorm(1, 30, 3)) %>%
          release("counter2", 1) %>%
          seize("counter3", 1) %>%
          timeout(function() rnorm(1, 5, 0.5)) %>%
          release("counter3", 1)
)

#plot(customer, verbose = T)

set.seed(123)
envs <- lapply(1:30, function(i) {
simmer("DMV") %>%
  add_resource("counter1", 3) %>%
  add_resource("counter2", 1) %>%
  add_resource("counter3", 1) %>%
  add_generator("Customer", customer, function() rexp(1, 1/5)) %>%
  run(360)
})

resource <- get_mon_resources(envs)

mean(resource$queue)

## [1] 1.466355

plot(resource, metric = "utilization")
```



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




```
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 7.320331e+00
## 2 counter2 4.123452e+01
## 3 counter3 9.733462e-17
```