

Object Oriented Simulation

Multi-Server Systems

José M. Garrido C.

Department of Computer Science
Kennesaw State University

November 13, 2016

Multiple Servers

Multi-server systems include more than one server, and these provide service to the customers arriving into the customer queue(s).

The models of multi-server systems can be designed with several similar servers or with different types of servers.

Queuing Models

The three categories of queuing models that can be analyzed with simulation are:

- 1 single-server models
- 2 multi-server models
- 3 queuing networks

Performance Metrics

Some of the performance measures to be computed for queuing models are:

- For every server, the percentage of the total time that the server is actually carrying out the servicing of the customers, the server utilization
- The average number of customers waiting (i.e., the number of customers in the queue)
- The average time that a customer spends in the system, the average sojourn time
- The total number of customers serviced in a given interval, the system throughput
- The percentage of customers rejected because of the limited queue size. (These customers arrive when the queue is full.)

Behavior of Multi-server Models

In simple multi-server models, any of the servers can provide the service demanded by a customers waiting in the customer queue.

- The servers that are not busy, are usually stored in server queue.
- An arriving customer will remove the server at the head of the server queue and reactivate the server.
- The system has several servers providing service to customers.
- All servers are alike, e.g., there is only one type of server.
- In a similar manner, all customers are alike, there is only one type of customer.

Behavior of Multi-server Models

In simple multi-server models, any of the servers can provide the service demanded by a customer waiting in the customer queue.

- The servers that are not busy, are usually stored in server queue.
- An arriving customer will remove the server at the head of the server queue and reactivate the server.
- The system has several servers providing service to customers.
- All servers are alike, e.g., there is only one type of server.
- In a similar manner, all customers are alike, there is only one type of customer.

Illustration of a Multi-server Model

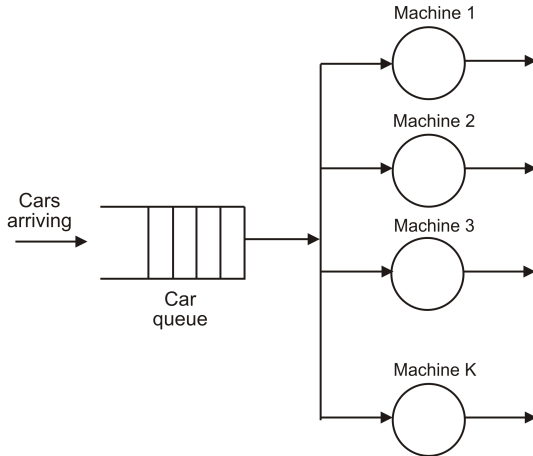


Figure: A multi-server model

Multi-server Carwash System

- The model of the Carwash system has K wash-machines.
 - A customer waits in the queue until any server becomes available.
 - The server then removes the customer from the customer queue and starts servicing to it.
-
- The multi-server model has an array of K server objects.
 - This array is needed first, so that customer can reference any server, second every server will compute its own to idle time.
 - There is a queue for idle servers.
 - The capacity of the server queue is K .

Queues in the Model

The model has two queues:

- One queue for cars waiting for service
- One queue for idle servers (waiting for cars)

As in the single-server model, the queues in this model are simple, i.e., they follow a first-in-first-out (FIFO) order.

General Behavior of the Carwash System

- 1 A new car object (customer) arrives and checks if there is any server available in the server queue. If so, the car removes the server at the head of the server queue and reactivates it.
- 2 The car joins the car queue to wait for service.
- 3 When a server becomes available, it removes the car from the car queue.
- 4 The server provides the service for service interval of the car object.
- 5 When the service has completed, the car object continues its own activities and the server becomes available.
- 6 If the server finds that the car queue is empty, the server object becomes available, is made idle, and joins the server queue.

The Server Object

- 1 Examine the customer queue and if it is not empty, perform the following activities:
 - 1 Remove the customer from the head of the customer queue
 - 2 Perform the service to the customer for the specified service interval
 - 3 Reactivate the customer object and let it continue its own activities
- 2 If the customer queue is empty, join the server queue and suspend itself to wait (in the idle state).

The Car Object

- ➊ Examine the car queue and if it is not full, join the queue.
 - ➊ Check whether there is an available server object by examining the server queue and if not empty, remove the server object from the head of the server queue.
 - ➋ Reactivate the server object.
 - ➌ Suspend itself and go into the idle state.
 - ➍ Reactivated by the server object after completed the service, continue with own individual behavior before terminating.
- ➋ If the car queue is full, terminate immediately , the car has been rejected.

The Arrivals Object

Repeat while the clock time is less than the close-arrival period.

- 1 Generate a random inter-arrival time interval.
- 2 Wait until the simulation clock advances by a time interval equal to the inter-arrival interval.
- 3 Generate a random customer service interval.
- 4 Create a customer object.

Implementation of the Multi-server Carwash Model

The OOSimL implementation of the multi-server Carwash model is stored in file `mcarwash.jar`. The source files included are: `MCarwash.osl`, `Car.osl`, `Server.osl`, and `Arrivals.osl`. The additional source file `SimModel.osl` is necessary to carry out the GUI I/O with the ACM classes.

First and Second GUIs of Multi-server Model

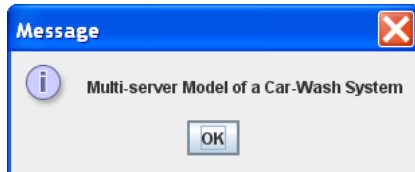


Figure: First GUI

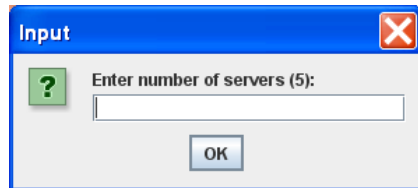


Figure: Second GUI

Results GUI of Multi-server Model

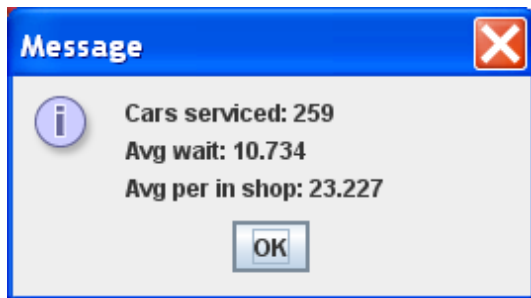


Figure: Results GUI of multi-server model

Partial Output of a Simulation Run

End Simulation of Multi-server Model of Carwash System
date: 2/12/2010 time: 10:20

Total number of cars serviced: 259
Car average wait period: 10.734
Average period car spends in the shop: 23.227
Machine 0 utilization: 0.984
Machine 1 utilization: 0.98
Machine 2 utilization: 0.984
Machine 3 utilization: 0.984
Machine 4 utilization: 0.976

Partail Code in class *Car*

```

if MCarwash.servQueue is not empty
then
    // get next available server
    remove object wmach of class Server from
        MCarwash.servQueue
    set servname = wmach.get_name()
    display cname, " activating server ", servname,
        " at ", simclock
    tracewrite cname, " activating ", servname
    reactivate wmach now
endif
//
suspend self                                // to wait for service

```

Partail Code in class *Server*

```
if MCarwash.car_queue is empty
then
    // car queue is empty
    set startIdle = get_clock() // start idle period
    tracewrite sname, " to idle state"
    // join available server queue
    insert self into MCarwash.servQueue
    suspend self // suspend server
    //
    // reactivated by a car object
    // queue must now be nonempty
    set simclock = get_clock()
    set idle_period = simclock- startIdle
    add idle_period to MCarwash.accum_idle[snum]
    display sname, " reactivated at " + simclock
endif
call serviceCustomer // service the car
```

Implementation of the Multi-server Barbershop Model

Very similar to the multi-server Barber model. The OOSimL implementation of the multi-server Barbershop model is stored in file `mbarber.jar`. The source files included are:
`MBarber.osl`, `Customer.osl`, `Server.osl`,
`Arrivals.osl`, `MBarberGUI.osl`, `MBarberPanel.osl`.

Input GUI of Multi-server Barbershop Model

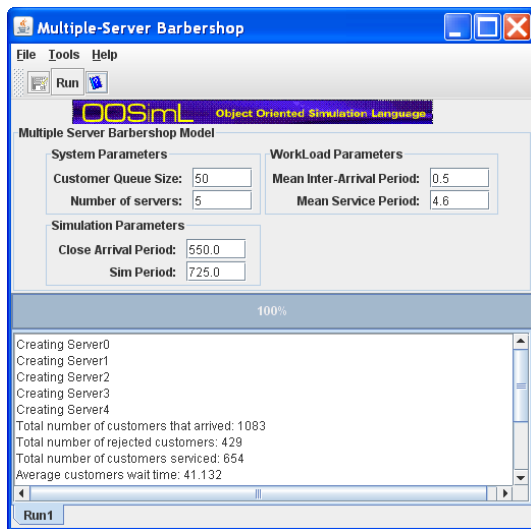
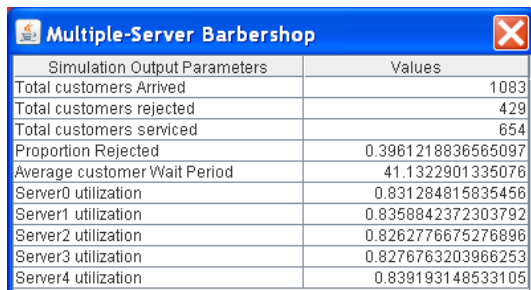


Figure: Input GUI of multi-server Barbershop model

Results GUI of Multi-server Barbershop Model



The screenshot shows a window titled "Multiple-Server Barbershop" with a blue header bar. Inside the window is a table with two columns: "Simulation Output Parameters" and "Values". The table contains the following data:

Simulation Output Parameters	Values
Total customers Arrived	1083
Total customers rejected	429
Total customers serviced	654
Proportion Rejected	0.3961218836565097
Average customer Wait Period	41.1322901335076
Server0 utilization	0.831284815835456
Server1 utilization	0.8358842372303792
Server2 utilization	0.8262776675276896
Server3 utilization	0.8276763203966253
Server4 utilization	0.839193148533105

Figure: Results GUI of multi-server Barbershop model

Models with Several Customer Queues

Multi-server models can include more than one customer queue and the simplest models have:

- A separate queue for every server.
- Each server object owns a queue and a customer object will receive service from only one server object.
- The customer queues are simple queues — FIFO queues.

Illustration of Model with Multiple Queues

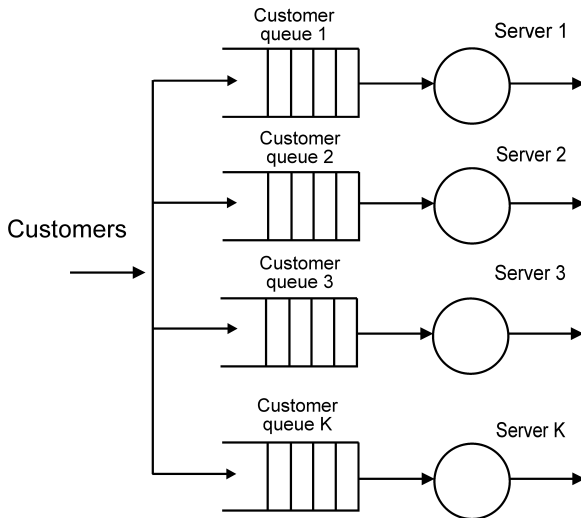


Figure: A multi-server model with multiple queues

Managing Multiple Queues

- Arriving customers normally select the station with the shortest queue.
- If all the queues have the same size, then an arriving customer joins the first queue.
- If a queue is full, the customer will ignore it.
- If all the queues are full, the customer is rejected and terminates.

Behavior of The Customer Object

- ➊ Join the shortest customer queue that is not full.
 - ➊ Reactivate the server object of the queue if the server is idle.
 - ➋ Suspend itself and go to the idle state.
 - ➌ After the service has been completed, continue own activities.
 - ➍ Terminate.
- ➋ If all the customer queues are full, terminate (e.g., the customer object is rejected).

Behavior of The Server Object

- ➊ Repeat while the customer queue for this server is not empty
 - ➊ Remove the customer object from the head of the corresponding customer queue
 - ➋ Perform the service to the customer object
 - ➌ Reactivate the customer object.
- ➋ If the customer queue is empty, suspend itself and wait in the idle state until the next customer object arrives.
- ➌ When reactivated by a customer object, start from step 1.

First GUI of Model with Multiple Queues

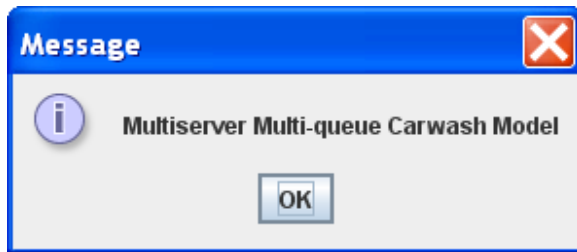


Figure: First GUI of multi-server model with multiple queues

Partial Output of a Simulation Run

```
End Simulation of Multiserver Multi-queue Carwash Model  
date: 2/15/2010 time: 13:48  
Elapsed computer time: 2328 millisec
```

```
Total number of cars serviced: 327  
Car average wait period: 153.004  
Average period car spends in the shop: 167.475  
Machine0 utilization: 0.998  
Machine1 utilization: 0.995  
Machine2 utilization: 0.989  
Machine3 utilization: 0.988  
Machine4 utilization: 0.979
```