REAMDE
COP 290
ASSIGNMENT 6 BANK SIMULATION
BY – Sagar Sharma 2018CS10378

TO RUN:
$ make clean
$ make

IMPLEMENTATION

SINGLE QUEUE:
All the simulation is handled by event queue. In the beginning I have initialised event queue by successfull teller events and customer arrival events.
Now the simulation starts. First event popped is teller completion event, he checks for customers in the single queue which is empty in the beginning. We push the same event after changing its type to teller idle event. This goes on untile first customer arrives. We pop the arrival event and insert the customer associated with it to the single queue. Now after this when the first teller idle event is popped, he sees a customer in queue and generates a service time and two event are pushed into event queue. First event is customer completion event in which we print out customer's stats, and the second is teller completion event.
The simulation keeps on running like this till no customer is left.

There are 4 types of events in event queue
// type- 1 Newly arrived customer
// type- 2 Customer with completed task
// type- 3 Teller with completed time
// type- 4 teller with idle time

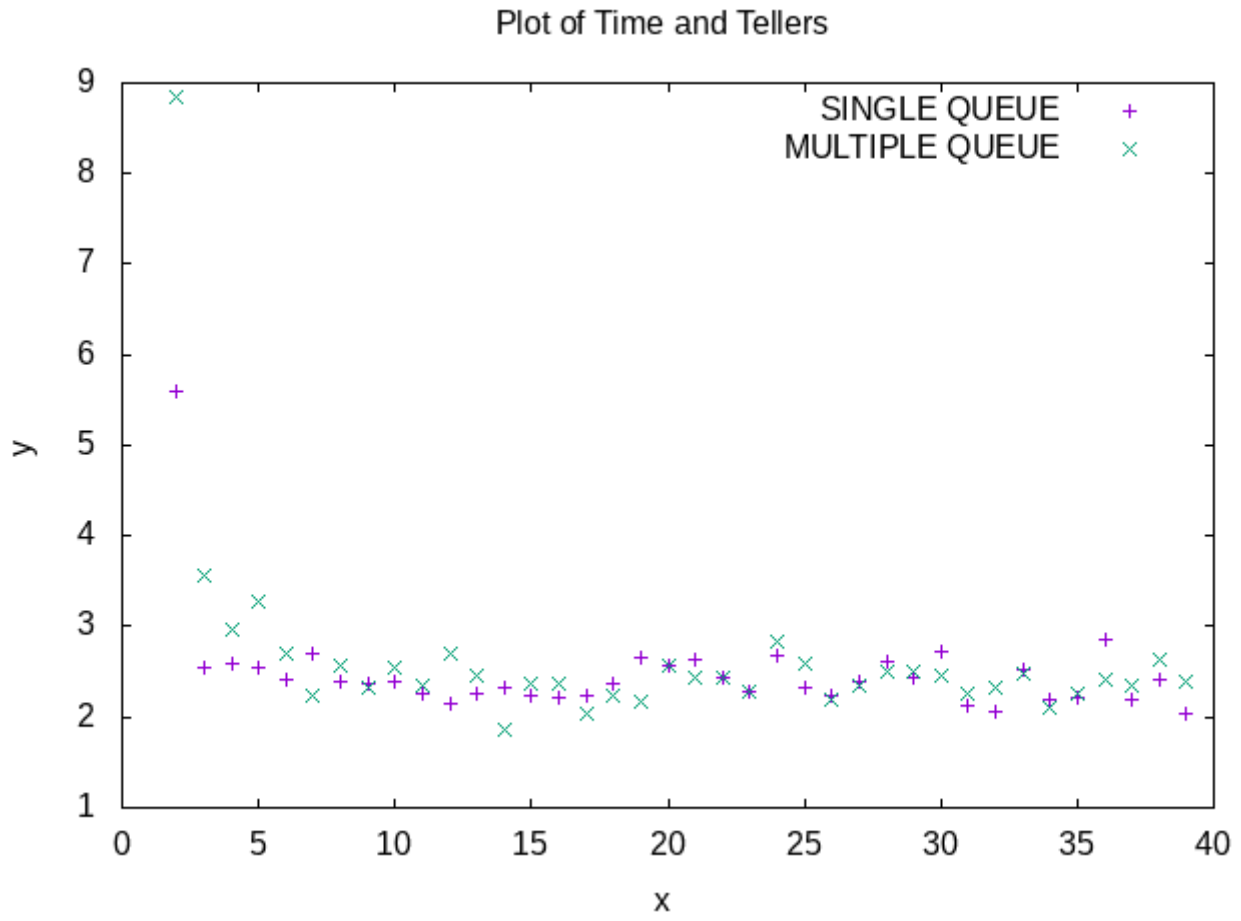
MULTIQUEUE:
All the simulation is handled by event queue. In the beginning we have inserted teller events with tellers having a field for line number they belong to. Then there are customer arrival events.
Now the simulation starts. First event popped is teller completion event but as all teller queues are empty, we push the same event into event queue after changing its type to teller idle event.
Now first customer arrives and as all queues are empty that all of them are shortest, he randomly selects one teller queue and is pushed into it. Now after this customers keep on arriving and randomly joining shortest queues. After this when first teller idle event is popped, he first checks his queue if it is non empty, a customer is popped and two events are generated just like in single queue. If his queue is empty then he looks for other non empty queue and randomly selects one them. A customer is popped from the selected queue and served and two events are pushed into event queue. In

case all queues are empty, an event of type teller idle time is inserted back into event queue.

GRAPHS AND INFERENCES

60 min Simulations

Graph 1: y-> avg time spent in bank x-> no of tellers in bank No of customers = 50

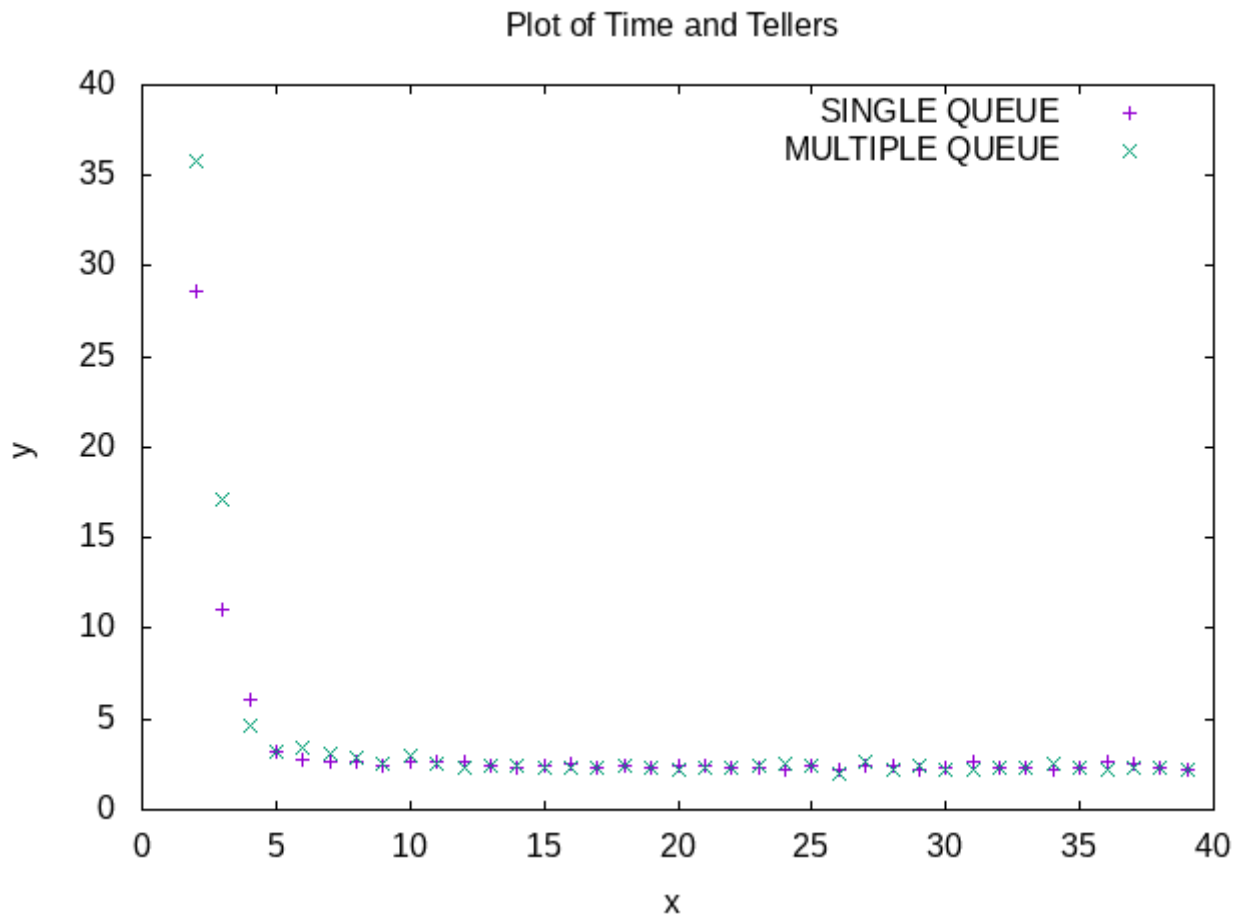


Inferences:

1. We can clearly see that less number of teller corresponds to more avg time spent in bank
2. There is no clear indication which model is better as most of the times customers arrive when the bank is empty. The model is quite random. **But at low number of tellers single queue performs better than multi**
3. After certain number it is useless to increase no of tellers in both cases as similar avg time is observed

Graph 2

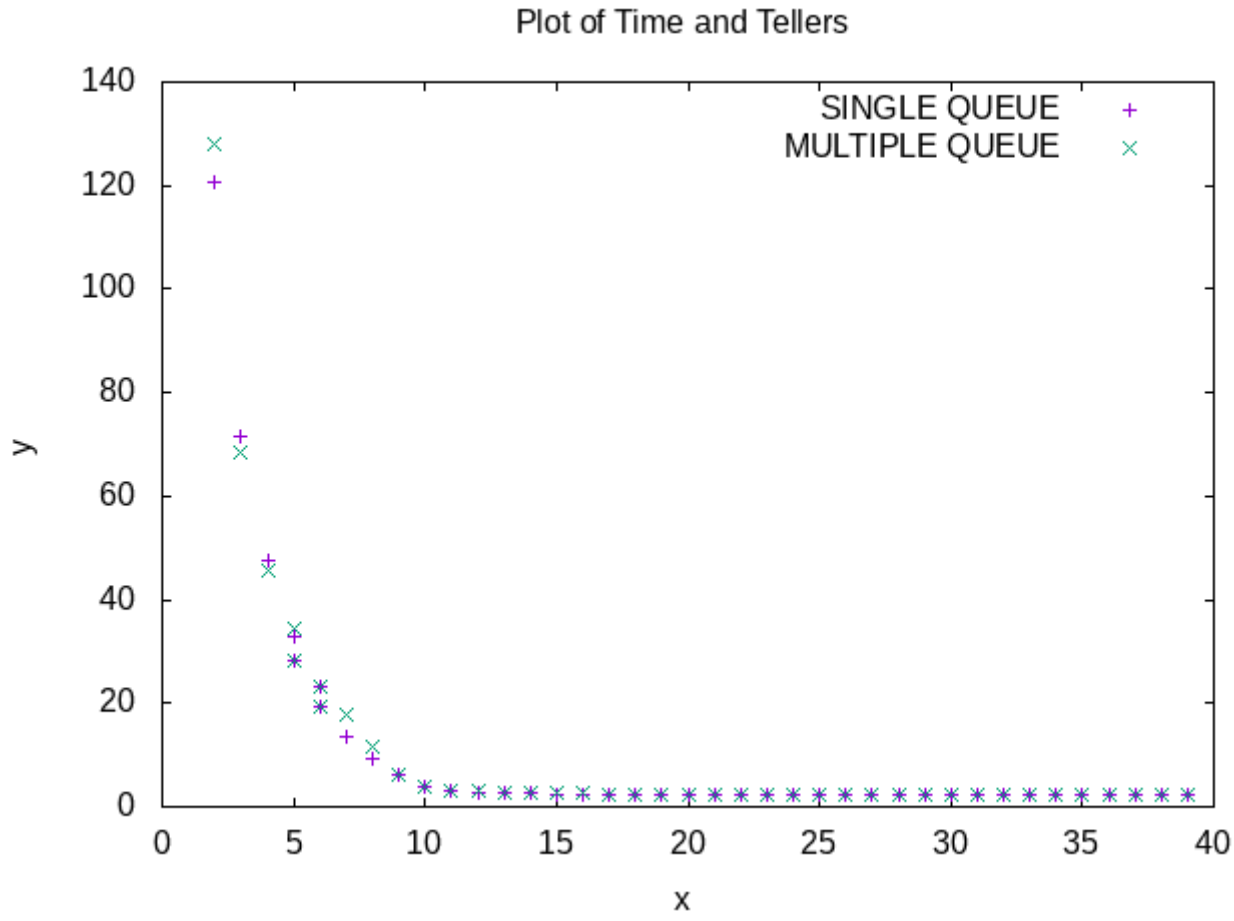
y-> avg time spent in bank x-> no of tellers in bank No of customers = 100



Inferences:

1. Now we can see a better curve. Avg time decrease as no of teller increase in both models.
2. After certain number the avg time is not changing that much. We can say 10 tellers is optimum in this case.
3. **Multiqueue performs lower than single queue when there are less number of tellers**

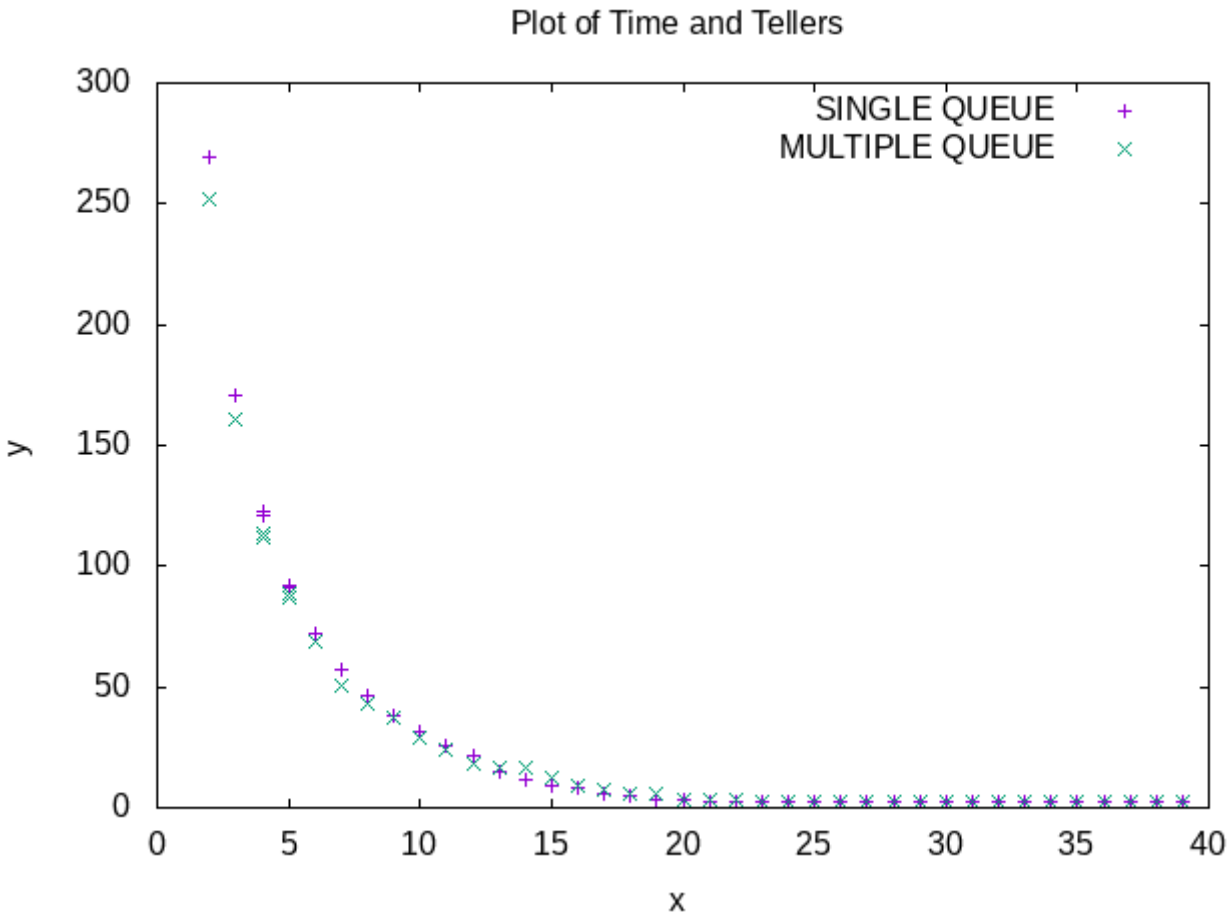
Graph 3
y-> avg time spent in bank x-> no of tellers in bank No of customers = 250



Inferences:

1. This is even more clear graph showing the behavior. For any no of tellers in both cases the avg time increases as customer increase.
2. Also the elbow point of graph has shifted towards right telling that more tellers are needed to give better time.
3. Same as above graph saturates after elbow point.
4. **Now one can see that even at low number of tellers, the gap between the two models is decreasing as number of customers are increasing.**

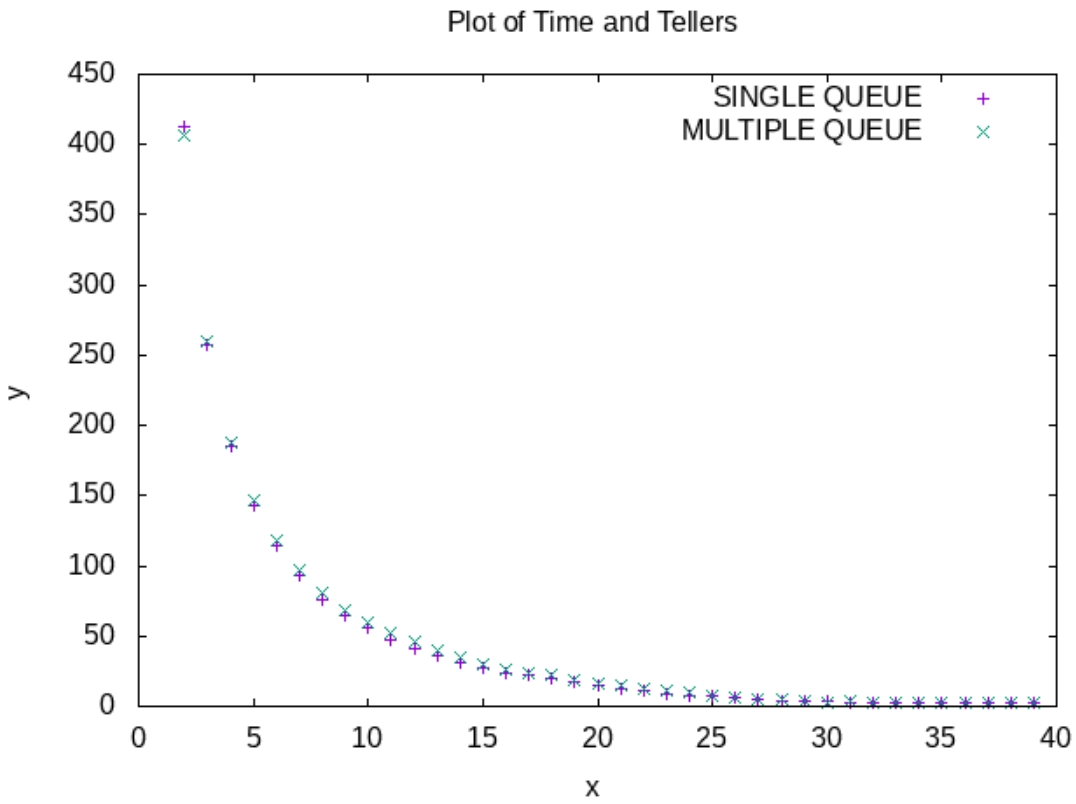
Graph 3
y-> avg time spent in bank x-> no of tellers in bank No of customers = 500



Inference:

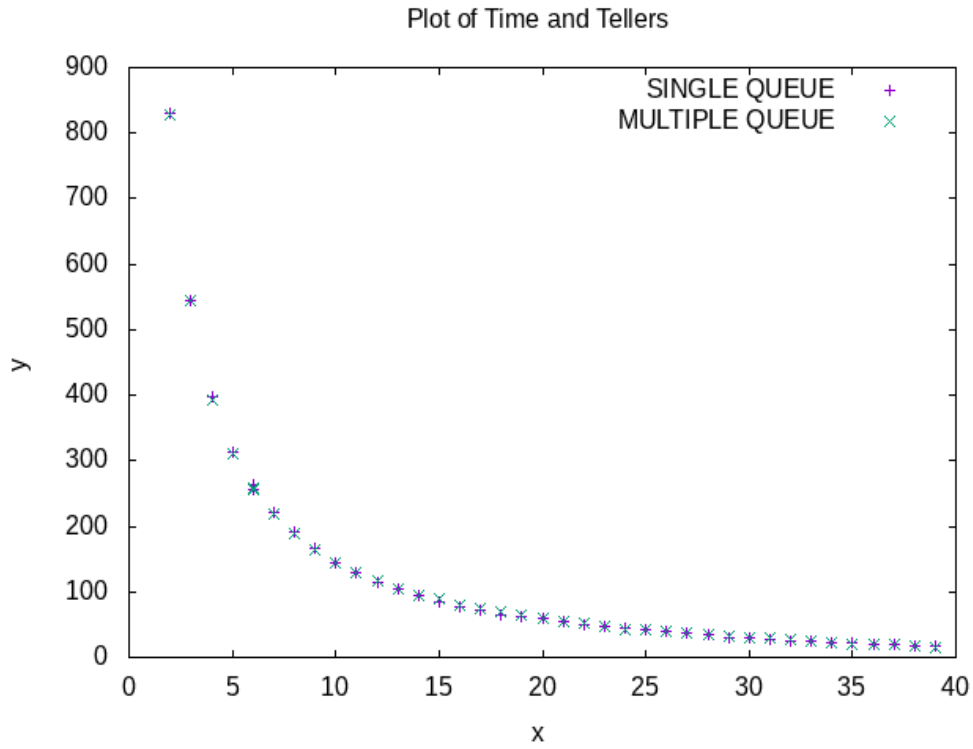
1. Elbow point shifted.
2. Multi queue is now performing better than single queue at low number of tellers because no of customers are now above a threshold.

Graph 4
y-> avg time spent in bank x-> no of tellers in bank No of customers = 750

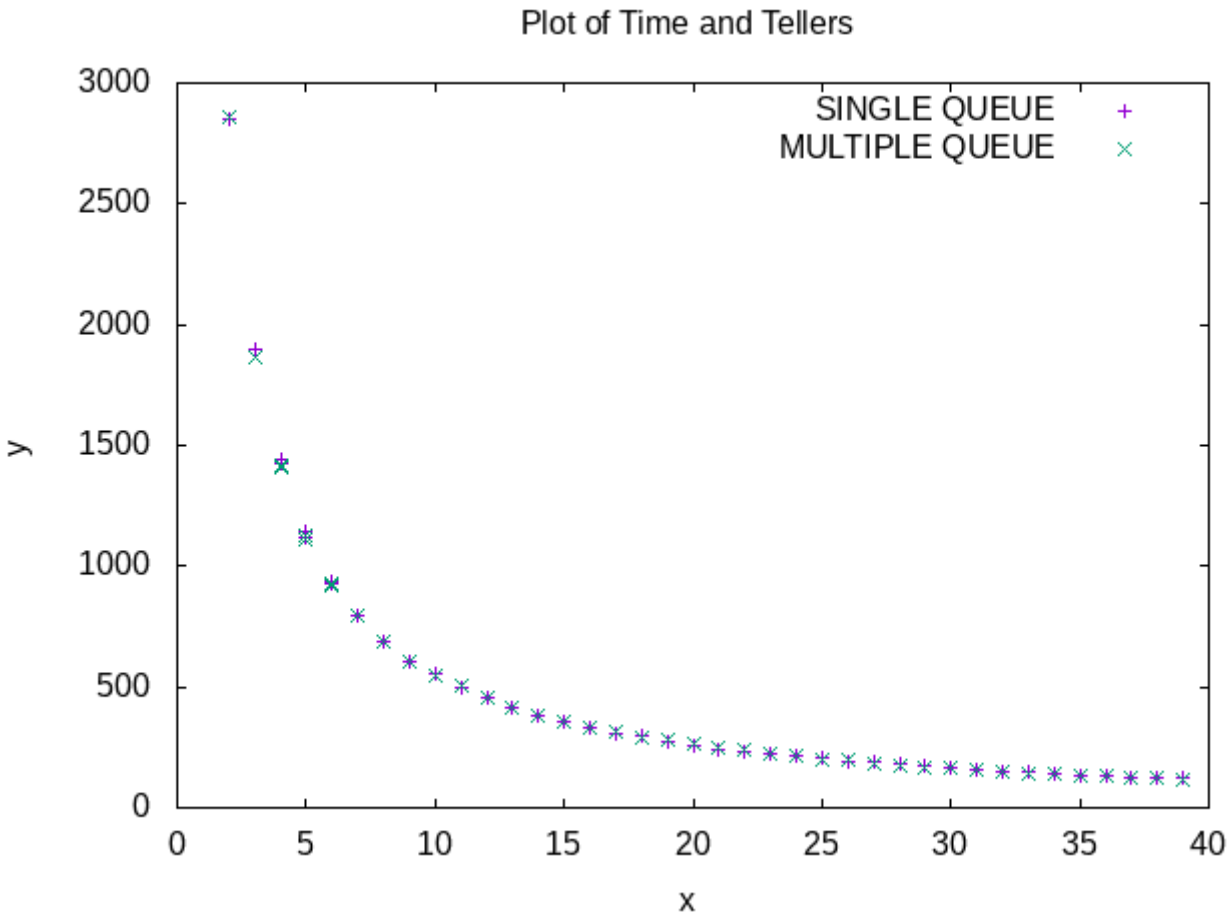


Both models overlap and elbow point shifted.

Graph 5 y-> avg time spent in bank x-> no of tellers in bank No of customers = 1500



Graph 5 y-> avg time spent in bank x-> no of tellers in bank No of customers = 5000



OUTPUTS:

./bin/qSim 100 4 60 2.3

FINAL STATS - SINGLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 9.114147
max_wait_time = 12.720654
total_amt_service_time = 254.434143
total_amt_idle_time = 18.866783
No of times fp used : 771

FINAL STATS - MULTIPLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 3.811847
max_wait_time = 5.009598
total_amt_service_time = 237.211288
total_amt_idle_time = 29.659966
No of times fp used : 794

./bin/qSim 100 6 60 2.3

FINAL STATS - SINGLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 2.885786
max_wait_time = 1.970360
total_amt_service_time = 244.777588
total_amt_idle_time = 142.558090
No of times fp used : 616

FINAL STATS - MULTIPLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 2.584434
max_wait_time = 2.634933
total_amt_service_time = 213.979080
total_amt_idle_time = 166.572525
No of times fp used : 954

./bin/qSim 100 9 60 2.3

FINAL STATS - SINGLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 2.468444
max_wait_time = 0.672941
total_amt_service_time = 226.643539
total_amt_idle_time = 362.761444
No of times fp used : 771

FINAL STATS - MULTIPLE QUEUE
total_no_cust_served = 100;
avg_time_spent_in_bank = 2.435224
max_wait_time = 1.100491
total_amt_service_time = 222.389725
total_amt_idle_time = 358.439575
No of times fp used : 1193