Computer aided simulations and performance evaluation - Lab 1

Ruben Berteletti - s277757

February 2021

1 Introduction

- In order to have affordable results, all the experiments done in this lab have been launched for 10 runs and on top of that a 95 % confidence interval has been built.
- The arrival rate is taken fixed at 5 s and the simulation time at 10000 s in order to have comparable results among the policies.
- The queues have only been tested in ergodic conditions.
- For replicability the seed is set to 22.

2 Exercise 1.1

The aim of this exercise is to use the simulation to compare analytical and empirical results for queues with *Poisson* arrival rate and different distribution for the service time (here exponential and uniform have been chosen). Moreover, the comparison is done also between finite or infinite capacity of the waiting line.
The main engine of the simulation consists in a nested loop following the *event scheduling* approach, where a class called *Measure* has the task of tracking all the performance metrics of the queue which during each event is updated.

2.1 Infinite capacity of the waiting line

In this setting the queuing theory provides all the analytical tool to inspect the behaviour both of M/M/1 queue and M/G/1 queue, then the simulation can work as a confirmation of the correctness of those formulas.
The simulation has been launched for different mean values for the service time both for the exponential distribution and uniform one, in order to test the queue under different condition of loads. The figure 1 shows the result of the script and many considerations could be made:

1. Service times uniformly distributed lead always in a faster queue, where both average number of customers and delays are less then those coming from the exponential distribution. This is explained looking at the coefficient of variation defined as $C_s^2 = var(S)/E^2[S]$ that the more is higher, the more it contributes to wasting time through the queue. For the exponential distribution the coefficient is $C_s^2 = 1$ always, while for the uniform one it ranges in the interval [0.01, 0.21].
2. Theoretical values are most of the time within the 95% CI, making the simulation affordable.
3. For both distributions when the load is close to 90% the performances start to get worse exponentially, until when close to 100% the curves converge.

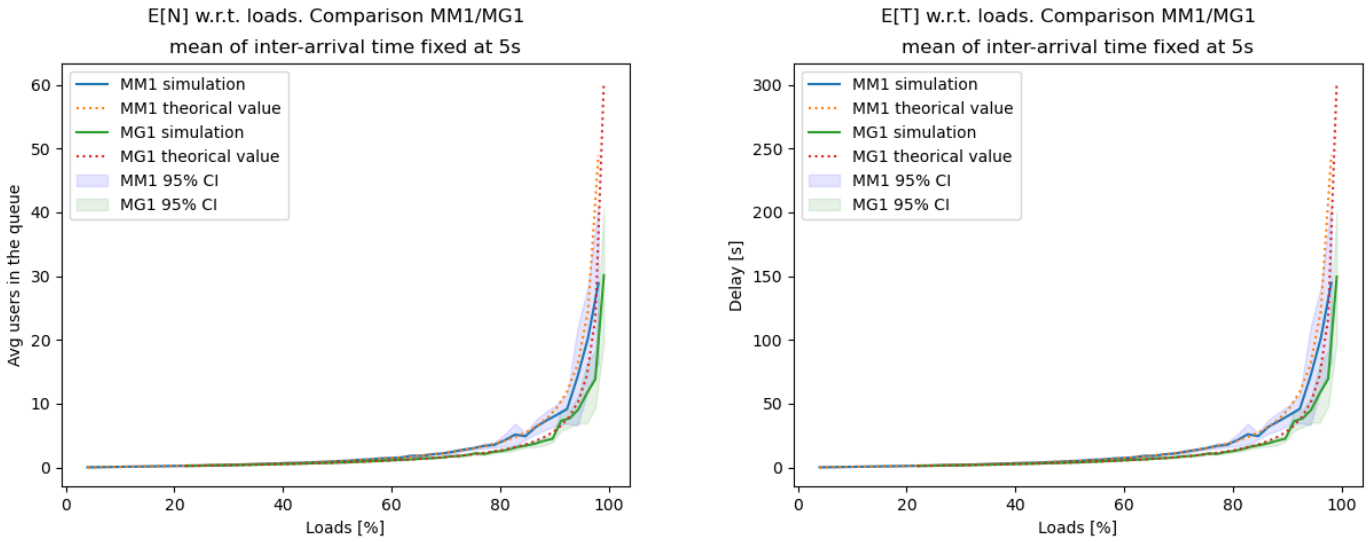

Figure 1: Average number of users (*left*) and delay (*right*) comparison between MM1 and MG1 queues.

2.2 Finite capacity of the waiting line

In this case theoretical formulas are not available for queue whose distribution of the service time is not exponential, therefore the analysis has been done only empirically.
Now, the queue's possible states are only B, where B is the capacity of the waiting line plus 1 that is the client already in service, then by construction the system is always ergodic. Introducing a finite capacity of the waiting line means allowing the queue to have losses, then as a new metric the loss probability can be analyzed to have a better view. The graphical comparison (figure 2 and figure 3) is made in this paper for simulation launched for $B = [2, 5, 7, 10, 20]$ and for simplicity, since the general trend for delays and average customers in the queue is the same, the former is avoided. The simulation shows:

1. The average number of users and delays of the two distributions are now similar, given the introduction of a factor that stabilizes the variance, which is the finite capacity of the waiting line; only for high value of B the smaller variance of the uniform distribution can be appreciated.

2. The loss probability grows as the capacity of the waiting line reduces, in this setting $B = 20$ guarantees performances close to the infinite capacity one.

3. The loss probability is smaller for the uniform distribution for all the loads and for all the capacities, this fact can be explained, once again, with the less variability that affect a client that cross the queue.

4. The effect of the finite waiting line is warned mostly when the load is huge (roughly $\geqslant 90\%$) allowing the queue to remain stable at the cost of losing a fraction of clients.

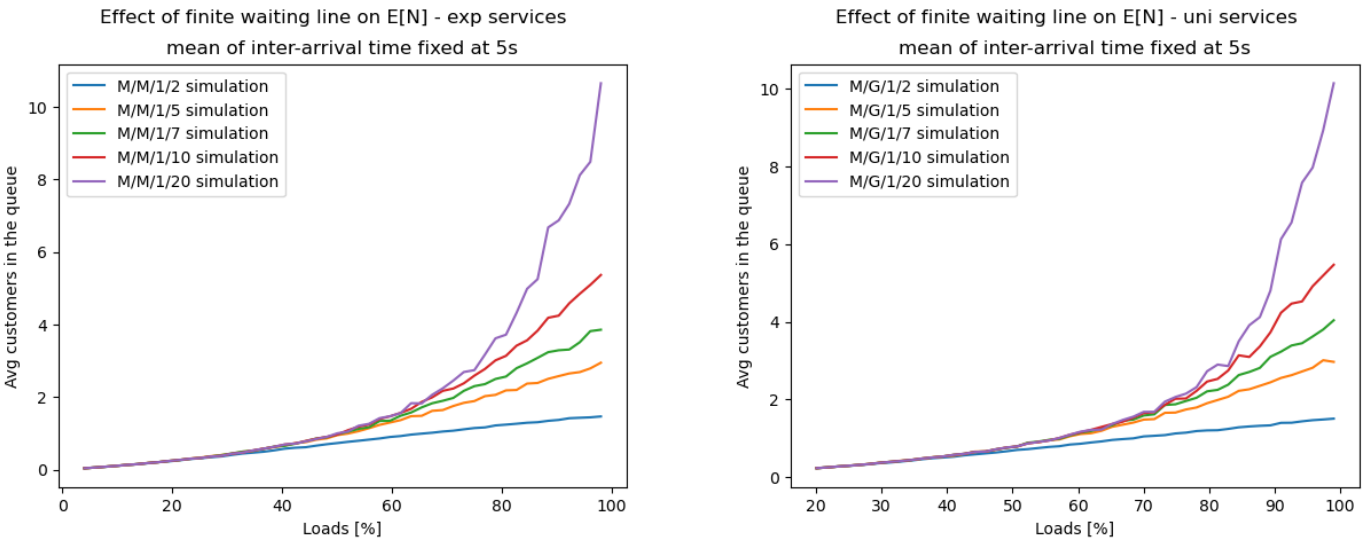

Figure 2: Average number of users for exponentially distributed service times (*left*) and uniformly distributed (*right*), given different capacities of the waiting line.

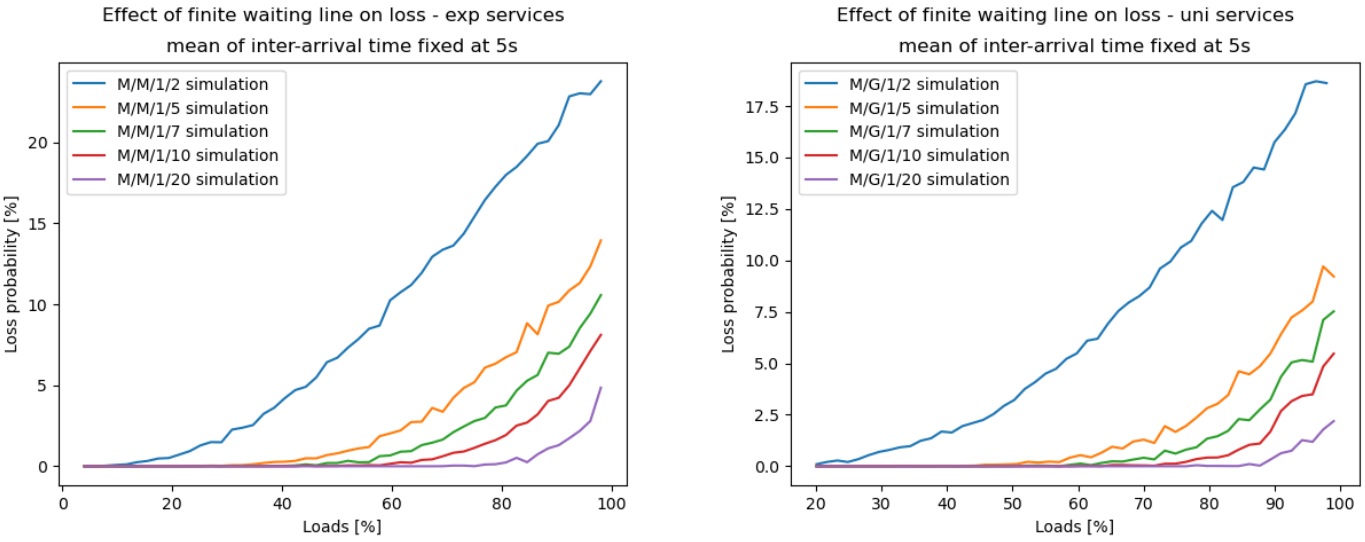

Figure 3: Loss probabilities for exponentially distributed service times (*left*) and uniformly distributed (*right*), given different capacities of the waiting line.

3 Exercise 1.2

Now the focus is on the performances of the queue in case of multiple servers (m), in particular with $m = 2$ which is enough to understand the behaviour trough different loads.
In the following sections two setting have been adopted:

- Servers with the same capacity (same service rate);
- Servers with different capacity, according to two policies in case of both servers idle, i.e. choosing a server at random or choosing the fastest one.

A new metric that can be analyzed is the distribution of calls of the two servers, in order to have insights about the the effect of the bottleneck (in case of different capacity).
Here the theoretical tools are not available for the MGm queues and not easy to computed for the MMm queues, then the results reported are only empirical, achieved then trough simulation.

3.1 Same server capacity

In this case both servers have the same distribution whose parameters are equal, this means that the the variation from state i to state *i-1* happens according to a distribution whose mean service rate is $m\mu$, where m is the number of servers.
The result is that given the same inter-arrival time and given same service time as before, the load is halved for 2 servers, in general the load becomes $load = \frac{servicetime}{inter-arrivaltime*n.servers}$ or, looking at the rates, $load = \frac{\lambda}{\mu * m}$.
Even here, delay chart and average users chart follow the same pattern, then only the users one is analyzed. Regarding the queues with finite capacity is interesting to look at the behaviour of the loss probability.

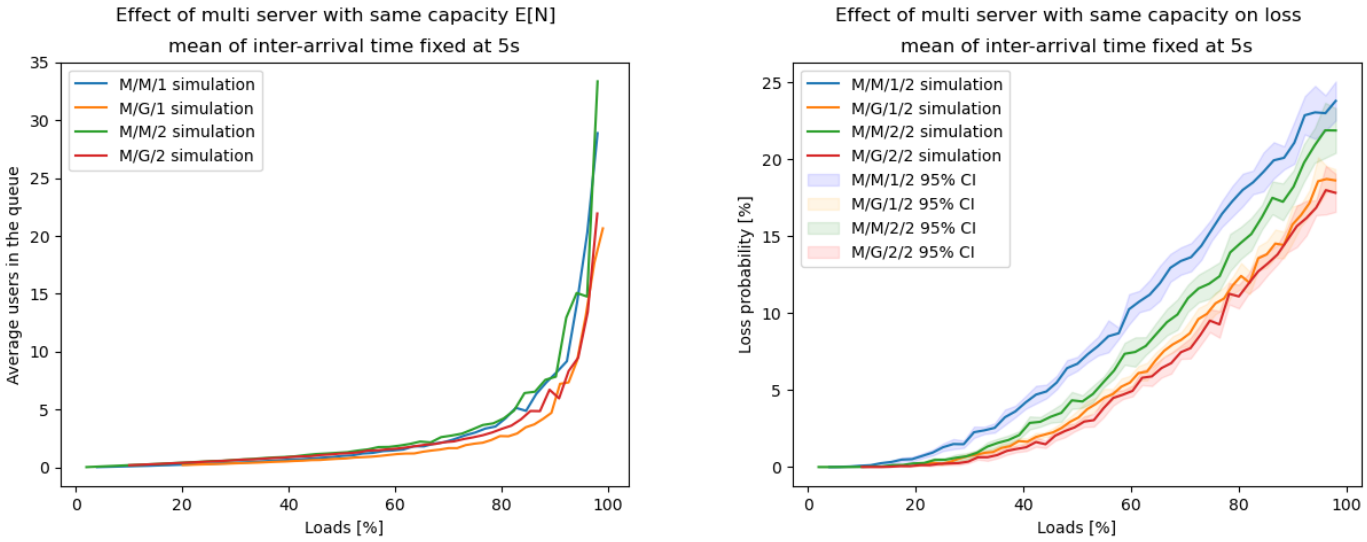

Figure 4: Multi server with same capacity. Effect on the average number of customers on infinite capacity queue (*left*) and effect on the loss probability on the finite capacity of the waiting line queue (*right*).

The analysis for $m = 2$ shows that the M/M/1 queue and the M/G/1 queue have for each load, a number of customer smaller then the M/M/2 and the M/G/2, explained intuitively trough the fact that there is always an additional client in service; the delays distribution follows the same trend.
Looking instead at the queues with finite capacity, the simulation shows that the additional server

is capable to reduce the loss probability for both distribution; must be noticed that the M/G/1/B queue has even less losses than the M/M/2/B.
The reasoning above is valid and then can be extended for any number of servers m.

3.2 Different server capacity

Having servers with different capacity means that a client could follow a policy for the path adopted and among those available, in this paper two of them are analyzed: choose the server at random and choose always the fastest one. Of course the client has 'decision power' only when both servers are idle, instead when there's only a server busy, he goes automatically in the other one. Moreover, the client is not allowed to change server during the waiting time.
Rather than explore the queues' behaviour under different loads, could be interesting focusing only on a single load, here 96%, in order to have a quantitative effect of the different policies. The results are displayed in the table below:

Load 96%	avg users	departures	delay	s1 calls	s2 calls
exp random	13.30	1956.50	67.70	797.90	1160.50
exp fastest	19.06	1992.00	94.60	796.30	1197.30
uni random	14.40	1990.50	69.41	1191.70	800.60
uni fastest	11.29	2000.70	55.76	1204.00	798.70

Table 1: Comparison between queues with infinite line capacity using two different policies: *random* and *fastest*. Values intended as mean of 10 runs.

Load 96%	avg users	departures	delay	losses	s1 calls	s2 calls
exp random	1.66	1452.20	11.47	526.30	614.50	839.00
exp fastest	1.66	1466.50	11.35	534.60	562.50	905.10
uni random	1.70	1521.40	11.22	481.30	888.10	634.50
uni fastest	1.70	1529.80	11.09	465.50	935.50	595.50

Table 2: Comparison between queues with finite line capacity using two different policies: *random* and *fastest*. Values intended as mean of 10 runs.

For both configurations the policy *fastest* allow the queue to be capable to handle better the clients, since the departures are more. As seen before, the effect of the waiting line is to regularize the flow of clients admitting some losses, that are less in case of uniform distribution for the service times (see consideration about coefficient of variation), but in general adding the finite capacity of the waiting line keeps similar the behaviour between the two setting .
The best configuration overall among those tested is the queue with service times uniformly distributed, where the policy in case of idle servers is to choose always the fastest one; this guarantees less average clients in the queue, less delay for them and in case of finite capacity, less losses. Once again the credit can be recognized in the smaller variance both in policy and in distribution.