15.MM1 and MM2 queues

print(f'p(N < 1) = {pNlei}')# Kendall's Notation
A special notation, composed by between 3 and 6 terms, is used to identify the behavior of simple systems characterized by a single queue service center.

$$A/S/c[/k[/p]][/Q]$$

Square brakets represents optionals values.

A/S

Two letters that characterise the arrival and service process of the system.

- M: exponential / Poisson (Markov)
- D: Deterministic
- E_k: Erlang with k stages
- G: General

c

Represent the number of service. Represent the number of jobs that can be done in parallel

[/k]

This represent the maximum capacity of the system. Whenever there is another arrival after the maximum number it will be blocked or lost.

[/p]

If the population from which the jobs that enter the system are taken is finite, its size can be specified as a fifth parameter. In this way, the arrival rate can be scaled proportionally. This become important if the population outside the system is comparable with the one inside. This component is used very rarely.

[/Q]

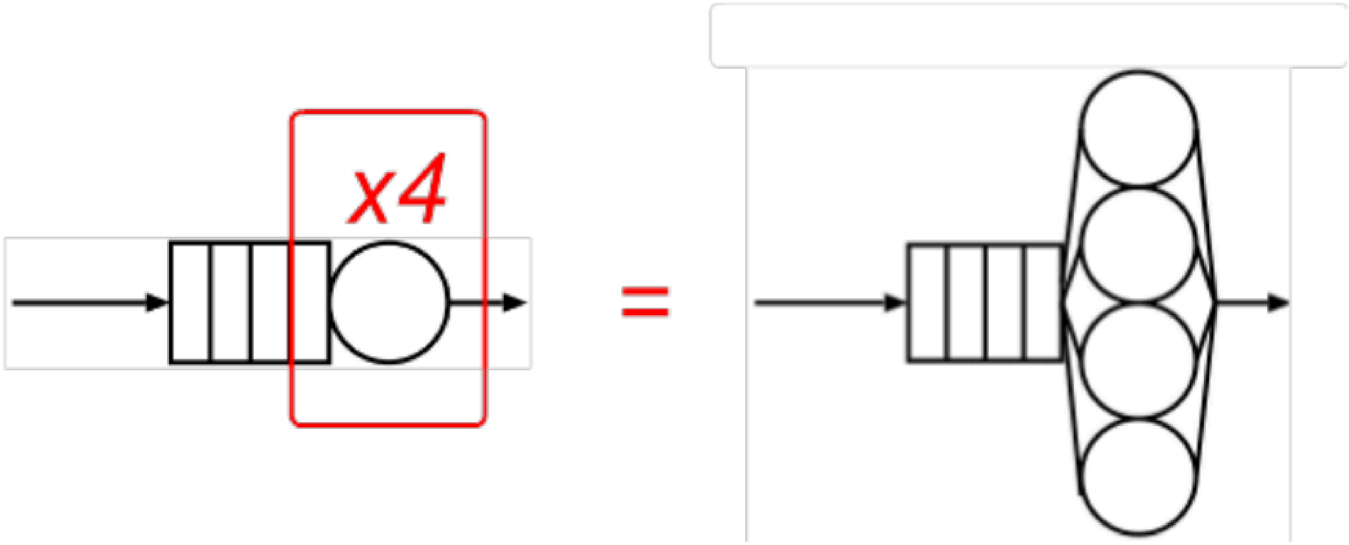
The final part indicates the service discipline: i.e. First Come First Served, Last Come First Served, etc…
The service discipline is generally identified by an acronym, which will be presented later in the course:

- FIFO or FCFS: First in first out
- LIFO or LCFS: Last in first out

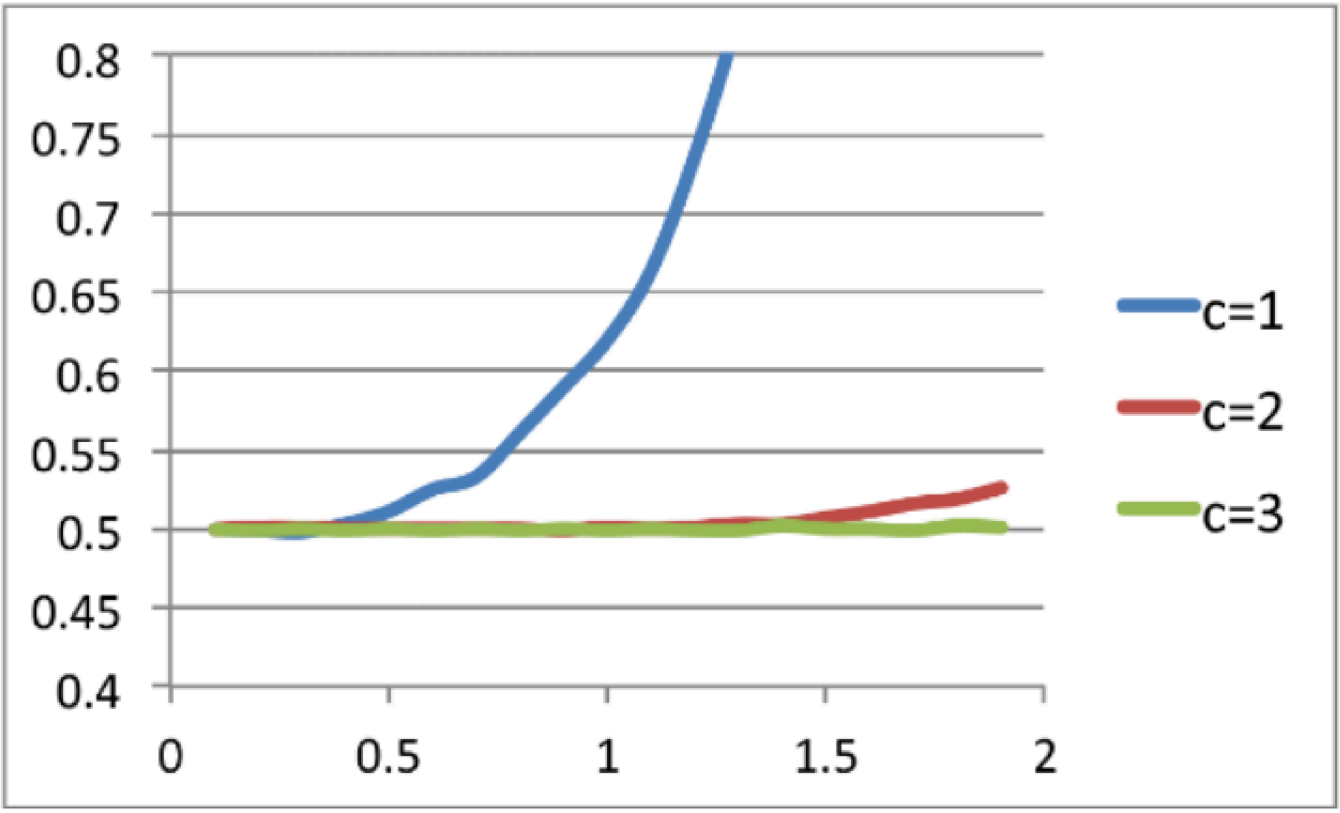
- SIRO: service in random order
- PS: process sharing
  Some examples of single queuing system specifications using Kendall's notation are the following:
- M/M/1: Poisson arrival, exponential service time, one server, infinite queue and infinite population.
- M/G/1/K: Poisson arrival, general service time, one server, queue capacity for K jobs and infinite population
- M/M/2/5/20: Poisson arrival, exponential service time, two servers, maximum 5 jobs in the system from a population of 20.
- D/G/1/LCFS: Deterministic arrival, general service time one server, infinite queue and population, last-in first-out (last-come first-served).

Multiple Servers

These are servers with multiple parallel server. This means we can have a single queue and multiple identical service on different servers. The servers are equivalent so they take the same time to do the same request. When a server ends its work, usually, the first job in the queue enter the server.



In the following example, we compare the response time of a server with exponential service time (S = 0.5 s.), and a deterministic inter-arrival, for different number of servers and different arrival rates.

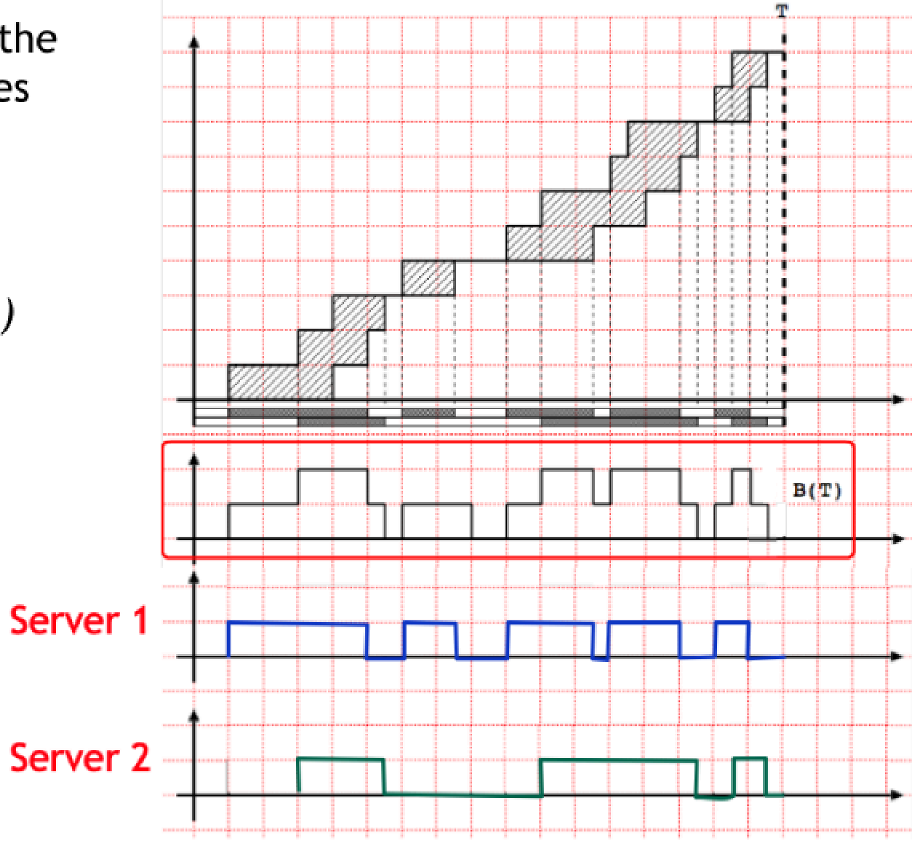


When considering multiple servers, the concept of utilization requires extra care. In particular, the utilization law must be used in the proper way.

The busy time is the sum of the total busy time of all the servers. We can also split the busy time considering all the servers with their busy time. The multiple server busy time is equal to how many servers are working.

B(T) can be defined as the sum of the busy times of the two servers.

$B(T) = min(2, A(T)\text{-}C(T))$



For the system the busy time is:

$$B(T) = \int_0^T min(2,\ \text{A(T) - C(T)})dt$$

In multiple servers models, the total utilization is a number between 0 and the number of servers c.

$$0 \leq U = \frac{B(T)}{T} \leq c$$

The average utilization focuses on the utilization of one of the c servers. It is a number between 0 and 1, and it is obtained dividing the total utilization by the number of servers c.

$$\bar{U} = \frac{U}{c} = \frac{B(T)}{cT},\ 0 \leq \bar{U} \leq 1$$

A system is stable if $\bar{U} \leq 1$ then $\lambda S < c$.
The utilization law assumes a different form when considering the average utilization.

$$U = XS, \bar{U} = \frac{XS}{c}$$

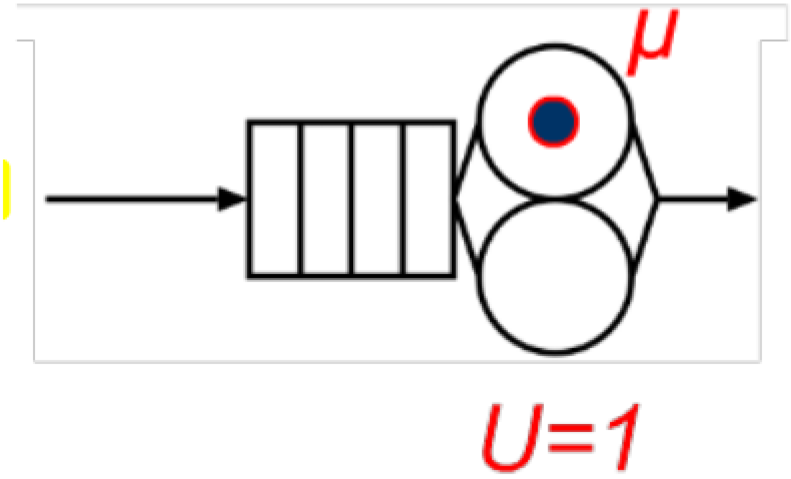
This allows us to determine a new bound to the maximum arrival rate that a c server system is able to handle before becoming unstable.

$$\lambda \leq \frac{c}{S}, S \leq \frac{c}{\lambda}, E[Y] \geq \frac{S}{c}$$

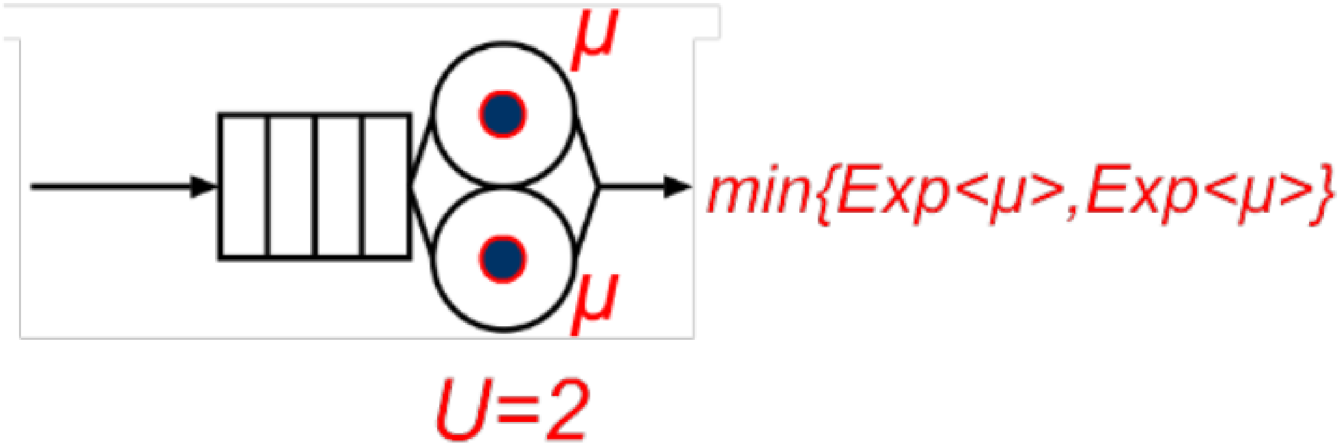
Here Y is the inter-arrival time distribution. As we can see, a c server queue can handle a workload that is c times higher than the one of a single server.

M/M/2

The M/M/2 queue is a station with two servers, where both inter-arrival times and service times are exponentially distributed. If there is just one costumer, it is immediately served by either of the servers. In this case the total utilization is 1 (that is, B(T) = 1), since only one server is not idle. The system will become empty after an exponentially distributed time, with rate μ.



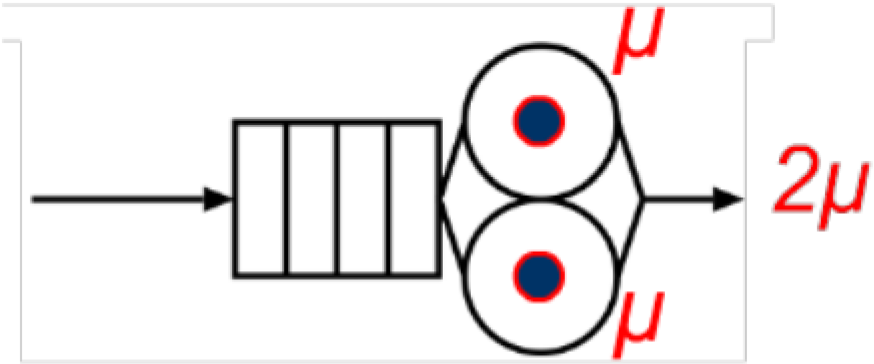
If two jobs are in the station, both servers are working. In this case the total utilization is 2 (that is, B(T) = 2), and the system decreases the number of jobs when the first of the two costumers being served at rate μ finishes. The time after which the first job leaves the system thus corresponds to the minimum of two exponential distributions of rate μ.



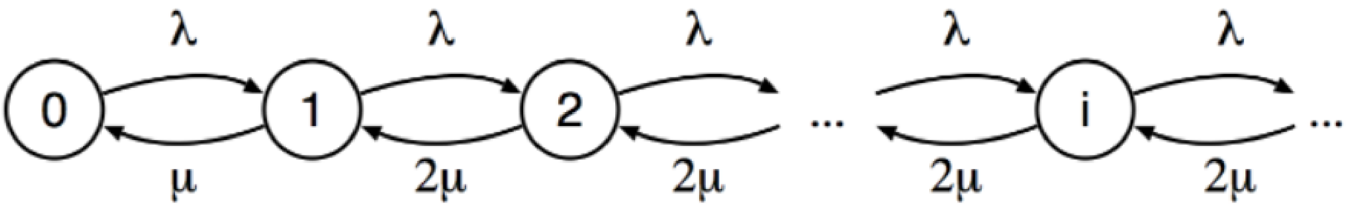
The minimum of a set of exponential distributions is still exponentially distributed, with the sum of the rates as parameter.

$$f_{min\{Exp_{<\lambda_1>},\dots,Exp_{<\lambda_n>}\}}(t) = f_{Exp<\lambda_1+\cdots+\lambda_n>}(t)$$

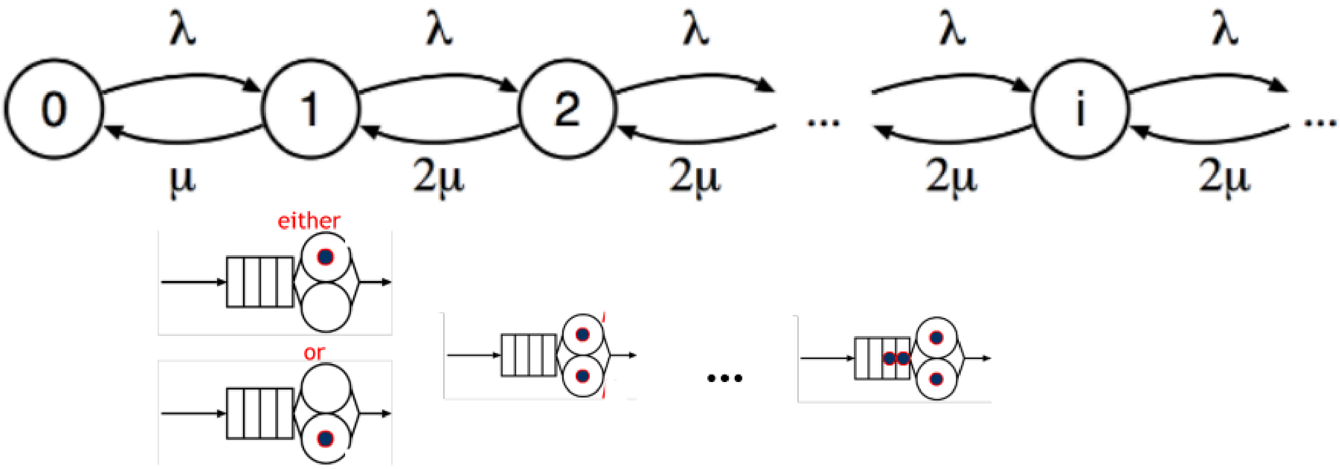
This means that the first job will leave the system after an exponentially distributed time of rate 2μ.



If more than two jobs are in the system, queue starts forming. In this case the system still has utilization equal to 2, and the number of jobs in the system decreases after an exponentially distributed time with rate 2μ.
The birth-death process that characterizes M/M/2 models, has the transition rate from state 1 to state 0 equal to μ (since in state 1 there is just one job in the system). The rate from state i to state i-1, for all other i > 1, is equal to 2μ since in these cases there are two jobs being served. Arrival rate is constant for all the states and equal to λ.



Note also that, thanks to the exponential assumption and the equivalence of the servers, there is no need to consider whether a job is being served by the first or by the second server. This greatly simplifies the state space, allowing to consider just the count of jobs in the system to fully describe its state.



Since the M/M/2 queue is a birth-death process, it can be analyzed with the formula previously seen:

$$\pi_n = \pi_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i}, \pi_n = \pi_0 \frac{\lambda}{\mu}(\frac{\lambda}{2\mu})^{n-1} = 2\pi_0(\frac{\lambda}{2\mu})^n \text{ for n} \leq 1$$

The empty system probability can be computed as:

$$\pi_0 = \frac{2\mu - \lambda}{2\mu + \lambda}$$

The utilization derived as:

$$U = \frac{2\mu - \lambda}{2\mu + \lambda}\frac{\lambda}{\mu}\frac{2\mu + \lambda}{2\mu - \lambda} = \frac{\lambda}{\mu} = \lambda D$$

Average utilization:

$$\bar{U} = \frac{U}{2} = \frac{\lambda}{2\mu}$$

Let us define ρ as:

$$\rho = \bar{U} = \frac{\lambda D}{2}$$

We then have:

$$\pi_0 = \frac{1-\rho}{1+\rho}, \pi_n = 2\frac{1-\rho}{1+\rho}\rho^n \text{ for n} \geq 1$$

The average number of jobs in the system can be derived as:

$$N = \frac{2\rho}{1-\rho^2}$$

Using Little's law, we can express the average response time:

$$R = \frac{N}{X} = \frac{D}{1-\rho^2}$$

We can also compute the average time spent in the queue:

$$\theta = R - D = \frac{\rho^2 D}{1-\rho^2}$$