POLITECNICO DI MILANO

Performance Evaluation and Applications





The importance of workload characterization

Motivating example

In order to improve its performance, a supermarket manager has decided to monitor her costumers, writing the time when they join the queue, the one when they leave, and the duration of the observation. From these she has determined:

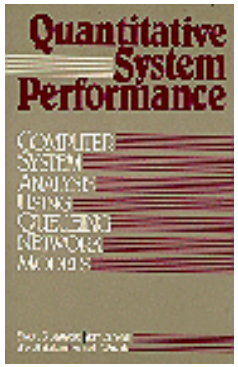
$T = 8h, \quad B = 6h{:}30m, \quad A = C = 780, \quad W = 123h{:}30m$

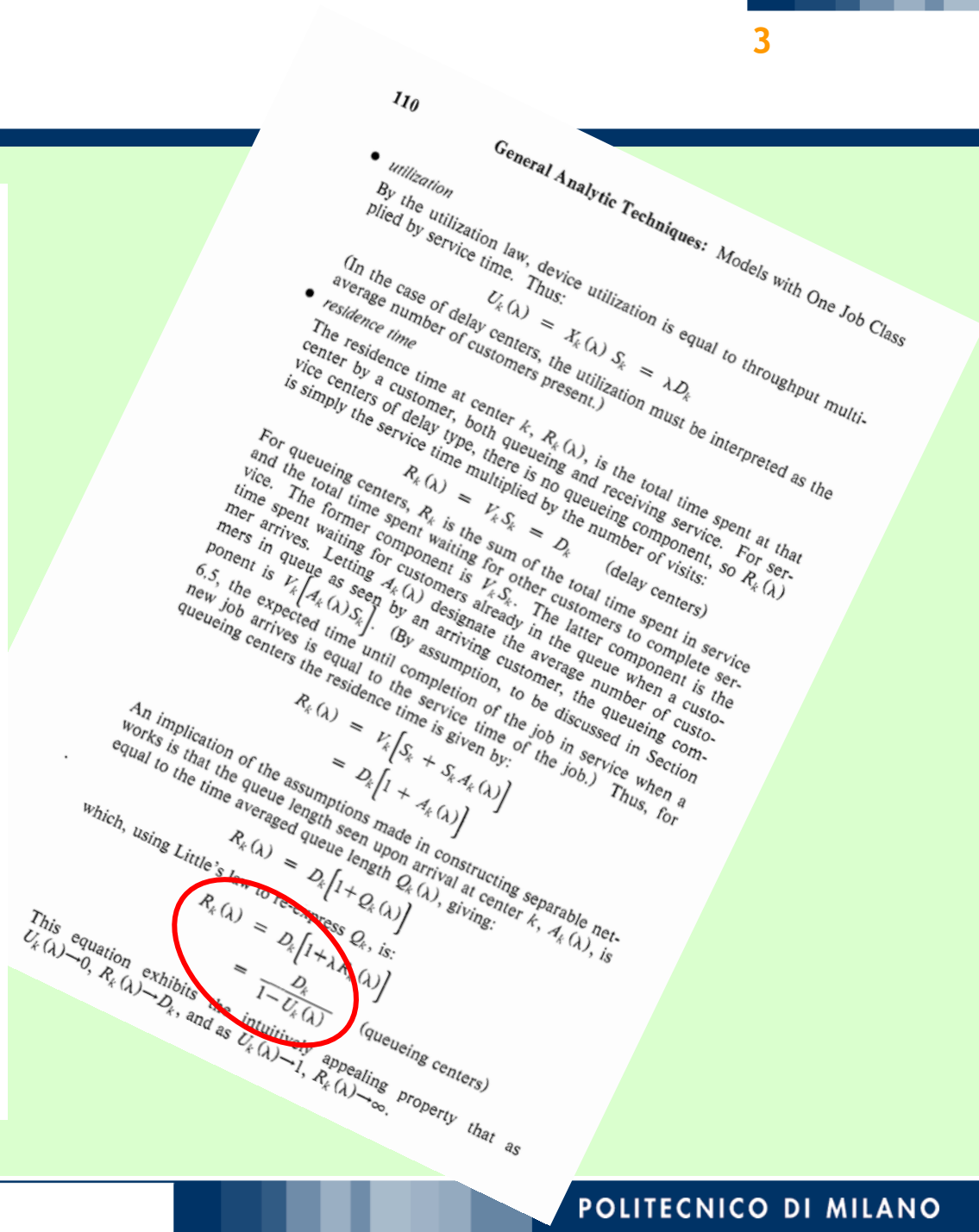
Motivating example

She first used the basic laws of performance, and she determined:

$\lambda = 97.5\ c/h,$
$S = 30s, \quad U = 0.8125,$
$R = 9m{:}30s$

Searching on a book, she found that $R = S / (1 - U)$




110

General Analytic Techniques: Models with One Job Class

• *utilization*

By the utilization law, device utilization is equal to throughput multiplied by service time. Thus:

$$U_k(\lambda) = X_k(\lambda) S_k = \lambda D_k$$

(In the case of delay centers, the utilization must be interpreted as the average number of customers present.)

• *residence time*

The residence time at center k, $R_k(\lambda)$, is the total time spent at that center by a customer, both queueing and receiving service. For service centers of delay type, there is no queueing component, so $R_k(\lambda)$ is simply the service time multiplied by the number of visits:

$$R_k(\lambda) = V_k S_k = D_k \qquad \text{(delay centers)}$$

For queueing centers, R_k is the sum of the total time spent in service and the total time spent waiting for other customers to complete service. The total time spent waiting for customers already in the queue when a customer arrives. Letting $A_k(\lambda)$ designate the average number of customers in queue as seen by an arriving customer, the queueing component is $V_k[A_k(\lambda) S_k]$. (By assumption, to be discussed in Section 6.5, the expected time until completion of the job in service when a new job arrives is equal to the service time of the job.) Thus, for queueing centers the residence time is given by:

$$R_k(\lambda) = V_k[S_k + S_k A_k(\lambda)]$$
$$= D_k[1 + A_k(\lambda)]$$

An implication of the assumptions made in constructing separable networks is that the queue length seen upon arrival at center k, $A_k(\lambda)$, is equal to the time averaged queue length $Q_k(\lambda)$, giving:

$$R_k(\lambda) = D_k[1 + Q_k(\lambda)]$$

which, using Little's law to re-express Q_k, is:

$$R_k(\lambda) = D_k[1 + \lambda R_k(\lambda)]$$
$$= \frac{D_k}{1 - U_k(\lambda)} \qquad \text{(queueing centers)}$$

This equation exhibits the intuitively appealing property that as $U_k(\lambda) \to 0$, $R_k(\lambda) \to D_k$, and as $U_k(\lambda) \to 1$, $R_k(\lambda) \to \infty$.

Motivating example

$\lambda = 97.5\ c/h, \quad S = 30s, \quad U = 0.8125, \quad R = 9m{:}30s$

$$R = S\ /\ (1 - U)$$

She applied that formula, and she obtained:

$$R = S/(1\text{-}U) = 0.5\ /\ (1 - 0.8125) = 2m{:}40s$$

Why results did not match? What is the mistake she did?

Goals of performance evaluation

As briefly introduced, the two main performance indices that fully characterize the perception of good behavior of a system are:

- **Throughput** – generally the system owners want to maximize it, since most of the time revenue is directly proportional to throughput
- **Response time** – in this case, the system users want to minimize it, since they have no time to waste and other things to do!

The two quantities however are *counterposed*:

- A higher throughput causes a higher response time
- A shorter response time, comes at the expense of a lower throughput.

Goals of performance evaluation

Initially, *let us focus on open systems*, where the throughput is equal to the arrival rate. Our goal is:

- Given the description of the arrivals to a system
- Given the service times of the system
  - **Determine the response time**

In this way, we can improve the system by either increase the maximum arrival rate (throughput) it can handle, or reduce the response time requests have to wait in the following way:

- Reducing the service times of the components, by replacing them with faster ones.
- Increasing the parallelism, so to have more components sharing the workload, and thus facing a smaller arrival rate.

Goals of performance evaluation

- Given the description of the arrivals to a system
- Given the service times of the system
  - **Determine the response time**

This step, however, is absolutely ***NOT TRIVIAL***

And it will be the main focus of this course!

Basic of Workload characterization

To predict the performance of a system, we must first be able to characterize its workload:

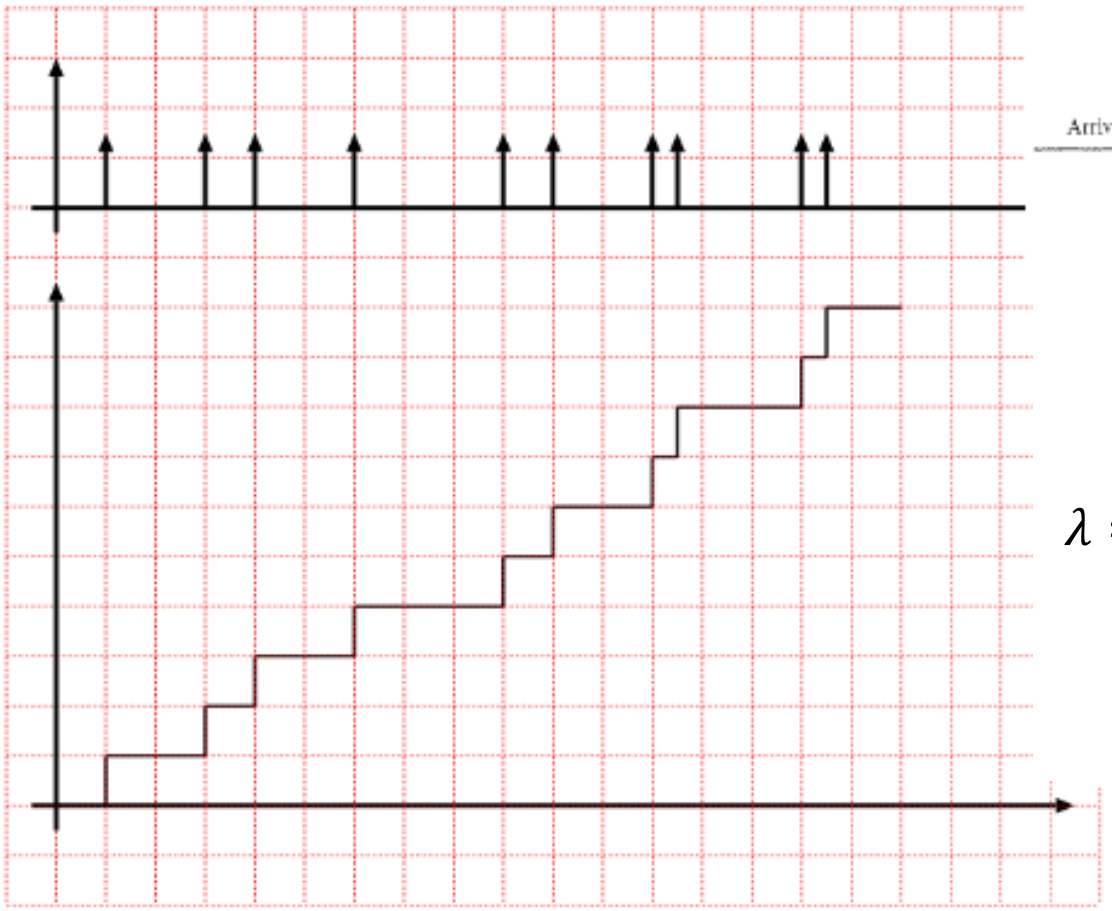
- The frequency and pattern at which things to do arrives to the system (Arrivals).
- The time required and the pattern with which requests are served (Services)

Arrivals

Arrivals describe the way in which jobs enter the system.

Arrivals

As we have seen, the *arrival rate* measures the speed at which jobs enter a queue. We have also seen that the arrival rate is closely related to *inter-arrival times* a_i.

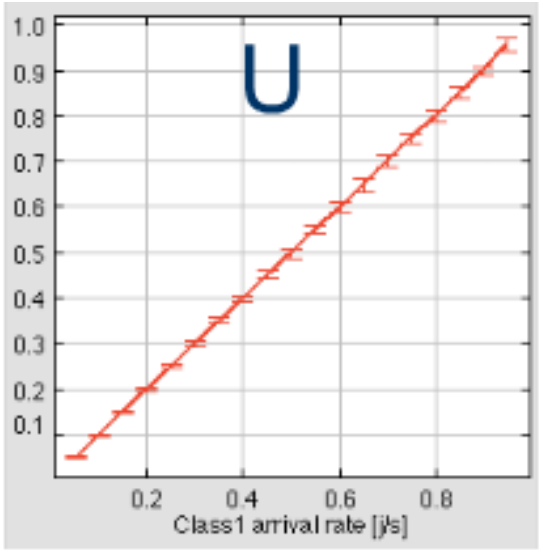


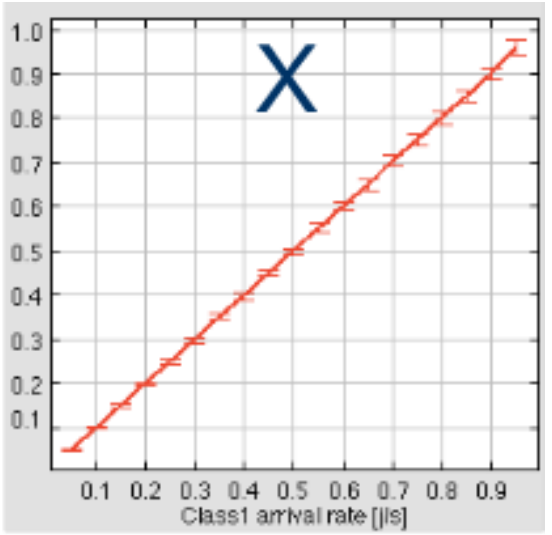
$$\lambda = \lim_{T \to \infty} \frac{A(T)}{T} = \lim_{A \to \infty} \frac{A}{\sum_{i=1}^{A} a_i}$$

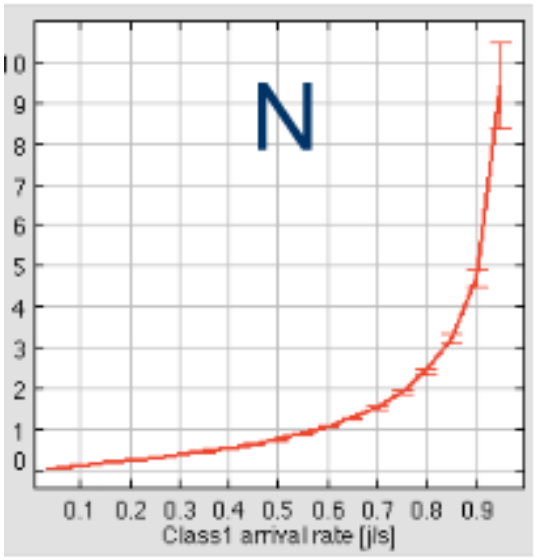
Arrivals

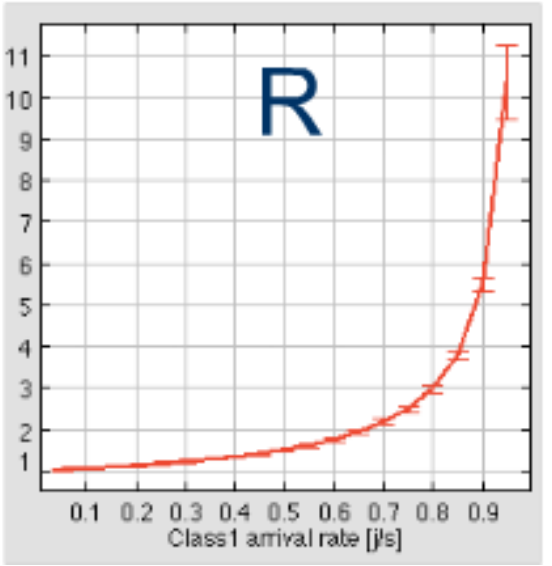
In open models, the arrival rate λ influences all the basic performance metrics: utilization, throughput, average number of jobs in the queue and average response time.

In the figures, the service time is constant *D=1* s. The arrival rate varies between *0.05 job/s*. to *0.95 job/s*.





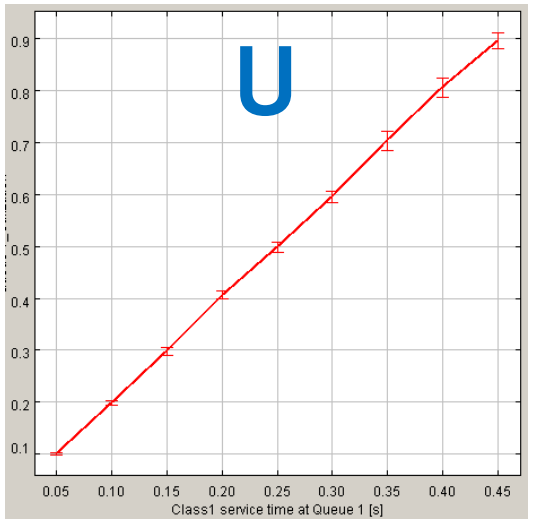


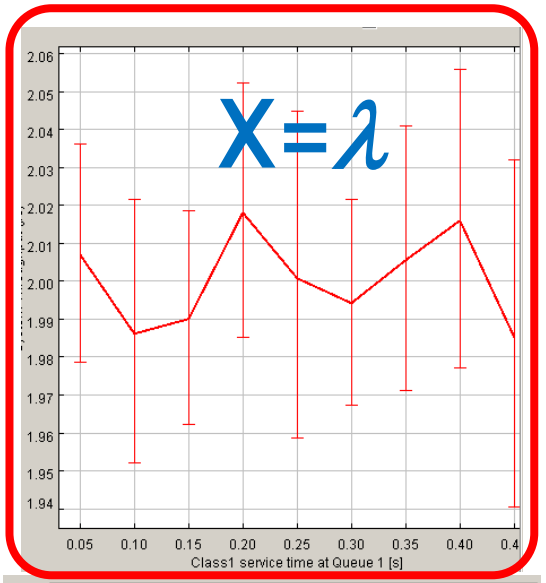


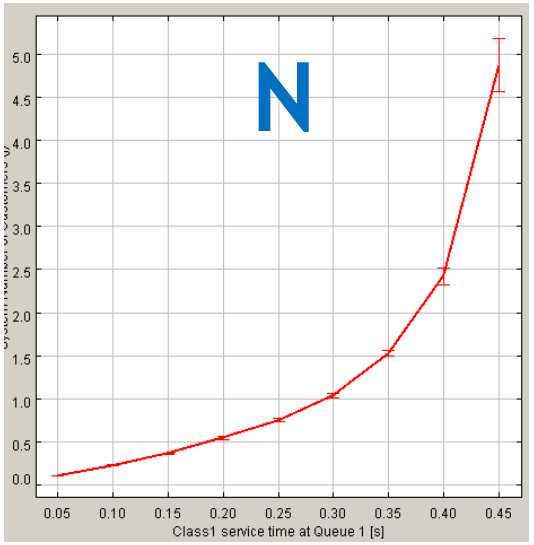
Inter-arrival time distribution

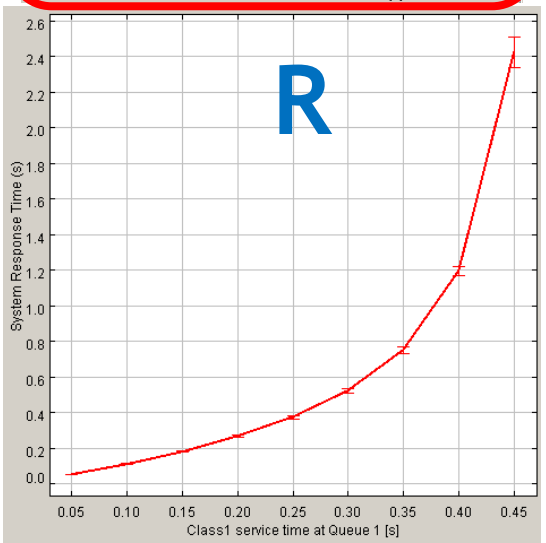
Even with a constant service time, the arrival *rate* is not the only parameter that influences the performances of a queue.

In the following, we will focus on different systems, all characterized by the same arrival rate (and throughput) of *2 job/s*.







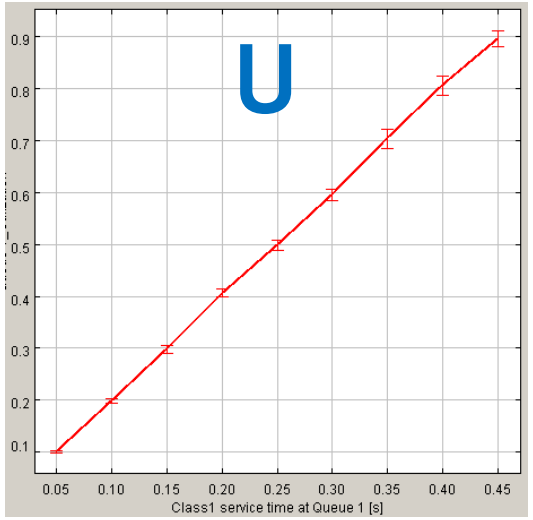


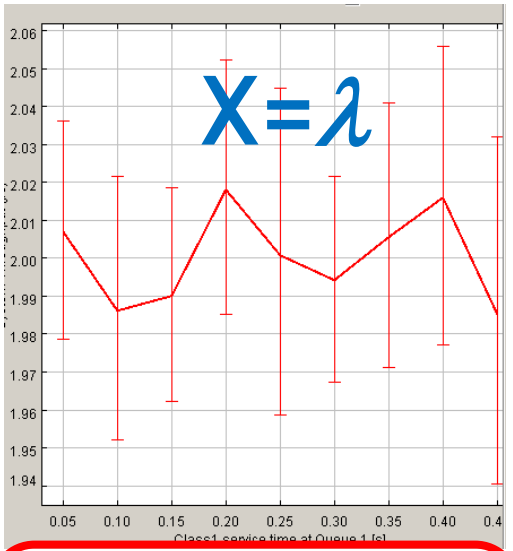
Inter-arrival time distribution

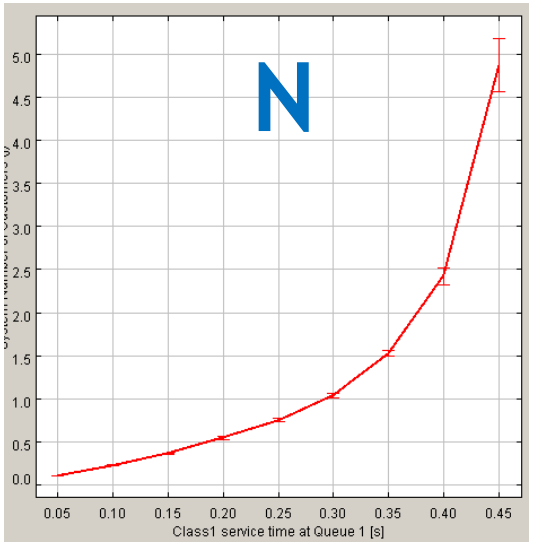
The service will be considered deterministic: every job always takes exactly the same service time.

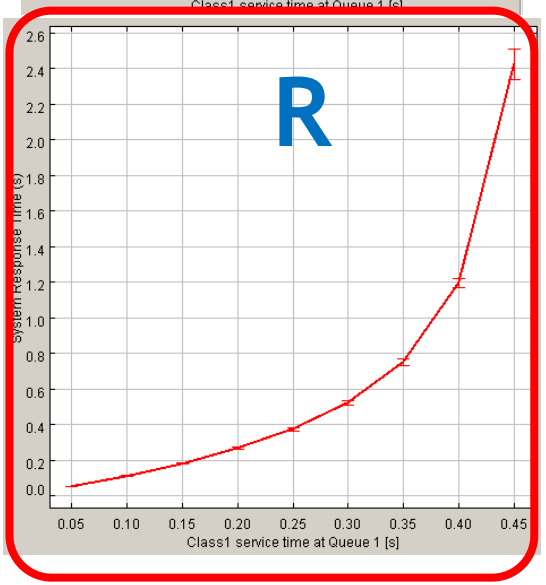
We will study the evolution of the *system response time*, for a service time S that varies from *0.05 s*. to *0.45 s*.

For the utilization law, this means that the utilization will always be between *10%* and *90%*, depending on the service time.



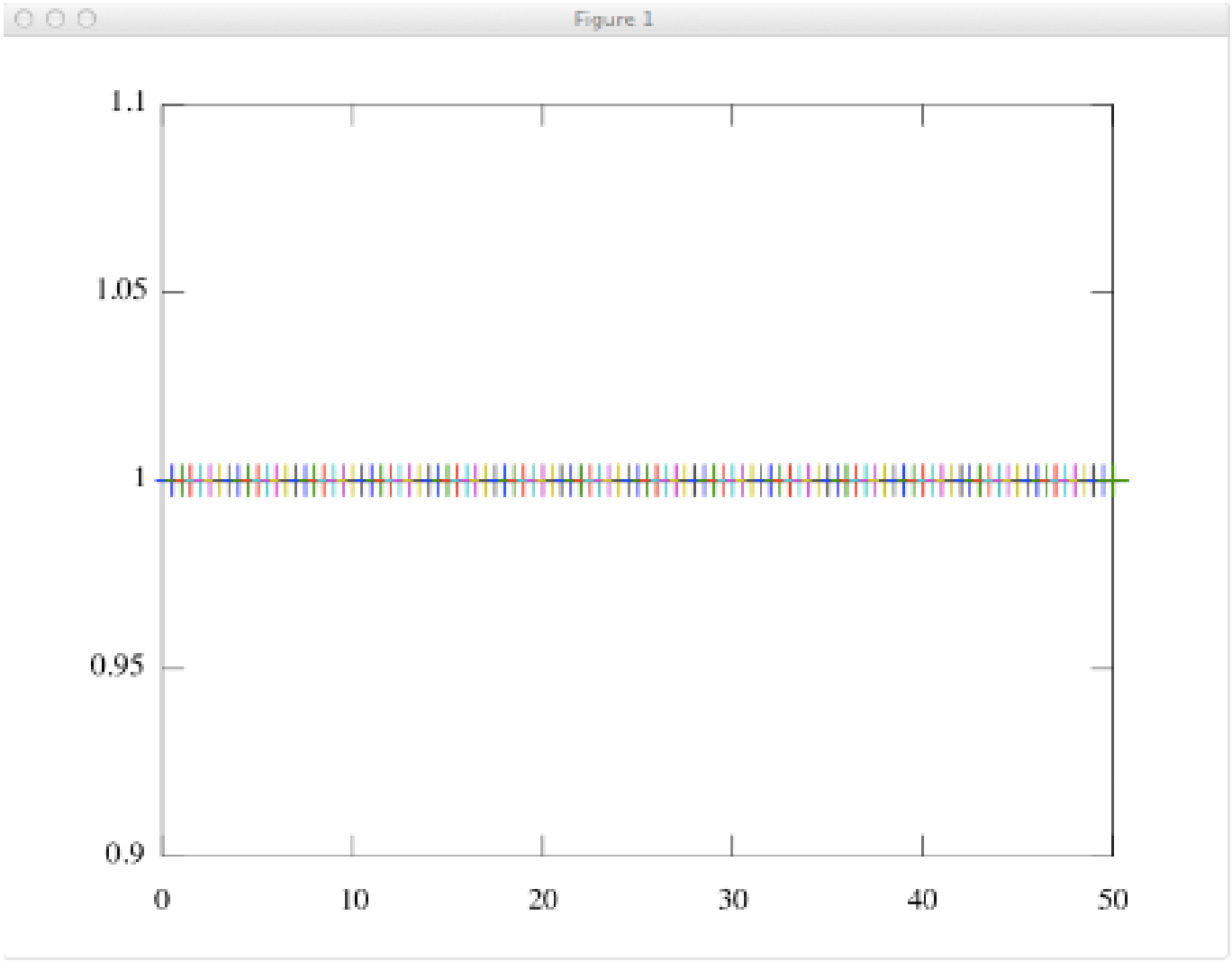






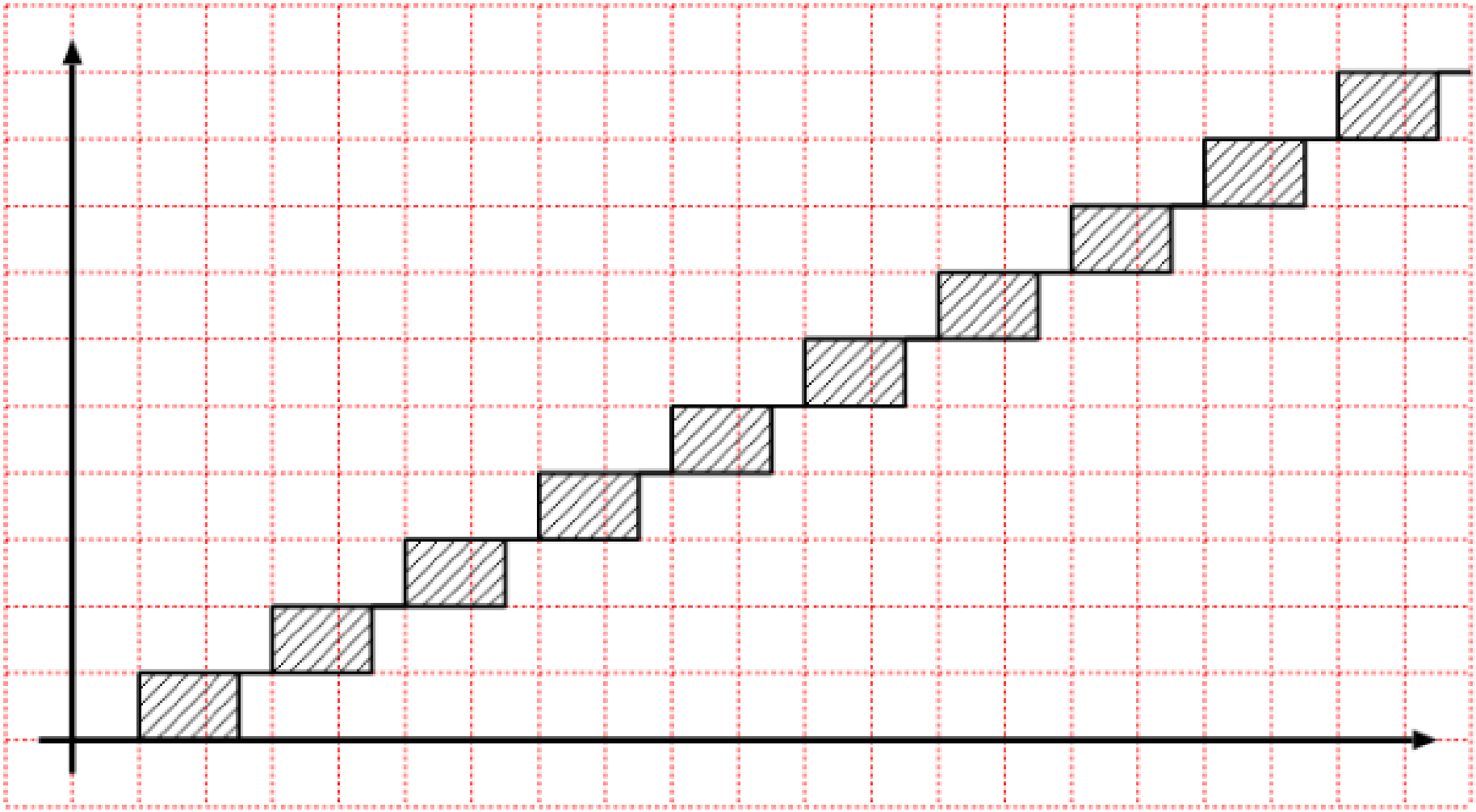
Deterministic inter-arrival time

Let us first consider new jobs arriving exactly every 0.5s.



Deterministic inter-arrival time

Both inter-arrival times and services are deterministic.



Deterministic inter-arrival time

Jobs never queue, and we can easily determine all the required performance indices:

$$W = B$$

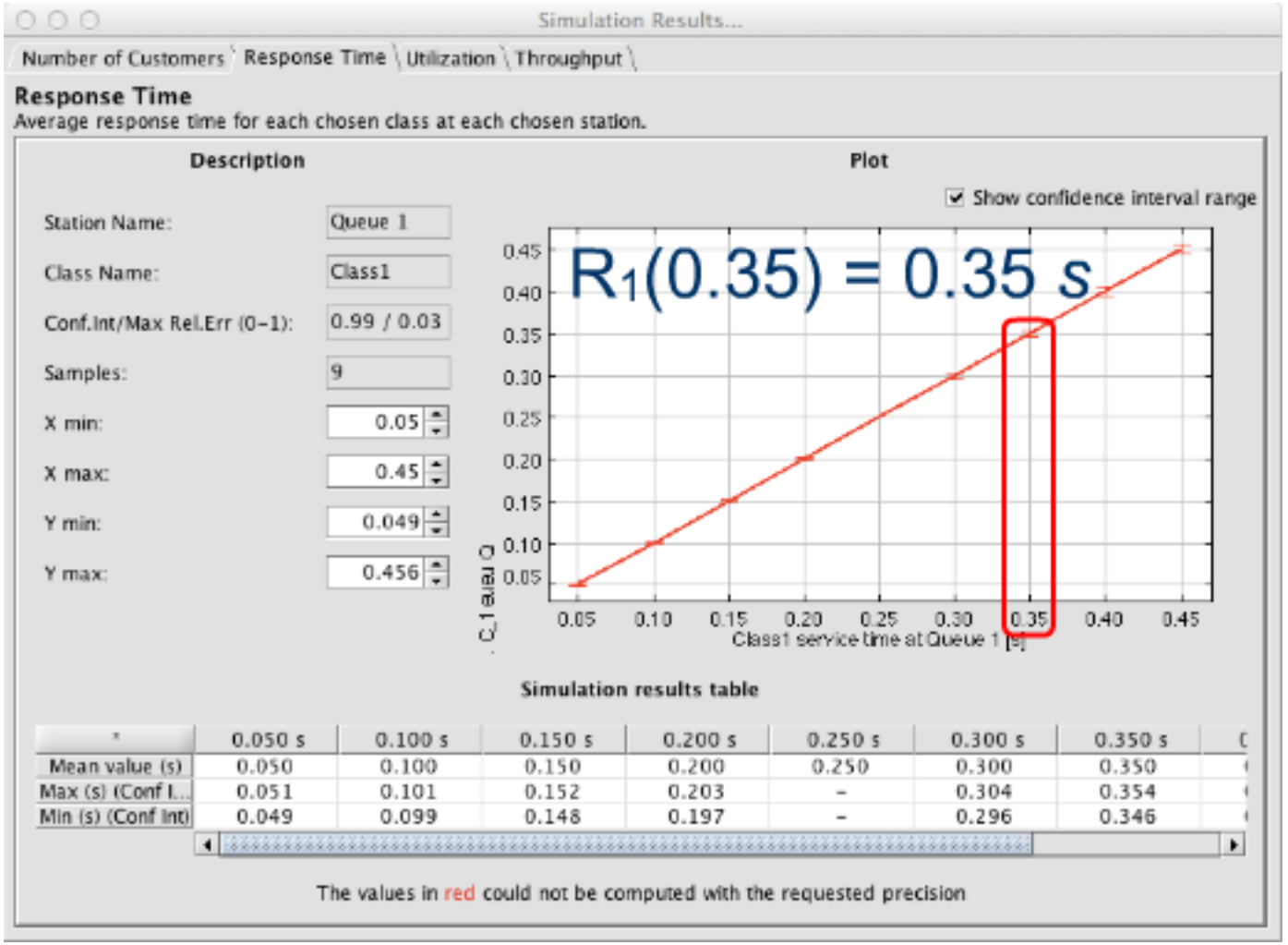
$$U = \lambda \cdot S \qquad N = \lambda \cdot S$$

$$X = \lambda \qquad R = S$$

since $\frac{W}{C} = \frac{B}{C}$

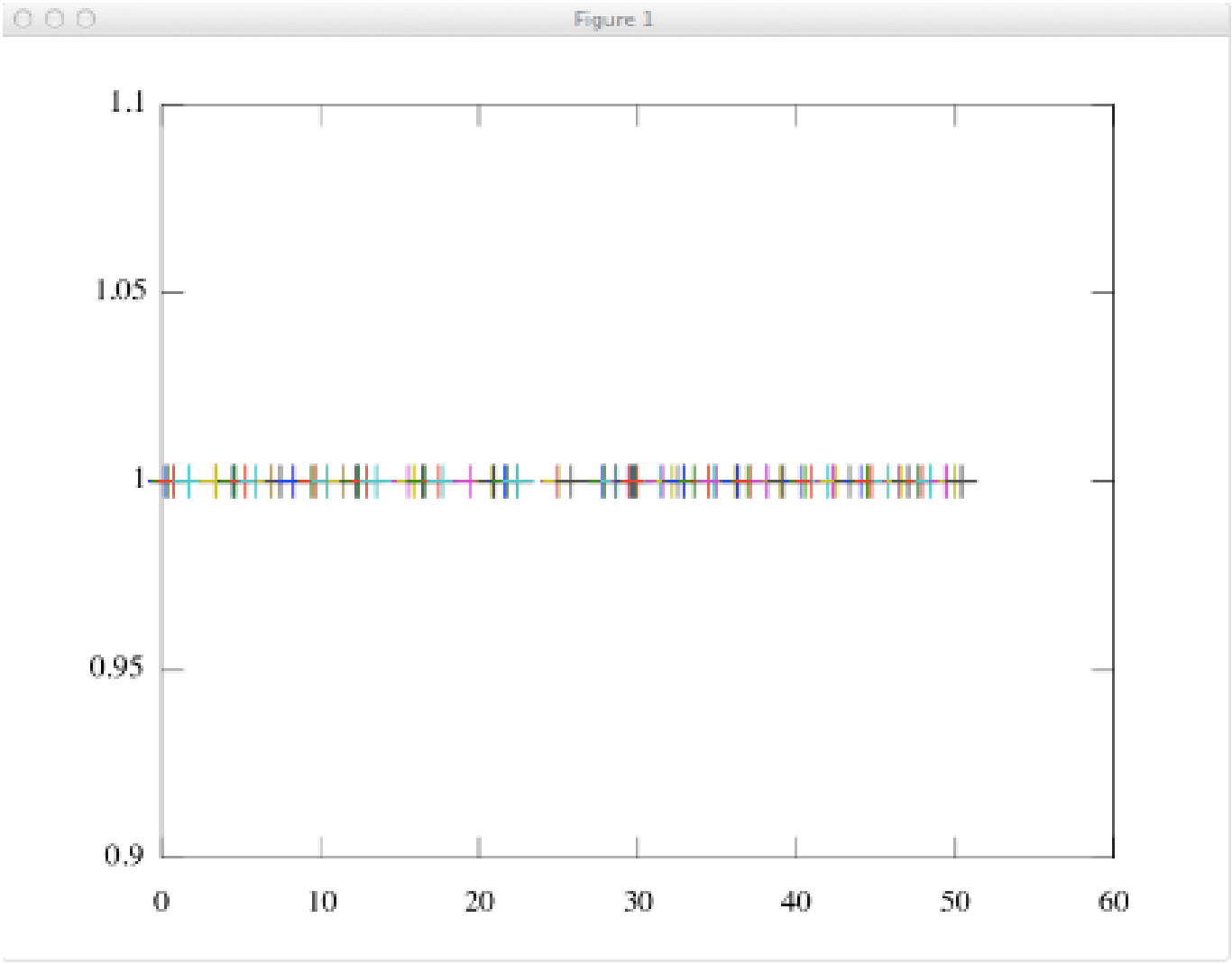
Deterministic inter-arrival time

If we plot the response time for the different service times, we see the following curve:



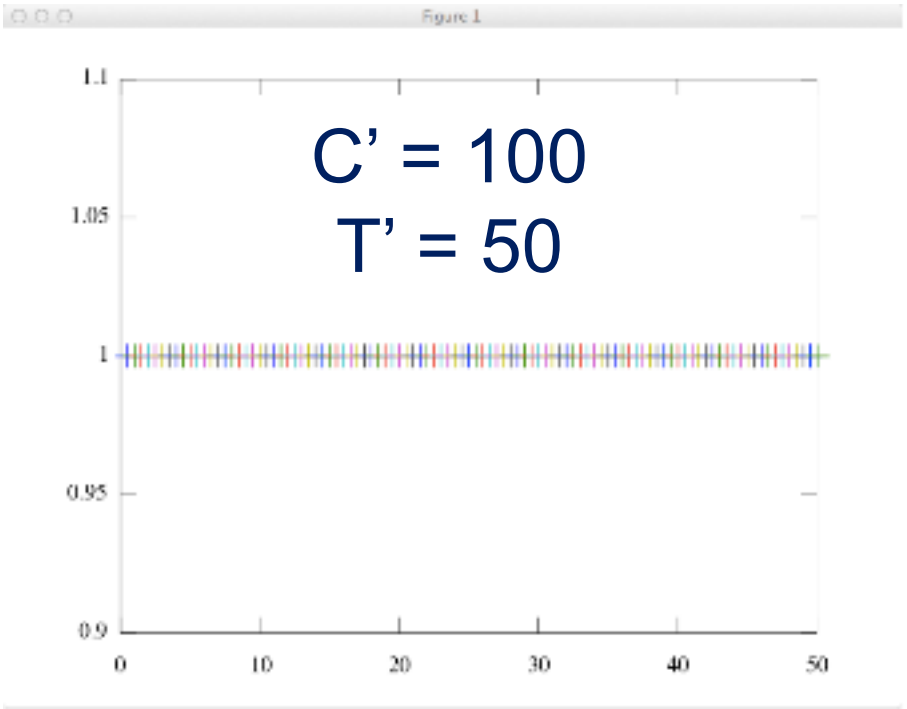
Exponential inter-arrival time

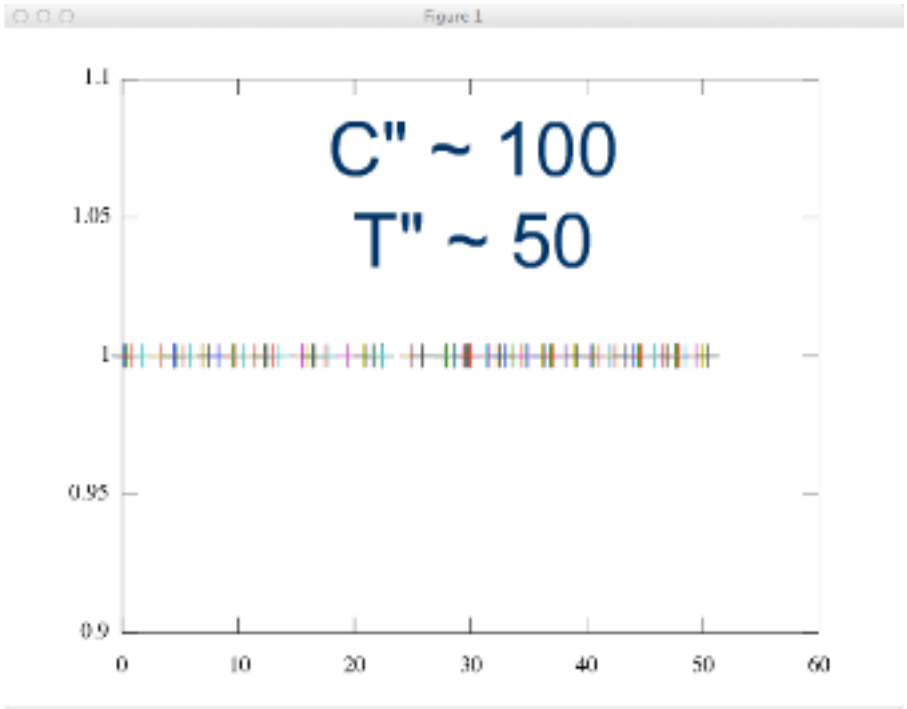
Let us now consider a random arrival.



Exponential inter-arrival time

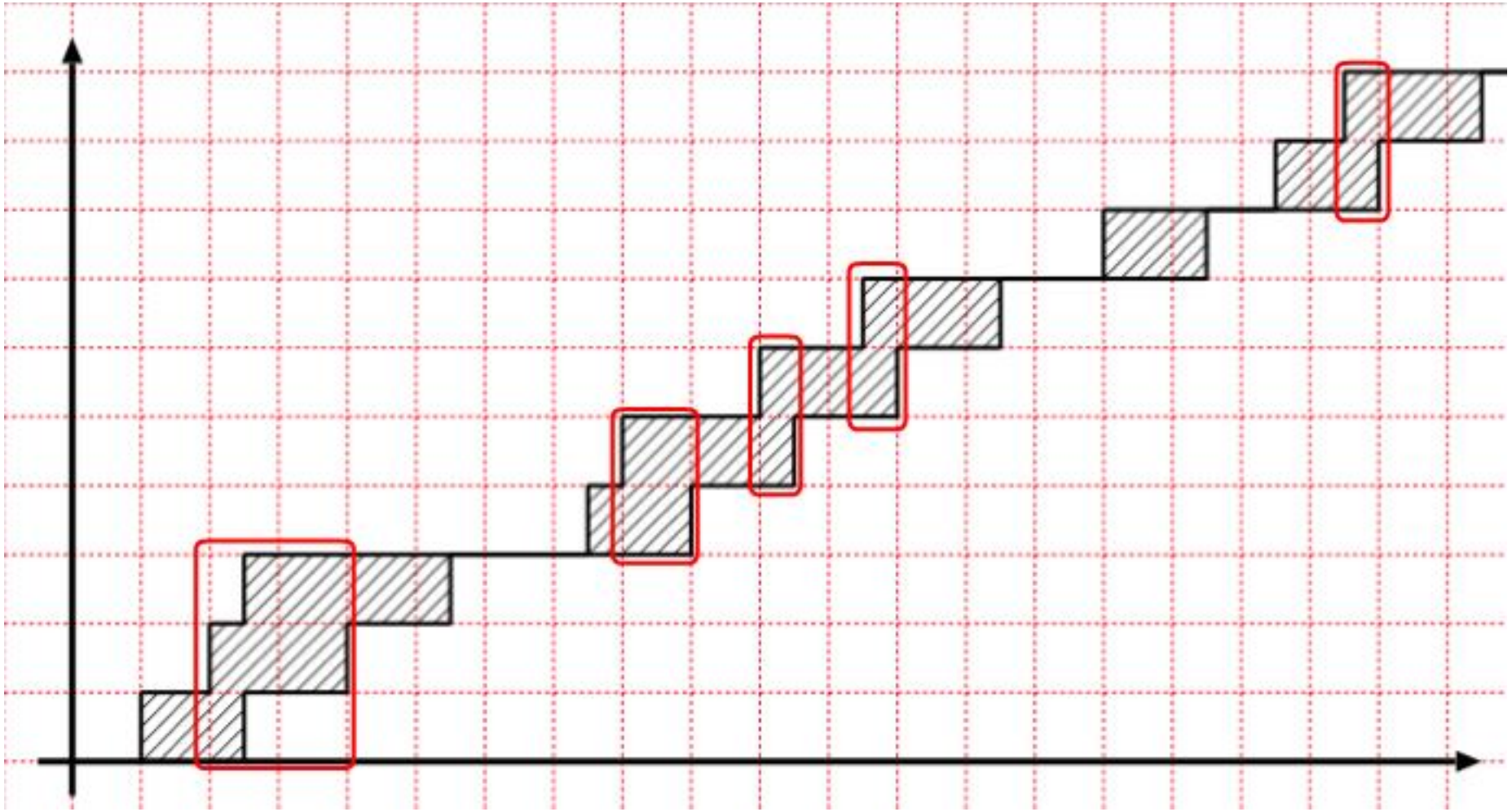
The arrival rate λ (in terms *of* $C(T) / T$) is still $\lambda = 2$ job / s.





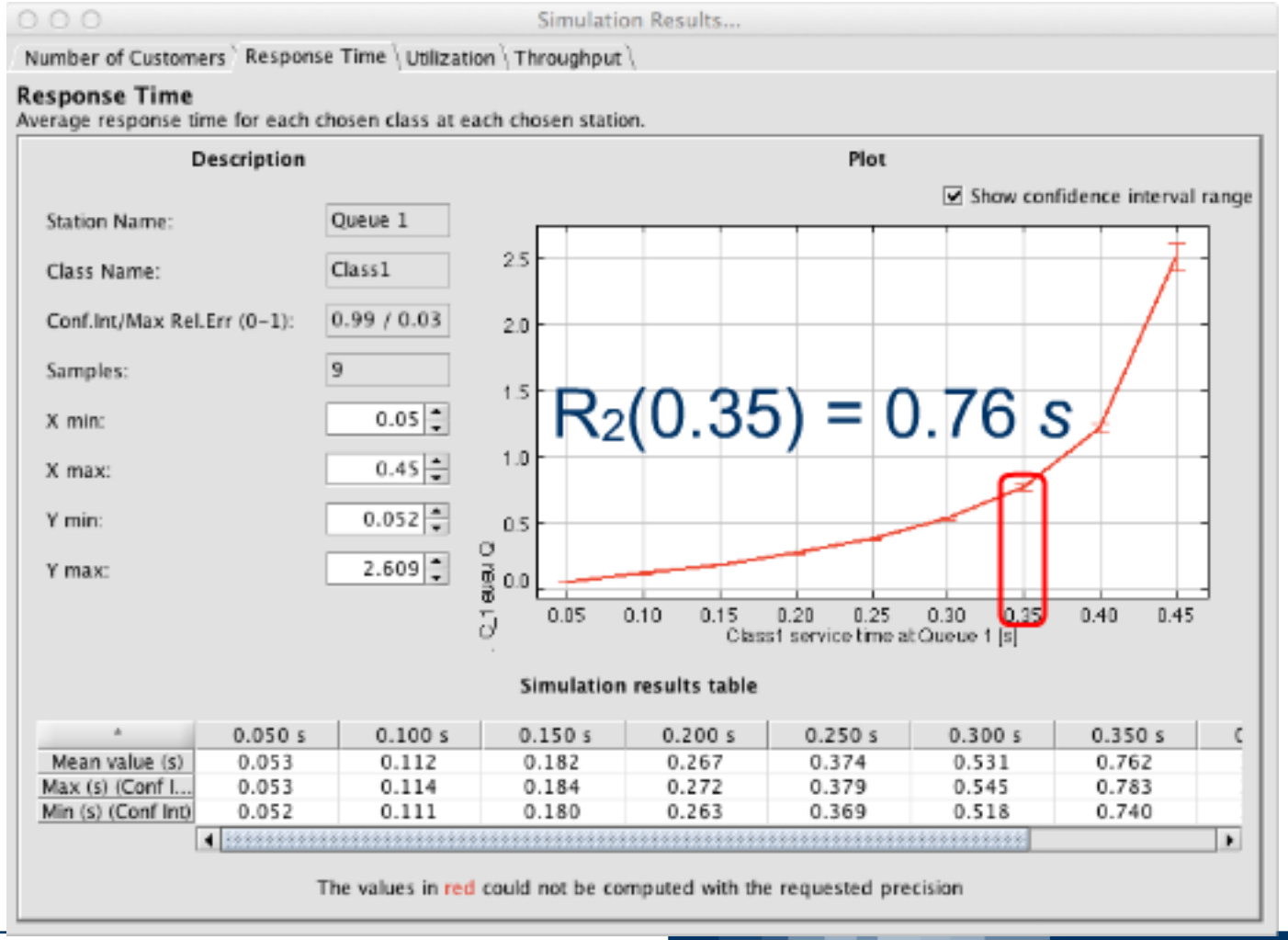
Exponential inter-arrival time

The non-determinism in the arrival creates queuing.



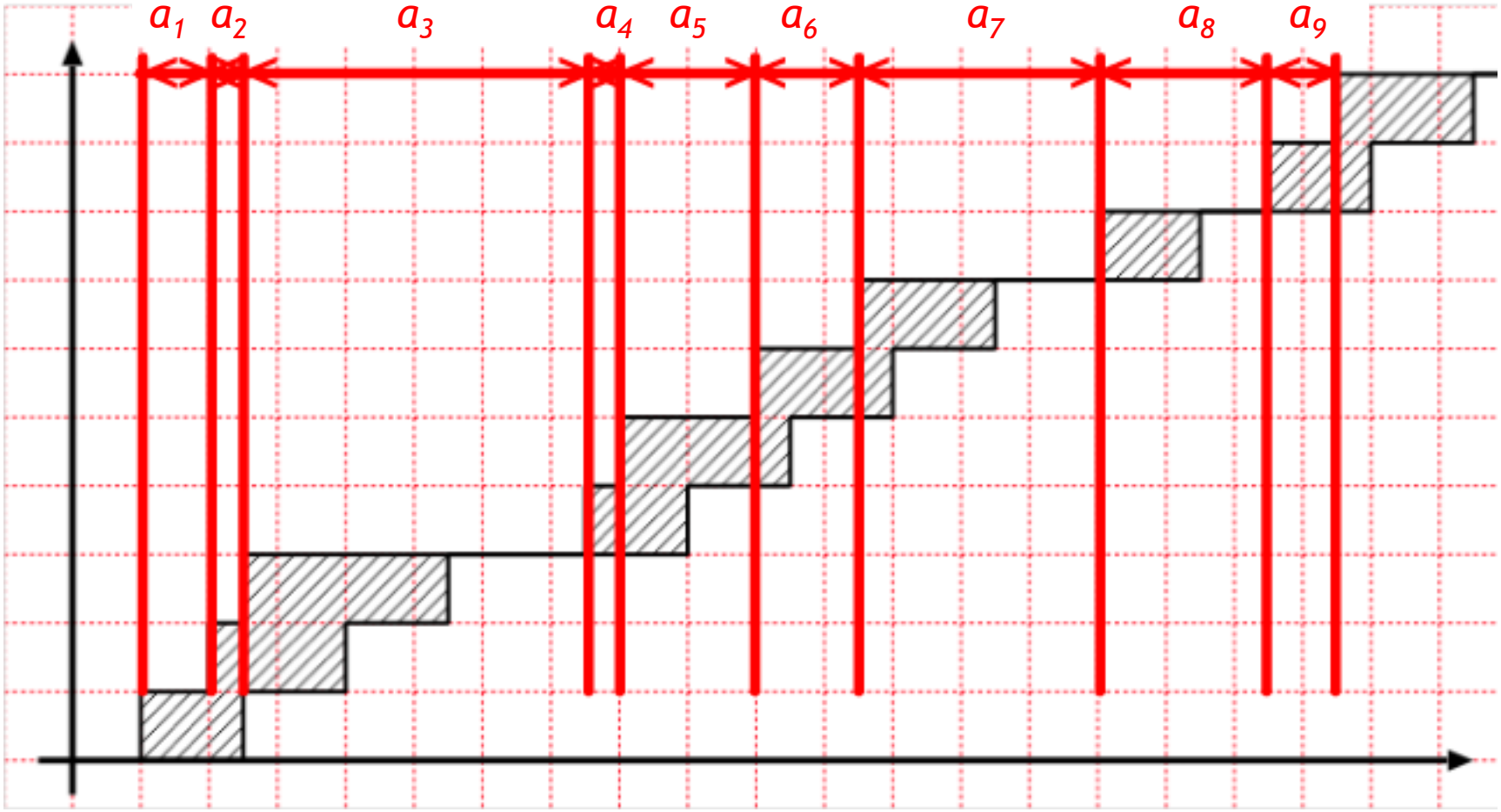
Exponential inter-arrival time

The response time now varies in a non-linear way as a function of the service time.



Inter-arrival time distribution

The randomness of the arrival can be described by *the inter-arrival time distribution*: the distribution of the time that passes between an arrival and the following one.

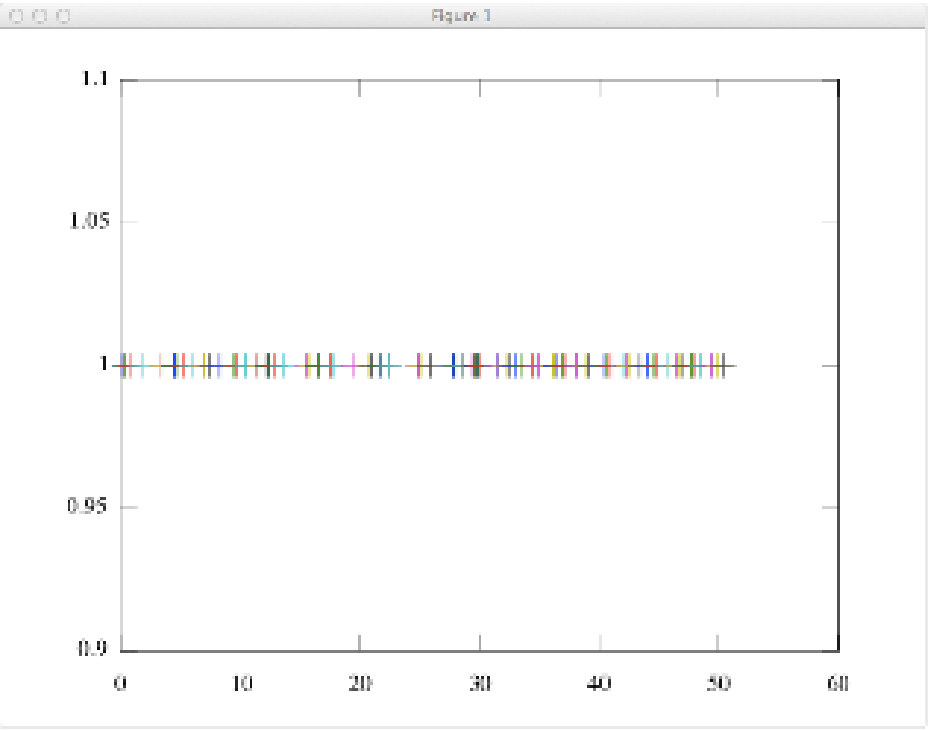


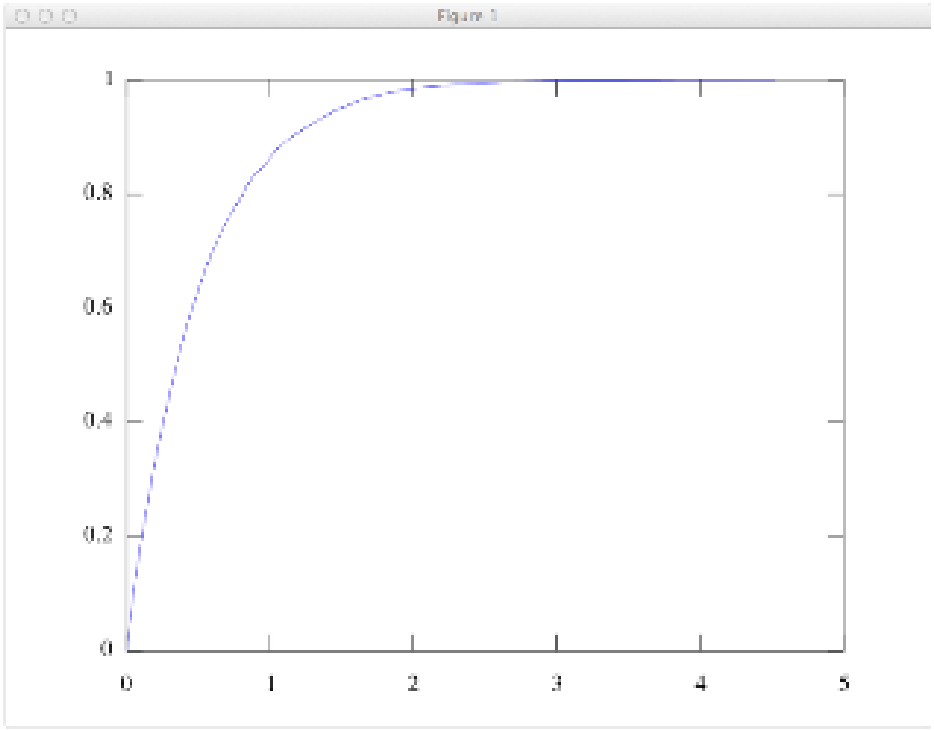
Exponential inter-arrival time

In this case, the inter-arrival time is *exponentially distributed*.
This input is called a *Poisson Process*.

$$a_i \sim Exp(2)$$

$$F_{Exp(2)}(t) = 1 - e^{-2t}$$





Exponential inter-arrival time

In this special case, it is possible to determine the performance indices of the system analytically considering it as an "*M/G/1*" queue (this topic will be covered later in the course).

$$U = \lambda \cdot S \qquad N = \lambda \cdot S \cdot \frac{2 - \lambda \cdot S}{2 \cdot (1 - \lambda \cdot S)}$$

$$X = \lambda \qquad R = S \cdot \frac{2 - \lambda \cdot S}{2 \cdot (1 - \lambda \cdot S)}$$

Inter-arrival time distribution

Note that the deterministic and the random models have the same *throughput* and *utilization*: this is because these two performance indices depends *only on the arrival rate* and on the *average service time*.

Deterministic

$$U = \lambda \cdot S \qquad N = \lambda \cdot S$$

$$X = \lambda \qquad R = S$$

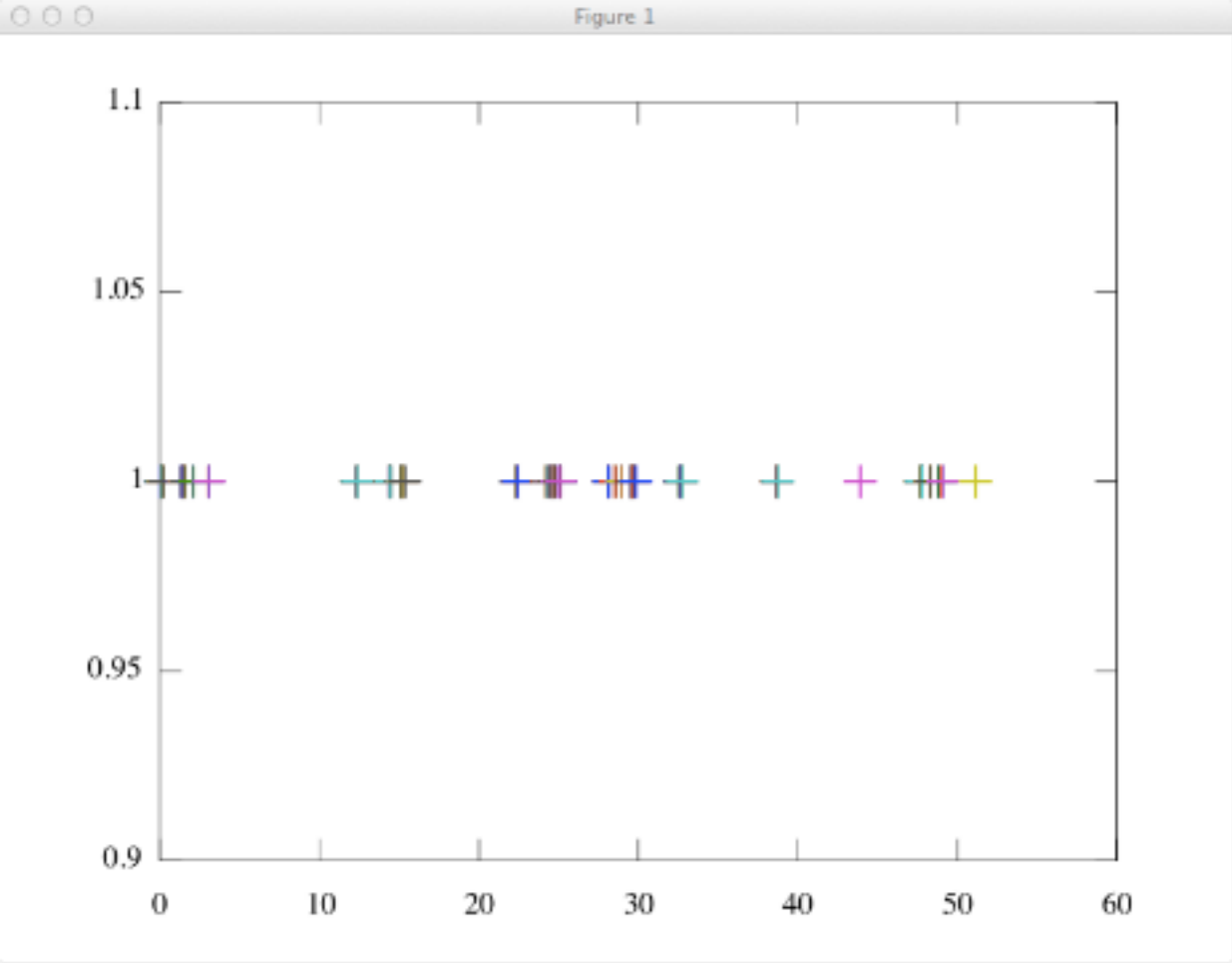
Exponential

$$U = \lambda \cdot S \qquad N = \lambda \cdot S \cdot \frac{2 - \lambda \cdot S}{2 \cdot (1 - \lambda \cdot S)}$$

$$X = \lambda \qquad R = S \cdot \frac{2 - \lambda \cdot S}{2 \cdot (1 - \lambda \cdot S)}$$

Hyper-exponential inter-arrival time

Let us now try another inter-arrival time distribution, with the same arrival rate of $\lambda = 2\ job\ /\ s$.



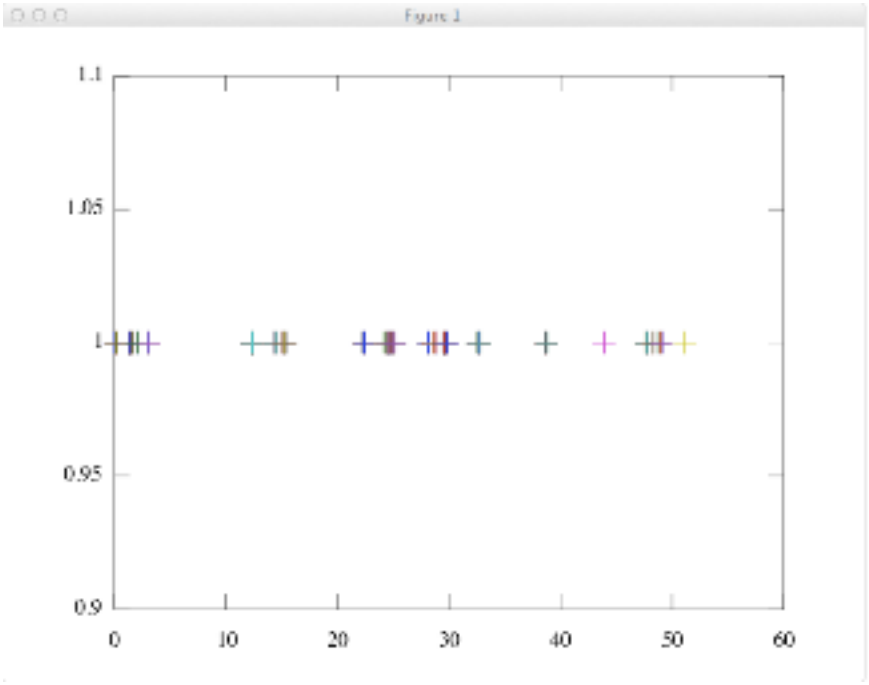
Hyper-exponential inter-arrival time

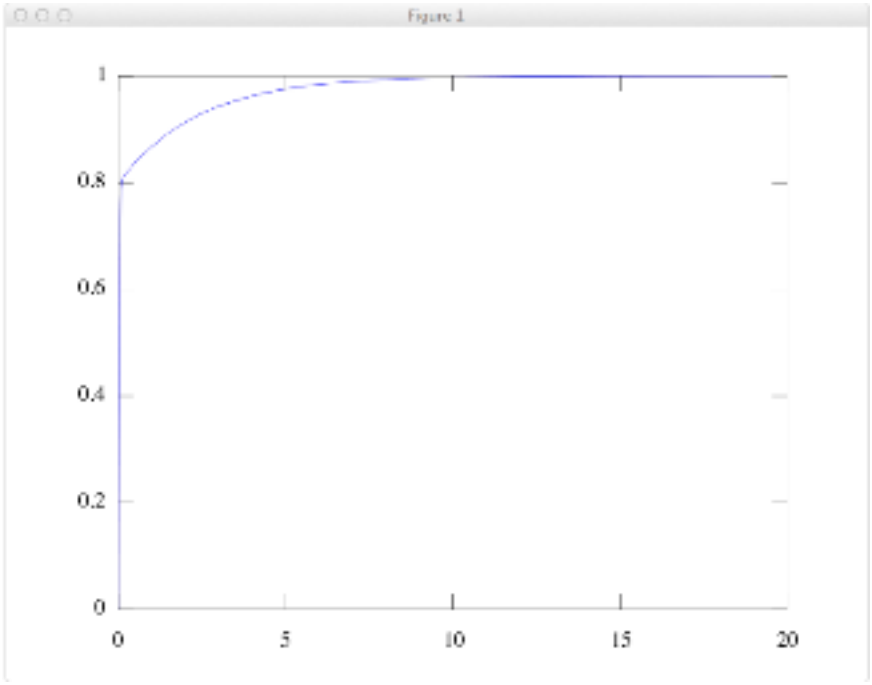
In this case the inter-arrival time follows a *Hyper-exponential distribution*.

$$a_i \sim Hyper(40, 0.41\bar{6}, p = 0.8)$$

We will soon return on the Hyper-Exponential distribution, and describe it in detail.

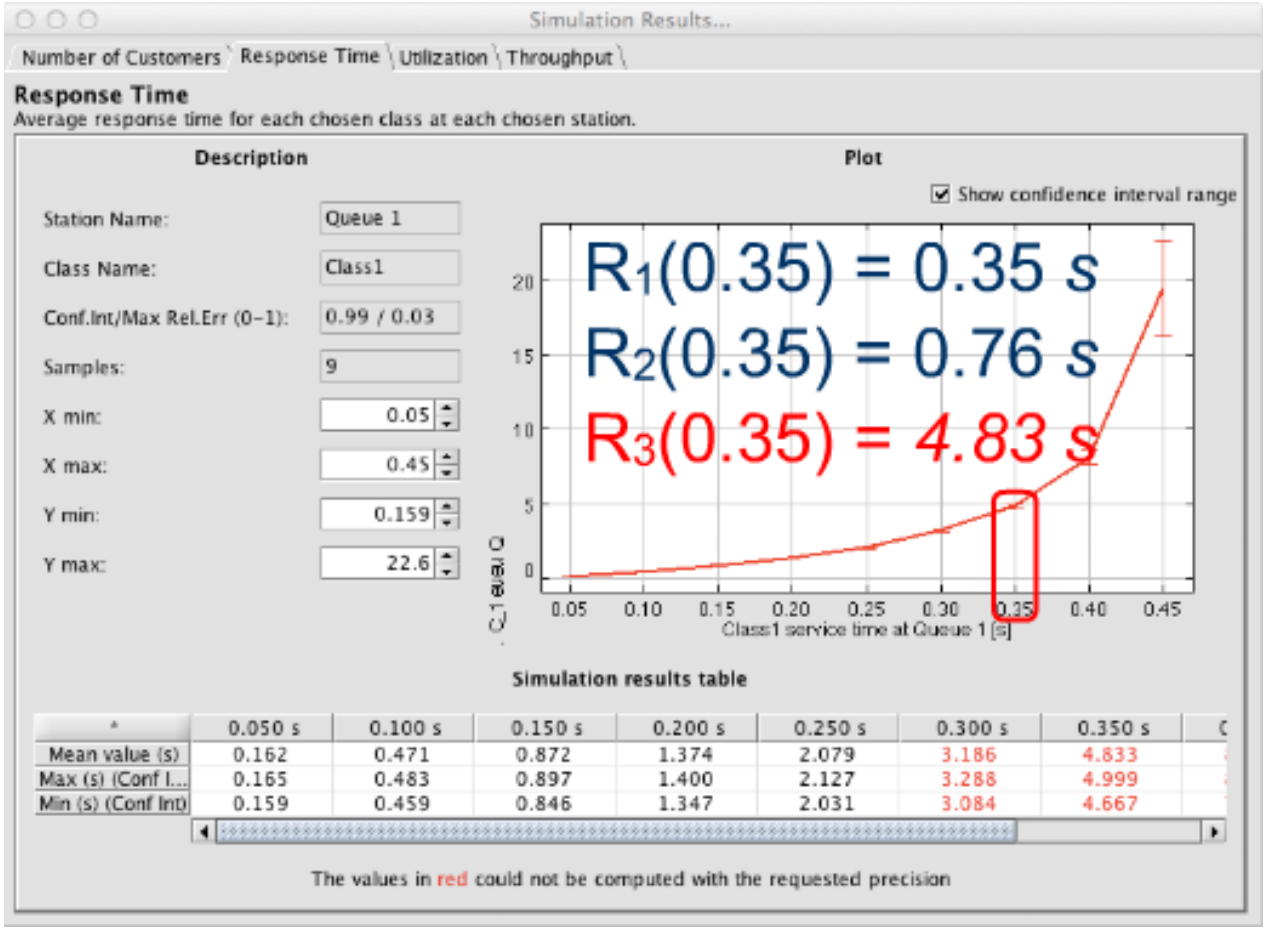
$$F_{Hyper}(t) = 1 - 0.8e^{-40t} - 0.2e^{-0.41\bar{6}t}$$





Hyper-exponential inter-arrival time

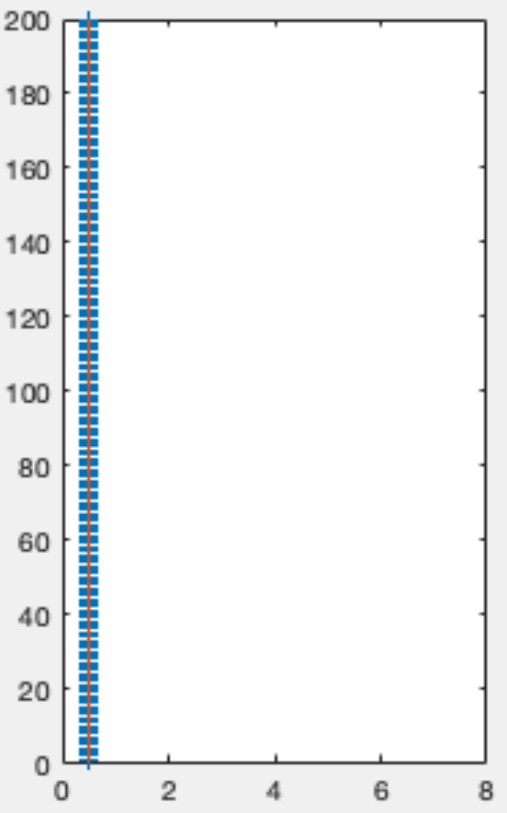
The resulting response time is much higher for the same service time with respect to both the deterministic and the exponential inter-arrival time distribution.



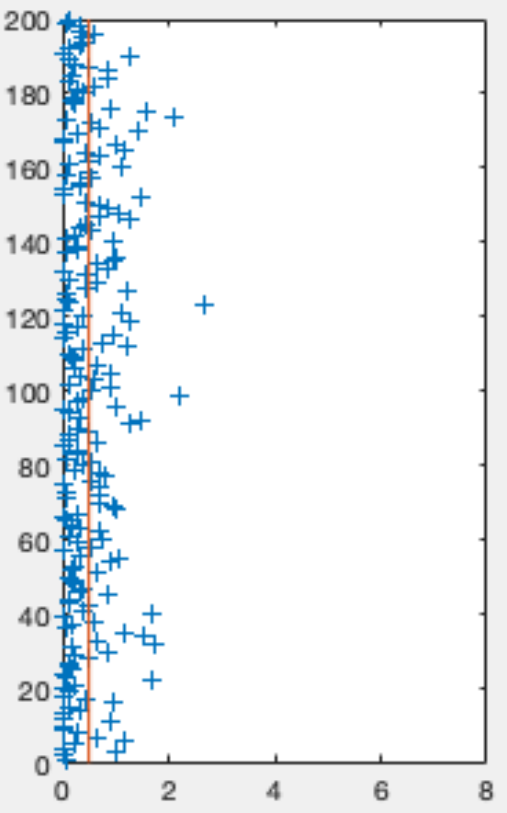
Hyper-exponential inter-arrival time

The three cases have inter-arrival time distributions characterized by the same *average*, but a different *variability*.

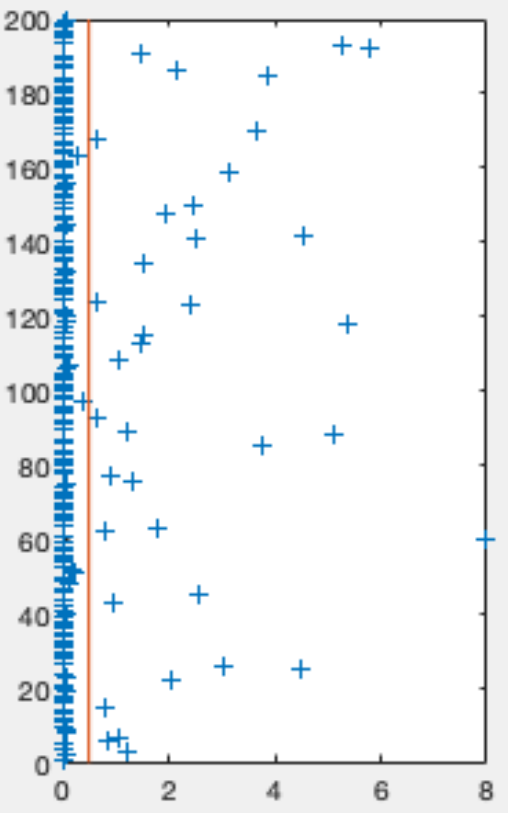
Deterministic



Exponential



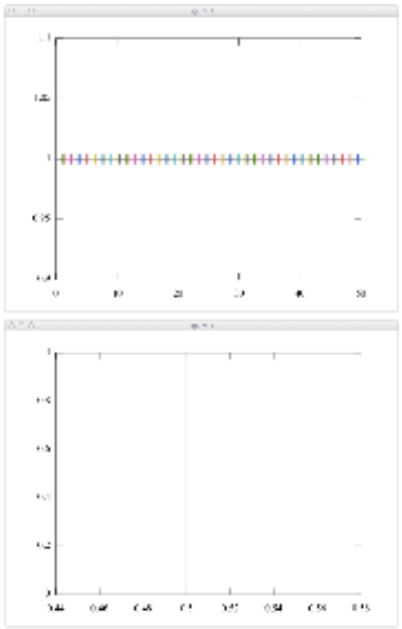
Hyper-Exponential



Effects of the inter-arrival time distribution

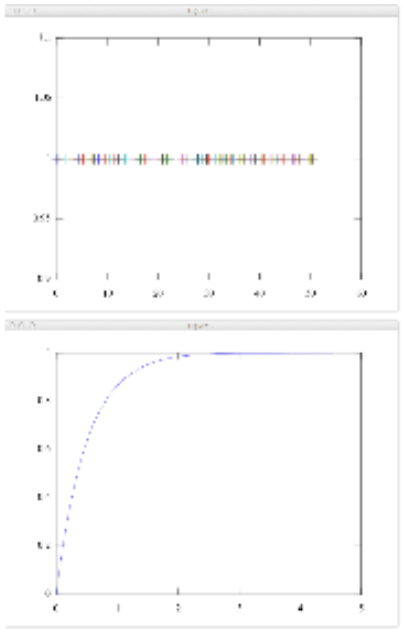
The different inter-arrival time distributions result in very different response times.

In particular: *a higher variability determines a worse performance*.



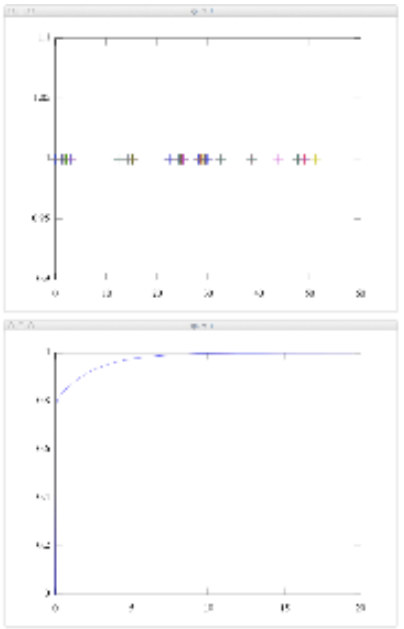
Det(0.5)

$R_1(0.35) = 0.35\ s$



Exp(2)

$R_2(0.35) = 0.76\ s$



Hyper(40,0.416,p=0.8)

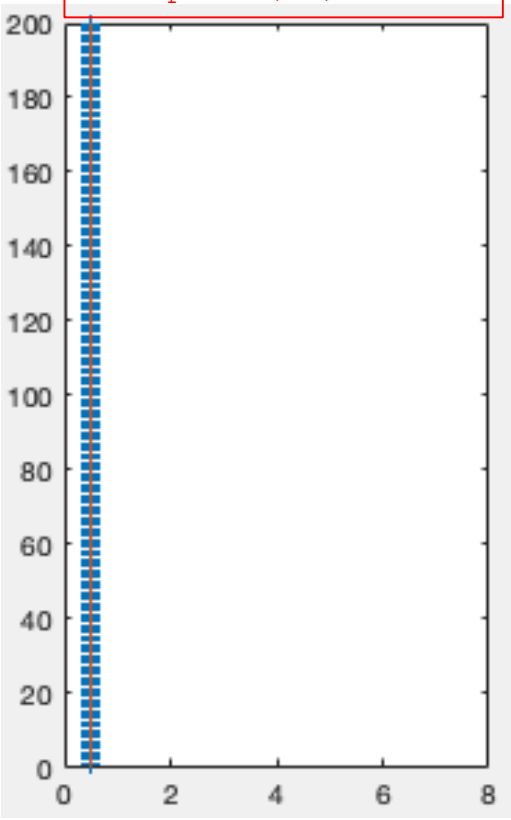
$R_3(0.35) = 4.83\ s$

Hyper-exponential inter-arrival time

The *Standard Deviation* (which can be automatically computed in many software – for example with `np.std()` in NumPy) is a simple way in which we can measure the variability of inter-arrival times.

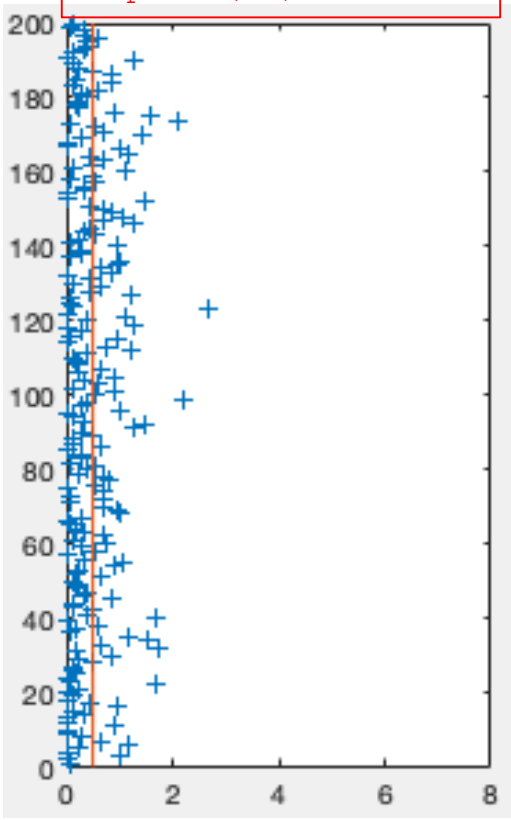
Deterministic

`np.std(Ai) = 0`



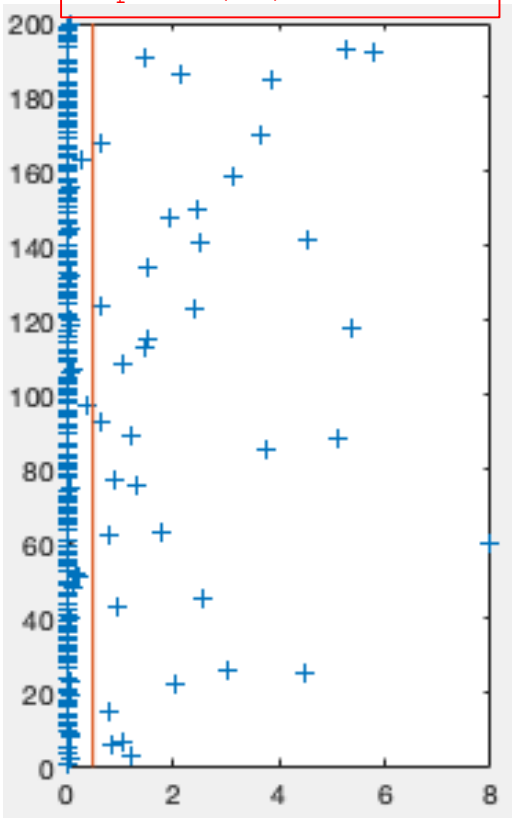
Exponential

`np.std(Ai) = 0.469`



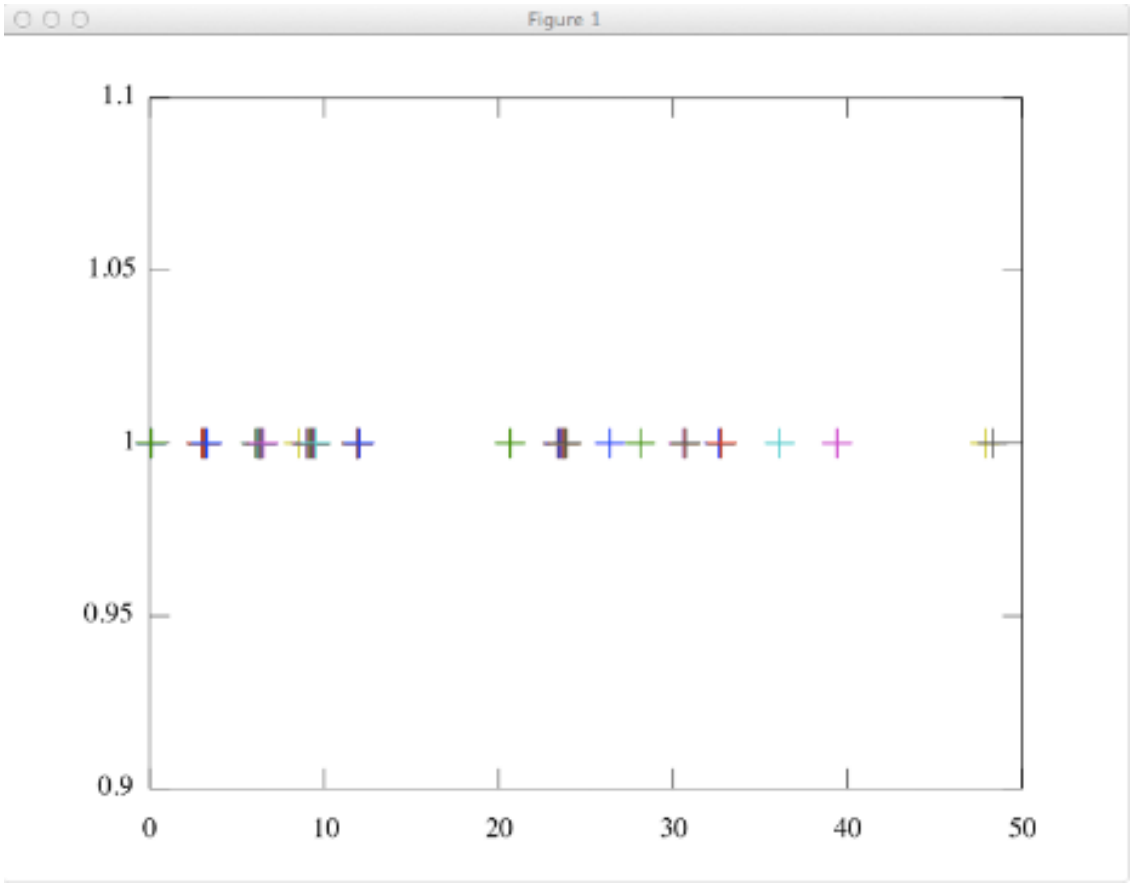
Hyper-Exponential

`np.std(Ai) = 1.2239`



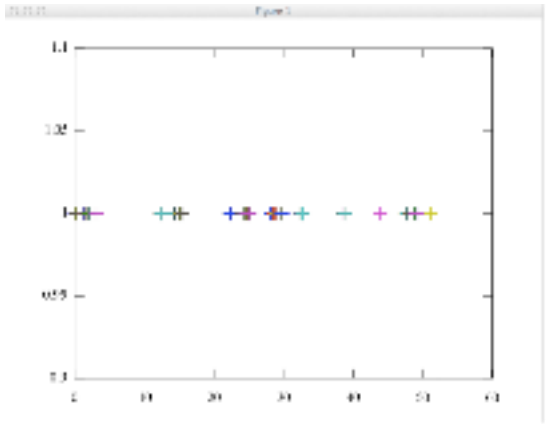
Correlation

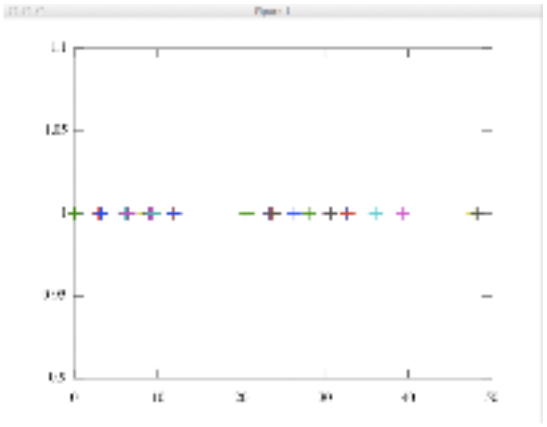
Variability of the inter-arrival time distribution is however not the only feature that characterizes arrivals to a queue. Let us consider the following arrival process:

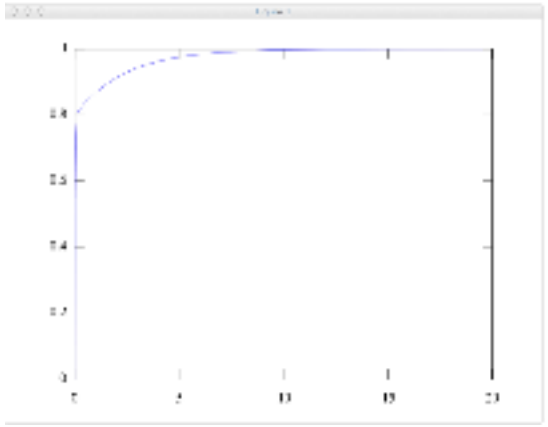


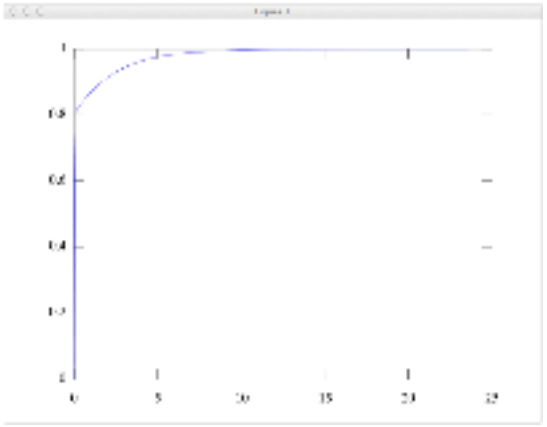
Correlation

This process has exactly the same inter-arrival time distribution as the *Hyper-exponentia*l case.



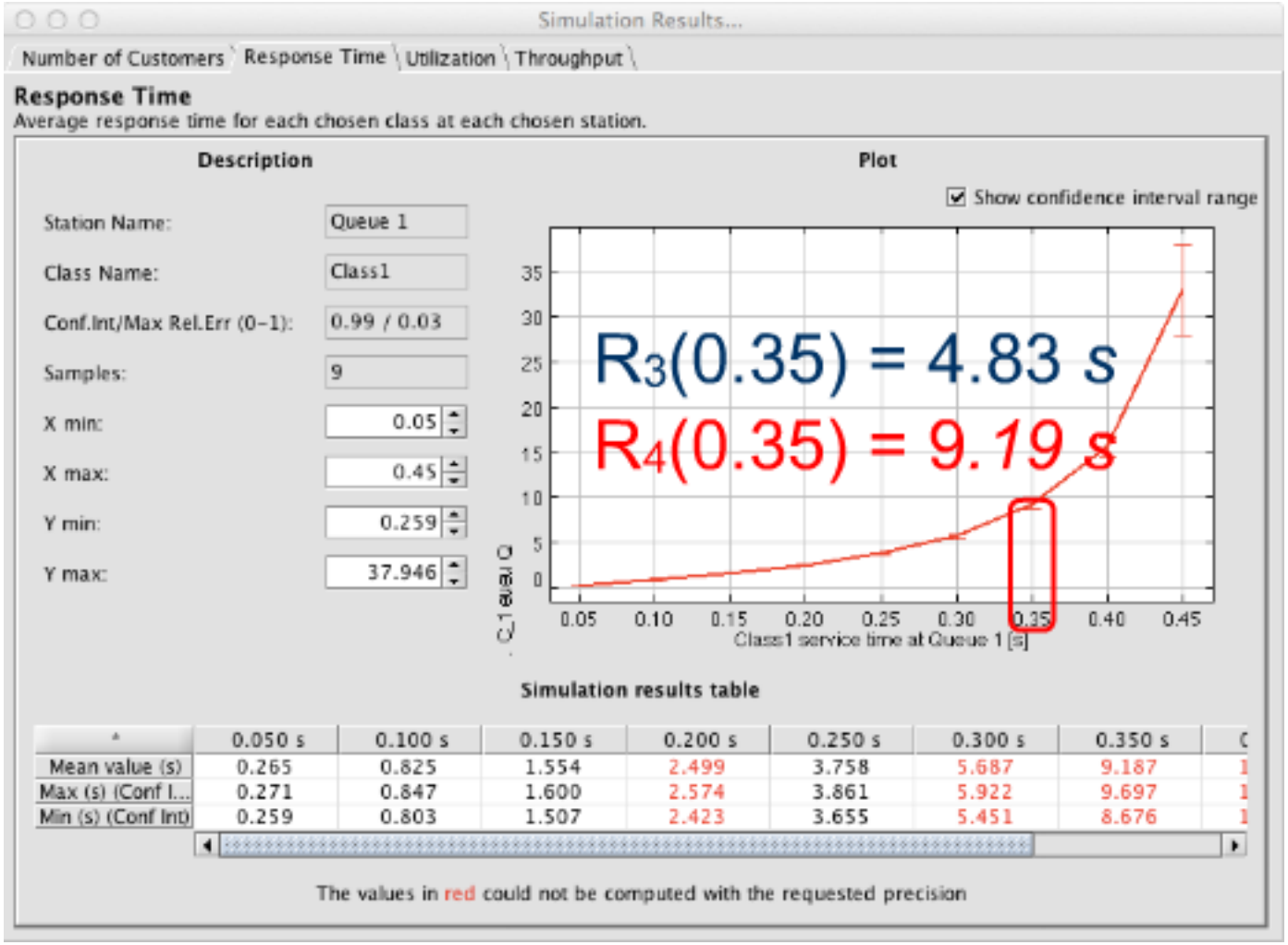






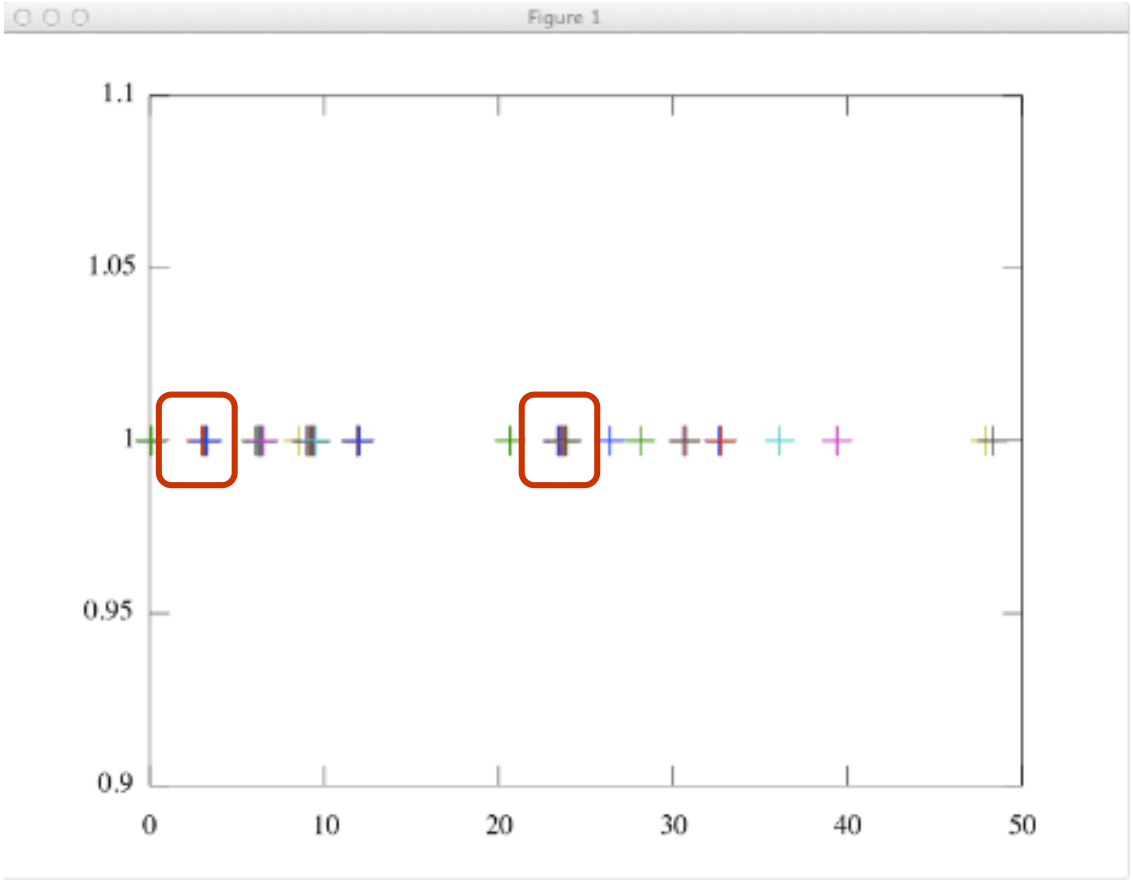
Correlation

However, if we measure the response time, we see that it is much higher than in the previous case.



Markov-modulated Poisson Process

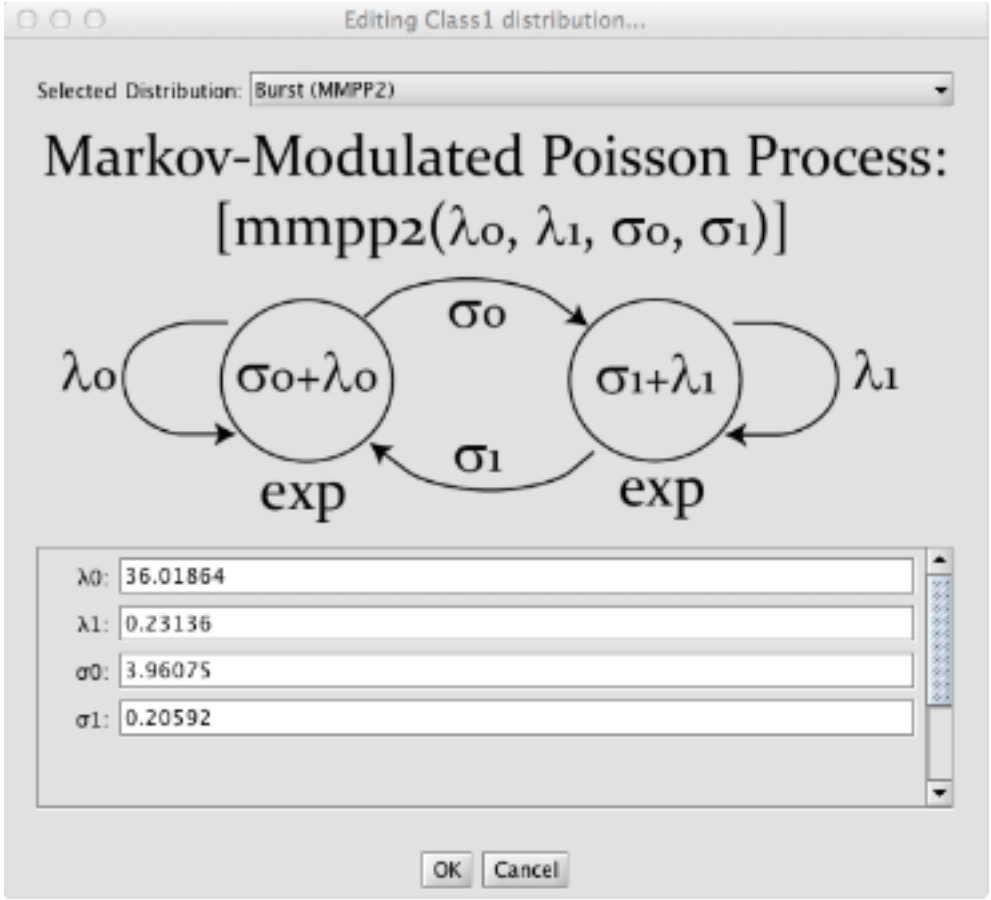
In this case, inter-arrival times are no longer independent: there is a strong *correlation* between arrivals that creates *bursts*.



In the following lessons, we will see how to detect correlation and bursts.

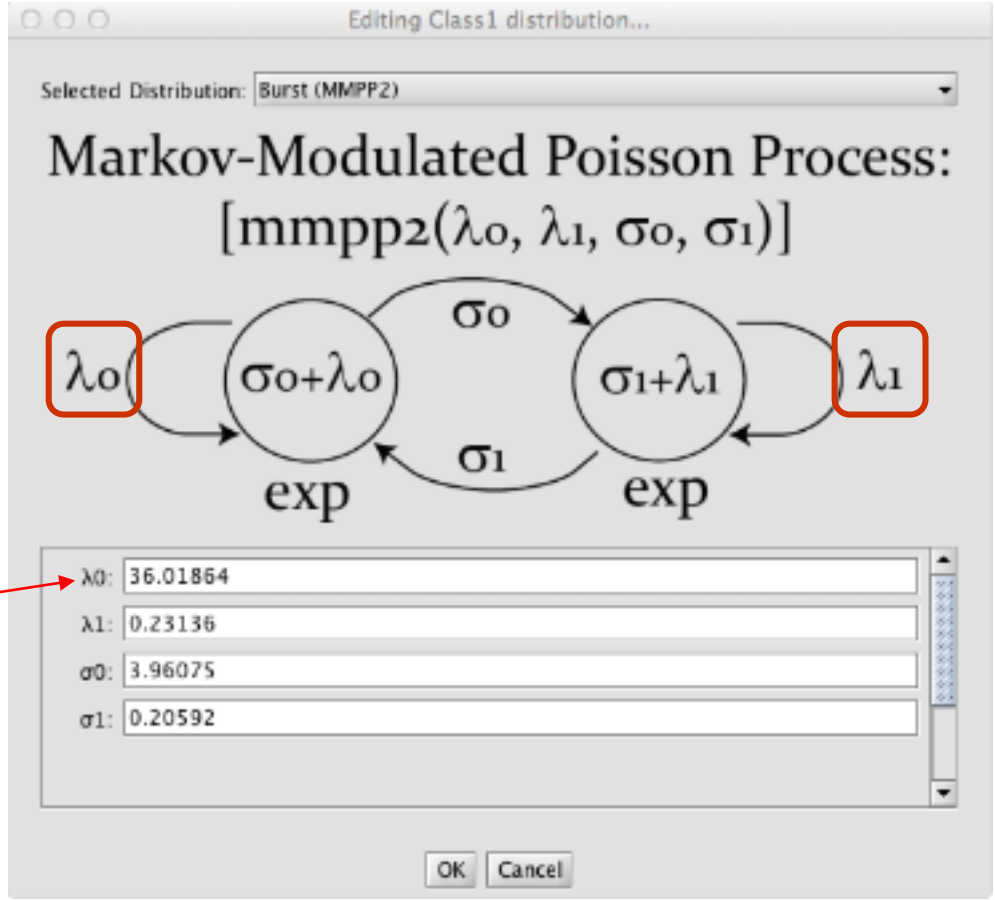
Markov-modulated Poisson Process

This particular arrival process is called "*Markov Modulated Poisson Process*" of *size 2*, and it is denoted as *MMPP(2)*.



Markov-modulated Poisson Process

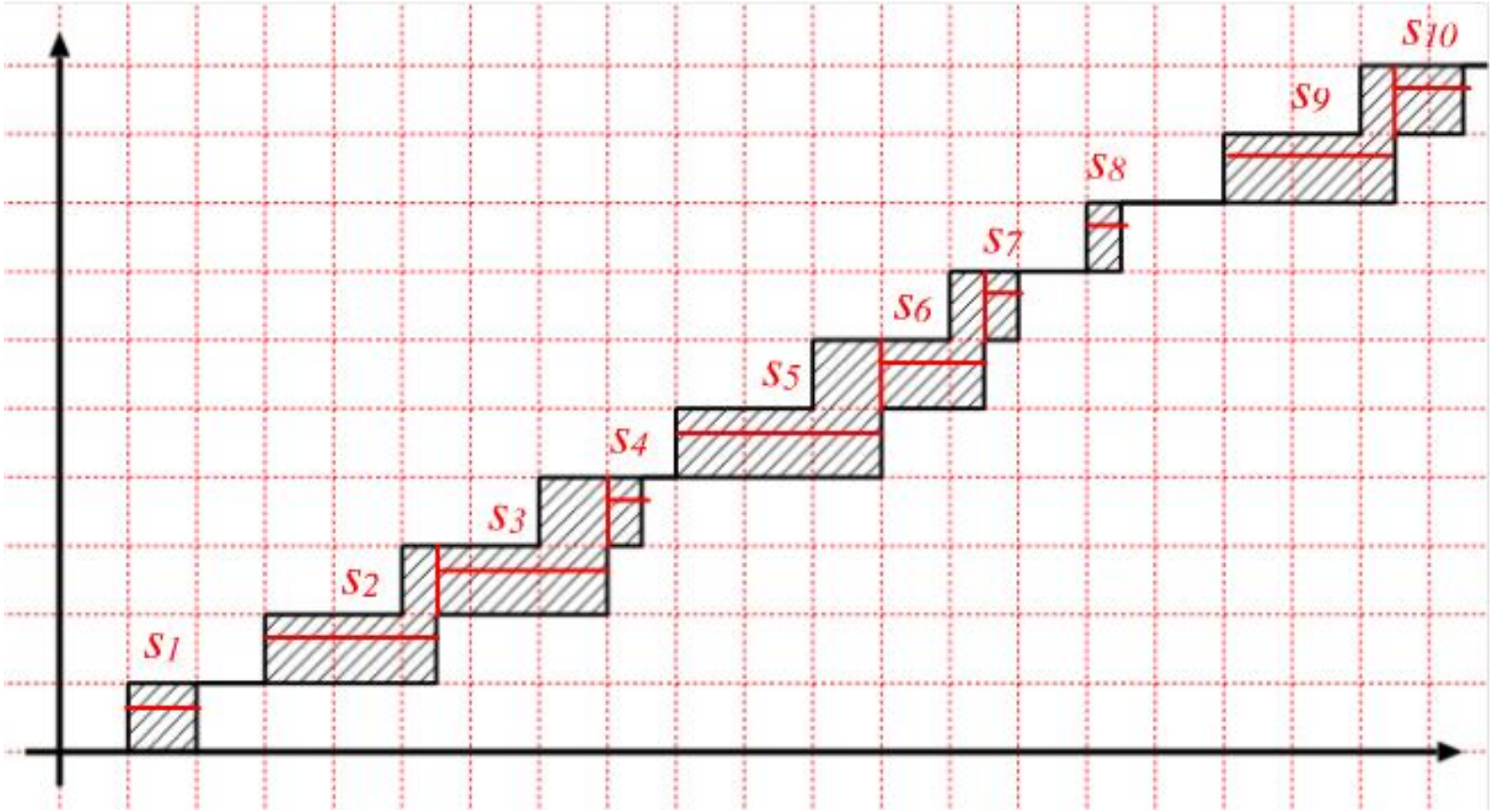
This process alternates between two phases in which jobs arrives at different rates (λ_0 and λ_1). Usually, one rate is small and the other is very high (the "burst").



burst

Service

The *service time distribution* defines the probability that the server will need a given time to complete a job.



Servive

Services describe the time required to complete a task.

Service

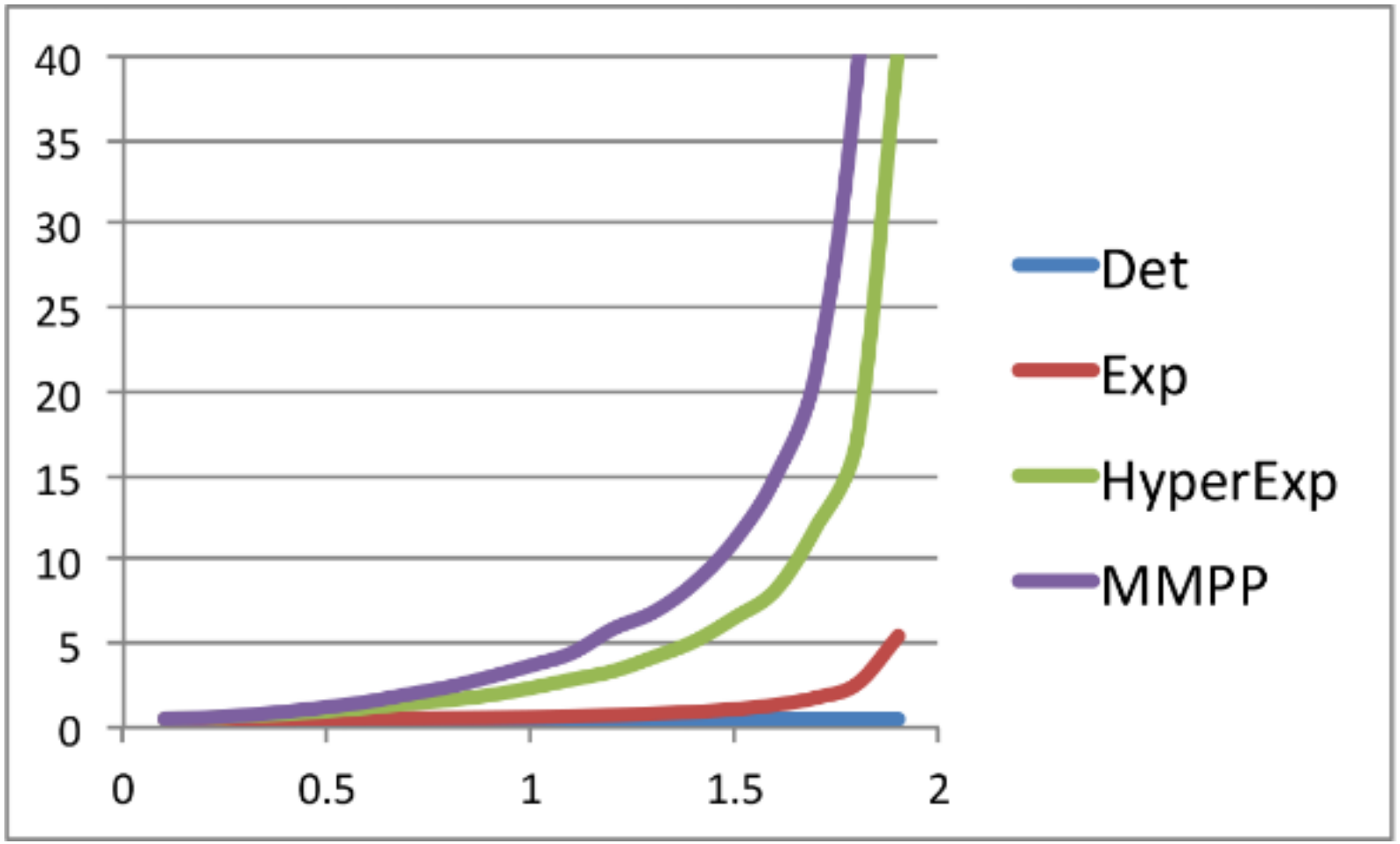
Again, randomness has an effect on the performance indices, which depends on the distribution: let us consider several service time distributions characterized by the same *average D = 0.5 s*.

For the utilization law, the system can serve an arrival rate up to λ = *2 job / s*.

- We will use this to perform consider several scenarios having a variable deterministic arrival rate from *0.1 job /s* to *1.9 job / s*.

Service

As we can see, the different distributions cause different response times. Correlation among services has also a visible impact.



Probability distributions

Arrival rates and *service times* are generally unpredictable, but they follow patterns that can be used to characterize them.

Probability distributions and *Stochastic processes* play an important role in performance modeling, since they are used to characterize most of the behaviors that cannot be deterministically predicted.

Defining the proper probabilistic characterization of external arrivals or service times is a crucial step for accurately modelling the behavior of a system.

Probability distribution in Performance Evaluation

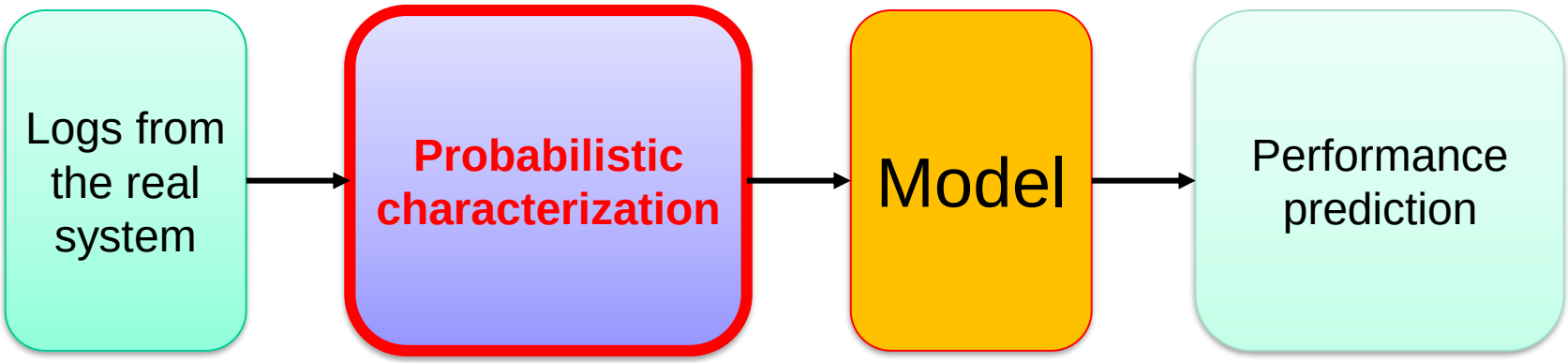
Using probabilistic descriptions we can:

- Predict the performance indices using analytical techniques (however the number of cases in which this can be done is limited).
- Generate synthetic traces to reproduce complex system behaviors.

Probability distributions

In general, measurements of the considered real system are used to derive its probabilistic characterization.

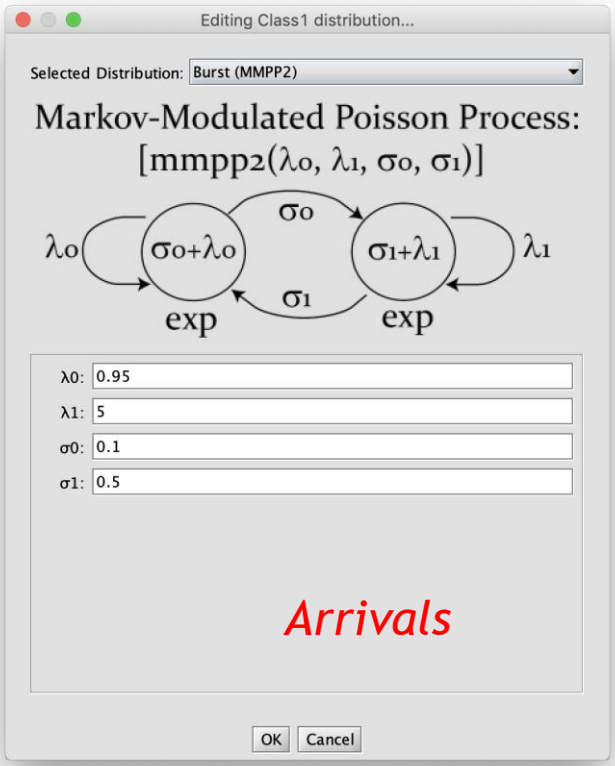
This is then used to parametrize a model, from which performances can be predicted.

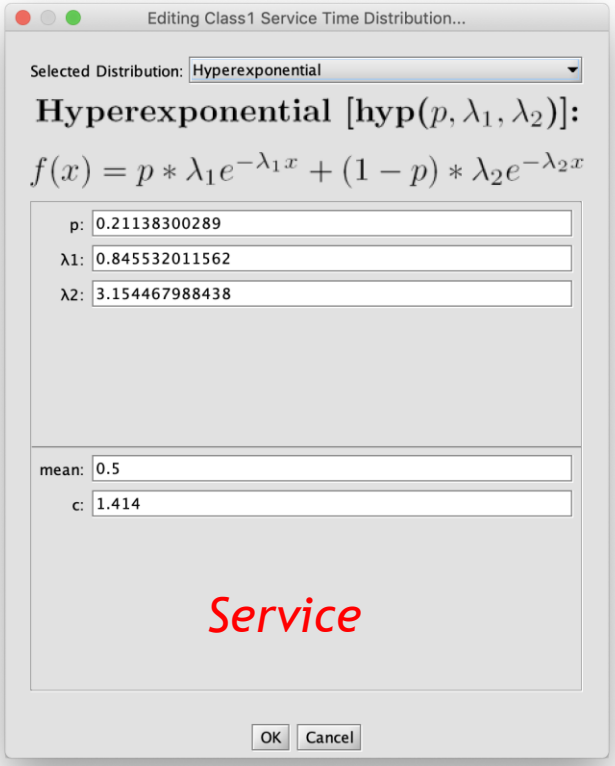


In the following lessons, we will see how to perform a proper probabilistic characterization of the workload of a system.

Analysis of Motivating Example

After a more careful observation, the manager found some extra statistical properties of both arrivals and services:





Analysis of Motivating Example

Performing the analysis using a suitable tool such as JMT, she was able, with the new specifications, to correctly match the response time estimation.