MsC in Computer Engineering
Course of Performance Evaluation of Systems and Networks

Project 13

Study of The Longest Job First Queue Policy in a single-queue single-server System

Yuri Iozzelli

Danilo Loffreno

Tommaso Zaccone

Introduction

The aim of this project is to study the response time of a queue+server system that serves jobs according to the longest job first policy.

The case studies that will be analyzed are the classic M/M/1 system and an M/G/1 system with a log-normal distribution for the service time (from here called M/L/1).

The Longest Job First Policy (LJF) is a policy that depends on the service time, and thus the theoretical results for the distribution of the response time do not hold.

In literature there is very little study of the LJF system, unlike its opposite SJF (Shortest Job First) system. The latter is proven to be optimal in respect to minimizing both the mean response time and the mean number of jobs in the system, thus we expect the LJF to have poor average performance in terms of response time.
Nonetheless it will be shown that in some cases the LJF policy may be the policy of choice.

As a meter of comparison for the response time performance a FIFO (First In First Out) queue will be used.

System Modelling

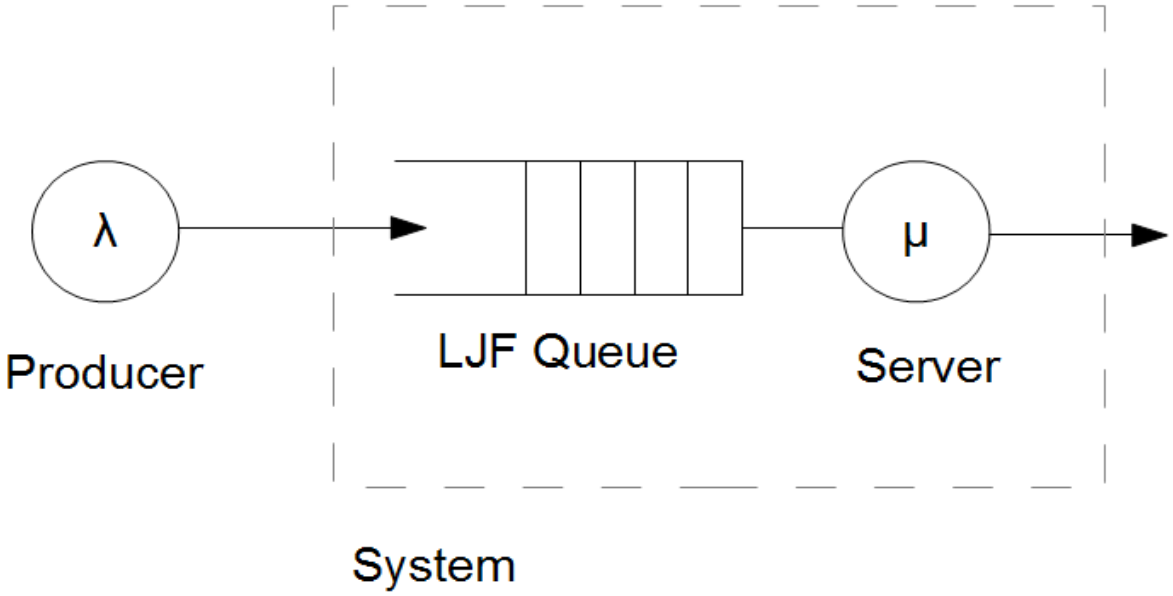

Fig.1 System Model

We consider a system with an infinite queue ruled by the LJF policy.

The inter-arrival Time is exponentially distributed (the Producer node in the figure dispatches jobs according to a Poisson Process) with rate λ. This value will be tuned to simulate several workloads.

The service is performed by a single server whose service rate is independent from the state of the queue and the arrival rate.

For the service time we consider two possible distributions:

- an exponential distribution with rate μ (M/M/1 system).
- A lognormal distribution with rate μ and standard deviation σ (M/L/1 system).

System Implementation

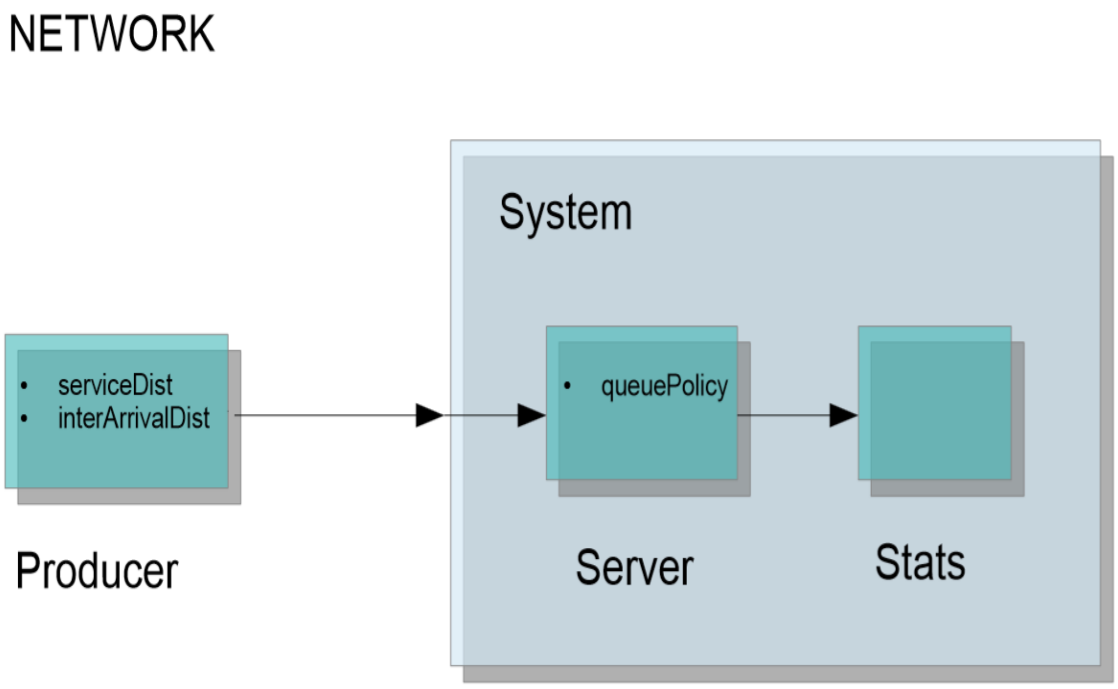

Fig. 2 - System Implementation in Omnet++

The implementation of the above model is achieved through the use of the C++ language and the Omnet++ simulator.

Omnet++ allows the developer to define the behaviour of the network nodes by subclassing the *cSimpleModule* C++ class. Several simple modules can be connected to form a *Compound Module,* which is functionally equivalent to a simple module from external nodes.

The implementation consist of 2 main modules:

- **Producer:** This is implemented as a Omnet++ simple module and it is responsible for the generation of the jobs with the given distributions.
- **System:** This is implemented as a compound module containing 2 simple submodules:
    - **Server:** Here both the server and the queue logic are simulated.
    - **Stats:** This module is used to collect the useful statistics.

The following is a detailed description of each module:

Producer

This module have no input and only one output connected to the system module, it takes 2 parameters:

volatile double interArrivalDist;
volatile double serviceDist;

In the configuration file these parameters are defined as the output of the corresponding distributions.

The volatile modifier allows to tell Omnet to re-compute the value every time it is accessed.

In this way the producer module is unaware of the kind of distribution we are using, which can be dynamically set in the configuration file.

The producer uses the interArrivalDist parameter to set the timer for the dispatch of the next job and the serviceDist parameter to set the job service time.

Server

In this module there are both the queue and server logic.

When a job arrives from the producer module it is pushed into the queue and ordered inside the queue according to the queue policy.

The processing of a job is simulated with a timer set accordingly to the service time set by the producer.

When a job is processed, it is sent to the Stats module and a new value is popped from the queue (if it is not empty).

The queue is implemented as an abstract class, and the actual class used is chosen based on the queuePolicy parameter (which can be set in the configuration file). This allow the module to work flawlessly with different policies.

The policies actually implemented are LJF and FIFO (which will be used for performance comparisons), but others are easily implementable.

Stats

This module is used to decouple the statistics collection from the system simulation.

It receives a job from the Server module, records the useful statistics,and deletes it.

Configuration File

In this file (omnet.ini) there are many parameters that define the behavior of the simulation:

- **sim-time-limit**: maximum time limit for the simulation. the value for this parameter needs to be high enough to allow the system to reach an acceptable number of states states.
- **repeat**: number of repetitions for a single experiment (with different random seeds). This is important for the computation of confidence intervals, because every run has a different random seed.
- **warmup-period:** time from which the system starts to collect statistics. It is set to a value for which the mean number of jobs in the system can be considered stable.
- **producer.serviceDist**: the distribution of the service time. This will be set accordingly to the specific case study, and depending on the particular distribution chosen it is influenced by other parameters (mean value, variance).
- **producer.interArrivalDist**: the distribution of the inter-arrival time. This is set to exponential for all the simulations. the mean value is modified to simulate a varying workload.
- **server.queuePolicy**: The selected queue policy. Possible values are FIFO and LJF.

For every experiment we will define other parameters to assign proper values to the above ones.

First Scenario: M/M/1 System

In this scenario we consider an exponential distribution for both inter-arrival time and service time.

In order to do this we use a fixed service rate μ and we simulate the system varying the utilization factor, and therefore the inter-arrival rate λ, according to the formula:

$$\rho = \frac{\lambda}{\mu}$$

The simulation is performed with both a LJF queue and a standard FIFO queue, that we will use for comparison.

We use FIFO for this job because it is the simplest and more common type of queue and therefore an obvious meter of comparison.

The following plot shows the mean response time of the system varying the utilization factor ρ:

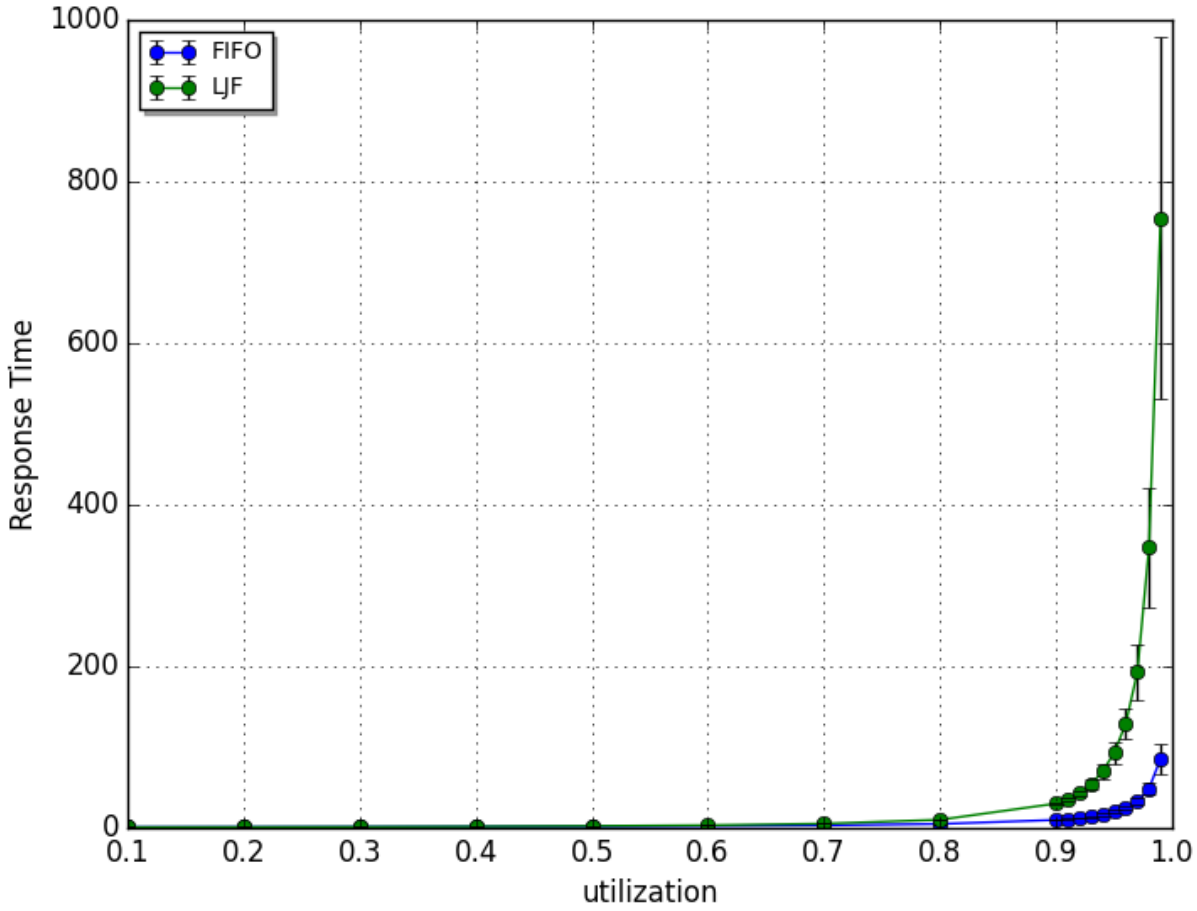

Fig.3

As the plot shows, in the LJF system the mean response time increases with ρ at a higher rate than in the FIFO system, and the difference is more and more evident as ρ approaches 1.

Even for low values of ρ , when the difference is little (because when the queue is empty the two behaviors are indistinguishable) the LJF curve is always above the FIFO one (see figure below), considering a confidence interval of 99% (see the Appendix for the method used to compute confidence intervals).

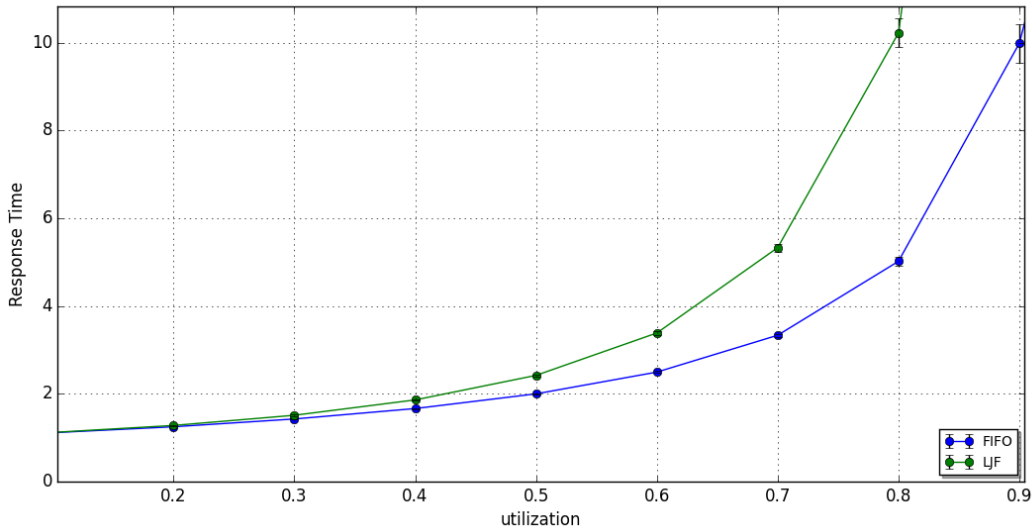

Fig.4

We can conclude that on average the response time of an LJF system is worse than the one of a standard FIFO system.

From this point on, all the plots are drawn for ρ=0.9: This value has been chosen because the behavior of the two distribution is sufficiently different to be characterized, but the tails of the distributions are not too much long compared to the duration of the simulation.

To measure the variability of the response time of the LJS system compared to the FIFO system we used a Lorenz Curve:

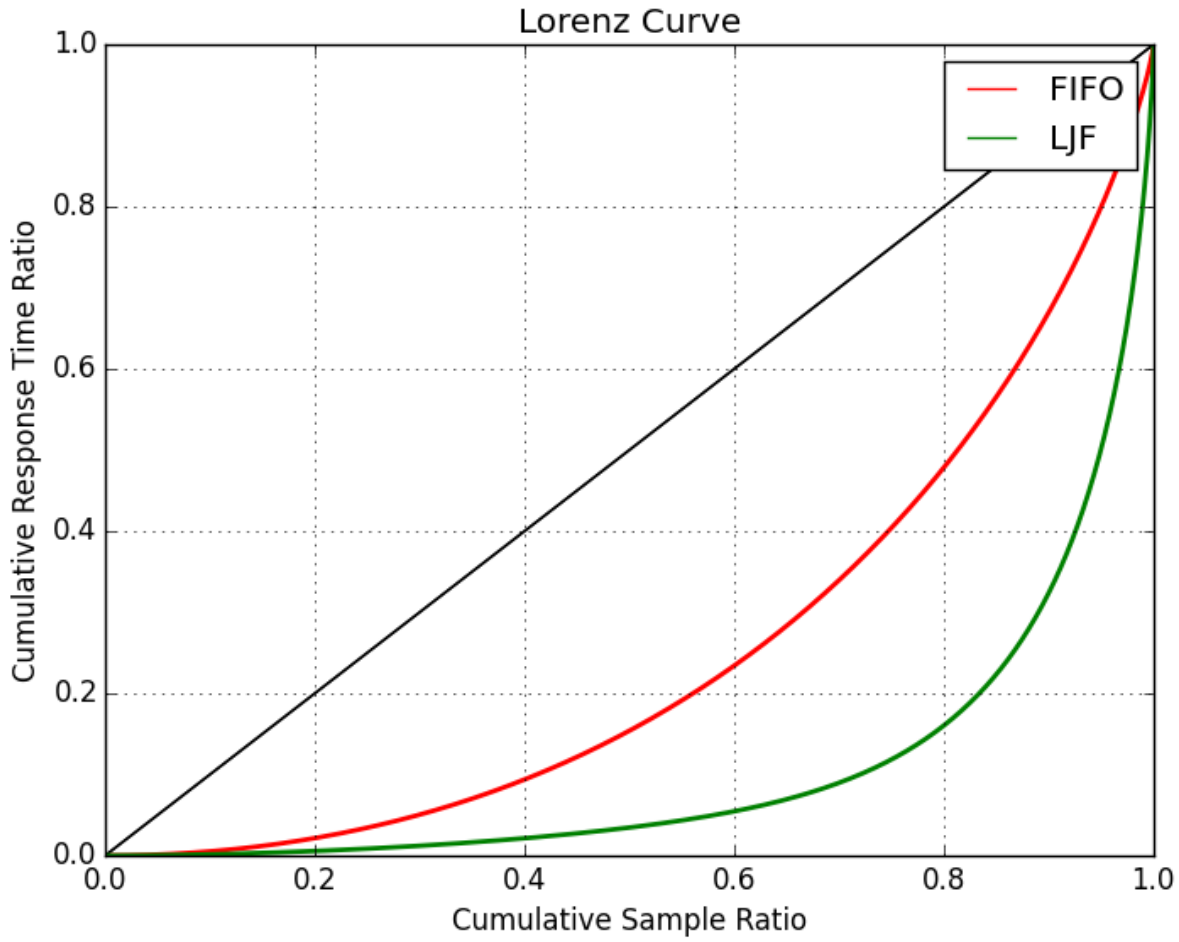

Fig.5

From the plot we can see that the LJF curve is always below the FIFO one. This means that not only that the variance is higher, but there are more jobs in the extreme values (very low and very high response times).

The following plot show the empirical probability density function (epdf):

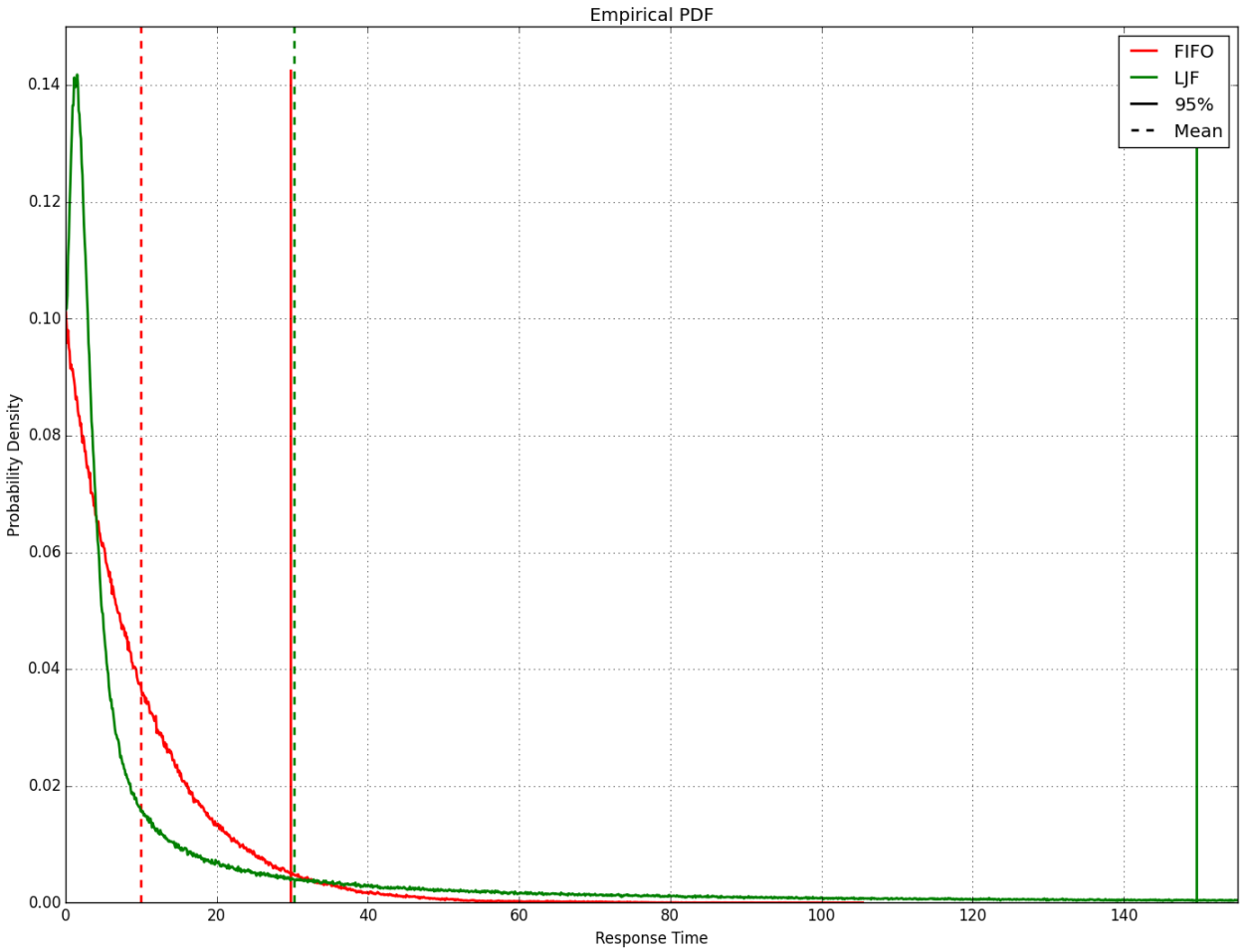

Fig.6

The continuous vertical lines represent the 95% of the sample, while the broken lines represent the mean value.

We notice that both distributions starts from the same point at x=0, and this is expected because the response time can be zero only if the queue is empty, and in this case FIFO and LJF behave in the same way.

We also notice that the tail of the LJF distribution is much longer than the FIFO one (and we can see it also by the 95% line).

The following plot shows the empirical cumulative distribution function (ecdf) with ρ=0.9:

Ff

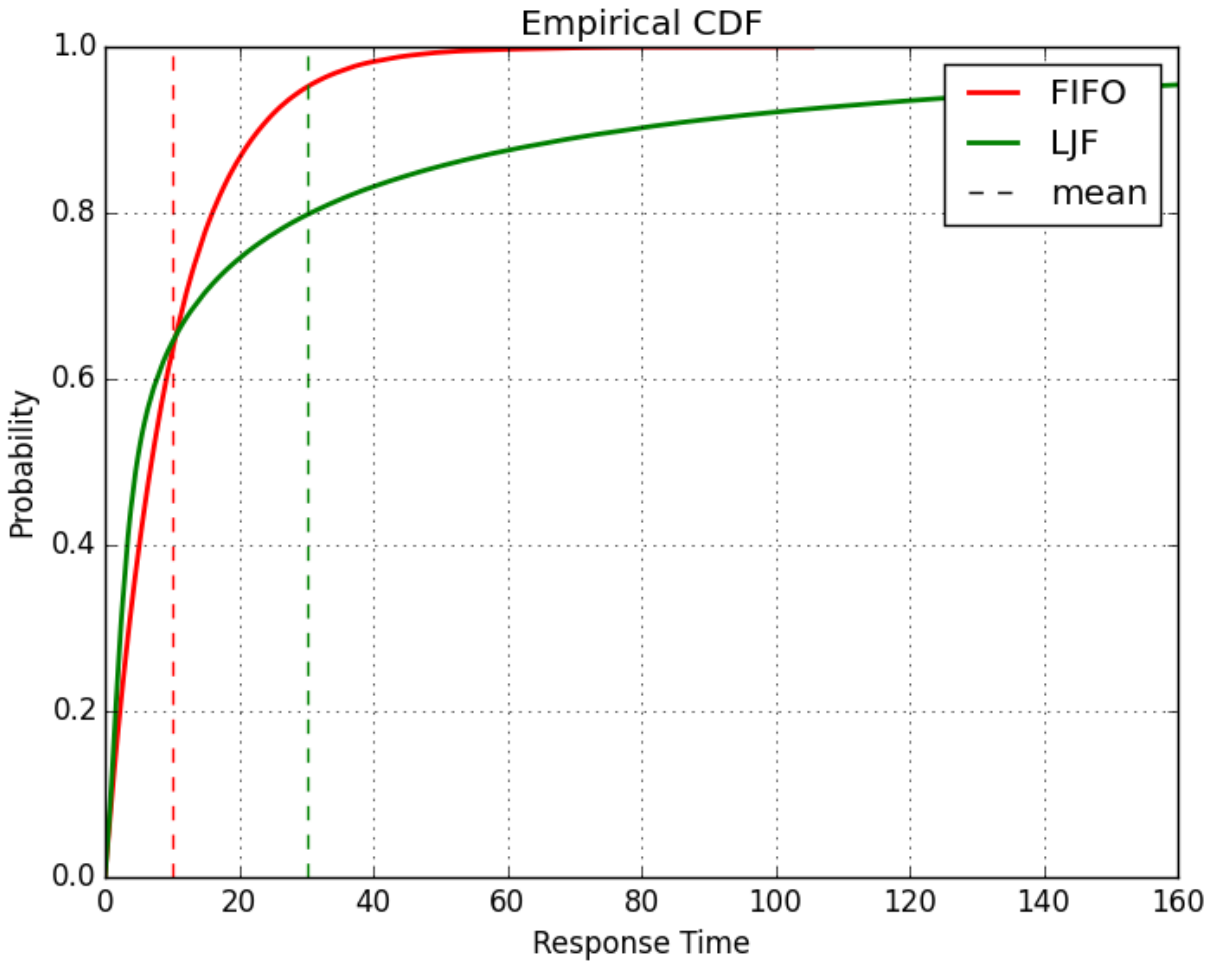

Fig.7

The broken lines represent the mean values of the distributions.

We can see that the LJS response time have an higher probability to take high values than the FIFO one.

We can also see something more unexpected: low values of the response time are more likely to be found in the LJS system, because the cdf for the LJF is above the FIFO one for low values.

This remains true for other values of ρ.

Second Scenario: M/L/1 System

In the M/L/1 scenario we consider the inter-arrival time still with exponential distribution and the service time with log-normal distribution.

In the log-normal distribution we need two parameters: ρ and σ.

The parameter ρ has the same meaning of the first case, and σ is the standard deviation of the service time. In our case it is more handy to use the coefficient of variation $c = \frac{\sigma}{m}$ (where m = mean) instead of σ because it is dimensionless (like the utilization factor).

The following plot shows the mean response time of the system varying both parameters:

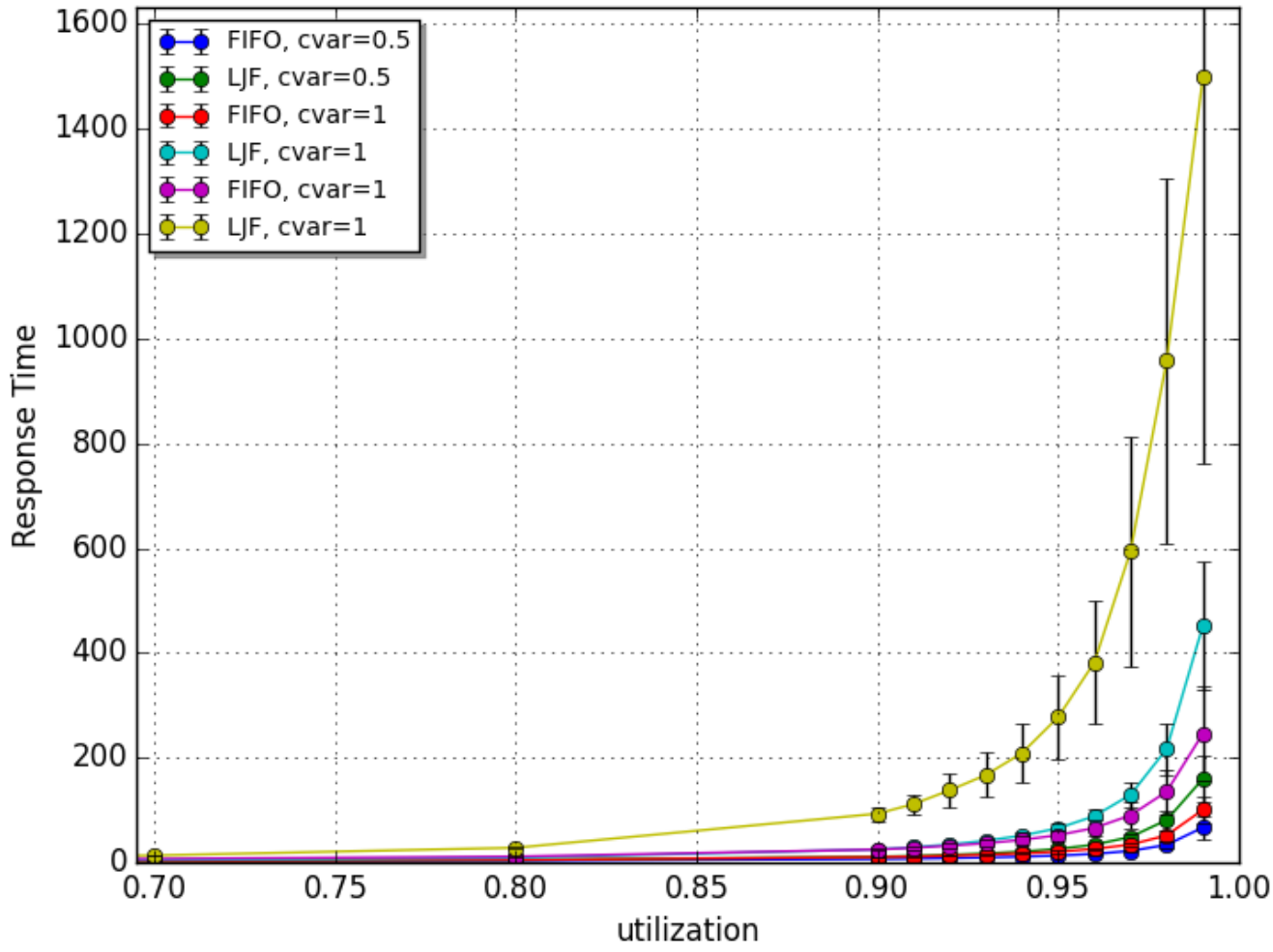

Fig.8

As in the exponential case the LJF curve is above the FIFO one, regardless the c parameter. We can notice, as expected, that increasing c also increase the mean response time on equal ρ values, but this effect is much more evident for the LJF queue.

The following are the Lorenz curves for some values of c:

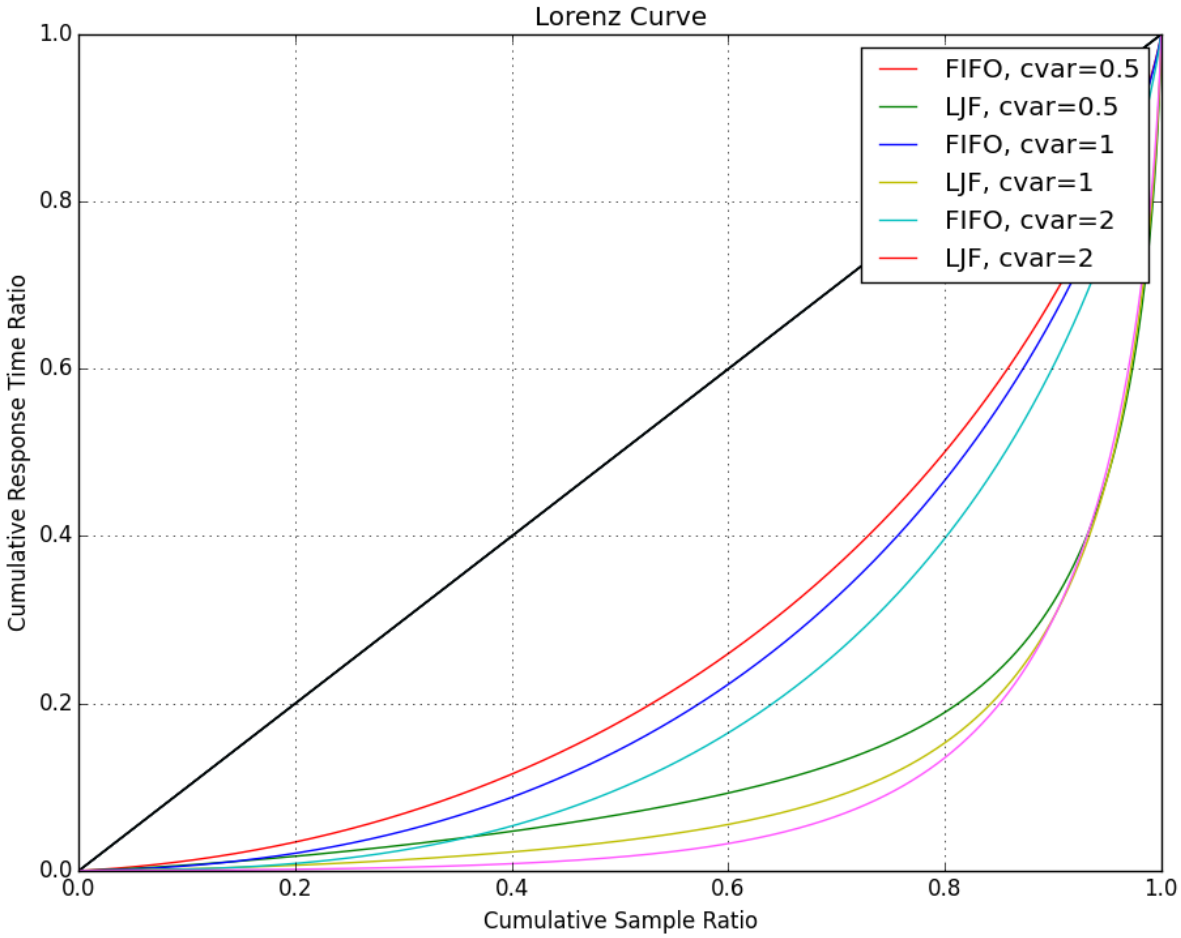

Fig.9

From this plot we can confirm that the variability of the response time in an LJF system is higher than in a FIFO system. It is peculiar to see that the last percentiles are not much affected by the value of the coefficient of variation in the LJF system, while in the FIFO one they are.

The following plot shows the empirical probability density function (epdf) for a fixed value of $c = 1$ (this is to have more clarity in the plot, but the general shapes doesn't change for the other values):

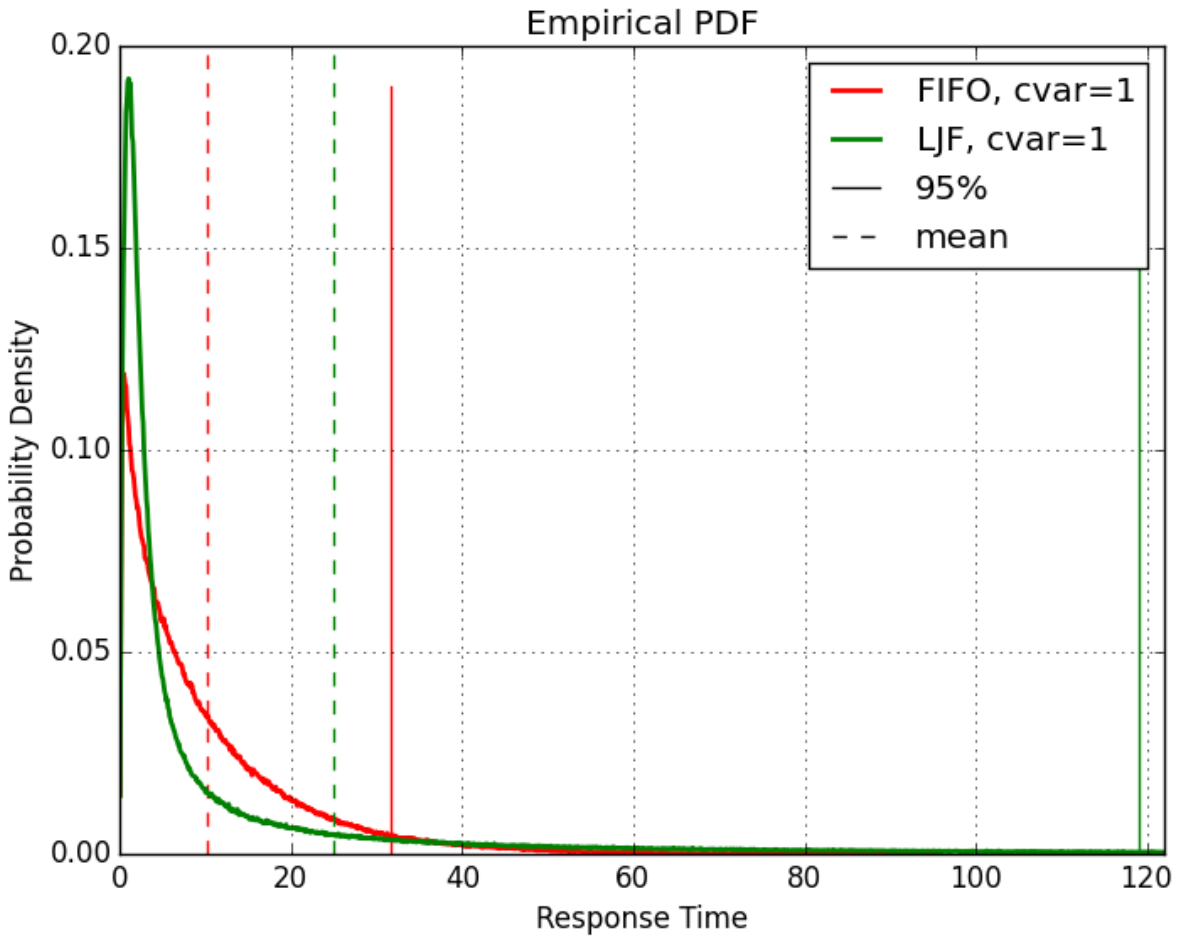

Fig.10

The continuous vertical lines represent the 95% of the sample, while the broken lines represent the mean value.

Both the curves start from the point (0,0) (even if it is not very clear with this level of zoom), and they are very similar until the FIFO curve stop rising, while the LJF continues to grow and achieves a higher mode. As usual the LJF has a significantly longer tail (as can be also seen by the 95% lines).

The variance of the service time influences the height and the position of the peak (mode), as shown in the following plot (which features only the LJF curve):

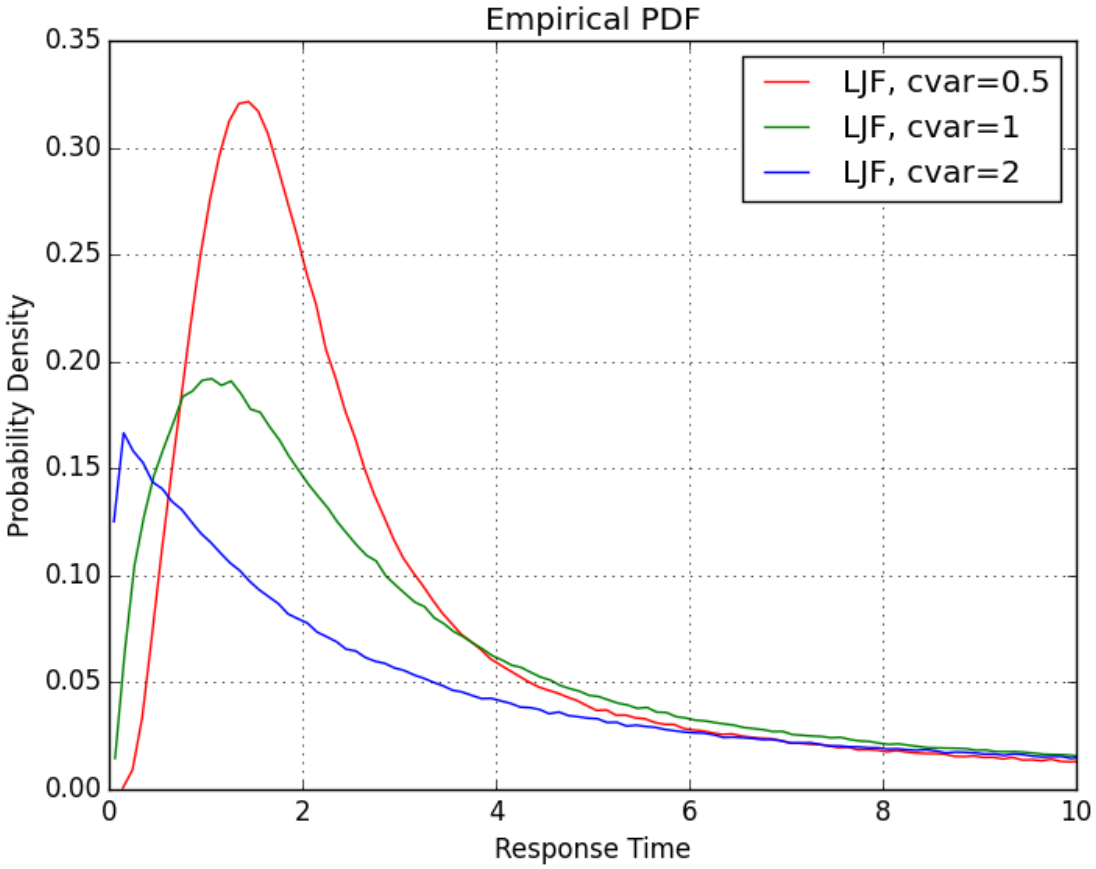

Fig.11

The following plot shows the empirical cumulative distribution function (ecdf) with ρ=0.9:

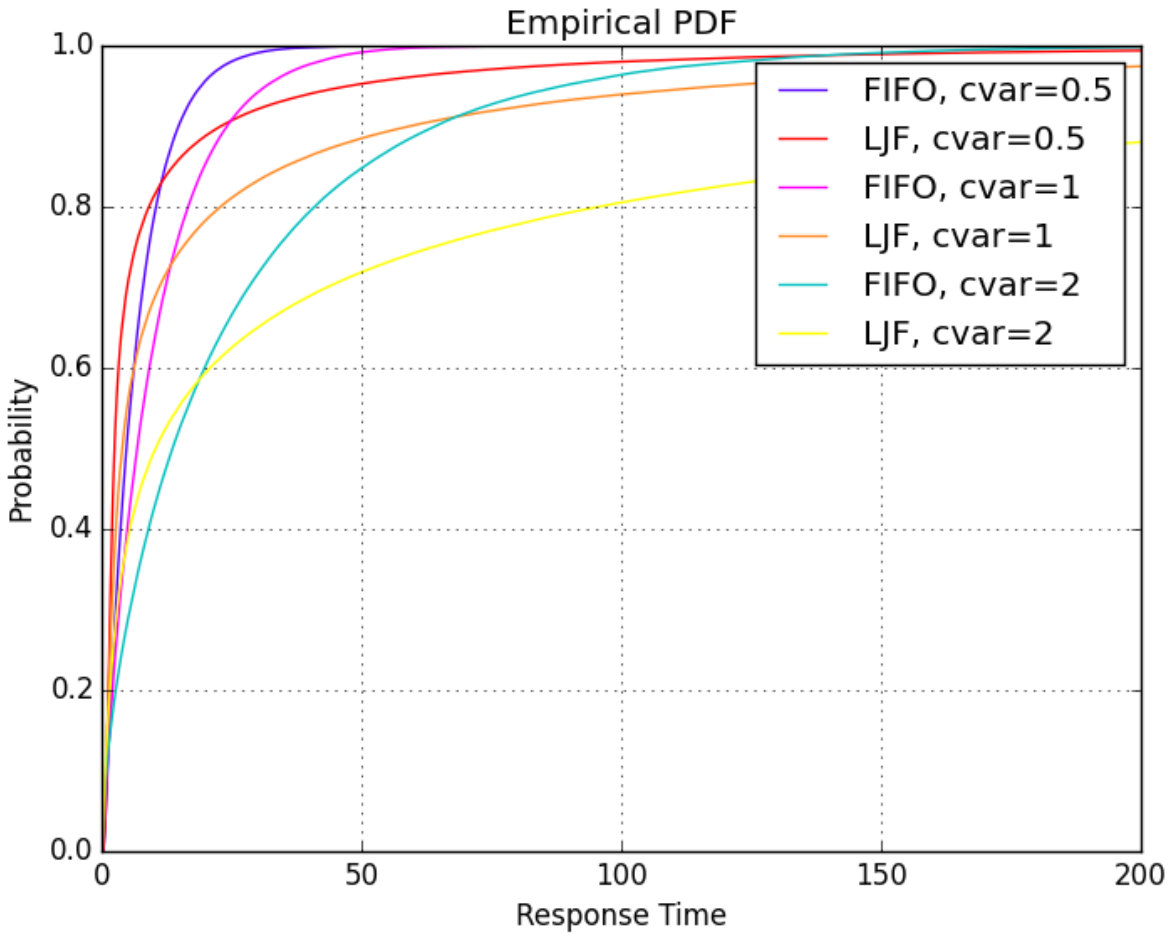

Fig.12

We can confirm what we discovered in the exponential case: the LJS response time have an higher probability to take high values than the FIFO one, but low values of the response time are also more likely to be found in the LJS system, because the cdf for the LJF is above the FIFO one for low values, and this is true for all the values of c (and ρ).

Conclusions

We can conclude that in general the LJF queue policy has worse response time performance on average, as expected. We recall indeed that the LJF policy is the opposite of the SJF policy, which is proven to be optimal in respect to the mean response time.

Nonetheless we noticed that in all the scenarios the LJF system not only has a larger number of jobs with higher response time, but also a larger number of jobs with a lower response time.

If the performance metric of a certain problem is not the mean response time, but the percentage of jobs which experience a response time below a certain threshold, the LJF system may perform better.

As we saw in the cdf plots (fig. 7, 12) there is an interval [0,x] in which the LJF curve is above the FIFO one. This means that if the threshold is in the interval [0,x] the LJF achieves a larger number of jobs that fall inside the threshold itself.

Possible Applications

A situation in which this may be applicable is, for example, the following:

A web service allows customer to submit requests, which are known to have a **uniform distribution** in [0,2] seconds.

A typical customer is not willing to wait more than 7 seconds, so a job with a response time higher than 7 means that the customer is lost. The goal of the provider is to have the maximum number of satisfied customer.

The average utilization factor is 0.5 and the server is dimensioned to provide a suitable response time for almost all the customers in this case:

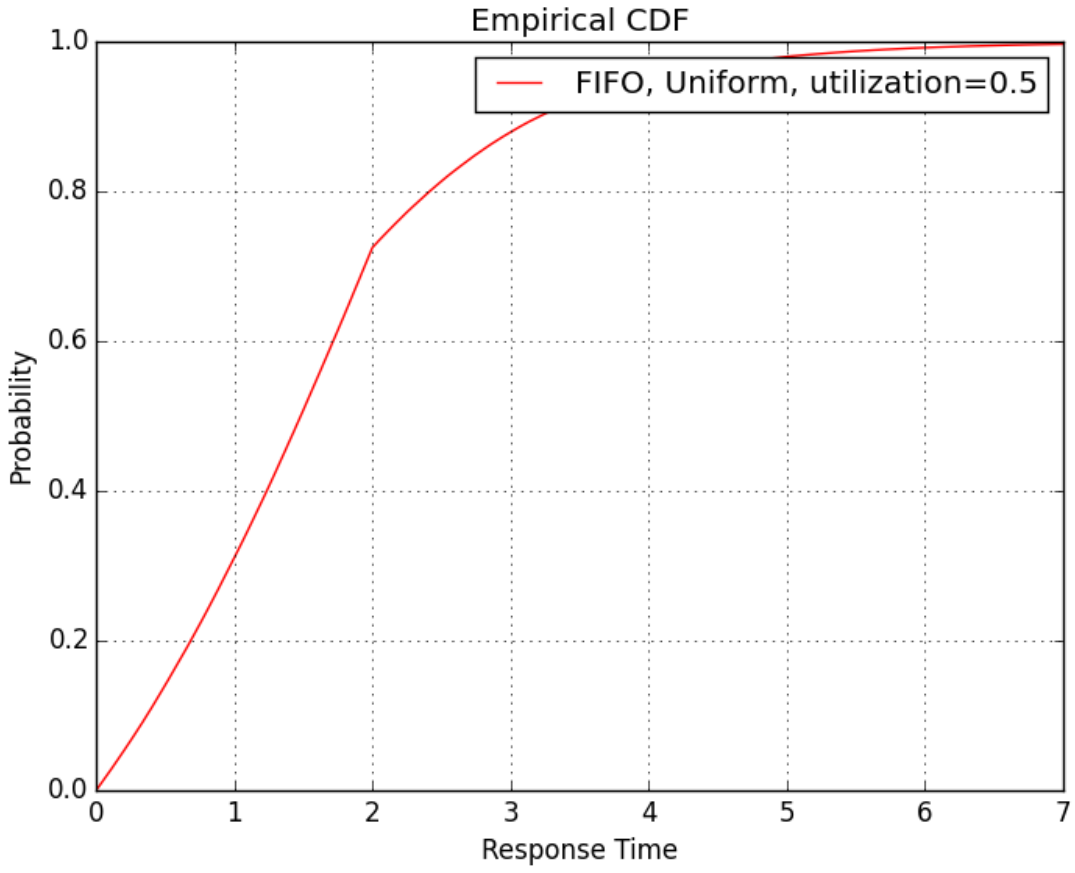

Fig.13

In case of a workload higher than this, some customers will have an unacceptable response time and will be lost.

The switch to a LJF policy in this case may increase the performance:

for example if the utilization factor in a given moment is 0.9, we have the following situation:

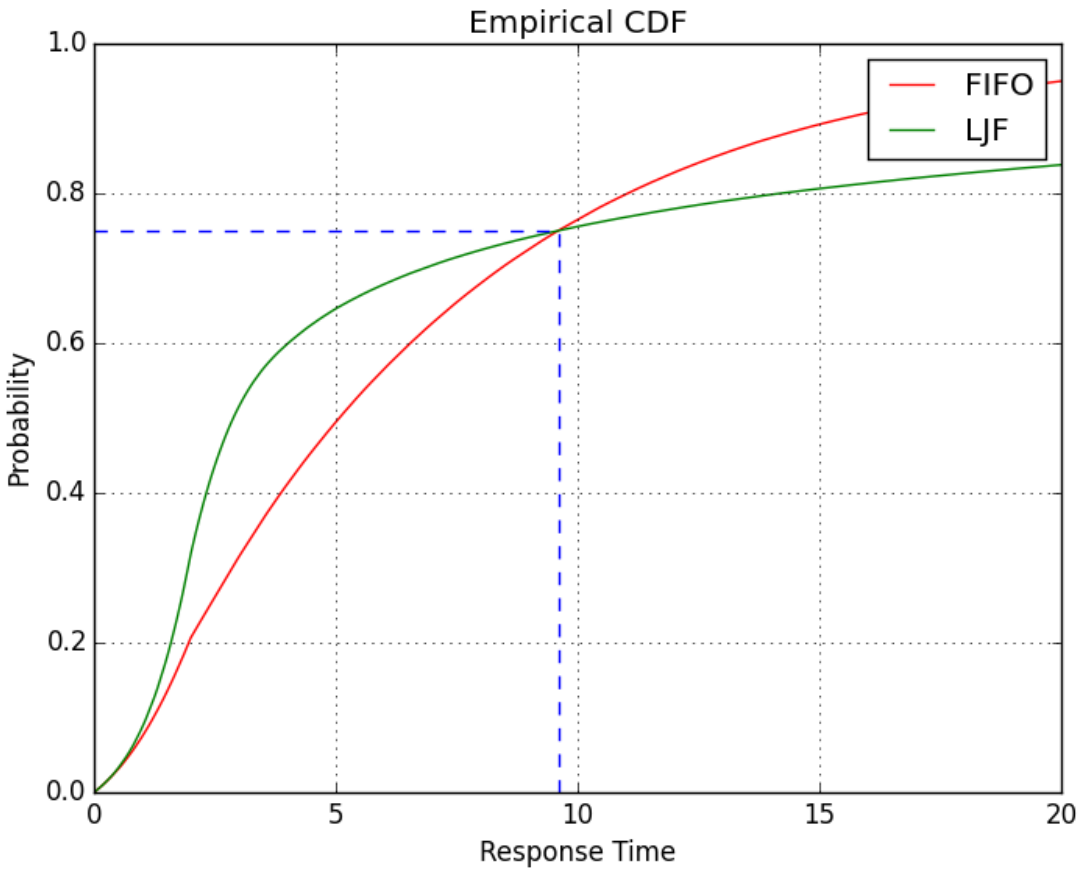

Fig.14

The intersection point between the two curves is higher than the threshold, so a larger number of customers will have an acceptable response time.

Appendix

In this appendix we discuss the validity of the computation of the confidence intervals for the plots of fig. 3,4 and 8.

Every set of parameters has been run for 30 times with different seeds.
The distribution of the random variable *sample mean X* has been approximated as a Student's T distribution with 29 degrees of freedom.
Thus the formula for the confident intervals is:

$$\left[X - \frac{S}{\sqrt{N}} \cdot t_{\alpha/2,N-1}, X - \frac{S}{\sqrt{N}} \cdot t_{\alpha/2,N-1}\right]$$

With S = sample variance, $t_{\alpha/2,N-1}$ = quantiles of the Student's T distribution with N degrees of freedom, N=29 and α=0.01.

The hypothesis of the Student's T distribution for the sample mean has to be tested, and this can be done via Q-Q plots.
The Q-Q plots have been checked for all the values reported on the plots, but here only the critical cases will be shown.
The first one is for parameters μ=0.99, queuePolicy=LJF,inter-arrival distribution=exponential:

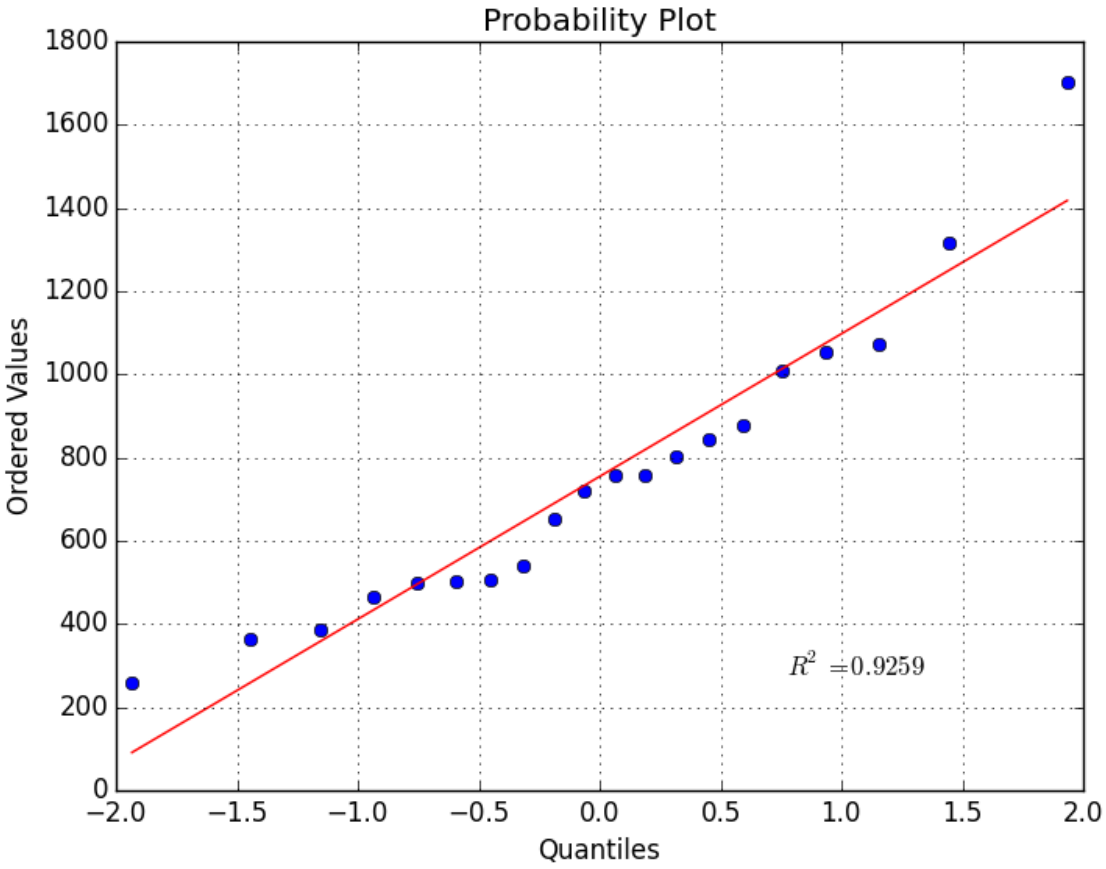

Fig.15

As it can be seen there is an outlier, but the overall behavior is linear.

The second critical case if is for parameters μ=0.99, cvar=2, queuePolicy=LJF,inter-arrival distribution=log-normal:

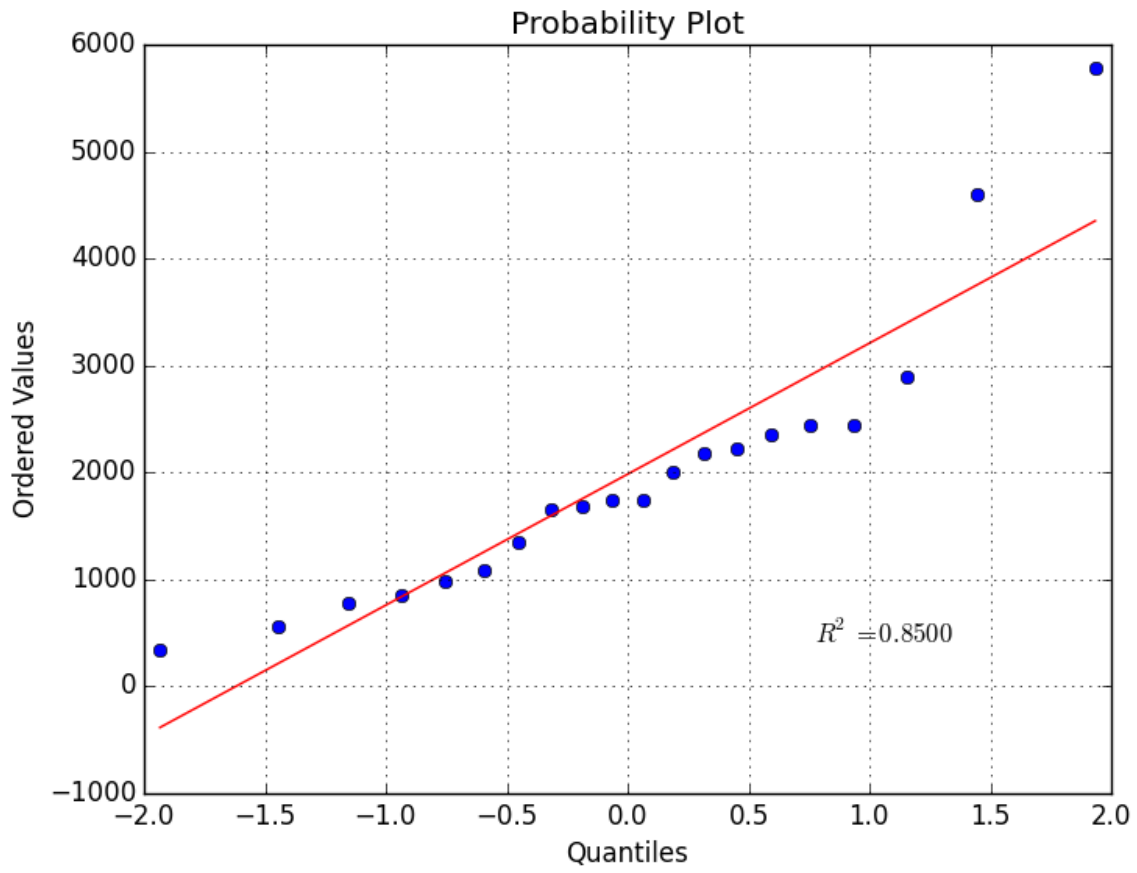

Fig.14

In this case there are more outliers, but the overall behavior is again linear.
Apart from these 2 cases all the mean value distributions show almost perfect linearity, thus the assumption of a Student's T distribution is acceptable (and of course the confidence intervals).