M. Sc. in Computer Engineering
Performance Evaluation of Computer System and Networks

# Project 2
# Quick checkout

Francesco Paolo Culcasi
Alessandro Martinelli
Nicola Messina

Year 2015/2016

# Contents

# 1. Modeling

## 1.1 Description and Assumptions

The project consists in modeling a supermarket with $N$ tills. The goal of this paper is to study the characterizing aspects of the supermarket, both from the point of view of customers (queueing time and response time experienced by each one) and from the standpoint of the owner of the supermarket (till idle time and load distribution). For a clearer analysis following assumptions are taken for granted:

- A percentage $P$ of tills, named *quick-checkout* tills, are forced to serve only customers whose shopping cart holds less then $K$ items;
- The time elapsing between the entrance of one customer and the following in the supermarket (from now on *inter-arrival time*) is described by an exponential random variable, while the number of items a customer will buy (from now on *sevice demand*) may be described by an exponential or lognormal random variable;
- Once a customer enters the supermarket he/she chooses the till with the smallest number of waiting clients among those where he/she is allowed to queue; more in details, a client bringing less than $K$ elements can be queued up to any till; instead a client with more than $K$ elements can be queued up only to non quick-checkout tills;
- If two or more tills have the same minimum number of clients waiting on them the new customer chooses one of them at random;
- If the number of waiting clients is 0, the arriving customer goes directly under service, instead if the till is not empty he/she will queue up on it. The queue length can grows to infinity and the customers cannot leave their queue;
- If the percentage of quick-checkout tills is 100%, customers with a number of items greater or equal to $K$ will be not allowed in the supermarket;
- The time an item needs to be elaborated (from now on *unit service time*) is constant, so the total time a customer need to be served is proportional to the number of items he/she is purchasing.

## 1.2 Building the System

### 1.2.1 Modeling

The supermarket model is composed by a Compound Module `Supermarket`, which contains three submodules, as shown in figure 1.1:

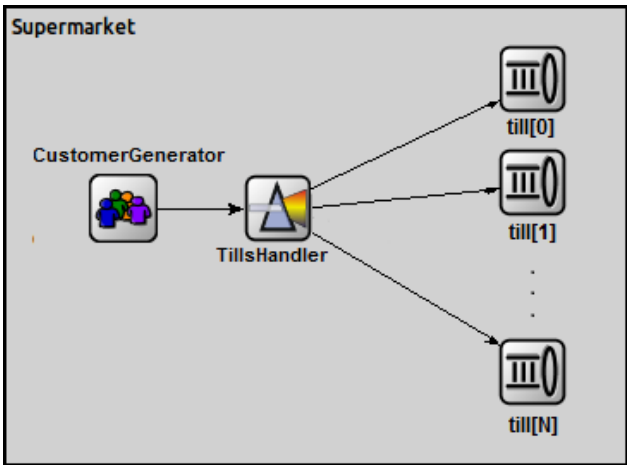


Figure 1.1: Supermarket Network

The `CustomerGenerator` module performs the task of creating customers, who are forwarded by `TillsHandler` module in the appropriate `Till`, where they are inserted in the FIFO queue and served at the appropriate time.

### 1.2.2 Implementation

Those three previous submodules are implemented in *Omnetpp++* as Simple Modules, having the following behavior:

#### CustomerGenerator

The purpose of `CustomerGenerator` module is to create customers with an exponential rate (inter-arrival time).

Customers are objects of the `Customer` class, that extends `cMessage`, and has two attributes, a `double` value that stores the number of items in the customer's cart and a `simtime_t` value for the simulation instant when the customer has been created. The `item number` field is filled with a random value (service demand) extracted from an aleatory variable with an exponential or lognormal distribution, according to the specific scenario.

The newly created customer will be sent to the `TillsHandler` module through the out gate and a new `Customer` creation is scheduled at an exponential random time after the current moment.

### Tills Handler

The tills handler looks up at the various queues and decides where to queue up the new customer, basing his decision on the number of items that the customer brings and on the number of customers already waiting in the queues (also called the *status of the till*). If the new customer brings less than $k$ elements, the tills handler addresses only the first $N \cdot p$ tills (the quick-checkout tills, as specified in the `Till` module).

The `Tills Handler` module is internally implemented as a `vector` that contains references to all the $N$ tills. For every arriving customer, the policy above is applied; to find the till with the minimum number of customers in queue, a `comparator` for the `vector` is used, applying the following policy: *till1 is lesser than till2 if the number of customers in till1 queue is lesser than number of customers in till2 queue.*

To avoid the problem that arises when there are two or more tills with the same minimum number of customers waiting on it – i.e. in that case only the first till among them is always returned – each till exposes, every time a new arrival occurs, a random number extracted from a uniform distribution to randomly break the parity. This way we obtain a fair distribution of the customers among the tills.

### Till

Tills are implemented internally with an `STL queue` of `Customer`. If the queue is not empty when a new arrival occurs, the incoming customer is queued up; otherwise he enters directly under service.

Considering the assumptions, items are processed with a constant unit service time, so that the total employed time for a customer is $unitServiceTime \cdot numberOfItems$. This quantity of time, increased by the current simulation time, is used within the `scheduleAt` function as wake up time argument, and it is calculated in the `getEndServiceTime()` function.

## 1.3 Verification

The System behavior has been tested under some interesting scenarios to proof the implementation correctness:

**Case $p = 0$:** only normal tills. In this scenario, at the steady state, all the tills receive approximately the same number of clients since our splitting policy ensures a new customer arrives in a till only if all the other tills have reached the same status. The Lorenz Curves 1.2 confirm this, showing an almost perfect bisector.

**Case $p = 1$:** only quick-checkout tills. No customers with more than $k$ items will be allowed in the supermarket, they will be dropped.

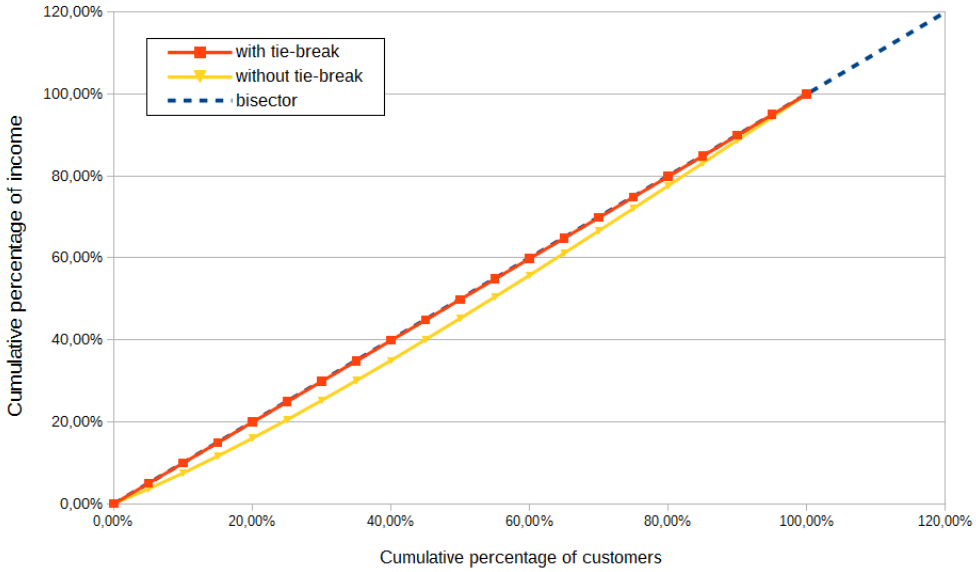


Figure 1.2: Lorenz Curve - customer dispersion among tills

**Case** $0 < p < 1$**:** both type of tills are present. In this scenario all customers will be queued up according to our splitting policy, and tills of the same type will approximately experience the same number of customers.

**Constant service demand**

- $SD \geq k$: in this case no customer will be queued up in the quick-checkout tills.
- $SD < k$: in this case every till could be accessed. Thus, all the tills will experience the same average number of arrivals. In this case p is not influent.
- Constant $IA$ with $p = 0$ and whatever $SD$: this scenario considers the simplified case of a probabilistic splitting and the limit condition for the saturation of the tills is going to be checked ($\rho = 1$), resulting in the formula

$$SD = \frac{IA \cdot N}{unitServiceTime} \tag{1.1}$$

Simulating the scenario using the following values, $IA = 2s$, $N = 20$ and $unitServiceTime = 2s$, we find $SD = 20$ as a limit condition for the saturation, which is the same result the above formula returns. In this case every customer enters in service immediately (waiting time of $0s$).

If we stress the system by setting $IA = 1.99s$ we're able to predict the waiting time for each customer: every client anticipates his arrival in the supermarket by $\frac{1}{100}s$, therefore each customer's arrival in the same till (1 each 20) will be anticipatet by $\frac{20}{100}s$. The waiting time of the $j^{th}$ customer is equal to $j \cdot \frac{20}{100}s$.

# 2. Calibration

In order to obtain reasonable results, some parameters have been set relying on a realistic supermarket environment (like the number of normal and quick checkout tills and the maximum number of items allowed in quick-checkout tills).

To estimate some parameters like the service demand and inter-arrival time, we take into account the $M/M/1$ model. Even though being a strong assumption, due to the deterministic splitting of our model, this is an important scenario: the inter-arrival time at a particular till of our scenario is always greater than or equal to the correspondent value in the $M/M/1$ scenario (Chapter 4 for more details). In this case, the number of customers queued up at each till is even larger in the $M/M/1$ model with respect to our system. Thus, the $M/M/1$ system sets an upper bound for our system backlog before saturation, so we can use it to verify the chosen parameters, i.e. the mean service demand and the mean inter-arrival time.

For the analysis of this system the number of tills is fixed to 20 and the unit service time (time required to process an item) to 2 seconds. The inter-arrival time has also a fixed value, 3 seconds. This is because, assuming a fair splitting, its variation will be strongly amplified by a factor $N$ (the number of tills).
The others parameters are chosen so that the system is in a stable state, with low probability to be empty.

The analysis for fairness and tills status are performed only on a specific value of $K$, when the quick checkout tills are present: as will be shown in the analysis, scenarios with high $K$ are not really significant, due to the fact that there would not be a meaningful behavioral separation between normal and quick checkout tills, thus causing the system to become unrealistic.

The $P$ value has been chosen so that also the quick checkout tills are still likely not empty. If $P$ is high and the probability that few items are generated is low, then on one hand the low number of normal tills will experience a lot of traffic, due to the fact that they have to serve a lot of clients (mandatory those who have $\#items > k$ and, with a certain probability, also the customers that have $\#items < k$). On the other hand, the large number of quick checkout tills would have to serve few clients, keeping themselves underutilized.

# 3. Analysis

## 3.1 Scenarios Definition

The analyzed scenarios relies on two orthogonal different cases:

- only normal tills ($P = 0$) $\leftrightarrow$ normal tills + quick-checkout tills
- exponential service demand $\leftrightarrow$ lognormal service demand

So 4 different analysis are performed, one for each set of the above cases. For each of them, warm-up analysis and fairness among the tills of the same type are performed. The results in every scenario show that the load is equally distributed among the tills, both in the number of total served customers, and in the statistical properties of the queueing and inter-arrival times, as proof of the fairness of the splitting policy. Thus, in prevision of collecting statistics, a till at random can be picked up to record statistical values.

The behavior of quick-checkout and normal tills couples can be compared considering different $K$; at the same time it is possible to compare normal tills of the normal and quick-checkout tills scenario with the ones of undistinguished tills scenario.

In order to compare scenarios differing from the service demand distribution, the mean of the lognormal and exponential distributions is kept the same in order to obtain a significant comparison.

## 3.2 Scenarios Analysis

### 3.2.1 Exponential service demand, undistringuished tills

Even though supermarket inter-arrivals are exponential, a till inter-arrival does not look like exponential. Increasing the load due to increase of the service demand, the probability that a customer arrives at very small times is lower than in an exponential PDF. This is due to the fact that a new customer is queued up to a till when all the other tills reaches the same till status. This is not obviously done in a small time, because the same reasoning must be done for all the other tills. In low load scenario this effect is amplified because the tills are more likely to be empty.

Although the non-probabilistic splitting results in non-exponential till's inter-arrivals, still turns out that the mean inter-arrival time in each till is equal to the supermarket inter-arrival time $1/\lambda$ multiplied by $N$.

As expected, the results confirm the intuitions: increasing the service demand the probability of having higher queueing time and response time grows up, thus the performances deteriorate.

Analyzing the different $p_0$ and $r_0$ probabilities, this is obviously a non-PASTA system: $r_0$ is even greater than $p_0$. It means that an arrived customer is more likely to find the queue empty at his arrival.

It is noteworthy that every EPDF for queueing time (Figure 3.1) shows a delta in 0 seconds with area equal to the probability of finding and empty system when a customer arrives, i.e. $r_0$.

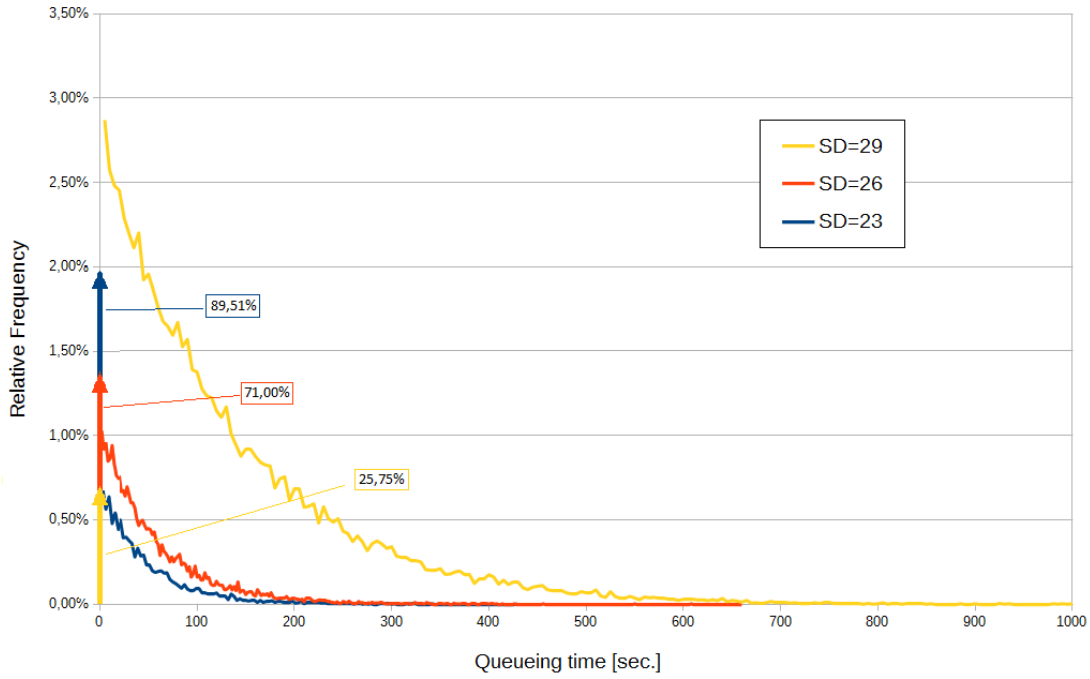


Figure 3.1: Queueing time EPDF on varying workload

### 3.2.2 Exponential service demand, normal and quick-checkout tills

From the analysis of queueing and time, increasing $K$, an increase for the queueing time of quick-checkout tills is observed. This denotes the progressive increase in load of the quick-checkout tills: in fact, increasing $K$, the probability for customers to have less than $K$ items increases exponentially, due to the exponential service demand distribution (Figure 3.2).

Looking at the chart on Figure 3.2, intuitively we can say that the normal tills queueing time changes faster than the quick-checkout one, at least for low values of $K$. To better understand this behavior we have to underly the difference between the probability to "hold less then $K$ items" (probability to be a "privileged customer")

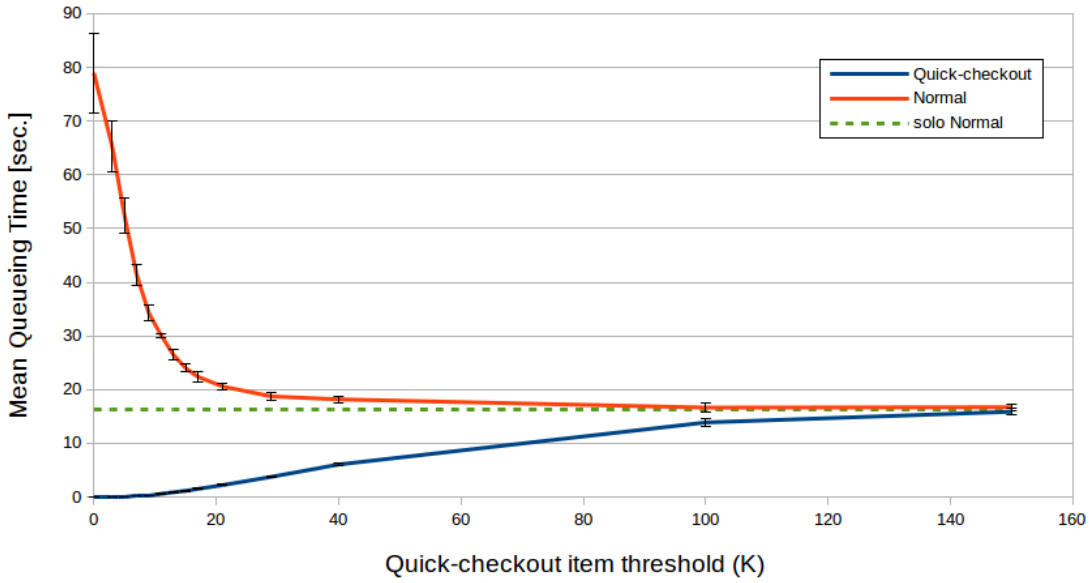


Figure 3.2: Queueing time on varying K

and the probability to "be queued in a quick-checkout till". An increase of K does not mean an higher load in quick-checkout tills, because privileged customers, relying on the current splitting policy, **may or may not** choose a quick-checkout till. This means that quick-checkout tills load does not grow as fast as expected. However the improvement of normal tills is boosted by privileged customers who choose normal tills, that, with their low service demand, reduces the average queueing time.

However, the trend is the conjunction between the queueing times of the 2 till types. In fact, increasing $K$, the quick-checkout tills behavior is more likely the normal tills one, due to the fact that they are distinguished by a certain probability to have still more than $K$ items, very low if K is high (due to factor $e^{-K/u}$). Both queueing time and response time for normal and quick-checkout tills converge to queueing and response time of the undistinguished tills scenario.

### 3.2.3 Lognormal service demand, undistinguished tills

This scenario is evaluated so that the mean service demand is the same as the *Exponential service demand with undistinguished tills* scenario, to better compare the conduct of the system. We observe that system behavior is very similar to the case with exponential service demand.

The only considerable difference consists in service demand EPDF shape: in lognormal scenario there is lower probability to obtain a number of items close to zero.

Studying this system for different workloads, under varying service demand distribution scale, an expected result is observed, i.e. with lower (higher) mean service demand we have higher (lower) probability to obtain low values of queueing time

(Figure 3.3) and a lighter (heavier) tail for response time.

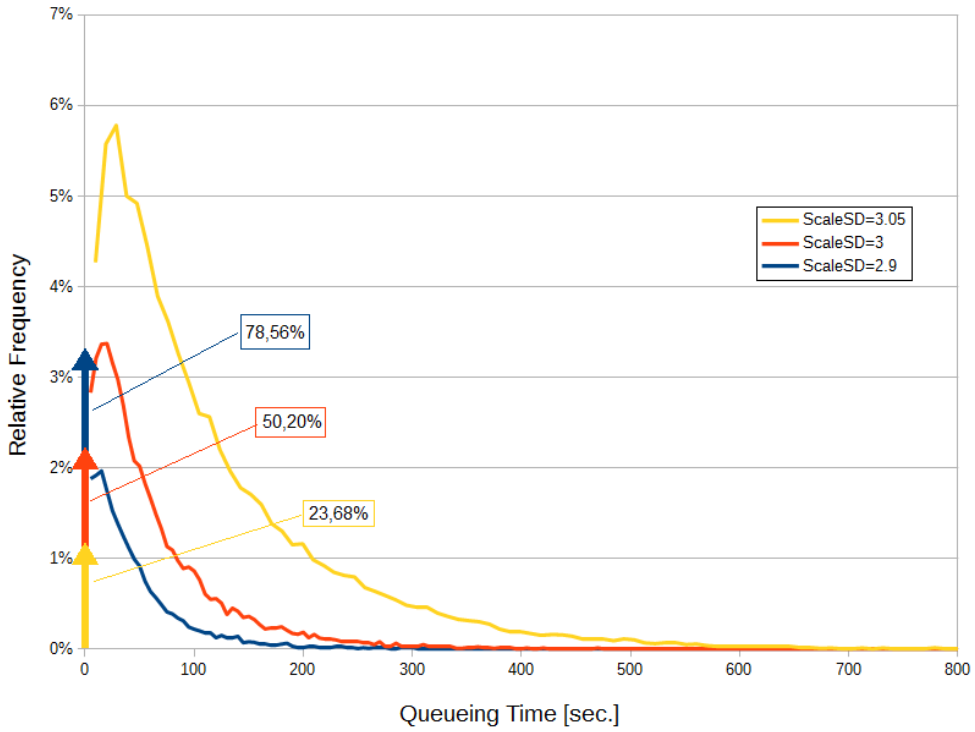


Figure 3.3: Queueing time EPDF on varying workload

### 3.2.4 Lognormal service demand, normal and quick-checkout tills

Also in this case, the behavior of the system is quite similar to the exponential with normal and quick-checkout tills. The accountable results concern the comparison of tills utilization and the total number of customer with respect to the exponential scenario.

The main difference between exponential and lognormal service demand distribution consists in a lower probability to obtain values to be near to zero in the latter. Considering this, quick-checkout tills receive a portion of the customers with service demand lower than $K$. The mean service demand for quick-checkout customers will be lower in lognormal service demand case compared to the exponential service demand (Figure 3.4). Thus, quick-checkout tills in exponential scenario look emptier and with a higher idle time, due to the fact that customers queued there, having a lower mean service demand, occupy the till for a shorter time (i.e. they have a higher idle time) compared to the scenario with a lognormal service demand.

This explains also why the percentage of customers queued up in quick-checkout tills is higher in the exponential scenario: on average customers are processed faster, thus tills are empty earlier and will be chosen by $TillsHandler$ beforehand than the lognormal scenario (Figure 3.5).

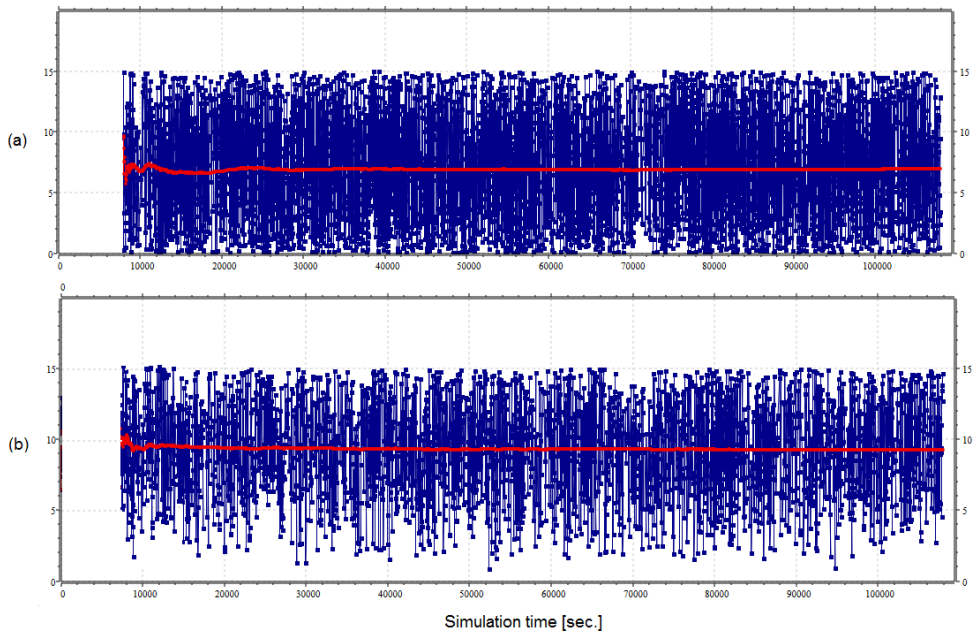


Figure 3.4: Service demand in quick-checkout tills with sliding window average: (a)Exponential SD (b)Lognormal SD

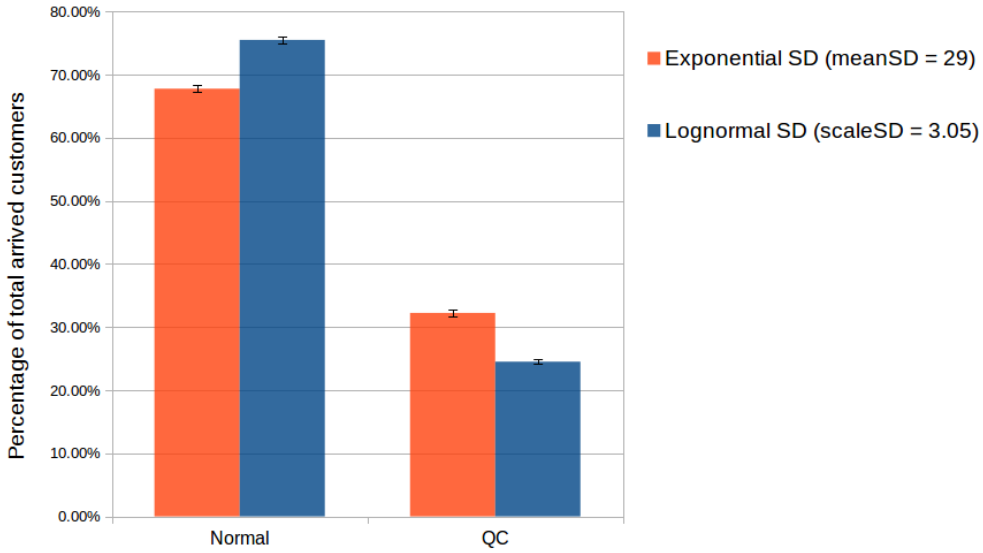


Figure 3.5: Total arrived customer of normal and quick-checkout tills: exponential versus lognormal

# 4. Calculations and Theoretical Results

## 4.1 Confidence Intervals

The confidence level in all the collected statistics are computed taking in consideration 30 repetitions of the same statistic. However, the IIDness of samples must be taken into account for the confidence intervals to be corrected calculated:

- For statistics directly collected from simulation from different repetitions, the IIDness is not checked: the different repetitions are IID pseudo ramdom variables generated by the RNG.
- Otherwise, the IIDness is explicitly checked through correlograms.

Using 30 repetitions, and having checked IIDness, we can assume that the 30 samples are normally distributed.

## 4.2 Model fitting attempts

Analysing the results and trying making some prevision using known models, it results that no noticeable theoretical distribution is involved in this particular model.

For instance, let's try to assume the supermarket as an Open Jackson Network model: thus, the splitting of customers among the various queues is assumed probabilistic, in this case with uniform probability. The various tills perhaps can be modeled as $M/M/1$ systems (initial hypotheses we used as an upper bound for system calibration).

Though if our model presents a quite good exponential inter-arrival times and service times, the expected values, such as $E[N]$ or $E[R]$ do not hold. The $r_0$ (tagged arrivals) and $p_0$(external observer) probabilities, for each n, are not the same. In fact, it is much more probable that an arriving customer finds the queue empty than an external customer finds the queueing empty at a random time.

Generalizing, all the $r_n$ and all the $p_n$ can be exploited, measuring them directly from the till status over time. We know that the ratio $r_n/p_n$ is exactly $\lambda_n/\bar{\lambda}.$, so we can extract $\lambda_n$ chart over n. The $\lambda_n$ sequence presents the shape of an hyperbole, so the discouraged arrival model can bring the analysis in the right direction. Fit doesn't give good results for standard discouraged model; instead, the sequence can

be fitted quite well using a variation of discouraged arrival model:

$$\lambda_n = \frac{\lambda}{(n+1)^\alpha}$$

where $\alpha$ is about 2. Using Chapman-Kolmogorov equations, we're able to find a closed form for $P_n$. From this we can calculate the mean number of clients in queue, in function of $\frac{\lambda}{\mu}$:

$$E[n] = \frac{\sum_{n=0}^{\infty} \frac{n \cdot \left(\frac{\lambda}{\mu}\right)^n}{(n!)^2}}{\sum_{k=0}^{\infty} \frac{\left(\frac{\lambda}{\mu}\right)^k}{(k!)^2}} = \frac{\sqrt{\frac{\lambda}{\mu}} I_1\left(2\sqrt{\frac{\lambda}{\mu}}\right)}{I_o\left(2\sqrt{\frac{\lambda}{\mu}}\right)}$$

where $I_n$ is the modified Bessel function of the first kind. $E[N]$ in function of $\frac{\lambda}{\mu}$ is quite different from the Kleinrock one, confirming our model to be quite different from M/M/1. $E[N]$, perhaps, confirm our initial assumption while comparing our system with $M/M/1$: our system will become busy at a larger utilization factor, as shown in Figure 4.1.

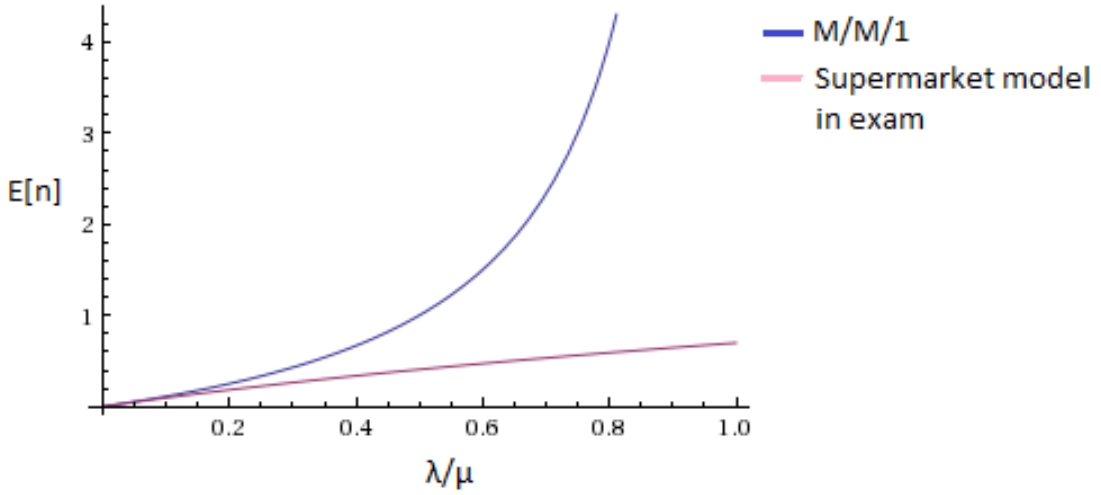


Figure 4.1: Mean number of customers in a till upon $\frac{\lambda}{\mu}$

Due to the fact that also the discouraged arrival exponent $\alpha$ presents fluctuations with change in service demand, the presented results are still not a perfect modelling of the system.