Computer Engineering

Performance Evaluation of Computer Systems and Networks

Evaluation of Supermarket Performance with quick checkout tills

TEAM MEMBERS:

Fabrizio Lanzillo

Riccardo Sagramoni

Niko Salamini

Academic Year: 2021/2022

Contents

Introduction:

Problem Description:

A supermarket has n tills. Two policies can be enforced for checkout:

a) **Undistinguished tills**: any customer can check out at any till.

b) **Quick-checkout tills**: a percentage p of tills is reserved for customers whose shopping cart holds less than k items (k being the quick-checkout threshold).

In both cases, a customer queues up at the till with the smallest queue among those where (s)he is allowed to queue. Consider the following workload: customer inter-arrival times are IID RVs (to be described later), their service demands (i.e., checkout times) are IID RV (to be described later). The percentage of quick-checkout tills can be varied (but stays constant in a single simulation), and so does k. Compare the queueing and response time of the two options under a varying workload. More in detail, at least the following scenarios must be evaluated:

- Exponential distribution of interarrival times and service demands.

- Lognormal distribution of service demands.

In all cases, it is up to the team to calibrate the scenarios so that meaningful results are obtained.

Objectives:

The objectives of the project are:

- Compare the case of “Undistinguished tills” against the “Quick-Checkout” case.
- Study the behavior of the quick-tills case with different combinations of p and K.

Performance indexes:

To assess the performance of the system we took as a reference:

- The Mean Response Time $\mathrm{E}[\mathrm{R}]$: defined as mean time between the arrival of a cart and its checkout.
- The Mean Waiting Time $\mathrm{E}[\mathrm{W}]$: defined as mean time between the arrival of a cart and its turn to be processed.

Modeling

General Assumptions

These are the general assumptions that have been made:

- The shopping carts of the supermarket coincide with the jobs of the system;
- The service time of the cart is defined as the number of the items multiplied by a unit of time (3 seconds).
- The time between the arrival of two different customers in the supermarket (inter-arrival time) is described by an exponential random variable, while the service times are distributed as exponential or lognormal random variable.
- A shopping cart is assigned to the till with the least number of carts in the queue, and it cannot change its till.
  The tills are represented as FIFO queues of unlimited capacity.
  In case of a tie between two or more tills, the till assigned will be the one with the minimum id.

Preliminary Validation

Before the implementation, a preliminary validation is necessary to verify that the model is correct. In order to do so, the previous general assumptions should be analyzed.

- The unlimited capacity of the tills is a reasonable assumption, because in the real world people would queue themselves even outside of the supermarket if necessary, thus making the queues approximately unlimited.
- Moreover, except for sporadic scenarios with high load (e.g. Christmas Eve), supermarkets usually don't exceed their maximum capacity. The assumption of the cart assignation to the shortest till, is reasonable, because a user, in most of the cases, choose the shortest till.
- Because we assume that each item is served with a specific unit of time, the service time of a chart is a sum of the times required to process every single item. We estimate the number of elements by dividing the response time by the unit of time and if necessary we approximate the last value by excess.

Factors

The following factors may affect the performance of the system:

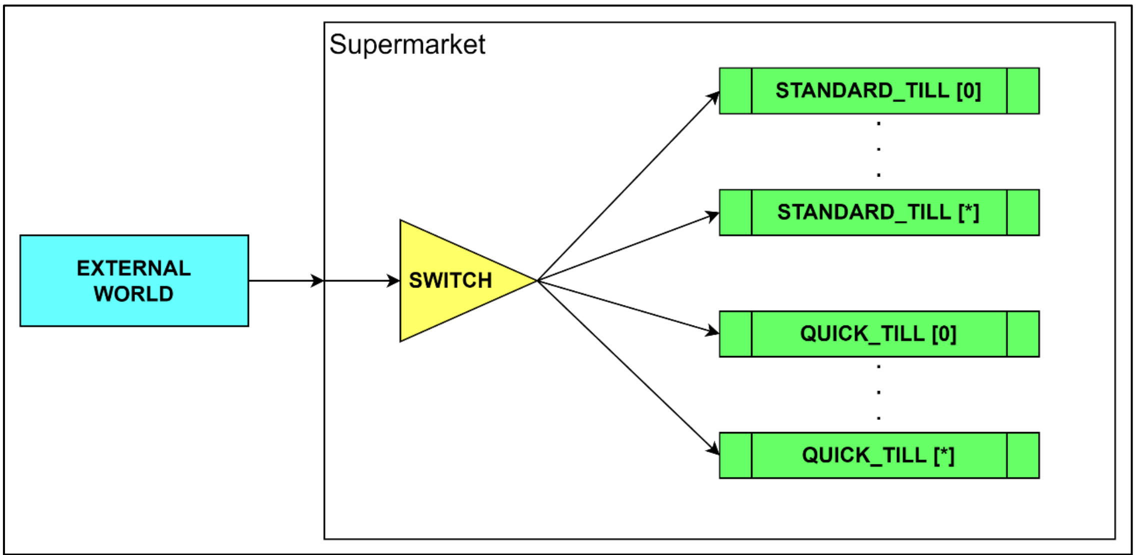
- **n**, number of tills in the system
- **k,** the quick-checkout threshold
- **p**, percentage of tills reserved for customers whose shopping cart holds less than k items
- $\frac{1}{\lambda}$, mean of the interarrival time distribution
- $\frac{1}{\mu}$**,** mean of the exponential service time distribution
- **M, σ,** lognormal distribution parameters

Implementation

Modules

The following modules have been defined:

- **ExternalWorld**: simple module which randomly generates the carts, according to the specified distributions, and send them to the Supermarket module as messages.
- **Supermarket**: compound module which represents the system under analysis. It composed by:
  - **Switch**: simple module which routes the received carts in a suitable till for its type.
  - **Till**: simple module that simulates the behavior of a supermarket till, by implementing a FIFS queue.



Design of Omnet++ project
"ExternalWorld" module is cyan; "Switch" module is yellow; "Till" modules are green

Messages

We defined a new format of messages to properly represent the carts. The new format, named "CartMessage", is a wrapper for a single simtime_t field, which represents the cart's service time.

Modules' behavior

External World

- Generate a random number from the selected interarrival distribution (exponential or constant for testing) and set a timer accordingly.
- When the timer goes off, generate a random number from the selected service time distribution (exponential, lognormal or constant for testing).
- Send a message to Supermarket with the generated service time
- Repeat

Switch

- When it receives a message from ExternalWorld, it reads its service time, checks it against the threshold (if in a quick-checkout scenario) and forwards the message to a Till object:
    - If optimized_routing is false, choose a random till between the suitable ones (*only for testing*).
    - If optimized_routing is true, choose the till with the smallest queue where the cart is allowed to queue.

- Depending on the scenario, the suitable tills where a cart is allowed to queue can be different:

	Service time **lower** than the threshold	Service time **higher** than the threshold
Undistinguished tills	All the tills	All the tills
Quick-checkout tills	All the tills	Only the “standard” tills

- The Switch is internally organized as two STL vector of Till pointers, one for each category of tills (standard and quick). During initialization phase, a pointer for each allocated Till is obtained and is put inside the corresponding vector. In order to select the smallest queue, the Switch calls the function getNumberOfJobs() provided by the Till class, which returns the current number of jobs of the till. The algorithm has linear complexity.

Till

- It provides the function getNumberOfJobs(), which returns the current number of jobs of the till.
- When it receives a message from Switch, the Till pushes it in a FIFO queue (std::queue). If the queue was empty before of that, start processing the message.
- In order to process a cart, the Till reads its service time and set a timer. When the timer goes off, **the Till extract the message/cart from the queue and delete it**. If the queue is not empty, it proceeds to process the new head of the queue.
- It also records statistics every time a message/cart/job is deleted (number of jobs and response time) or becomes the head and start to be processed by the Till (number of jobs in the queue and waiting time)

Verification

Degeneracy Test

The degeneracy test was used to analyze the behavior of the system when the parameters are set to 0 or to other extreme values. In every test the model worked in a correct way.
In particular, some observations can be made:

- If $p = 0$, we are in the case where we have only standard tills. In this case every till receives approximately the same number of shopping cart, because of our model polices.
  While if $p = 1$, then, we have only quicky tills. Therefore, in this case, no cart with more that K items will be in the supermarket.
- If $K = 0$, and $0 < p < 1$, there are both standard and quick tills. However, no cart will be queued in the quick ones, because every cart must have at least one item. On the other hand, if K is very big, then every cart can choose every till in an undistinguished way. Therefore, in this case, the role of quick tills loses its meaning.

Consistency Test

The consistency test verifies that the system and the output react consistently.
In particular, this test consists of two different sub-tests, that are described here, in which different values for the parameters were set:

Test 1

- $\boldsymbol{N}$ (Number of tills) = 4, $\boldsymbol{K}$ (quick-checkout threshold) = 1,
  $\boldsymbol{p}$ (Percentage of quick-checkout tills) = 0.4, $\frac{1}{\lambda}$ (Mean of the interarrival time) = 1.1,
  $\frac{1}{\mu}$ (Mean of the service time) = 1

Test 2

- $\boldsymbol{N}$ = 4, $\boldsymbol{K}$ = 1, $\boldsymbol{p}$ = 0.4, $\frac{1}{\lambda}$ = 2.2, $\frac{1}{\mu}$ = 2

We expect the behavior of the system to be more or less the same in both tests, because a system where a cart arrives every 1.1 seconds and will be served every 1 seconds, in equal to another one where a cart that arrives every 2.2 seconds is served every 2 seconds.
We set the number of tills to 4 and the percentage to 0.4, to have 1 quick till and 3 standard tills. We only changed the mean of the interarrival time and the mean of the service time between the 2 tests.

The plot in figure shows the results of the two different tests. We can see that the behavior of the system in the two cases is very similar, therefore, the system works as expected.

Continuity Test:

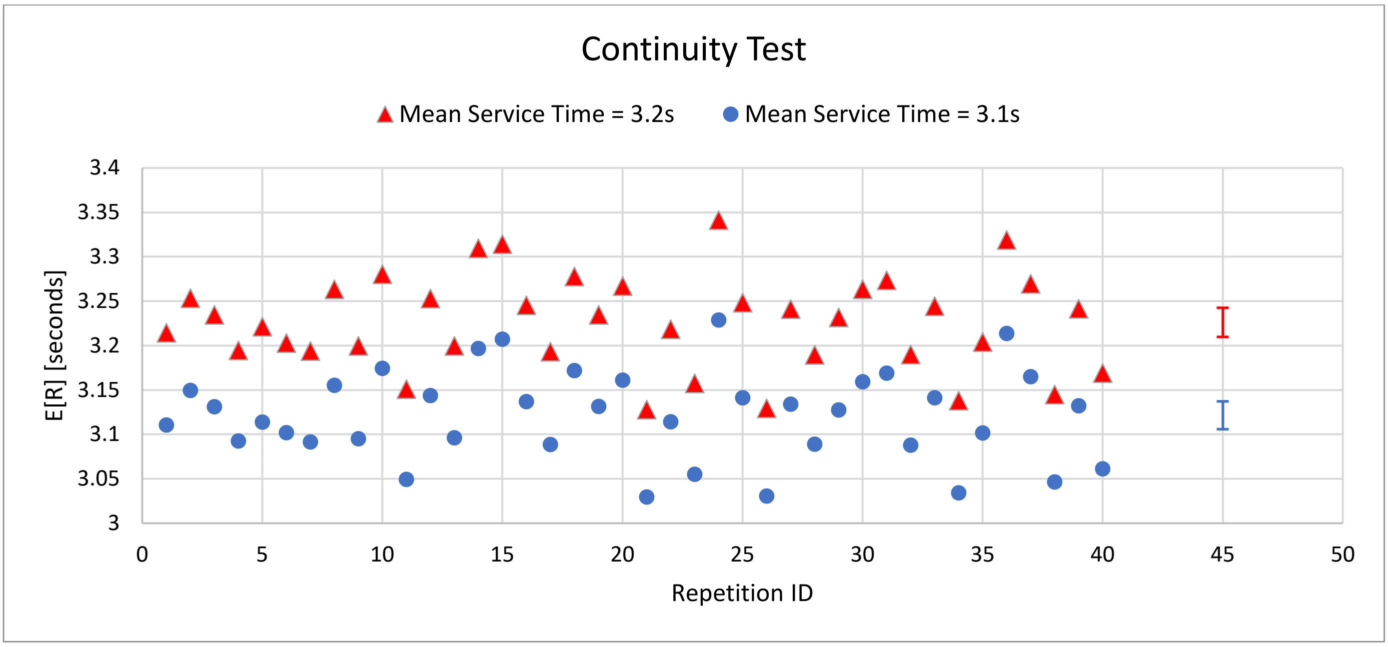
To assess the correctness of the system we must carry out an experiment to verify if changing slightly the input affects slightly the output. For example, we can compare 2 different configurations with a very similar service time and observe the output response time:

Configuration	**Mean Service Time**
1	3.1
2	3.2

All the other parameters have the same values for the 2 experiments:

- Number of tills: 5
- Service distribution: exponential
- Mean interarrival time $\frac{1}{\lambda}$: 3s
- Percentage of quick tills **p**: 0.4
- Quick checkout threshold **K**: 5s

The experiments consist in 40 repetitions:



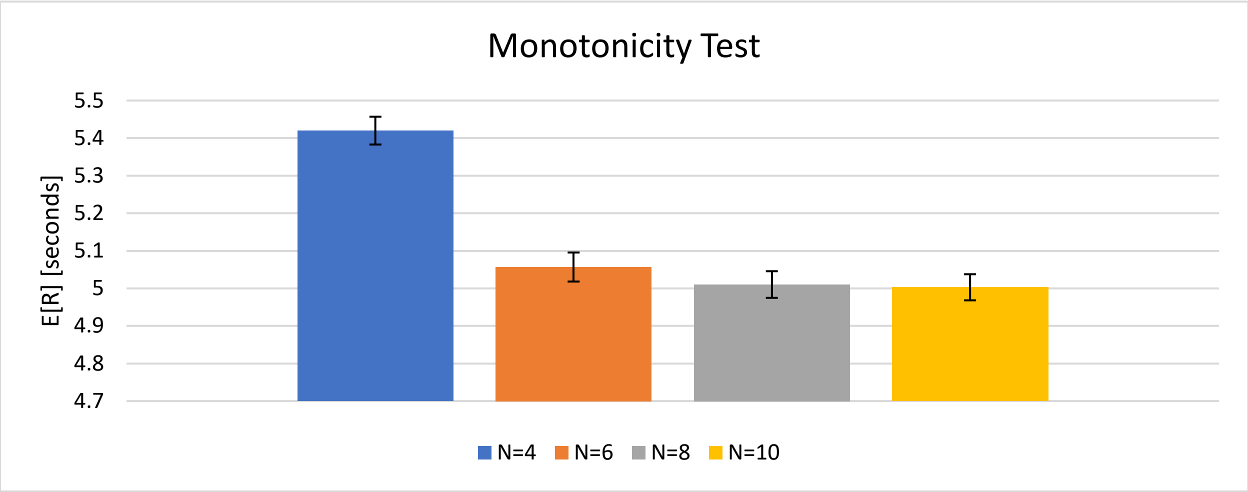
Observing the 95% confidence intervals we can see that the mean response time changes slightly between the 2 cases and so the system does not show any unexpected behavior. The same happens with a lognormal distribution of service time.

Monotonicity Test:

A second experiment to verify the correctness of the system is the monotonicity test. It consists in assessing the monotonicity of some performance indexes using different combinations of factors. In our case we can verify if increasing the number of tills, fixing the arrival and the service time, the mean response time decreases. We expect this behavior because with more tills the jobs are better distributed and so the mean response time fall off.

The fixed parameters in this case:

- Service distribution: exponential
- Mean interarrival time $\frac{1}{\lambda}$: 3s
- Percentage of quick tills **p**: 0.4
- Quick checkout threshold **K**: 6s



We can observe that with increasing N the mean response time decreases as expected and so the test is passed. The same happens with a lognormal distribution of service time.

Verification against theoretical model

The goal of this test is to compare the output of a simplified version of our system against a theoretical model.

In order to create a queuing model of our system, we made the following assumptions:

- All the tills are equals (undistinguished scenario).
- Random routing policy (i.e. a cart is allocated to an equally random till instead of one with the shortest queue).
- Exponential interarrival
- Exponential **or** lognormal service times

The main simplification from the real model is the random routing policy. This assumption holds because a non-optimized routing policy will have worse performance than an optimized one. Our theoretical results, then, can be generally considered as an upper limit of our actual performance.

We can model this simplified context as **n M/M/1** service centers (or **M/G/1** when the service times are lognormally distributed), where n is the number of tills. Thus, we can compute the expected metrics (E[N], E[Nq], E[R], E[W]) and compare them to the results from the simulations.

Since the tills are probabilistically chosen, each till receives cart distributed as an exponential with rate $\boldsymbol{\lambda/n}$, where λ is the interarrival rate of carts to the supermarket

Test 1: exponential service demands

In this scenario, the distribution of the service times is

$$\mathrm{f}(x) = \mu\, e^{\mu\,x}$$

where μ is the rate of the exponential distribution.

Thus, from the queuing theory we get for each till:

$$\rho = \frac{\lambda}{n\mu},\ \ \mathbf{E[N]} = \frac{\rho}{1-\rho},\ \ \mathbf{E[Nq]} = \mathrm{E[N]} - \rho,\ \ \mathbf{E[R]} = \frac{\mathrm{E[N]}}{\frac{\lambda}{\mathrm{n}}},\ \ \mathbf{E[W]} = \mathrm{E[R]} - \frac{1}{\mu}$$

We tested the model in the following scenario:

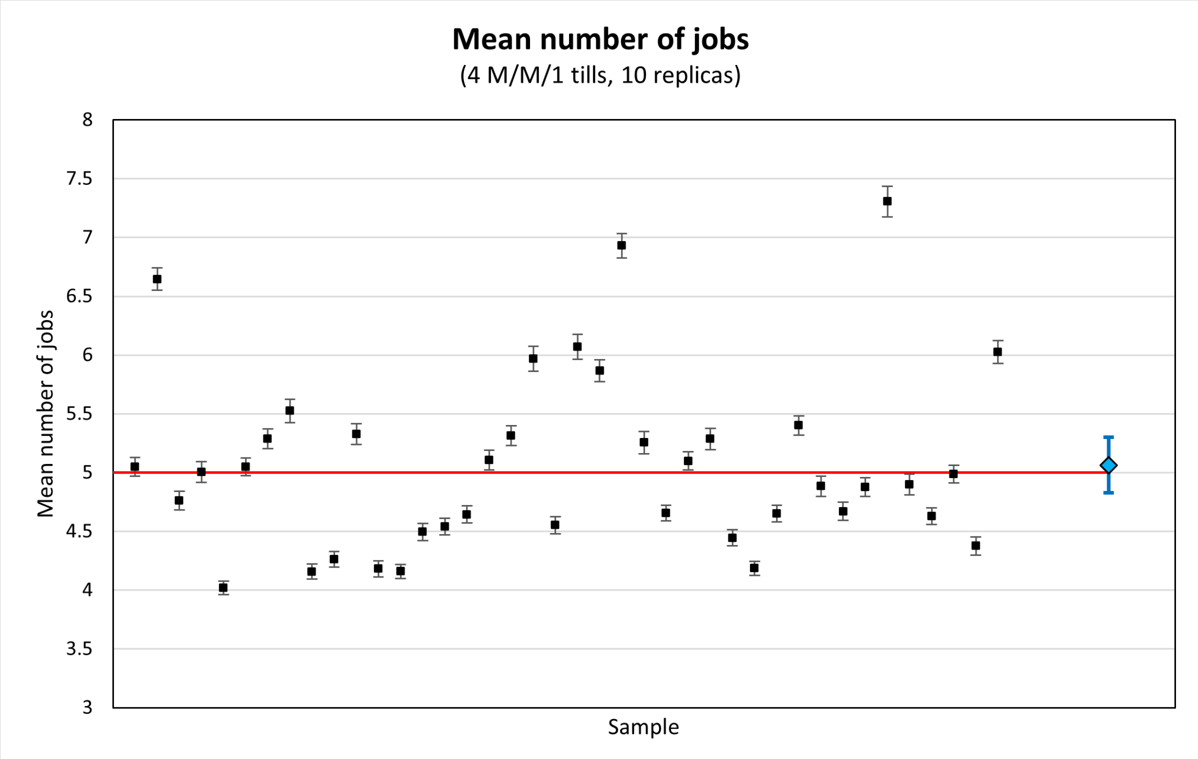
- Number of tills (**n**): 4
- Percentage of quick-checkout tills (**π**): 0%
- Mean inter-arrival time ($\mathbf{1/\lambda}$): 0.3
- Mean service demands ($\mathbf{1/\mu}$): 1.0
- Duration of the simulation: 10,000s
- Replicas: 10
- Seed-set: ${repetition}
- Confidence interval: 95%

$$\rho = 0.8333,\ \ \mathbf{E[N]} = 5,\ \ \mathbf{E[Nq]} = 4.1667,\ \ \mathbf{E[R]} = 6,\ \ \mathbf{E[W]} = 5$$

Since every till is modeled as a M/M/1 system, we computed the previously stated metrics for 40 different M/M/1 and we computed their mean and CI values between the tills.

The results we got are in line with our expectations, as showed in the following table and graphs:

Metric	Value	Confidence Interval	Metric	Value	Confidence Interval
E[N]	5.0632	0.2364	**E[R]**	6.0597	0.2712
E[Nq]	4.2278	0.2332	**E[W]**	5.0585	0.2691



Mean number of jobs of 40 M/M/1 with 95% confidence interval.
The black squares are the E[N] with its CI of the individual tills. The red line is the expected result from the theoretical model. The blue diamond is the mean of E[N] with its CI.

Test 2: lognormal service demands

In this scenario, the service time distribution is lognormal, i.e.

$$f(t_s) = \frac{1}{t_s\,\sigma\sqrt{2\pi}}\,\exp\left(-\frac{[\ln(t_s)-\mu]^2}{2\sigma^2}\right)$$

where μ and σ are the mean and the standard deviation of its logarithm, which is normally distributed.

Thus, we can model each till as a **M/G/1** system. According to the **Pollaczek-Khinchin formula**, the mean number of jobs in the system is:

$$E[N] = \rho + \frac{\rho^2 + \left(\frac{\lambda}{n}\right)^2 \cdot Var(t_s)}{2\cdot(1-\rho)}$$

where

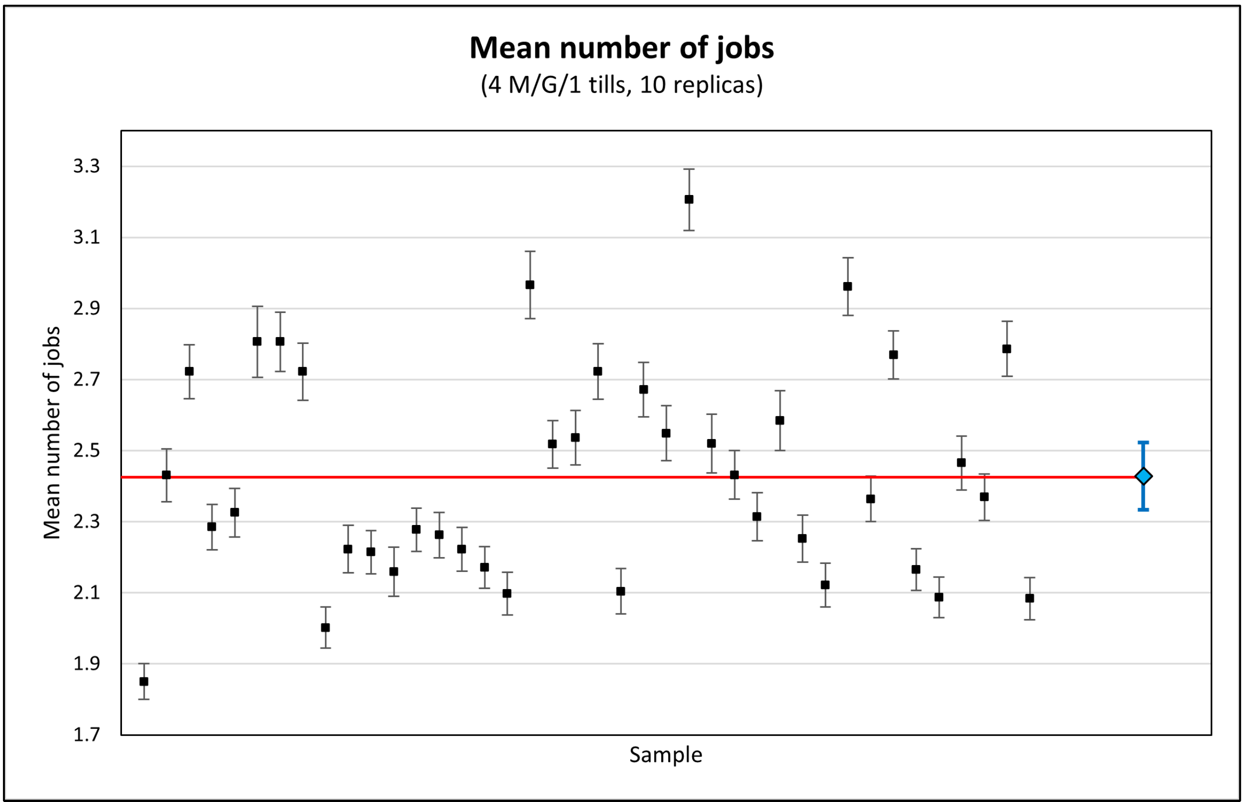
$$\rho = \frac{\lambda}{n}\cdot E[t_s],\;\; E[t_s] = e^{\left(\mu+\frac{\sigma^2}{2}\right)},\;\; \mathrm{Var}(t_s) = \left(e^{\sigma^2}-1\right)\cdot e^{(2\mu+\sigma^2)}$$

We tested the model in the following scenario:

- Number of tills ($\mathbf{n}$): 4
- Percentage of quick-checkout tills ($\boldsymbol{\pi}$): 0%
- Mean inter-arrival time ($\mathbf{1/\lambda}$): 1.0
- Mean of the logarithm of the service times ($\mathbf{1/\mu}$): 1.0
- Standard deviation of the logarithm of the service times ($\boldsymbol{\sigma}$): 0.5
- Duration of the simulation: 10,000s
- Replicas: 10
- Seed-set: ${repetition}
- Confidence interval: 95%

$$\rho = 0.7701,\ \ \mathbf{E[N]} = 2.4257,\ \ \mathbf{E[Nq]} = 1.6556,\ \ \mathbf{E[R]} = 9.7027,\ \ \mathbf{E[W]} = 6.6225$$

Since every till is modeled as a M/G/1 system, we computed the chosen metrics for 40 different M/G/1 and we computed their mean and CI between the tills.



Mean number of jobs of 40 M/G/1 with 95% confidence interval.
The black squares are the E[N] with its CI of the individual tills. The red line is the expected result from the theoretical model. The blue diamond is the mean of E[N] with its CI.

The results we got are in line with our expectations, as showed in the following table:

Metric	Value	Confidence Interval	Metric	Value	Confidence Interval
E[N]	2.4280	0.0945	**E[R]**	9.6670	0.3279
E[Nq]	1.6565	0.0899	**E[W]**	6.5873	0.3242

Calibration

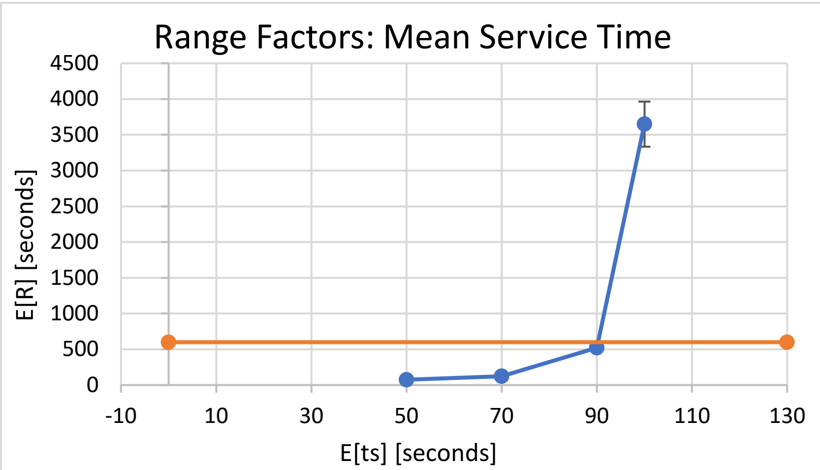
Factors Calibration

The aim of this part is to fix the intervals of the factors to correctly reproduce the behavior of the real system. We start from the observation that according to “Waiting time delays and customer satisfaction in supermarkets”, a study carried out by Gail Tom and Scott Lucey in 1995, the average checkout time was estimated at 4.38 minutes.

Making the hypothesis that this behavior is expected in supermarkets of average size we can carry out some experiments fixing the number of tills at 10 (medium size supermarket). The idea is to set the mean service time of the carts according to the fact that the checkout time cannot be over a certain threshold, we fix this threshold at 10 minutes (600 seconds). The experiments are carried out using the exponential distribution and with incremental values of the mean service time to find the explosion point (some confidence intervals were too tiny to be visible):

The fixed parameters in this case:

- Service distribution: exponential
- Mean interarrival time $\frac{1}{\lambda}$: 20s
- Percentage of quick tills **p**: 0.2
- Quick checkout threshold **K**: 60s



We can observe that the mean response time, in other words the checkout time, does not resemble the real case after a value of $\mathrm{E}[t_s]$ of 90 seconds where it becomes in the order of 3600 seconds (1 hour). We can set the mean service time’s upper bound to 90 seconds.

What about the lower bound? It can be any value over 0 but, recalling the hypothesis that each object takes 3 seconds to be served, we can set the lower bound to 15 seconds so the mean number of objects in a cart will be 5 in an extreme case.

All the other factors are set trying to resemble the real case:

- Mean Service Time $\mathbf{E}[\mathbf{t_s}]$: $[15\text{s}, 90\text{s}]$
- Mean Arrival Time $\frac{1}{\lambda}$: $[10\text{s}, 50\text{s}]$
- Quick_Checkout Threshold $\mathbf{K}$: $[6\text{s}, 60\text{s}]$
- Percentage of quick tills $\mathbf{p}$: $[0.2, 0.5]$
- Lognormal Parameters: $\boldsymbol{\mu} = [2, 3], \boldsymbol{\sigma} = [1, 1.75]$
- Number of tills: [5, 20]

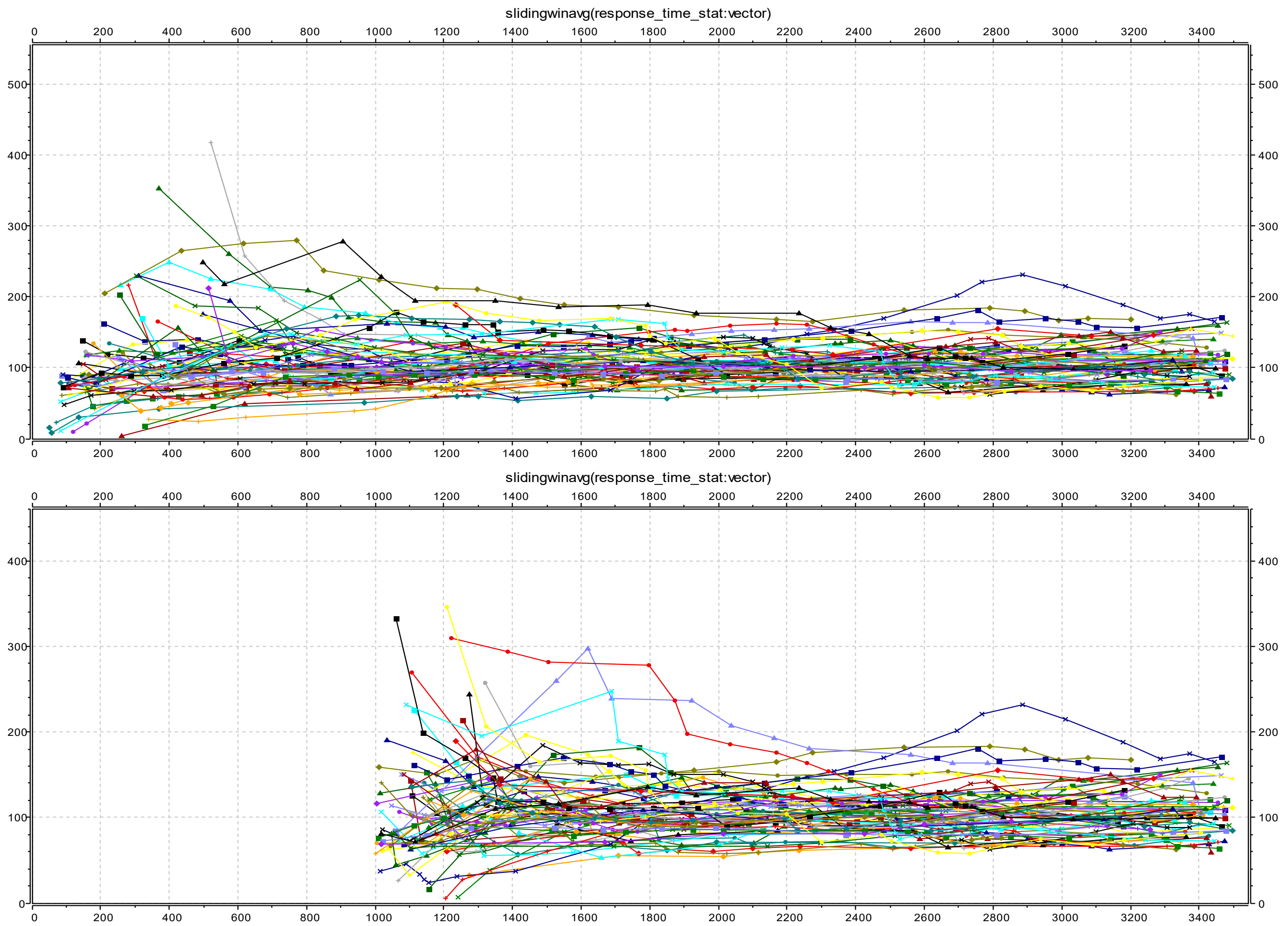
The lognormal parameters are set to fit the case studied on the exponential distribution, so to have a mean response time under the 600 seconds threshold.

Calibration of warm-up time and simulation time

In order to calibrate the warm-up time, we studied the evolution of the performance indexes during the early stage of the simulation. During our analysis, we found that the **response time** is the most reliable index for the warm-up, since it's never null (unlike the waiting time).

We tried different scenarios by changing the values of the factors and we found that a general warm-up time good enough for every scenario under study is **1,000 seconds**.

In fact, as shown in the graph below, the behavior of the response time before 1,000 seconds is unstable and significantly different from the behavior after the threshold. Even if the response time never actually becomes stable, it keeps oscillating inside a well-defined band.



Sliding average of response time (only standard_tills[*]):
no warm-up vs 1,000s warm-up time
10 tills; quick-checkout (p=30%, k=60); exponential interarrival (mean=10);
exponential service (mean=60); sliding window size = 10

For all the experiments we conducted, we decided to set the simulation time to **10,000s**. That would allow us to collect enough data to draw our conclusions.

Simulation Experiments

About the out-of-bound values of p:

The experiments are carried out considering out-of-range values of p, this allowed us to better explain why it is not realistic to use such percentages.

About the number of tills used in the experiments:

Given the hypothesis that the load of a supermarket is proportional to its size, meaning that a larger supermarket will have more customers and a higher mean number of items in the carts, the behavior of the system will be the same independently from its size. Because of this we can study a single case of size and apply the same reasoning for all the others, we choose the medium case corresponding to a number of tills equal to 10.

About the confidence interval:

We used a confidence interval of 95%, in the experiments we do not show them because they are too tiny to be visible if compared to the values of the KPIs.

The effects of the Quick-Checkout Threshold K on the system

The experiments consist in testing the effects of the quick-checkout threshold K with different percentages p of quick-tills for different loads on the system for both lognormal and exponential mean service time. The tests also consider the case of parameters out of the range set in the “Range Factor” paragraph, this is to show what happens. To reduce the number of possible experiments we focused on 3 different loads:

- Low: $\frac{1}{\lambda} = 50\text{s},\ \frac{1}{\mu} = 15\text{s}$
- Medium: $\frac{1}{\lambda} = 30\text{s},\ \frac{1}{\mu} = 54\text{s}$
- High: $\frac{1}{\lambda} = 10\text{s},\ \frac{1}{\mu} = 90\text{s}$

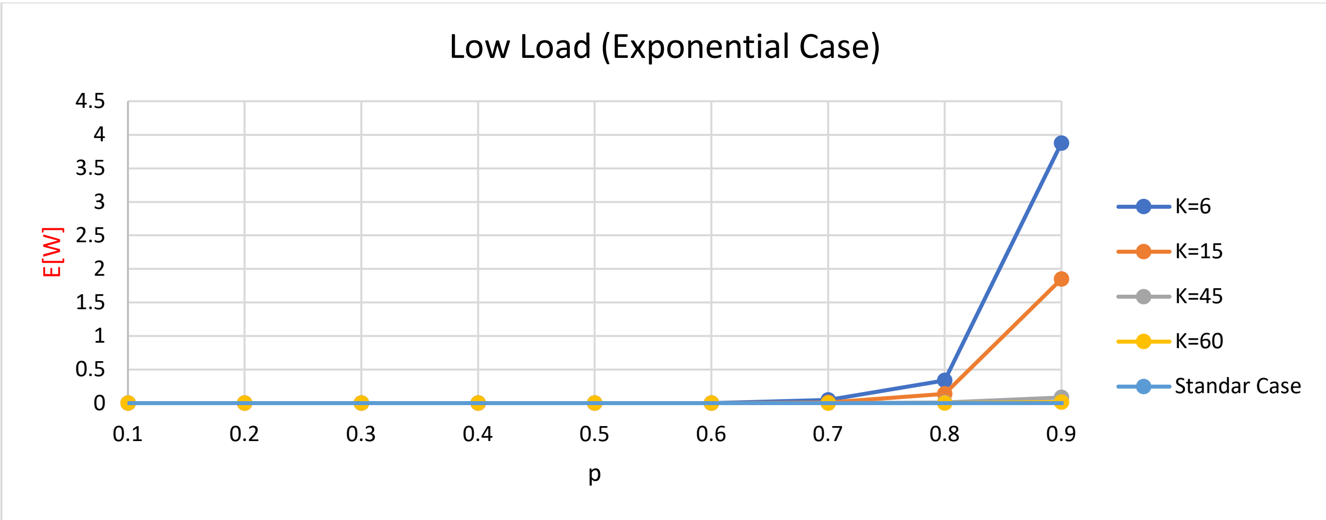
Regarding the quick-tills case parameters we consider:

$K = \{6s, 15s, 45s, 60s\}$

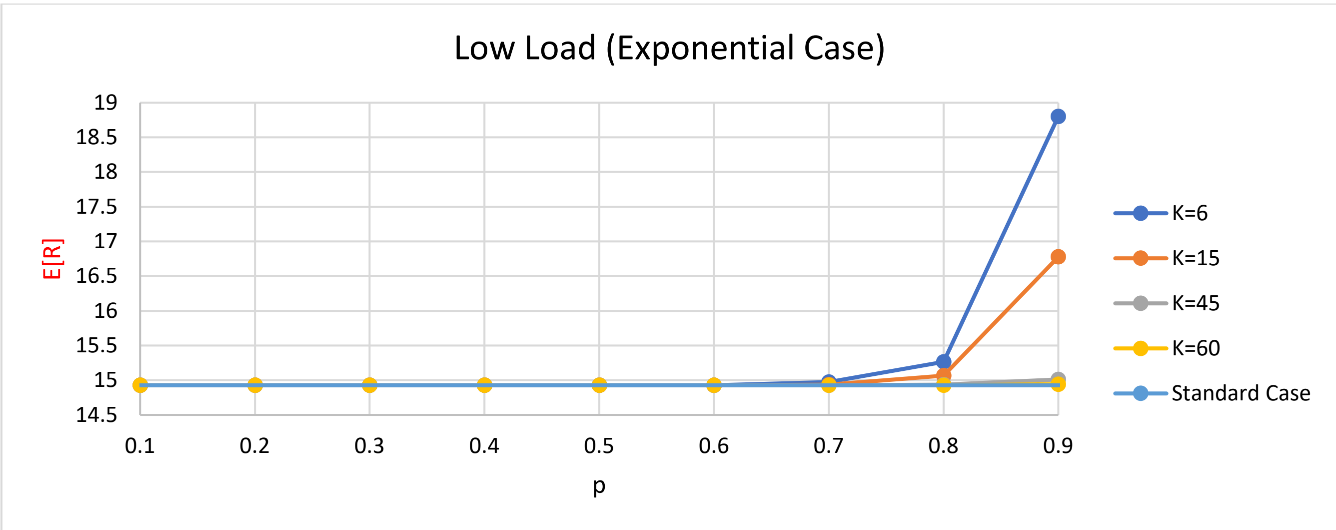
$p = [0.1, 0.9]$ (Out of range)

Starting from the low case we get the following plots:

- Low: $\frac{1}{\lambda} = 50\text{s}$, $\frac{1}{\mu} = 15\text{s}$



Waiting Time on Low Load (Exponential Case)

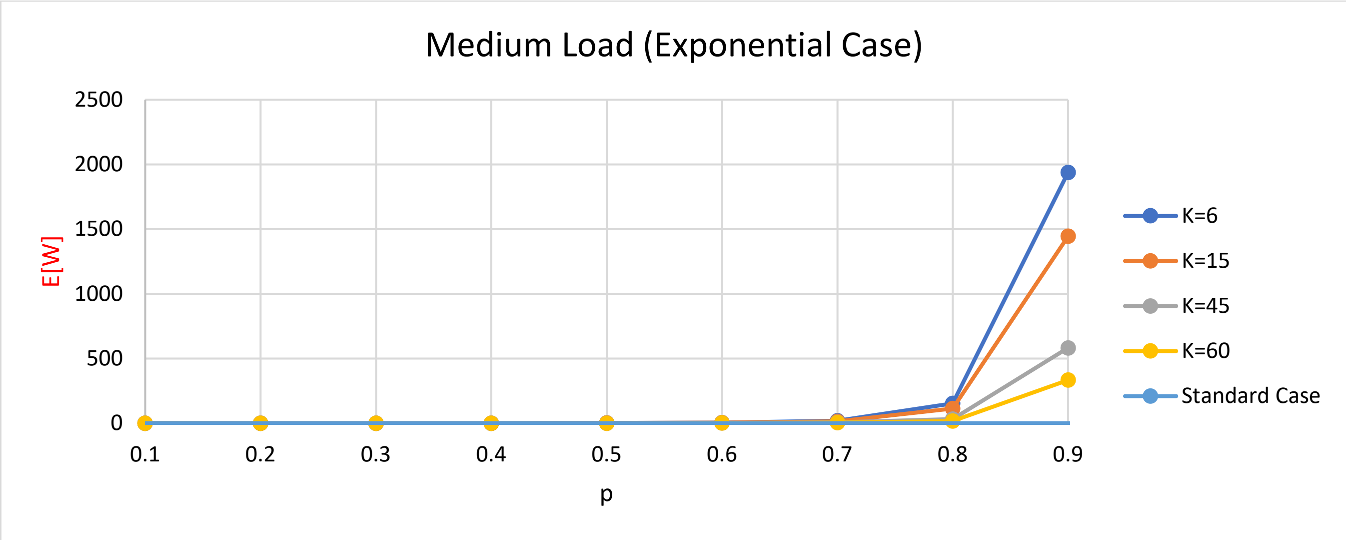


Response Time on Low Load (Exponential Case)

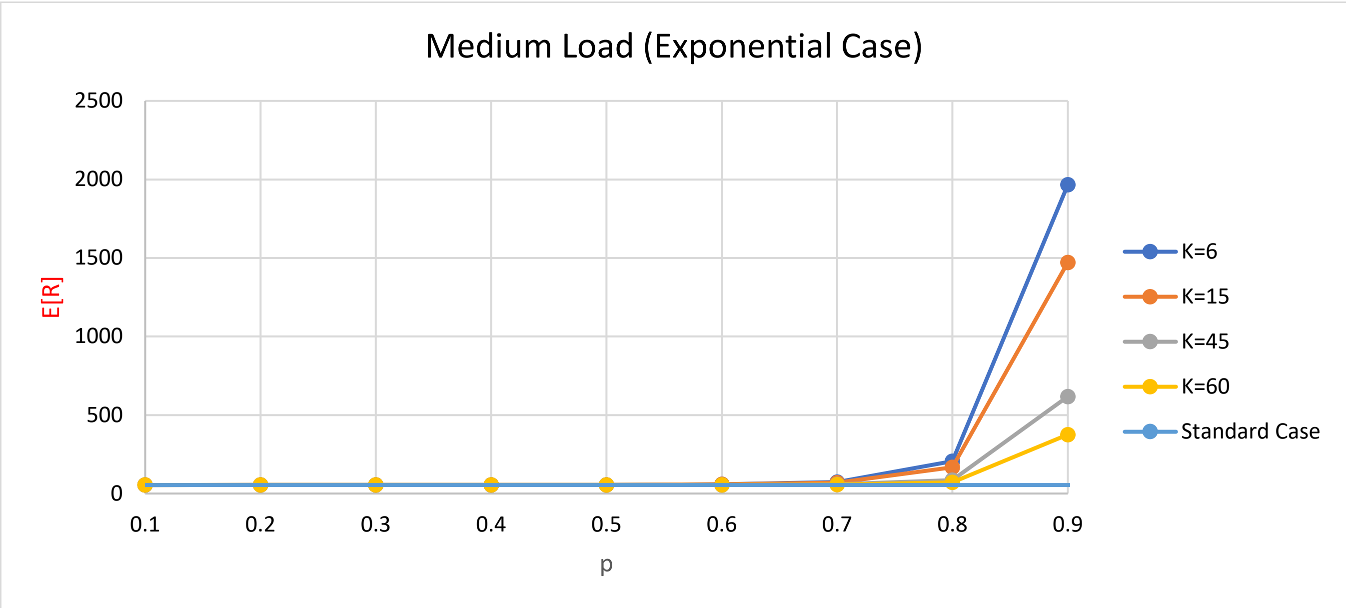
The KPI are coherent and performing the tests on the lognormal distribution shows the same behavior (omitted). As K decreases less carts have access to the quick tills that becomes rarely used, the result is that only a portion of the total number of tills are exploited and the response (waiting) time increases.

In the medium case we are in the same situation:

- Medium: $\frac{1}{\lambda} = 30\text{s},\ \frac{1}{\mu} = 54\text{s}$



Waiting Time on Medium Load (Exponential Case)

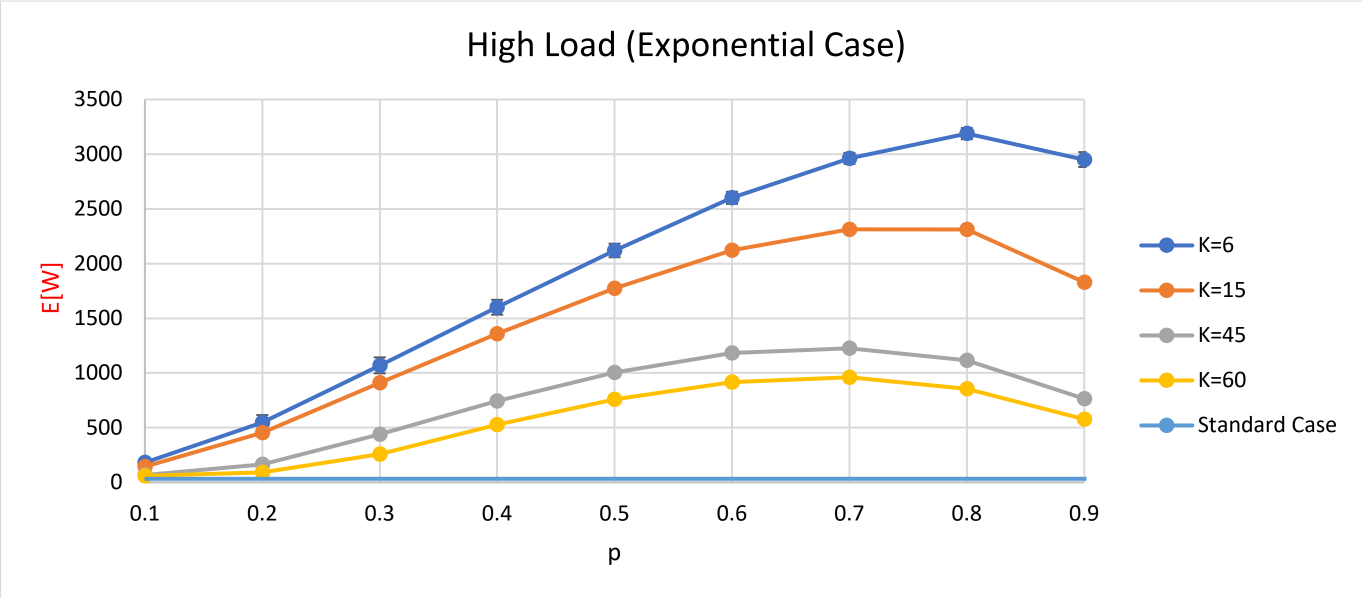


Response Time on Medium Load (Exponential Case)

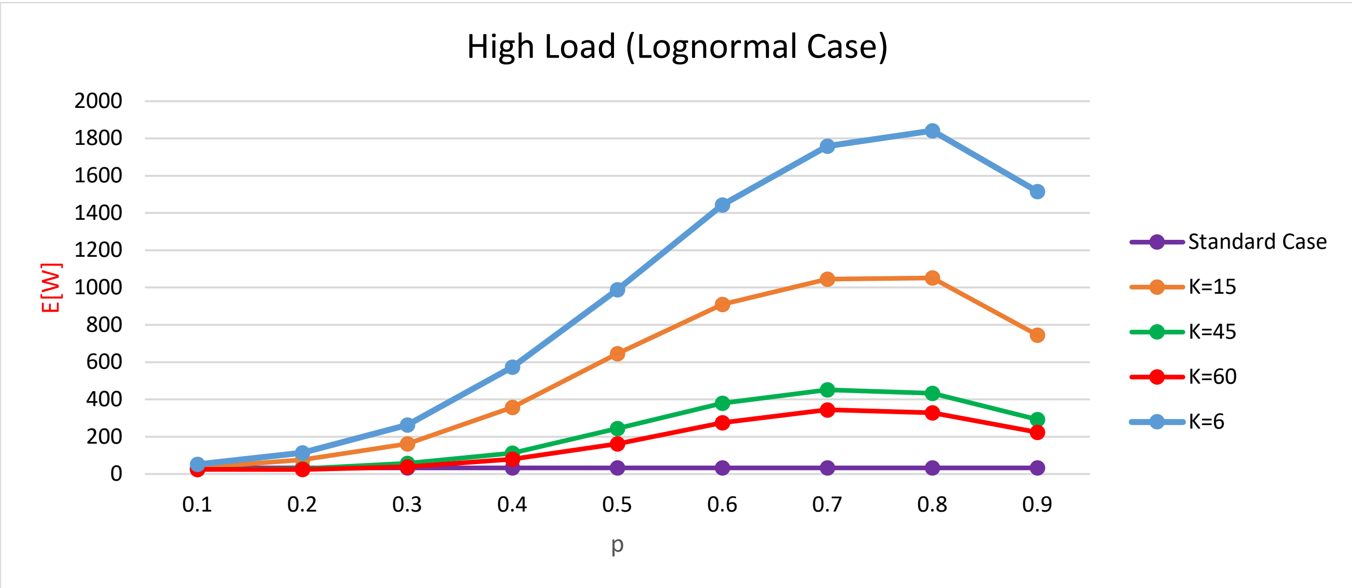
The case of the lognormal distribution shows the same behavior.

The high case shows a strange behavior for both lognormal and exponential distribution:

- High: $\frac{1}{\lambda} = 10\text{s}, \ \frac{1}{\mu} = 90\text{s}$



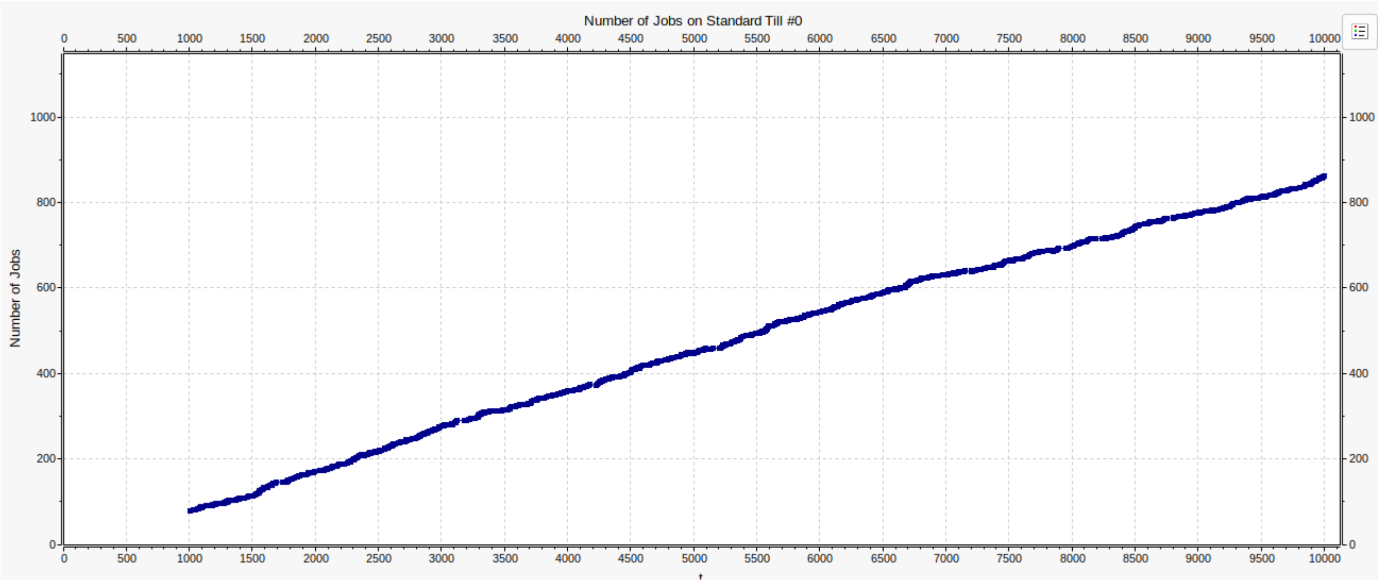
Waiting Time on High Load (Exponential Case)



Waiting Time on High Load (Lognormal Case)

In this case the mean waiting time shows a peak, this strange behavior is because on high load, when the percentage increases and reach a certain value (in this case between 0.7 and 0.8), the cart arrivals become unsustainable for the system that starts to accumulate all the jobs in few tills such that no one will never leave the system before the “sim-time-limit”. So those values will never be counted in statistics, that’s why the mean waiting time decreases.

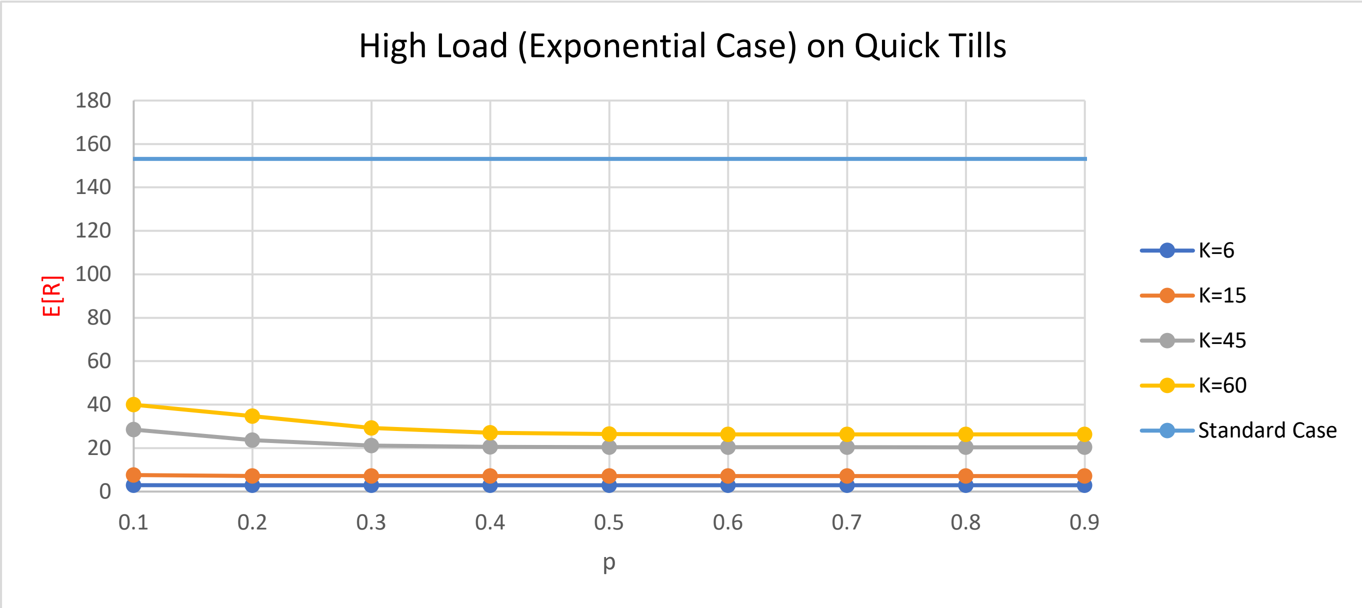
This plot shows what happens in one of the overloaded tills:



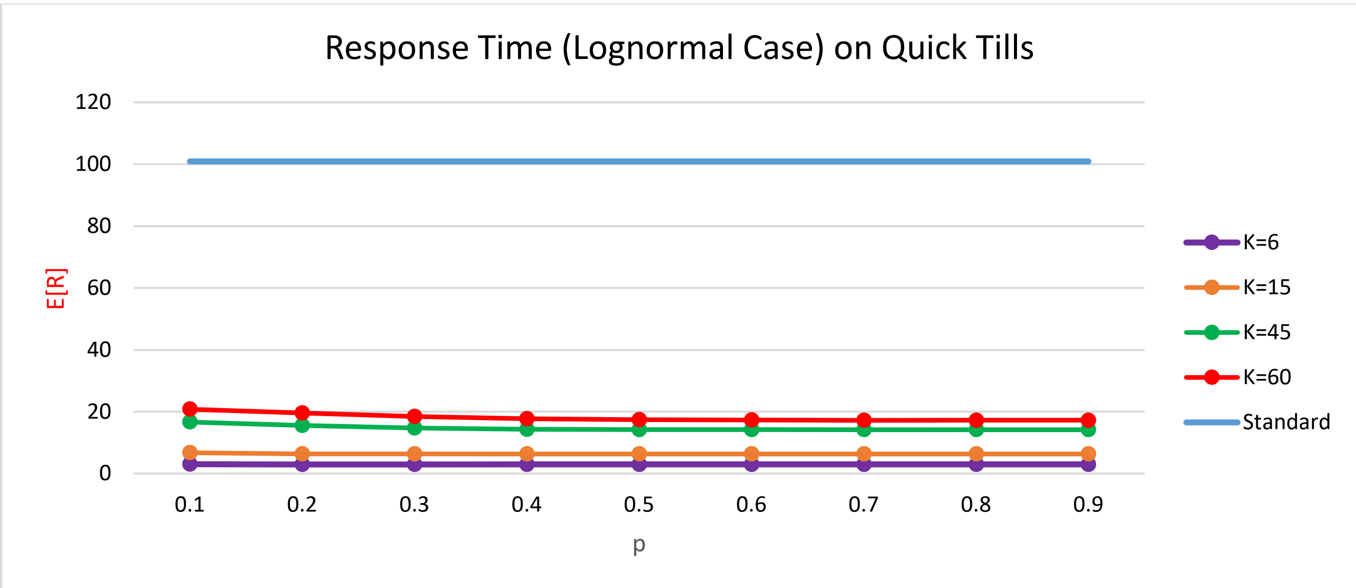
Number of Jobs on a Standard Till for High Load and High quick till percentage

This to emphasize why using high percentages of quick tills is not a good idea and the reason is that the standard tills for the will not be enough to sustain a high load of traffic.

The objective of our study is to compare the standard case and the quick-tills case, further, we must understand how the parameters of the second case influence the system. The previous experiments clearly show that in terms of global mean waiting time (or response time) the standard case is the clear winner in most cases, but we have not discussed what happens on the response time of the customers in the quick tills.



High Load (Exponential Case) on Quick Tills



High Load (Lognormal Case) on Quick Tills

We can observe how the response time in the quick tills is much lower respect to the standard case, this is a big advantage for customers who brings carts with few objects according to fixed threshold K.

Conclusions:

The introduction of quick tills heavily affects the system, we have seen that using high percentages of them could result in a waste of tills because customer with larger carts cannot join. We have also observed that the global response time does not show big improvements from the standard case because the amount of people that can join the quick tills are far less than the total. The question is:

What is the purpose of having quick tills in a supermarket?

The answer is that with the same global performances we can reduce the mean waiting time of the customers having few items in their carts. In fact, in the standard case, all the customers are queued on the same type of tills and people with tiny carts must wait for the bigger ones to be processed, resulting in a worse user experience.

Appendix: Failed attempt of 2kr factorial analysis

Before proceeding with the experiments, we tried to carry out a **2kr factorial analysis** to get some insights and a deeper understanding of the system under analysis. We aimed to understand the contribution of factors on the carts' response and waiting time.

We considered the following parameters and ranges:

- Number of replicas: 5
- Number of tills: 10
- Interarrival mean: [10, 50] (**A**)
- Service mean (exponential): [15, 90] (**B**)
- Service mean logarithm (lognormal): [2, 3] (**C**)
- Service standard deviation logarithm (lognormal): [1, 1.75] (**D**)
- Threshold: [6, 60] (**E**)
- Percentage of quick-checkout tills: [0.2, 0.5] (**F**)

We analyzed two macro-configurations related to the quick-checkout tills, since they have both load and structural factors and have a higher number of factors to deal with:

1. Exponential service times, quick-checkout tills (**A, B, E, F**)
2. Lognormal service times, quick-checkout tills (**A, C, D, E, F**)

The results of the 2kr factorial analysis showed that in both configurations, **the main factors are those related to the load** of the system (A, B, C, D). On the other hand, the structural factors have a negligible impact (E, F) on the performance.

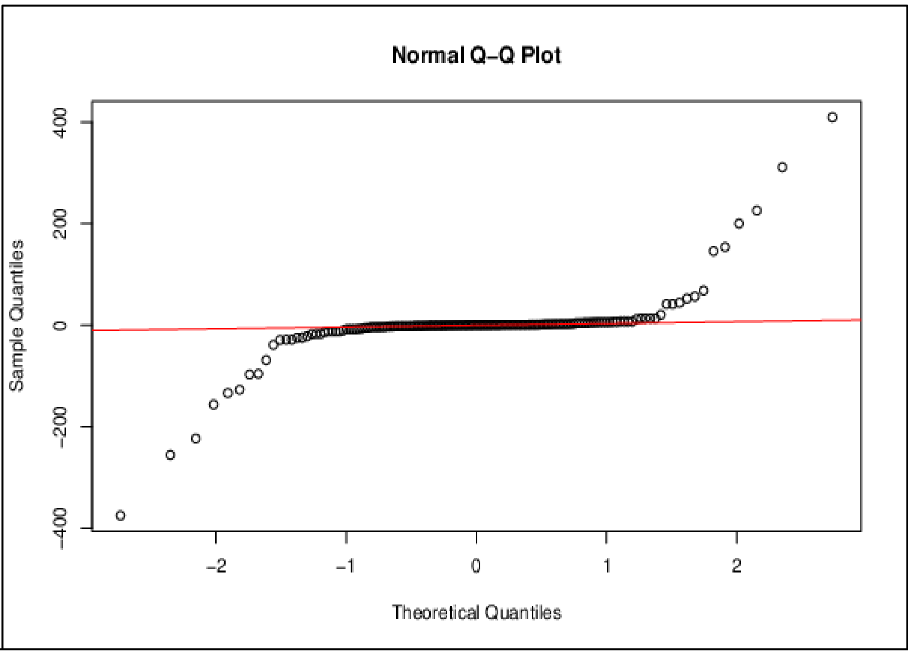
It's good to notice that the contribution of errors was also negligible.

Configuration 1 (A, B, E, F) Response time	
FACTOR	**CONTRIBUTION**
B	22.81%
A	15.84%
AB	15.77%
others	all < 8%
errors	0.02%

Configuration 2 (A, C, D, E, F) Response time	
FACTOR	**CONTRIBUTION**
C	15.38%
D	14.16%
CD	11.92%
ACD	8.99%
A	8.59%
AC	8.06%
AD	7.17%
others	all < 3%
errors	0.05%

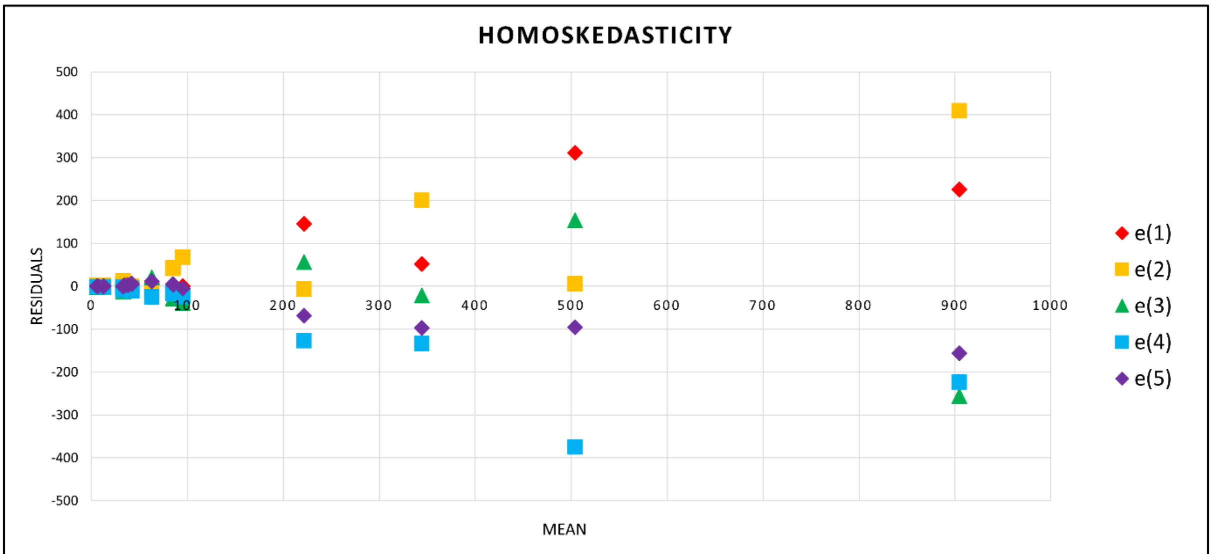
Unfortunately, our factorial analysis was a **failure**, because **the hypothesis** (residuals are IID normally distributed with null mean and constant standard deviation) **were NOT met**. This obviously forbade us from using its results in our project.

For what concerns the normal hypothesis, the **QQ plot** of the residuals vs normal shows doesn't show a linear tendency. The skyrocketing behavior suggests that residuals could be fitted in a heavy-tail distribution. Even after a log transformation of the data, no linear tendency appears.



QQ plot of the residuals of the 5 replicas.
On the horizontal axis, we have the theoretical quantiles of a normal distribution, whereas on the vertical axis we have the sample quantiles from the residuals. The red line is a linear fitting of the data.

For the homoskedasticity (i.e. constant standard deviation), the **scatterplot of residuals vs predicted response** shows that there is a **clear and not negligible trend**, indicating that standard deviation is not constant.



Scatterplot for homoskedacity. Average response (x axis) vs residuals (y axis, one color per replica)