COMP9334 – 17s2

Report: Getting the best response time out of your power bill

zID: z5112209
Name: Darshika Fernando

1. Introduction

This work focuses on simulating a processor sharing server in a server farm with the objective of optimising the response time with regards to its power consumption.

2. Objective

The behaviour of power consumption is represented through the service time distribution. Therefore, by calculating the mean response time for each server configurations (the number of servers that can be kept switched on at any one time), the minimum mean response time that can be achieved by this server farm is calculated.

3. Method

The server farm has 10 servers to whom jobs are assigned according to a round robin sequence, hence only one server is simulated as specified in the project specification.

3.1. About the python code

Programming was done with python 3.6.
Python modules used include: numpy, scipy, matplotlib.pyplot, csv, math, random

The final results can be produced by running main.py
Plots used for visual inspections can be produced by visualise.py and transientReomval.py
More information about the code itself is included in the script as comments.

3.2. Modelling arrival and service times

Modelling of arrival time sequence for the high-speed router resulted in the following distributions. Figure 1 (a) and (b) show the exponential distribution a_{1k} and uniform distribution a_{2k}. Figure 2 shows the inter arrival time distribution of a_k.

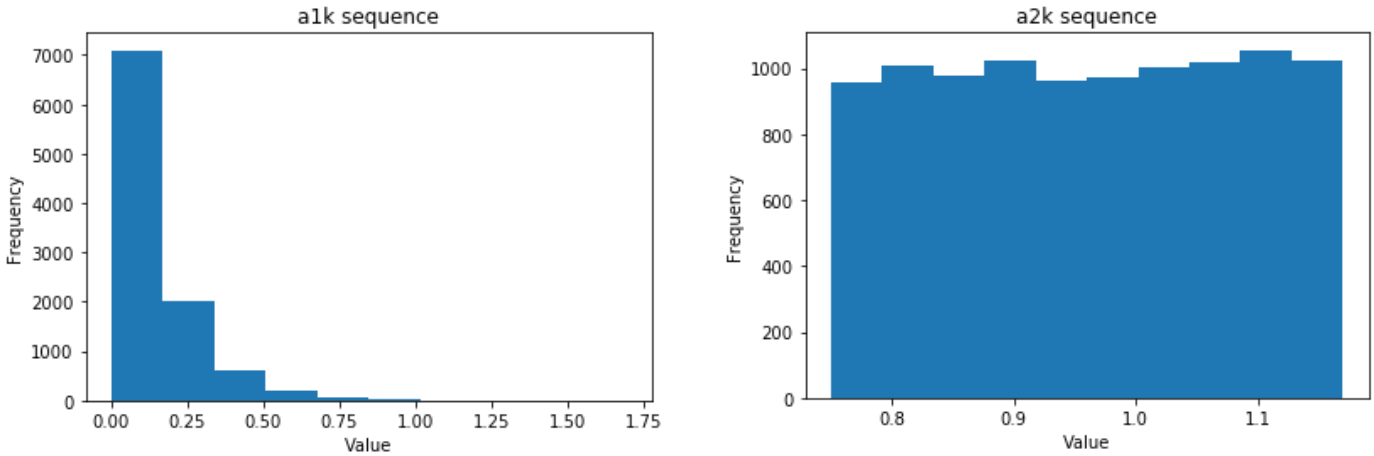

Figure 1(a) and 1(b): a1k, a2k sequence

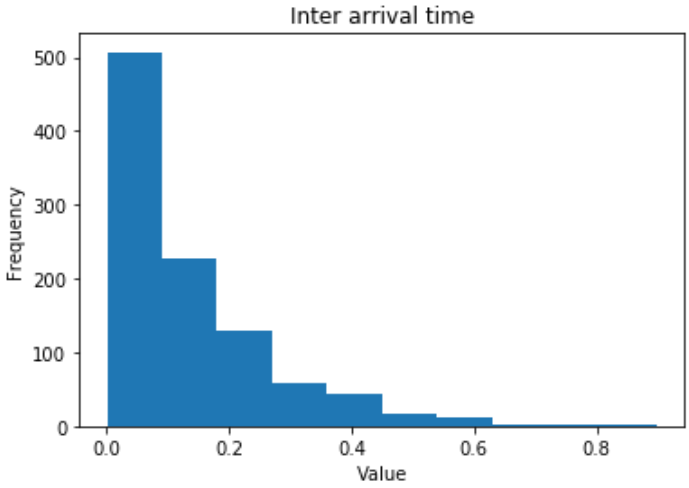

Figure 2: Inter arrival time

Figure 3 shows the distribution of service time for one instance where 5 servers are switched on.

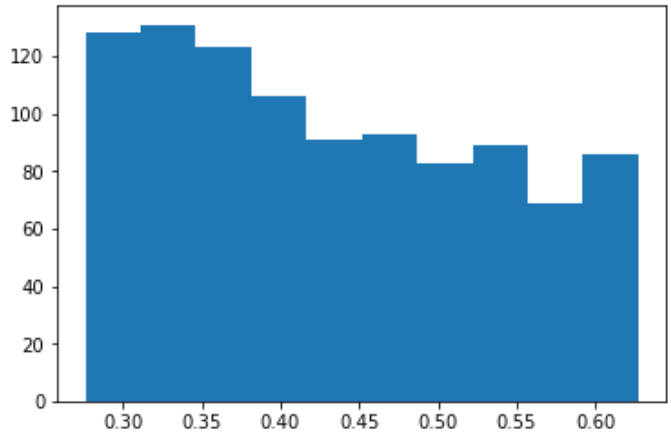

Figure 3: Service time distribution

3.3. Simulation Parameters

For this project, the following simulation parameters were used:

1. Length of simulation - length of simulation is varied by changing the number of job requests sent to the high-speed router
2. Number of replications – the number of replications were varied from 5 to 10 to determine the optimised number of replications under the limited computing resources.
3. Accuracy - a confidence interval of 95% was used

4. Statistical Analysis

4.1. Selecting the Length of Simulation

The number of servers switched on, was varied from 3 to 10. For each instance, 5000, 50000, 75000, 80000, 90000, 100,000 and 150,000 jobs were sent to the high-speed router to change the length of the simulation. The time it takes for the response time to reach a steady state was observed visually and it was decided to use 90000 job requests for simulation because further increasing the number of jobs did not resulted in significant marginal improvement. Also, the amount of data left after removing the transient part and the amount of time and computing resources it takes to process the jobs were considered in making this decision.

The visual representation used is as illustrated in Figure 4(a) and 4(b). As shown in Figure 4(a), the amount of information that can be retained after removing the transient part, is not sufficient for further calculations.
The mean response times for 90,000 jobs is illustrated in Figure 5(a) and 5(b).

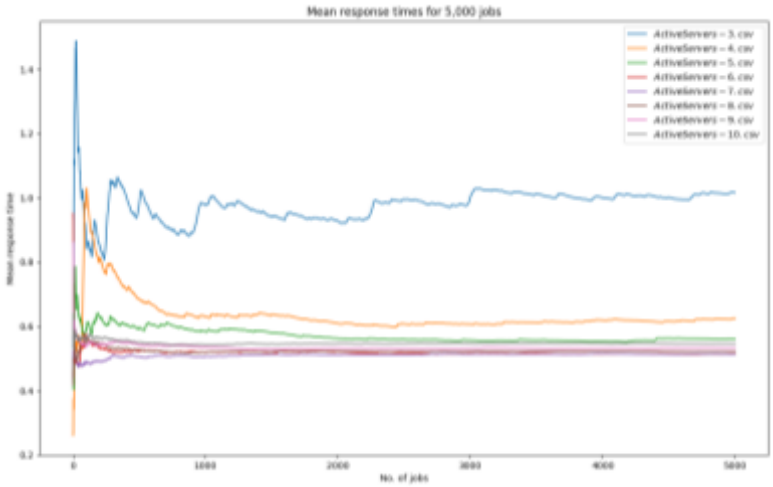

Figure 4(a): Mean response times of Servers 3 – 10 for 5000 jobs

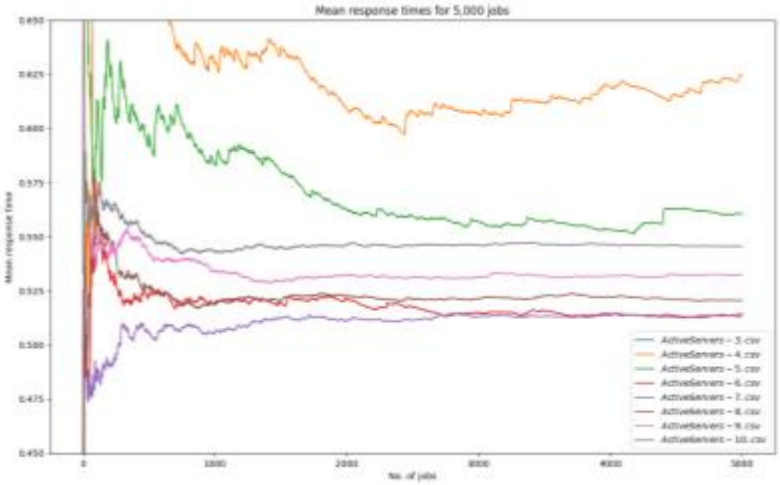

Figure 4(b): Mean response times for 5000 jobs (zoomed out)

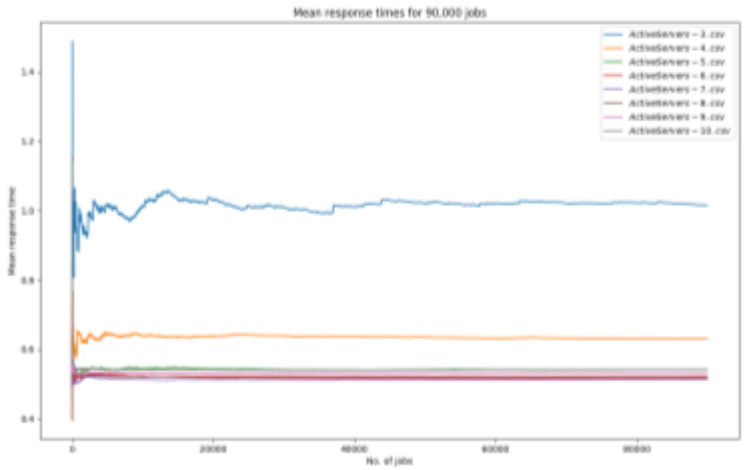

Figure 5(a): Mean response times of Servers 3 – 10 for 90,000 jobs

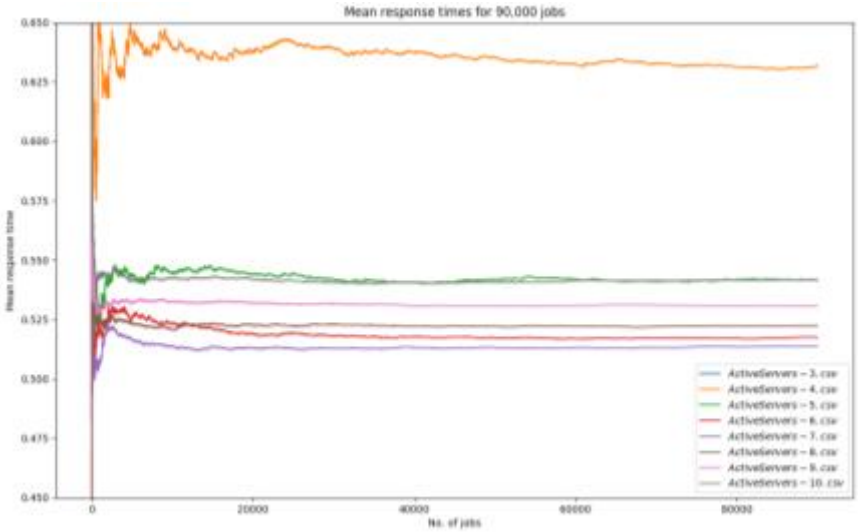

Figure 5(b): Mean response times of Servers for 90,000 jobs (Zoomed out)

Observation

By observing the running mean response time for each server configurations (the number of servers switched on at each experiment) it can be concluded that the minimum mean response time is observed when 7 servers are switched on. Therefore, the next steps of analysis only focus on the instance when 7 servers are switched on.

4.2. Transient removal

After selecting 90,000 jobs as the parameter for simulation length, the transient component was removed so that an estimated mean response time can be calculated.

Transient component of the response time was detected using pairwise difference between running mean response times recorded by the simulation program ps.py. That is, the difference between each ordered pair of response times were calculated so that the point when these differences become negligible can be identified. The pairwise differences for the instance when 7 servers are active is illustrated in Figure 6.

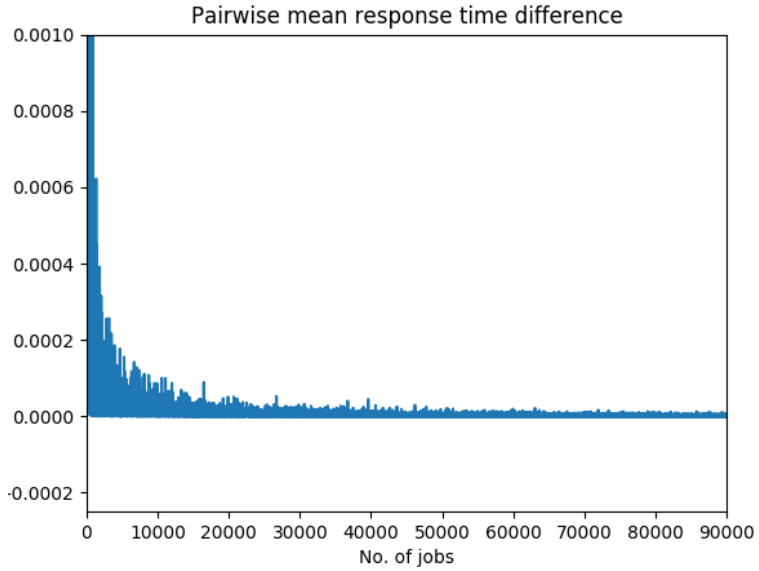

Figure 6: Pairwise Mean Response Time Differences for 90000 jobs

From visual inspection, it is evident that the pairwise difference in mean response times are negligible after 30,000 jobs. Therefore, steady state is considered from 30,000 jobs to 90,000.

The simulation program has already recorded the running mean for each server. Therefore,

$$Estimated\ steady\ state\ mean\ response\ time = \frac{Sum\ of\ running\ mean\ responses\ in\ steady\ state}{Total\ Number\ of\ jobs\ in\ steady\ state}$$

4.3. Independent replications of steady state mean response time

To maintain generality, the above process is repeated 10 times using different random seeds. For each time, the estimate steady state mean response time is recorded in a list.

Since the state where 7 servers are switched on is considered, the experiment only calculates the replications for 7 servers. However, if required it can be used to calculate independent replications for all server configurations.

4.4. Confidence Level

After recording the estimated response times for all the replications, the final mean response time can be calculated with a confidence interval of 95%.