# COMP9334 Project Report

**Name: Xuhua Le**

**Student number: z5047516**

## 1. Simulation Design.

In this project, we are going to simulate a three-layer model of fog, network and cloud. As the fog and cloud both are PS server, and the time interval for each job is relatively short, it's important to treat job in these two servers are working in the same.

In the simulation design process, there are five events that needs analyzed:

a. job arrivals at fog.
b. job left fog (finishes).
c. job left fog to network.
d. job left network to cloud.
e. job left job (finishes).

PS: event 'b' and 'c' could be considered in a similar way.

In order to analyze the working process, I have set a number of variables to store relative information, and some of the key variables are: fog_jobs, network_jobs and cloud_jobs, which are used to record the job information in fog, network and fog; fog_dep, net_dep, cloud_dep, which are used to record the job departure time at fog, network and cloud.

When a job arrivals at fog, we need to update all the running jobs' work left as during the time interval they have been served for some time, and update fog_jobs as there is a new comer, and we also need to update the next departure time. Besides, we need to decide which event is going to happen in the next stage.

When a job departs from fog, there are two possible situations: the job finishes or goes to network. We need to update relative information as well, such as: fog_jobs, network_jobs (possible), work left time for each job, and so on. Besides, we need to decide which event is going to happen in the next stage.

When a job departs from network to cloud, we need to update network_jobs, cloud_jobs, work left time for each job, and so on. Besides, we need to decide which event is going to happen in the next stage.

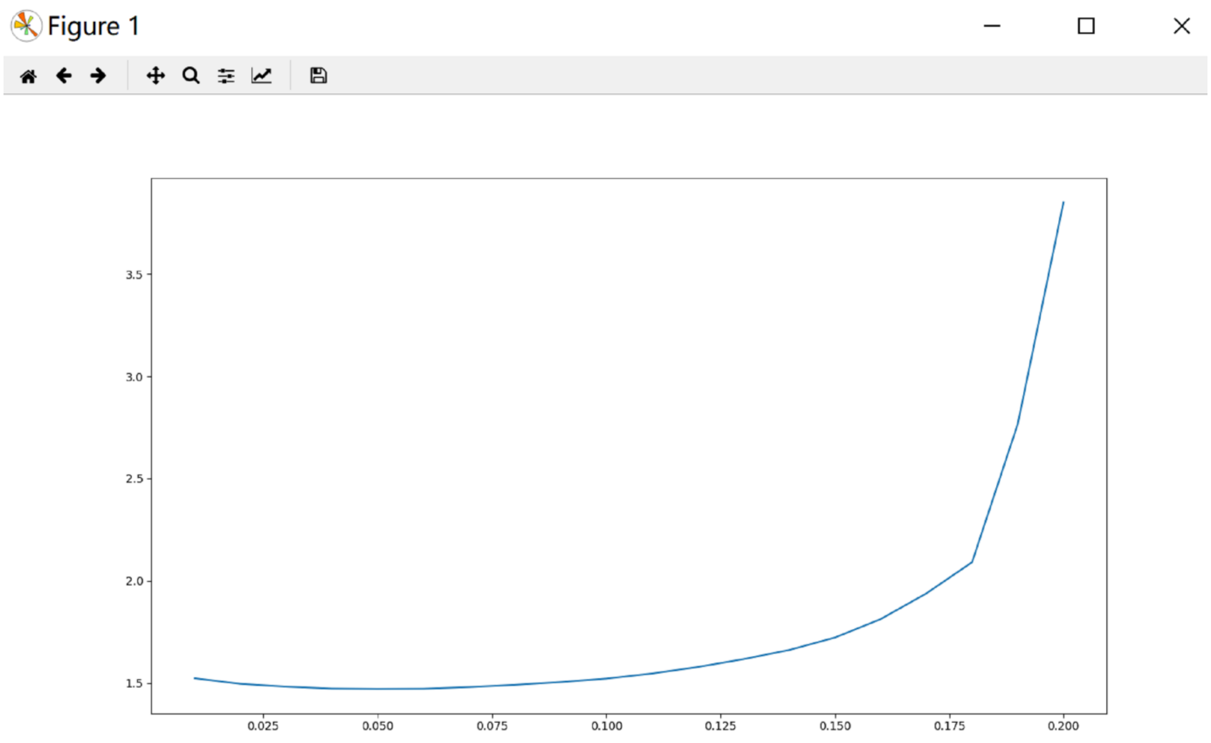
When a job departs from cloud, we need to update cloud_jobs, work left time for each job, and so on. Besides, we need to decide which event is going to happen in the next stage.

These processes are clearly to analyze when the mode is "trace". But the essential logic is the same when the mode is "random", except that we need to decide the arrival time, service time and network latency. When the mode is 'random', as the inter-arrival probability distribution is exponentially distributed with parameter λ, it's a Possion distribution, hence the time interval is 1/λ, and we could get the arrival time. For the service time, we could deduce the formula for time t as an inverse function of the probability density function g(t), and we could get the service time. For the network latency, as it's uniformly distributed, we could use corresponding formula to get the network latency. Hence, we could use these values as parameters for the "trace" mode to get corresponding outputs.

**2. Design Problem**

Our target is to determine a suitable value for "fogTimeLimit" in order to get the minimized mean response time "mrt" for the random mode. If fogTimeLimit == 0, it means all the jobs are going to be served at cloud. If fogTimeLimit == max(service time in fog unit), it means all the job are going to be served at fog. Its's clearly that both are not the case, hence, 0 < fogTimeLimit < max(service time in fog unit).

The key idea is to set an upper limit for fogtimelimit, and from 0 to the fogtimelimit, for every interval of 0.01 for fogtimelimit, we need to run the simulation and get the corresponding mean response time, and plot the graph. After we get the graph, its's important for us to remove the transient stage as it diverge the theoretical values a lot, and we are only interested in the steady states. After that, we could visually inspect the graph to get the suitable range of fogTimeLimit to get minimized mean response time. And I have got the following graph though the program “optimization.py”, and I used the sample parameters in the project specification.



In this graph, the x-axis is fogTimeLimit while the y-axis is the mean response time. We could see that when fogTimeLimit in (0.4, 0.6), we could get the minimized mean response time.