Student id: 5166715
Name: Zeyuan Ma

Report of fog/cloud computer system

1. **Introduction**

   The report is about the simulation of an Analytic Model for Fog/Cloud Computing including two part. The first part is about the implement about simulation program for Fog/Cloud Computing including the function how to simulate the trace model and random model, the correctness of the simulation code and the probability distribution of input for random model. The second part is about how to determining a suitable value of fogTimeLimit including the statistically sound methods for analysis and how to choose the parameters, such as lengths of your simulation, number of replications, end of transient.

2. **simulation program for Fog/Cloud Computing**

   (1), trace model

   The idea of simulation program for trace model is that updating the related variables according to the occurrence of events.

   The program of trace is enclosed in a function named trace_modle() which input is arrival, service, network, para and can be found in trace_model.py

   It can be easily seen that there five event that need to be noticed.

   They are:

   a. Next arrival at fog.
   b. Next departure from fog without cloud
   c. Next departure from fog with cloud
   d. Next departure from network
   e. Next departure from cloud

   (The figure 1 shows five event)

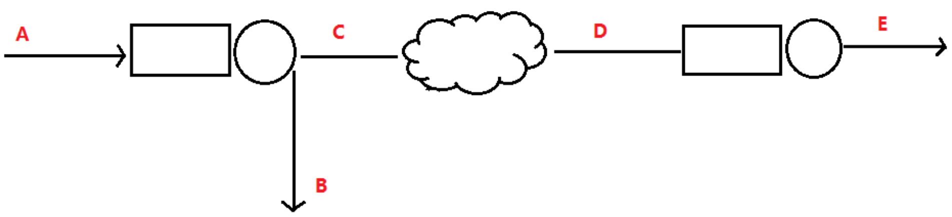

Figure 1

The program relate each event to a variable.

They are next_arr_fog

next_dep_fog_woC

next_dep_fog_wC

next_dep_net

next_dep_cloud

Those five variables will be update when an event occurred

The program have a main loop which the end condition is the the number of completed request and the number of request in the arrival are equal.

Then, for each occurred event, the job lists in the system will be updated, and the value of next event variable is updated according to the updated job lists.

After an event is finished, all the related variables are updated, it find the minimum value of event variable as the next occurred event.

(2), random model

Because the input of random model is different from trace model. We have to generate the lists of arrival, service, network.

The time of interval arrivals should be the exponential distribution.

Then it can generate a set of numbers, U, that are uniformly distributed in (0, 1),

And Compute the value of inverse function for CDF which is:

$$invF(u) = \frac{-log(1 - U_k)}{\lambda}$$

The figure 2 show it is an exponential distribution.

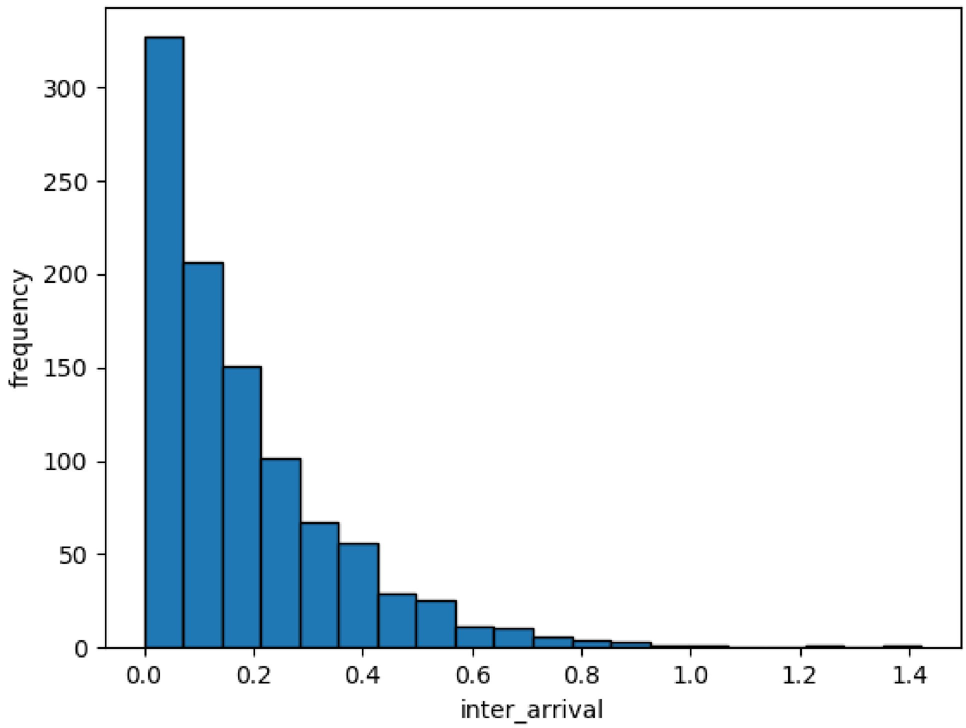

Figure 2

For service time, the probability density function is

$$g(t) = \begin{cases} 0 & \text{for } t \le \alpha_1 \\ \frac{\gamma}{t^{\beta}} & \text{for } \alpha_1 \le t \le \alpha_2 \\ 0 & \text{for } t \ge \alpha_2 \end{cases}$$

where

$$\gamma = \frac{1-\beta}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}}$$

Then its CDF is

$$CDF = \int_{-\infty}^{t} g(x)dx,$$

$$G(t) = \begin{cases} 0 & for\ t \le \alpha_1 \\ \frac{\gamma}{1-\beta}(t^{1-\beta} - \alpha_1^{1-\beta}) & for\ \alpha_1 \le t \le \alpha_2 \\ 1 & for\ t \ge \alpha_2 \end{cases}$$

The inverse function of G(t) is:

$$invG(\mu) = [\frac{1-\beta}{\gamma}\left(\mu + \frac{\gamma}{1-\beta}\alpha_1^{1-\beta}\right)]^{\frac{1}{1-\beta}} \quad where\ \mu \in [0,1]$$

μ is uniformly distributed between [0, 1].

From figure 3, it can be seen that the service time is also an exponential distribution.

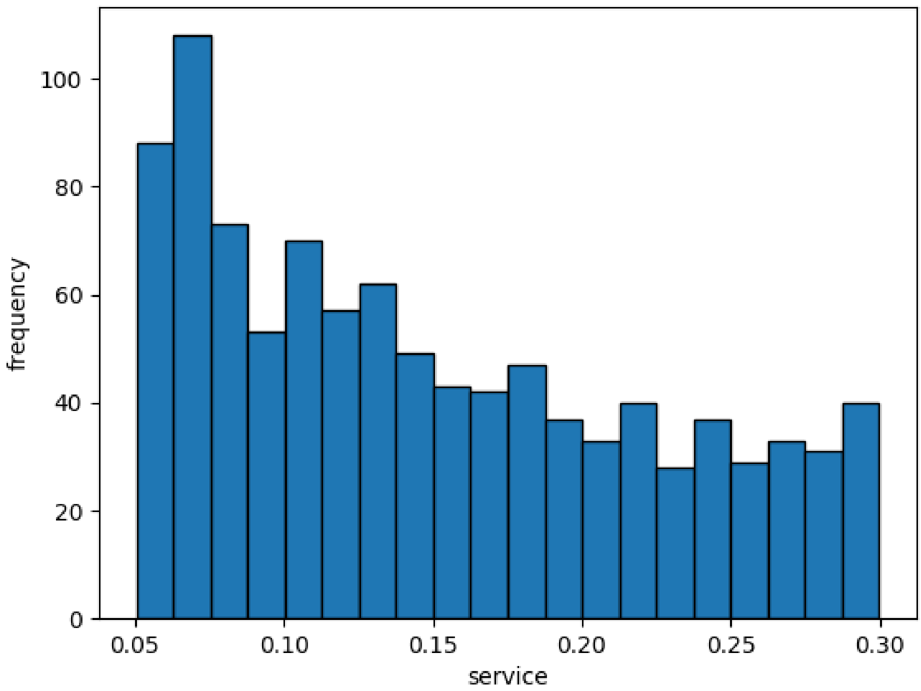

Figure 3

For the network, The network latency is uniformly distributed in the open interval (ν1,ν2) where ν2 > ν1 > 0. It can be seen in the Figure 4.

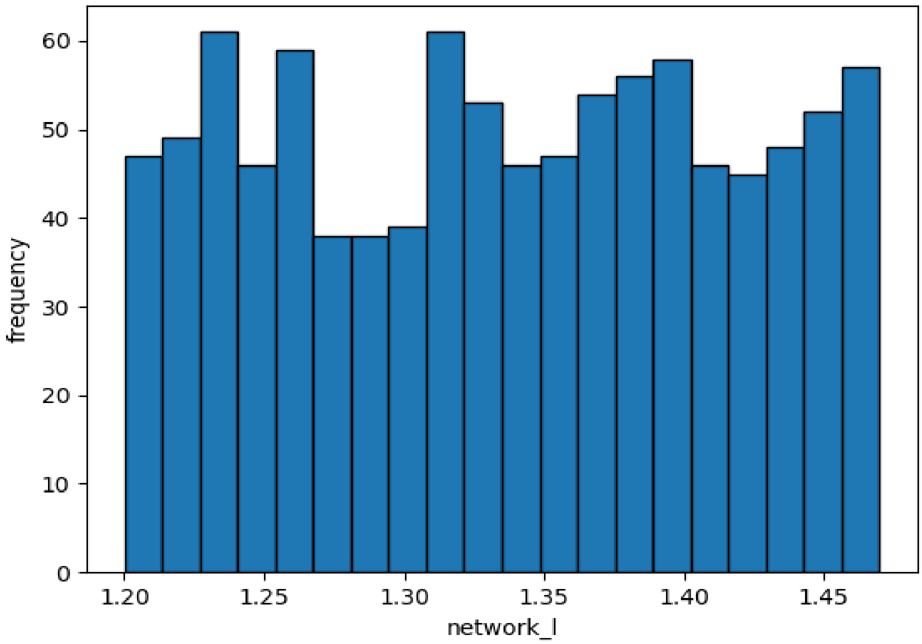

Figure 4

After generate those three lists, we can use trace model to output correct results. But it should be noticed that the condition of main loop should be the (master clock < end_time).

With running the cf_output_with_ref.py, it can be seen that my simulation program output the correct results.

3. The best Fog time limit

In order to determine a suitable value of fogTimeLimit, we have to use statistically sound methods, with parameter values: $\lambda = 9.72$, $\alpha1 = 0.01$, $\alpha2 = 0.4$, $\beta = 0.86$, $\nu1 = 1.2$, $\nu2 = 1.47$ and fogTimeToCloudTime is 0.6.

First we have to find the steady state for the random simulation model. From running part2_findsteadyState.py, we can get Figure 5

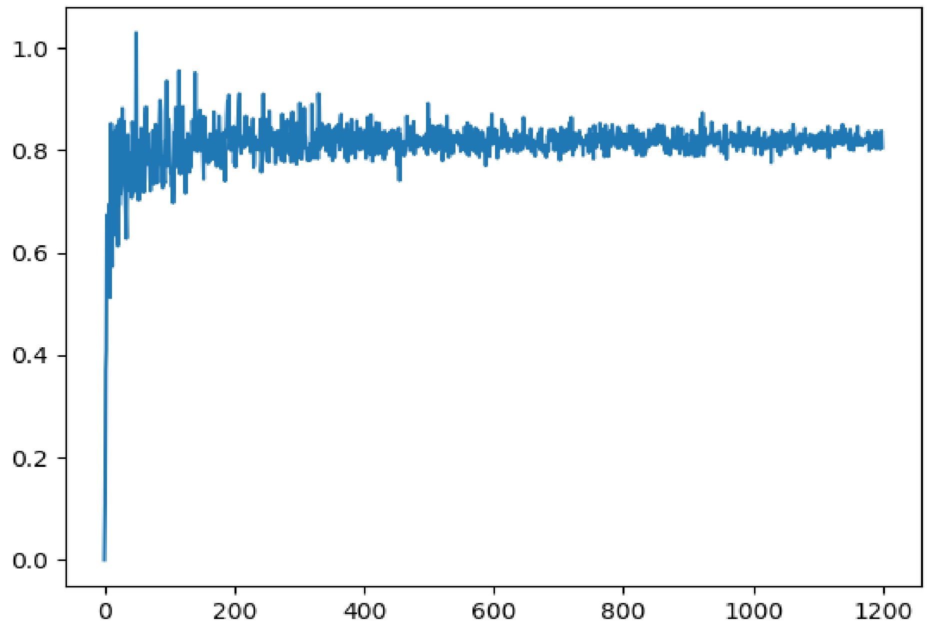

Figure 5

It can be seen that after 500 seconds the random model will be steady.

Then we need to use number of Transient removal to compute mean response time.
The formula is:

$$\frac{X(m+1)+X(m+2)+...+X(N)}{N-m}$$

Where m is the end of transient. N is the number of requests.

Because for the random simulation model we can control the number of requests in the system by the value of time_end. Assume we set time_end is 1000, then the lengths of simulation is 1000s and end of transient is 500
Therefore we only need to compute the mean response time from 500 to 1000(500<t<1000).

Then we choose different value of FogTimeLimits. We need to do the Independent replications.
Therefore, first we set number of replications is 10, we use seed(0-10) control the output of random simulation model.
Then we can find that the best FogTimeLimits is between 0.10 and 0.12

For the number of replications is 10, the confidence interval of mean response time for 0.10, 0.11, 0.12. it can be seen in figure 6 by running part2_findbestFTL.py

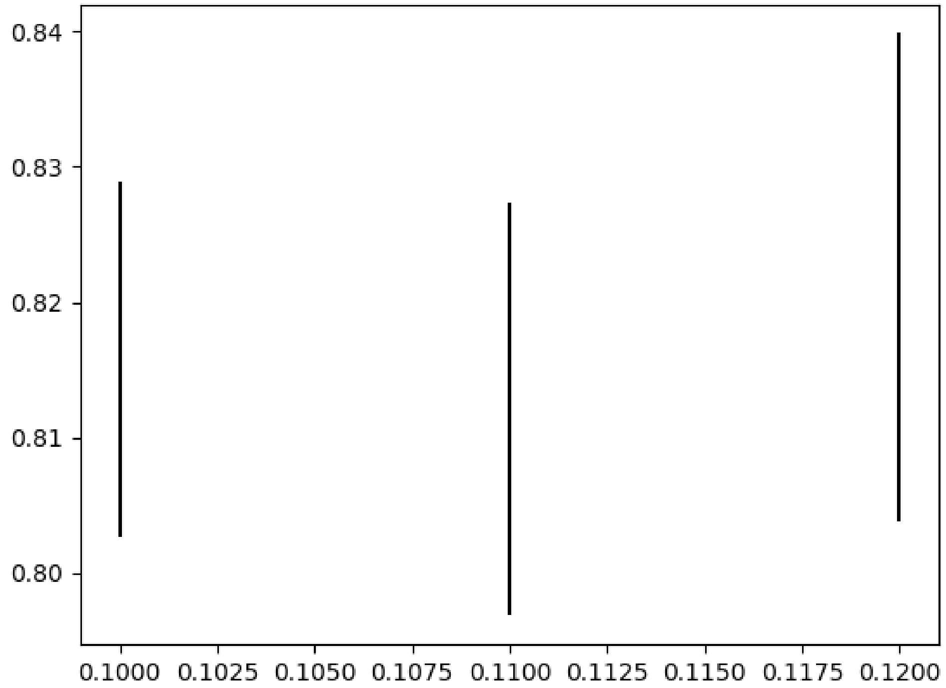

Figure 6

Because it does not separate the confidence interval we need more replications, set the number of replications is 50. We get figure 7.

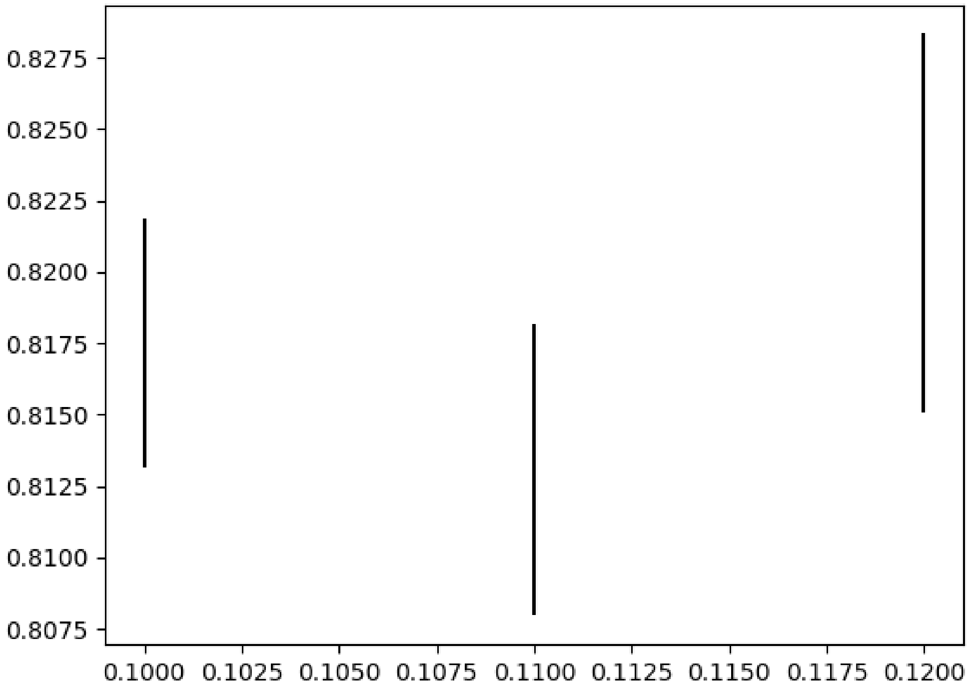

Figure 7

It can been seen the coincident parts is smaller for the three confidence interval but still not separate the confidence interval.

Because the computation is too large if we do more replications. And from figure6 and figure7, we can noticed that the confidence interval of 0.11 will be smaller than 0.10 and 0.12 with the replications increase. Therefore, we can get the best fog time limit is about 0.11.

4. Conclusion

With simulated the fog/cloud system and use the statistically sound methods, we find the best fog time limit is about 0.11.