Trace mode:

For the simulation correctness, it is important that narrow the time error. For example, the way to deal with the time unit. In order to improve the accuracy of result, I am trying to use more smaller time unit in my FOR loop. Thus, in my simulation program, the time unit is 1/1000 unit, which means the judgement algorithm would detect the request if or not finished in Fog/Cloud. Therefore, the error with the theory time could be narrowed tremendously. As for departure time of fog, it needs a function to find the request could achieve in Foglimit time. Then send it to cloud processor through the network propagation. This step would be network departure time. Finally, the request would arrival the cloud to finish the test of task.

Random mode:

Most importantly, generating the simulation data from provided file. First and foremost, the arrival time for request is the starting point. As for the interval-arrival time, it should fellow the current the clock time and end_time, which means it can not smaller than the 0 unit, and finish after the end_time. In addition, the arrival time would obey the order by time. Furthermore , it should be an exponentially distributed with parameter λ. In consideration of above, it is generated by python built-in, random.expovariate(λ), function.

Secondly, it is the service time. After some math calculation blow:

$$g(t) = \begin{cases} 0 & t \le a_1 \\ \frac{\gamma}{t^\beta} & a_1 \le t \le a_2 \\ 0 & t \ge a_2 \end{cases} \qquad \gamma = \frac{1-\beta}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}}$$

Because of $a_1, \alpha_2, \beta \in R$.

$$g(t) = \frac{\gamma}{t^\beta} \quad a_1 \le t \le a_2 \text{ is definite intergral}, \ t \ge a_1, \ t \le a_1$$

$$\Rightarrow G'(t) = g(t) \Rightarrow g(t) = \gamma \cdot t^{-\beta}$$

$$\Rightarrow \begin{cases} G(\alpha_2) - G(\alpha_1) \\ G(t) - G(\alpha_1) \\ 0 \end{cases}$$

$$\Rightarrow G(t) = \gamma / 1-\beta \cdot t^{1-\beta} \Rightarrow \frac{1-\beta}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}} / 1-\beta \cdot t^{1-\beta}$$

$$\Rightarrow G(\alpha_2) - G(\alpha_1) = \frac{\alpha_2^{1-\beta}}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}} - \frac{\alpha_1^{1-\beta}}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}}$$

$$= 1$$

Then:

$$G(t) - G(t_1) = \frac{t^{1-\beta} - \alpha_1^{1-\beta}}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}}$$

$\therefore$ To find t in this function.

$G(t) - G(\alpha_1)$ is random provide by assignment spec.

Assume

$$\therefore \text{random} = \frac{t^{1-\beta} - \alpha_1^{1-\beta}}{\alpha_2^{1-\beta} - \alpha_1^{1-\beta}}$$

$$(\text{random}) \cdot (\alpha_2^{1-\beta} - \alpha_1^{1-\beta}) = t^{1-\beta} - \alpha_1^{1-\beta}$$

$$\text{random} \cdot (\alpha_2^{1-\beta} - \alpha_1^{1-\beta}) + \alpha_1^{1-\beta} = t^{1-\beta}$$

$$\Rightarrow \quad t = (\text{random} \cdot (\alpha_2^{1-\beta} - \alpha_1^{1-\beta}) + \alpha_1^{1-\beta})^{\frac{1}{1-\beta}}$$

And t means service time.

Then, using random.uniform(v1,v2), it is easy to generate the network time for some requests to cloud. After all of them, I could have arrival, service and network file. Fellowing the same algorithm as trace, it would calculate to mean response time.