COMP9334 project report

Fog/Cloud Computing

Content

Part 1

Description of python files

1. proj.py
   This is the main simulation program. If you simply command: python proj.py, it will generate 3 sets of output files for test 4 (random mode, seed = 0,1,2) which is required by question 6.5 (testing reproducibility).
2. wrapper.py
   This is a wrapper file that calling function from proj.py to generate output files for all the tests. The random seed for random mode is 0.
3. question5_2_0.py
   Plot mean response time for first k jobs.
4. question5_2_1.py
   Generate image_1.png which describe the distribution of 1000 service time value.
5. question5_2_2.py
   Plot mean response time over FogTimeLimit in range 0 to 0.4 step 0.01, save picture as image_2.png
6. question5_2_3.py
   Plot mean response time over FogTimeLimit in range 0 to 0.2 step 0.01, save picture as image_3
7. question5_2_4.py
   Plot mean response time over FogTimeLimit in [0.08,0.09,0.10,0.11,0.12] with 30 different random seeds, save picture as image_4, plot the confidence interval upper bound and lower bound of different FogTimeLimit, save as image_5
8. question5_2_5.py
   Plot mean response time over FogTimeLimit in [0.10,0.11,0.12] with 300 different random seeds, plot the confidence interval upper bound and lower bound of different FogTimeLimit, save as image_6

Part2

Explaining the service time computation

(In proj.py line 61)

The service time in the fog time unit t is generated by the probability density function $g(t)$ where:

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

where

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

Let $G(t) = \int g(t)dt$

$$\text{random value} = \begin{cases} \int_{\alpha 1}^{\alpha 2} g(t)dt, \text{where } t > \alpha 2 \\ \int_{\alpha 1}^{t} g(t)dt, \text{ where } \alpha 1 < t < \alpha 2 \\ 0, \text{where } t < \alpha 1 \end{cases}$$

$$G(t) = \frac{\gamma}{1-\beta} * (t ** (1-\beta))$$

Random value = G(α2) – G(α1)

$= \frac{\gamma}{1-\beta} * (\alpha 2 ** (1-\beta)) - \frac{\gamma}{1-\beta} * (\alpha 1 ** (1-\beta))$

$= \frac{\gamma}{1-\beta} * (\alpha 2 ** (1-\beta) - \alpha 1 ** (1-\beta))$

= 1 t > α2

So when α1 < t < α2, 0 < random value < 1

So random(0,1) = G(t) – G(α1)

$= \frac{\gamma}{1-\beta} * (t ** (1-\beta) - \alpha 1 ** (1-\beta))$

So $t ** (1-\beta) = \frac{1-\beta}{\gamma} * random(0,1) + \alpha 1 ** (1-\beta)$

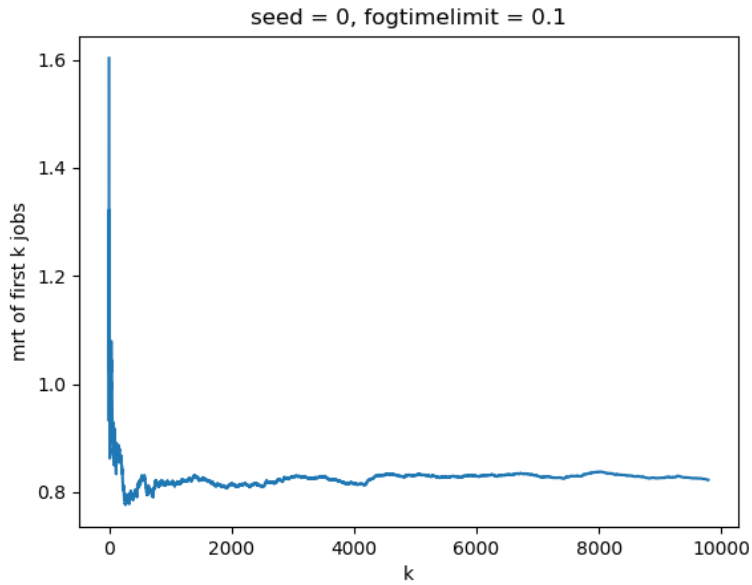
So $t = (\frac{1-\beta}{\gamma} * random(0,1) + \alpha 1 ** (1-\beta)) ** \frac{1}{1-\beta}$

Part 3

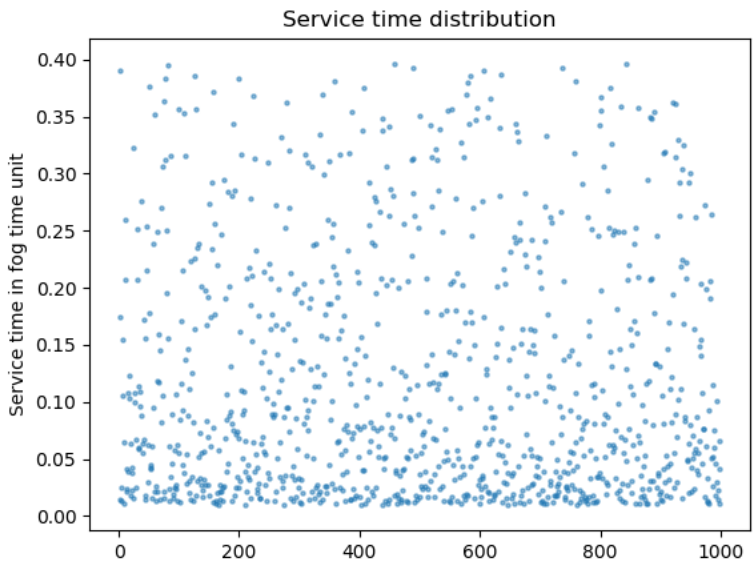
Question 5.2 Determining a suitable value of fogTimeLimit

Given parameter values: λ = 9.72, α 1 = 0.01, α 2 = 0.4, β = 0.86, v 1 = 1.2, v 2 = 1.47 and fogTimeToCloudTime is 0.6. The TimeEnd value is not defined, I set it to be 1000. The random seed was set to be 0.

Firstly, I plotted a graph recording the mrt of first k jobs. It turned out that after k = 500, the transient behaviour trend to steady. In terms of that, I set M = 500. For comparing systems over different fogtimelimit, I remove first 500 jobs.
Run with command **python question5_2_0.py** and get the result.

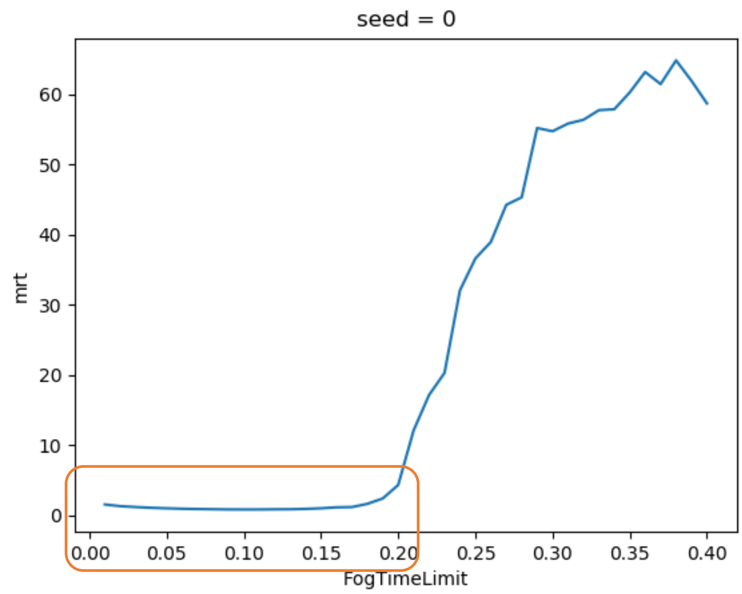


Secondly, I wanted to determine the distribution of service time. I generated 1000 service time in fog time unit by using α 1, α 2 and β.
Run with command **python question5_2_1.py** and get the result.

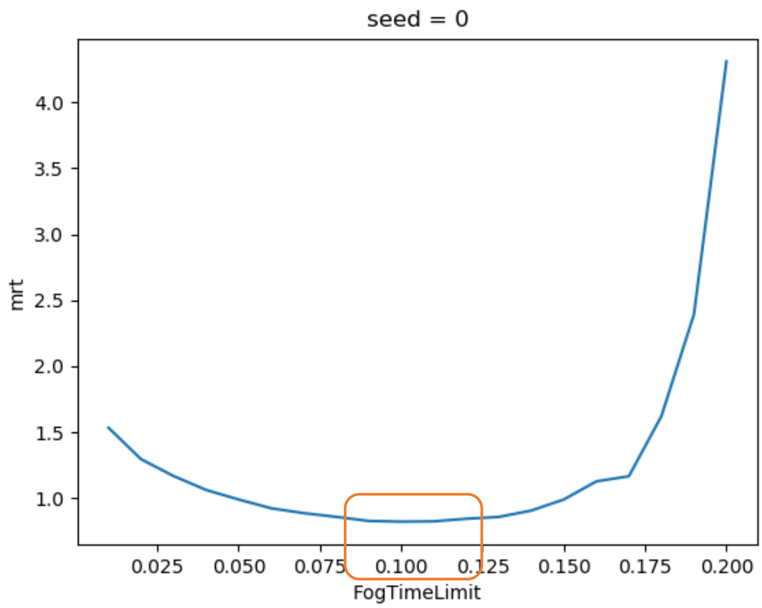


From the figure, we can find that service time < 0.4. Which means if fog time value larger than 0.4, the jobs would be completed at fog and never go to cloud. The mean response time will linearly increase based on FogTimeLimit value.
In order to find the most suitable value of FogTimeLimit that minimize the mean response time (I will use 'mrt' to stand for mean response time in the following text.). The best FogTimeLimit must be in interval [0, 0.4].

Then I plotted a mrt over FogTimeLimit in interval [0,0.4] step 0.01. When calculating mrt, I remove the first 500 jobs to make sure that the state is steady. (Transient removal)
Run with command **python question5_2_2.py** and get the result.



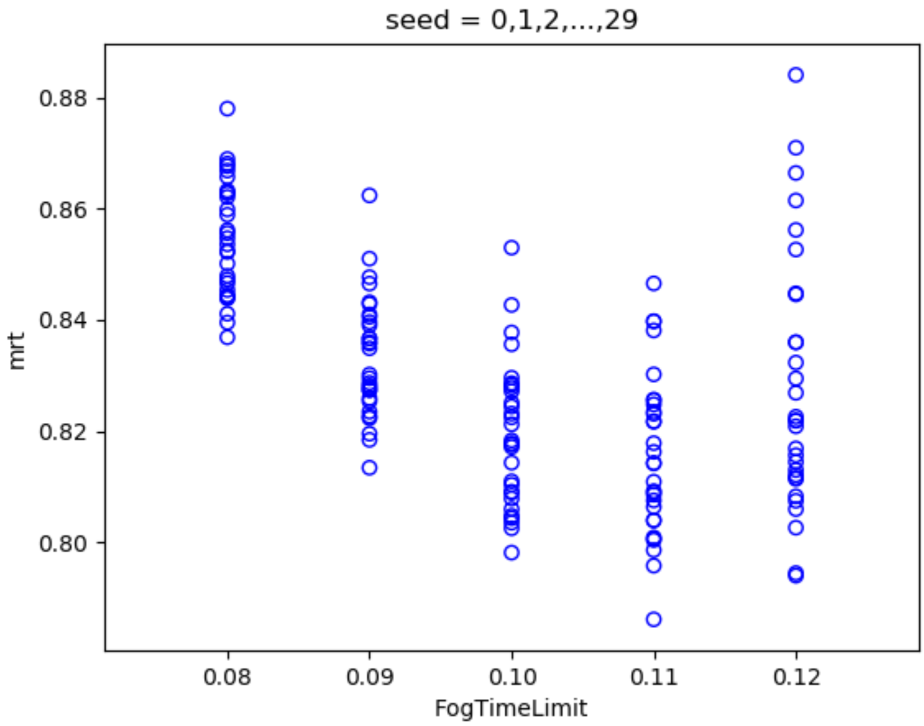
From figure 2 we can see that after FogTimeLimit greater than 0.2, the mrt increases rapidly. Hence we can narrow down the candidate interval to [0, 0.2] and plot again.
Run with command **python question5_2_3.py** and get the result.

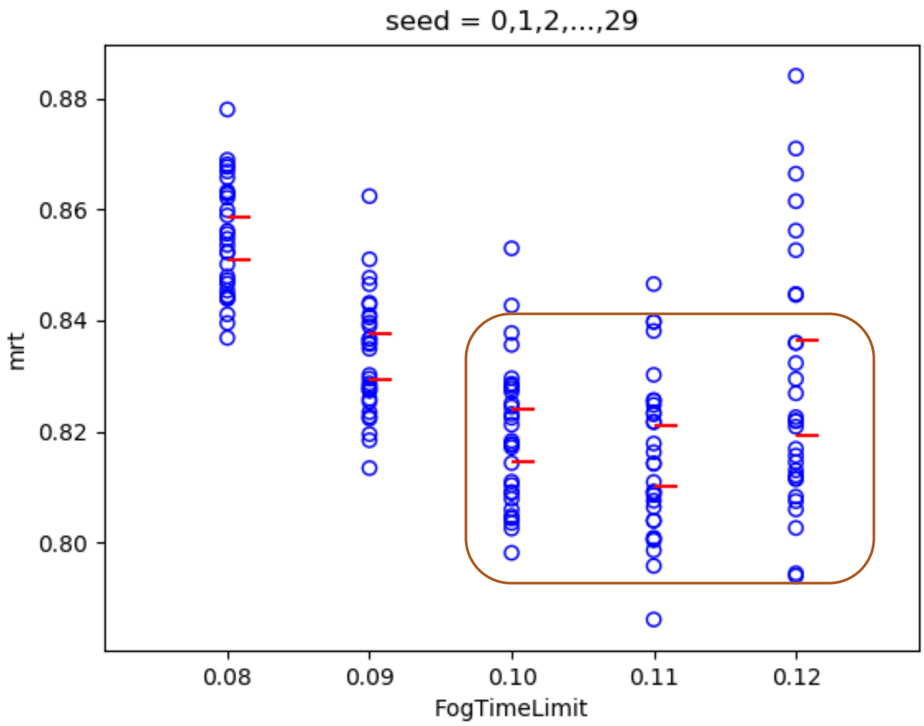


By observing figure 3, it turned out that the mrt is minimal when FogTimeLimit is in interval [0.08,0.12].
Next step is compare FogTimeLimit = [0.08,0.09,0.10,0.11,0.12]. I will experiment with different seeds and compute the 95% confidence interval of each FogTimeLimit. (Sample mean $\pm$ t * sample standard deviation / $\sqrt{n}$), t = 2.045 (look up in the student t distribution form, referenced from https://en.wikipedia.org/wiki/Student%27s_t-distribution)

One-sided	75%	80%	85%	90%	95%	97.5%	99%	99.5%	99.75%	99.9%	99.95%
Two-sided	50%	60%	70%	80%	90%	95%	98%	99%	99.5%	99.8%	99.9%
1	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	127.3	318.3	636.6
2	0.816	1.080	1.386	1.886	2.920	4.303	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.767
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	0.683	0.854	1.055	1.311	1.699	$\boxed{2.045}$	2.462	2.756	3.038	3.396	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.403	2.678	2.937	3.261	3.496
60	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
80	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	2.887	3.195	3.416
100	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	2.871	3.174	3.390
120	0.677	0.845	1.041	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	0.674	0.842	1.036	1.282	1.645	$\boxed{1.960}$	2.326	2.576	2.807	3.090	3.291

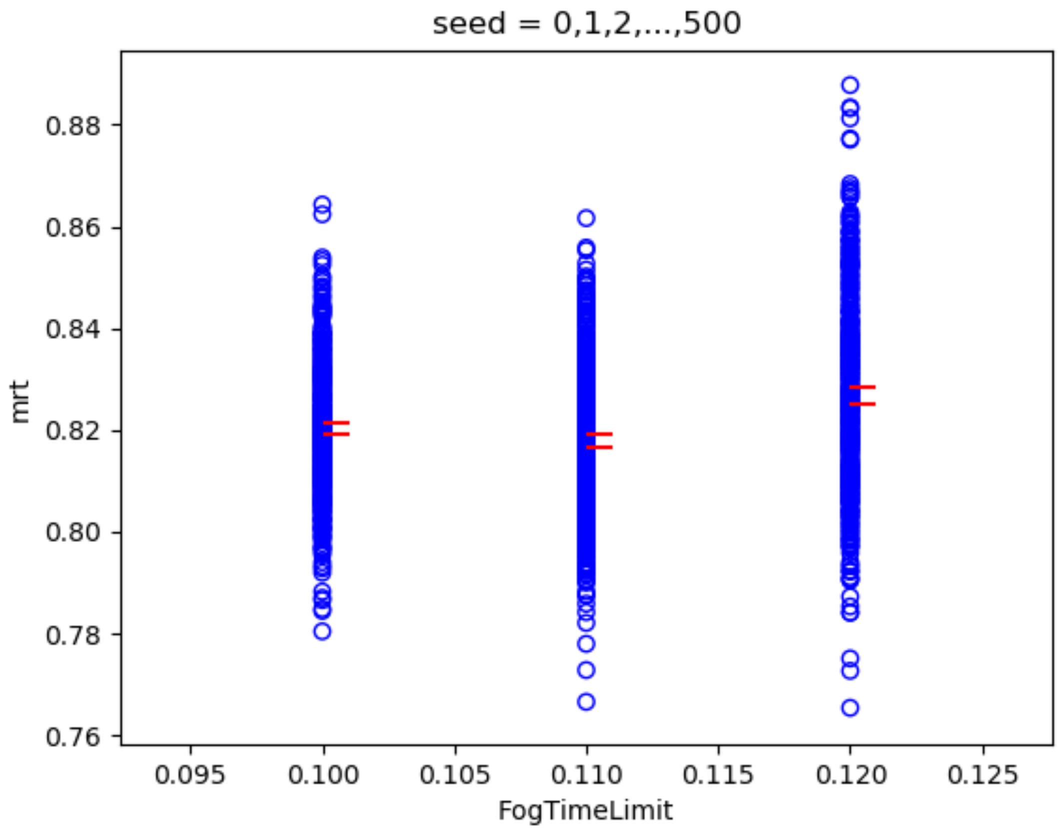
Set FogTimeLimit = [0.08,0.09,0.10,0.11,0.12];
Seed range from 0 to 30.
Run with command **python question5_2_4.py** and get the result.





In the above figure, the blue circles represent different mrt generated by different FogTimeLimit and different random seeds while the red lines represent upper bound and lower bound of confidence interval (95%) of different FogTimeLimit. Since 0.08 and 0.09's lower bound is greater than Min(upper bounds), these two value are not suitable to be the best value that minimize the mrt.

Next step is to compare these three values [0.10,0.11,0.12] and choose a best one.
Set FogTimeLimit = [0.10,0.11,0.12]
Seed range from 0 to 500, t = 1.960
Run with command $\boxed{\textbf{python question5\_2\_5.py}}$ and get the result.




```
0.1 [ 0.8190878034287118 0.8213273842281511 ]
0.11 [ 0.8166584770631462 0.8192313360380572 ]
0.12 [ 0.8250607782741 0.8283897451546822 ]
```

These are the confidence interval for the three fogtimelimits. 0.12's lower bound is greater than 0.11's upper bound, so we can exclude 0.12.

In conclusion, the fogtimelimit set to be in range [0.10, 0.11] will gives the best system response time.