Z5211414

Jin-ao Olson Zhang

COMP9334 23T1 Project Report

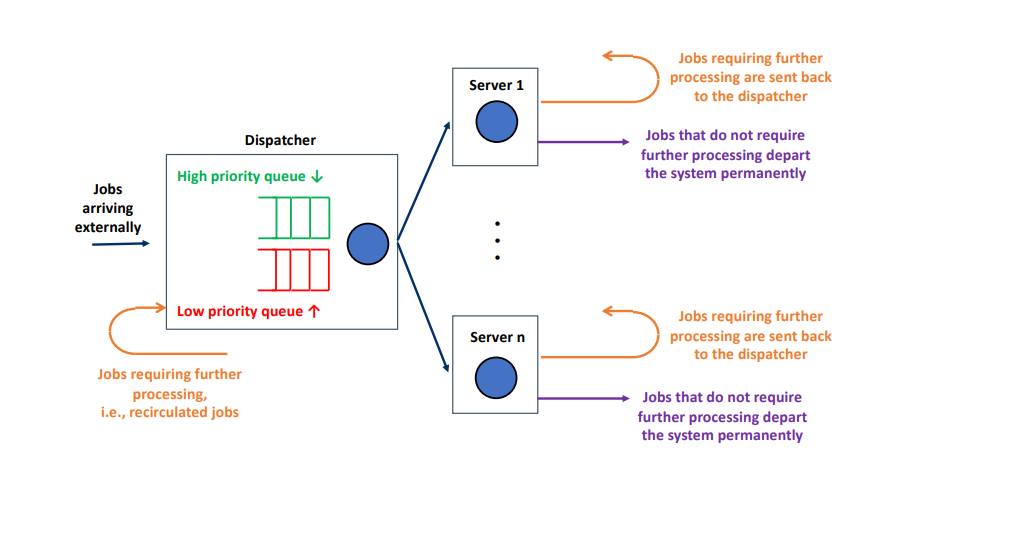
**Declaration for 6.6**

All code is written by myself , except for the use of simulation, which refers to the logic of sim\_mmm\_lib.py in solution4b and also uses master\_clock to monitor the actions of jobs.

There are two main file, main.py and jobclass.py, jobclass file contains all the classed that may use when doing the simulation,which according to the system for this project shows in Figure 1, they are Jobs:

Dispatcher: hold servers and jobs

Server: hold jobs



1. **Simulation program**
   1. **test case correctness**

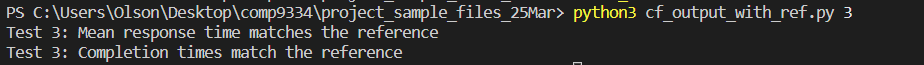
For test the correctness of my code, I used given shell run\_test.sh by

./runtest.sh $index\_of\_test\_case

to get the simulation output,$index\_of\_test\_case was a integer which helps us to find the corresponding reference that used to simulate such as paramater/simulation mode(in 6.1).

After that our code will generate two files dep\_ $index.txt (which contains the completion times of of the server visits from the servers). and mrt\_$index.txt (The mean response time)

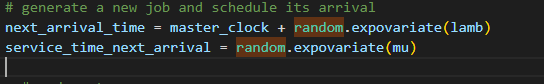
to test the correctness we can run the project given correctness check file cf\_output\_with\_ref.py to know our output was equal or not to the expected output,if so it will print



* 1. **Correctness of the inter-arrival probability distribution**

inter-arrival probability distribution is exponentially distributed with parameter λ. This

means the mean arrival rate of the jobs is λ,in sim\_mmm\_lib.py it used random.expovariate to generate a series of pseudo-random numbers



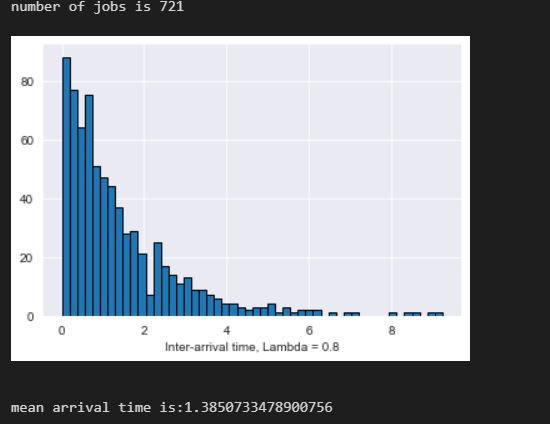
But our project recommend me to generate the inter-arrival times is to multiply an exponentially distributed random number with the given rate and a uniformly distributed random number in the given range,so my new arrival was generated by



lets using test\_case 7 to show the result of inter-arrival probability distribution, we have

|  |  |
| --- | --- |
| End Time | 1000 |
| Lamda | 0.8 |
| Mode | Random |
| Num of jobs | 721 |

By using those argument as the input and plot the distribution of inter arrival time and number of jobs,we got



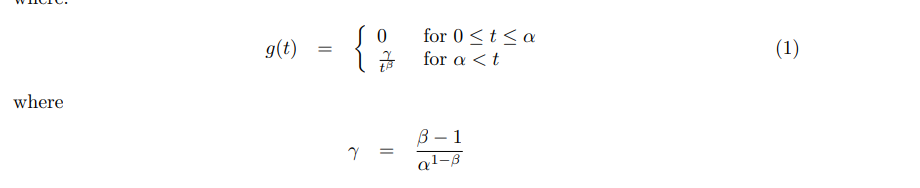
Where lamda here equals to 1.3850

|  |  |  |
| --- | --- | --- |
|  | Expected value | Actual value |
| Inter arrival | 1/lambd\*(alpha2l+alpha2u/2)=1.25\*1.09=1.3625 | 1.385 |

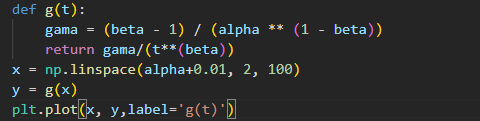
They are almost the same, thus, we can prove the inter-arrival probability distribution of my code are correct.

**1.3 correctness of probability distribution of the number of server visits, and service time distribution**

First know the server time in generate by g(t) followed this distribution



So I printed g(t) as the Comparator of the correctness of my code.

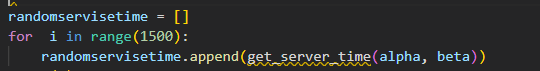


Where x = np.linspace(alpha+0.01,2,100) generate a list of number from alpha+0.01 to 2 with same distance in 100steps.

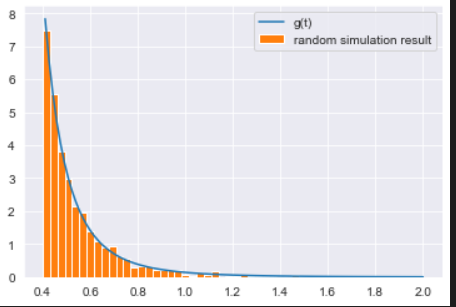
In my code I got a get\_server\_time(alpha,beta) function in main.py to get random server time

Here I also used the reference of test\_case\_7

when alpha = 0.4,beta = 4.5,here I get chose to get 15000 values( No real requirement, but the more times the more accurate)



By plotting both of g(t) and randomservisetime, I got

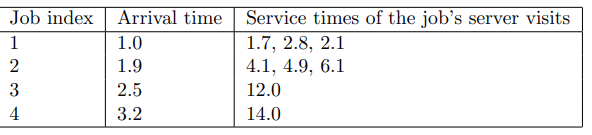


Its easy to find that they are almost the same, thus, we can prove the probability distribution of the number of server visits, and service time distribution of my code are correct.

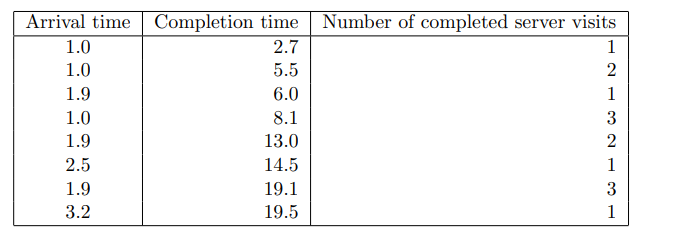
* 1. **simulation correctness**

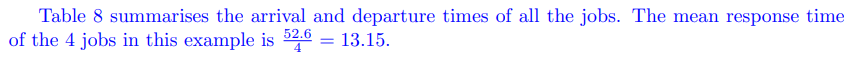
to check the correctness of my simulation,in ”trace” mode I chose one of the examples in section 4 which is example 3

number of servers n = 3, threshold h = 1



As I said befor,cf\_output\_with\_ref.py can help me to check the ouput is or not equal to the expected output, for this example,the expected output was





And here is my code’s output

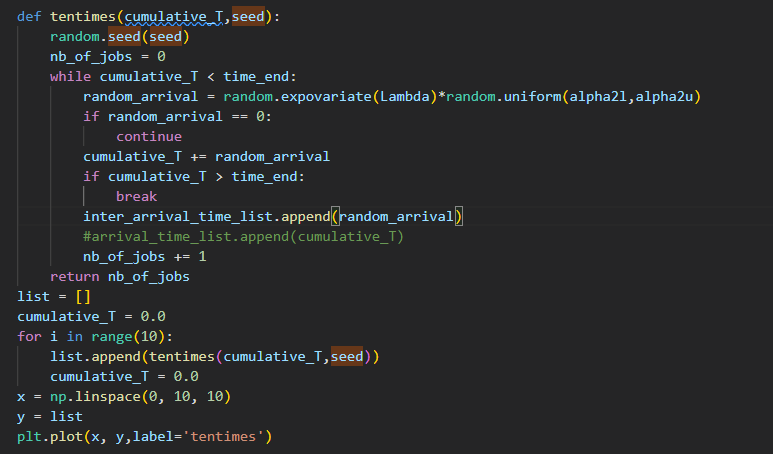


Well for random mode ,I'm not sure how to verify the correctness of the code for now

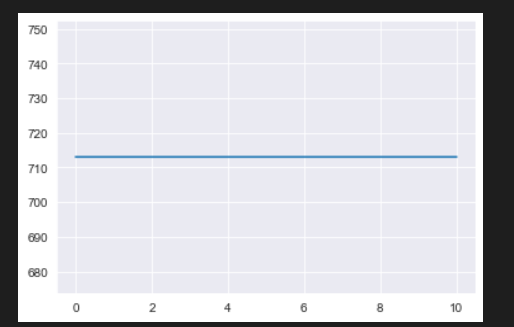
**2.1 reproducible**

In the code, I used random.expovariate() and random.uniform(), its seems that the value generated everytime cannot be same,but to make sure my code is reproducible, I added a random seed=1 into random function, which change it to pseudo-random,so even multiple time used( with the same input) ,the output will remain the same.

Here is a function that plot the changes of jobs in ten different random test with same input.（all the reference are from test case 7 which already shows before)

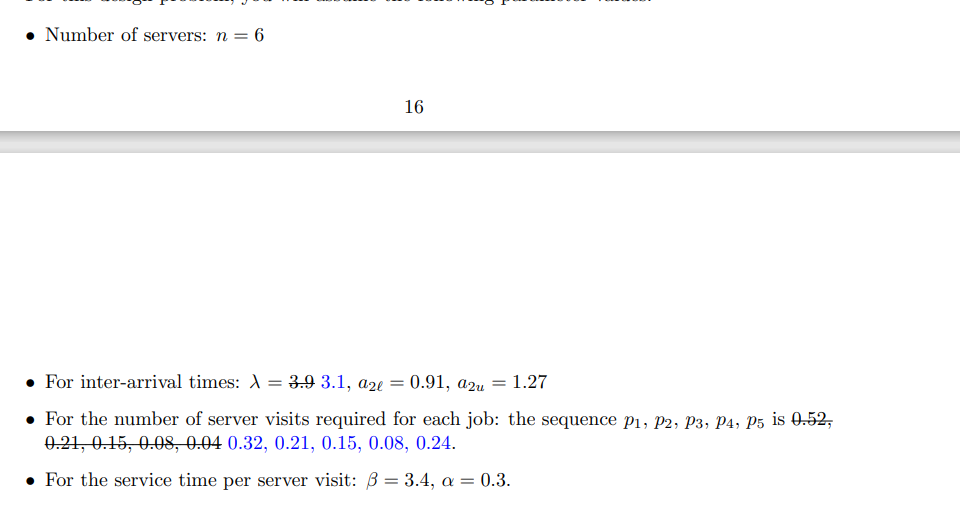


The plot show that it didnt change in 10 time test,so its reproducible

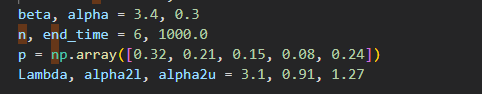


**3.1 determining a suitable value of the threshold h**

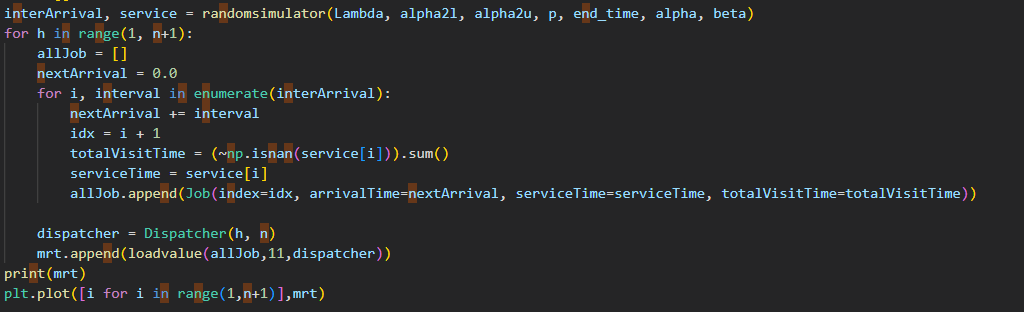
First of all,the 5.2 part of project description want me to assume that



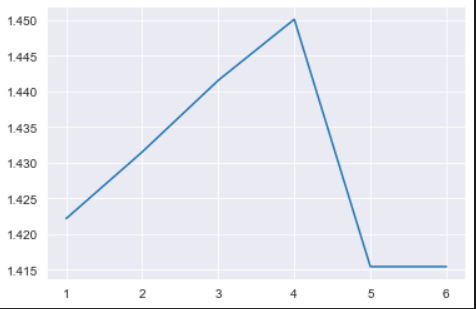
On this basis, I think the best way to find the suitable threshold h is to use the control variables in the statistics，do not change any parameters other than h and observe the change in mean response time for different h



For h from 1 to n， calculate the mrt of simulation



In random seed = 1, the output is



We can find when h=5 to h=6, mrt becomes to the minimum and remain the same.

END of Report