THE UNIVERSITY OF NEW SOUTH WALES

Comp9334

Student number: Z5085805

Student name: Sun YongXin

Report

System Design:

I design four modules in the fork-join system : Server, Processor, Working_Server, Working_Processor.

Depicting the computer system which you will be simulating in this project. The system consists of a pre-procesor, m servers (labelled as Server 1, Server 2, ..., Server m in Figure 1) and a join point (which is shown as a triangle in Figure 1).

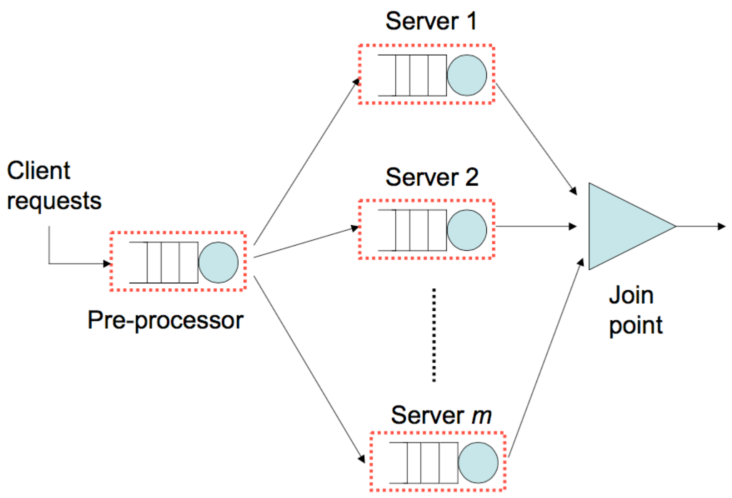


Figure 1: The multi-server system to be simulated in this project.

This fork-join system is mainly about how to handle the association between the pre-processor and several servers. Note that no matter the pre-processor or the servers, they are all based Fist-In-First-Server mode.

Therefore, there are based M/M/1 model.

The entry point of the client request to the system is the pre-processor. The pre-processor has a few functions:

tasks assigned to them by the pre-processor. Once a server has finished processing a sub-task, it will send the result to the join point. You may assume that it takes negligible time to send the results of the sub-tasks from the servers to the join point.

class Server:

I set up a class that have some information about this servers, (server name, arrival time about this departure, subtask_request, number_subtask). The programe (is_busy) record condition of server.

class Processor:

Same as the code **class Server**. Some details with **Processor**.

class Working_Server:

Include arrival next time of server. Easy compute.

class Working_Processor:

Include value of processor. Easy compute.

other:

The following is how get the service time.

$$f(t_s) \quad = \quad \begin{cases} 0 & \text{for } t_s < \frac{t_m}{n^{1.65}} \\ \frac{1}{n^{1.65k}} \frac{k t_m^k}{t_s^{k+1}} & \text{for } t_s \geq \frac{t_m}{n^{1.65}} \end{cases} \qquad (1)$$

where $t_m = 20\frac{k-1}{k} \approx 10.3846$, $k = 2.08$ and n is the number of servers chosen to serve the client request. Note that with the above probability density function, t_s can only take values in the range $[\frac{t_m}{n^{1.65}}, \infty)$.

so, I design this following program:

```
def worktime(n):
    tm = 10.3846
    k = 2.08

    sub_t = (tm ** k) / (n ** (1.65 * k))
    service_time_subtask = (sub_t / (1 - random.uniform(0,1))) ** (1 / k)

    return service_time_subtask
```

How using the system:

```
DanielsdeMacBook-Pro:Desktop danielsapple$ python2.7 projectsyx.py
```

hit Enter

```
Tend is total serve time , set Tend =
```

Input Tend value

For example, Tend = 1000, if you want.

```
Selects n distinct servers out of m servers , set n =
```

Input n value, cannot out 1~10.

For example, n = 5

```
Test No.
```

Input test number, record testing number, because code (random) produce different random number.

Then result is :

```
major_time: 998.997316853
major_time: 999.177514513
major_time: 999.873381039
major_time: 1000.51181403
```

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

Remove transient:

Mean response time: $\hat{T} = \frac{\sum_{i=1}^{n} T(i)}{n}$,

Standard deviation: $\hat{S} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{T}-T(i))^2}{n-1}}$,

Confident interval: $[\hat{T} - t_{n-1,1-\frac{\alpha}{2}}\frac{\hat{S}}{\sqrt{n}}, \hat{T} + t_{n-1,1-\frac{\alpha}{2}}\frac{\hat{S}}{\sqrt{n}}]$,

And I make a graph that the mean response time changes based on the simulation for each n (1~10).
Then by looking at the graph, I find a stable state in this period. Because some graph’ s response time too longer, so I choose three shorter n to compare and analysis.
Choosing one steady state simulation time, and repeat the simulation time. Because the different random number, I try five times, and compute the mean response time, standard deviation and 95% confidence intervals.
Finally, the minimum response time of n is the result.

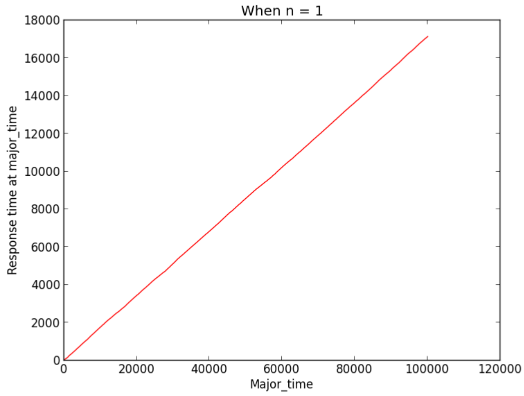
n	tend	T	N	response
1	100000	855100923.1	49893	17138.69527
2	100000	38938269.66	75192	517.8512286
3	100000	2393290.575	74899	31.95357181
4	100000	1343438.576	75526	17.78776283
5	100000	815399.5355	75469	10.8044301
6	100000	803409.7731	75251	10.67639996
7	100000	715222.3656	75160	9.515997414
8	100000	557482.695	75420	7.391709029
9	100000	524090.2398	75277	6.9621563
10	100000	486734.7991	75502	6.446447759

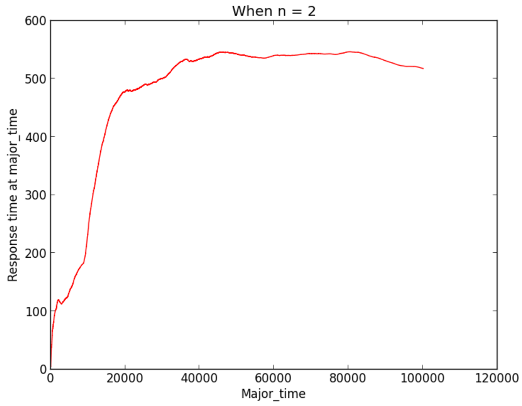
T_o=($T^o_{tend=4500}$-$T^o_{tend=5500}$)/($N^o_{tend=4500}$-$N^o_{tend=5500}$)

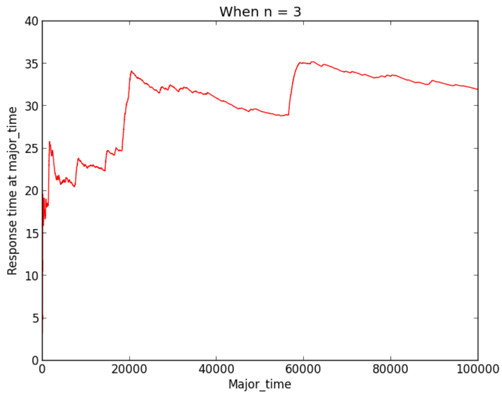
T=(T1+T2+T3)/3

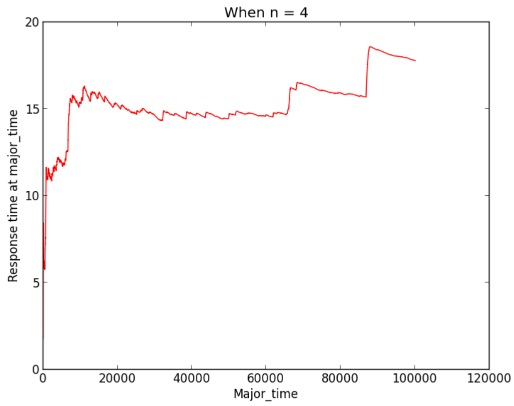
S=(T-T_1)**2+(T-T_2)**2+..+(T-T_o)**2

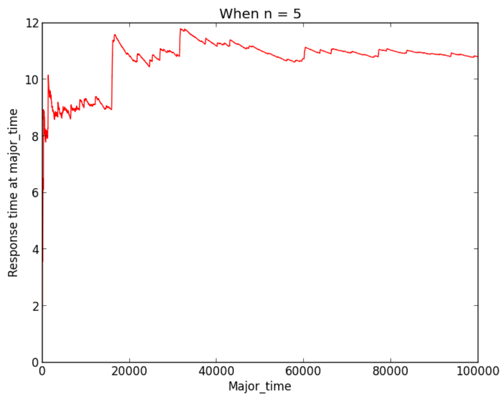
[T-4.303*(S/((3)**(1.0/2)),T+4.303*(S/((3)**(1.0/2)))

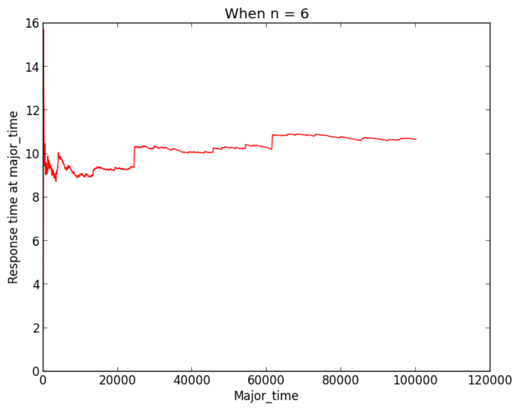


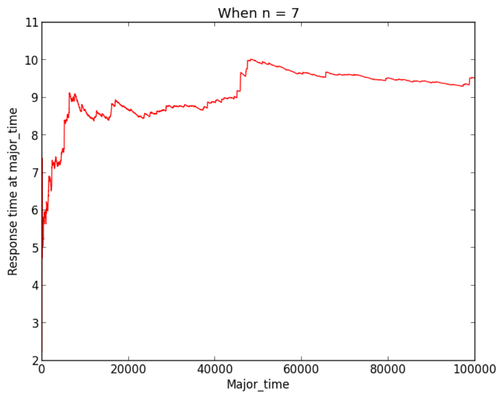




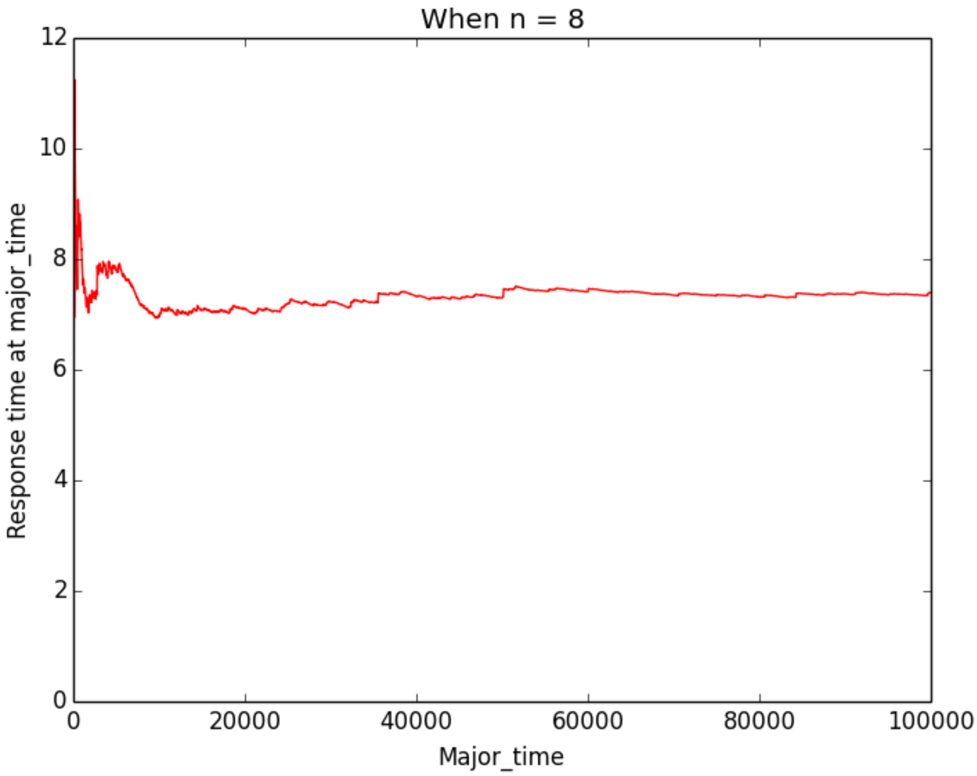








when n = (1,2,3,4,5,6,7)
The response time bigger, so I choose three smaller response time.
Analysis three case, when n =8,9,10.



We can find that the state is steady after Major_time=10000, I choose tend 45000 and 55000, repeat the simulation 3 times(same seed) with different random.

o	n	tend	T	N	response
1	8	45000	319438.2923	35840	8.91289878
2	8	45000	216561.6641	31893	6.790256925
3	8	45000	297467.653	35317	8.422789393
1	8	55000	387001.7539	43793	8.837068799
2	8	55000	273054.4797	38884	7.022283707
3	8	55000	367267.5973	43206	8.500384143

T_o=($T^o_{tend=4500}$-$T^o_{tend=5500}$)/($N^o_{tend=4500}$-$N^o_{tend=5500}$)

T1=8.495342842

T2=8.080791813

T3=8.847755644

T=(T1+T2+T3)/3

=8.4746301

The sample mean of 3 replications = **8.4746301**

S=(((T-T_1)**2+(T-T_2)**2+..+(T-T_o)**2)/2)**(1.0/2)

=0.383901215

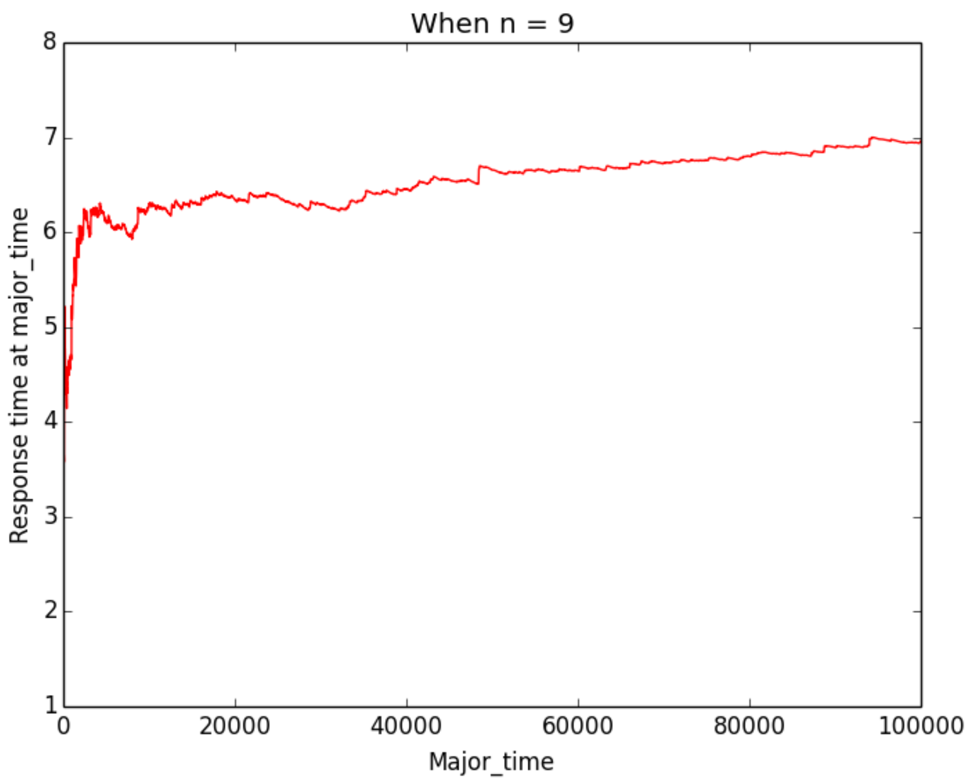
The sample standard deviation of 3 replications is 0.383901215

I want to compute the 95% confidence interval, a = 0.05

Since we did 3 independent experiments and want to 95% confidence interval, I use t(2,0.975)

From the t-distribution table, the value of t(9, 0.975) = 4.303,

therefore the 95% confidence interval:



T_o=($T^o_{tend=4500}$-$T^o_{tend=5500}$)/($N^o_{tend=4500}$-$N^o_{tend=5500}$)

T1=8.084020056

T2=6.078238181

T3=7.56384522

T=(T1+T2+T3)/3

=7.242034486

The sample mean of 3 replications = **7.88646062**

S=(((T-T_1)**2+(T-T_2)**2+..+(T-T_o)**2)/2)**(1.0/2)

=**0.281735178**

The sample standard deviation of 3 replications is **0.281735178**

I want to compute the 95% confidence interval, a = 0.05

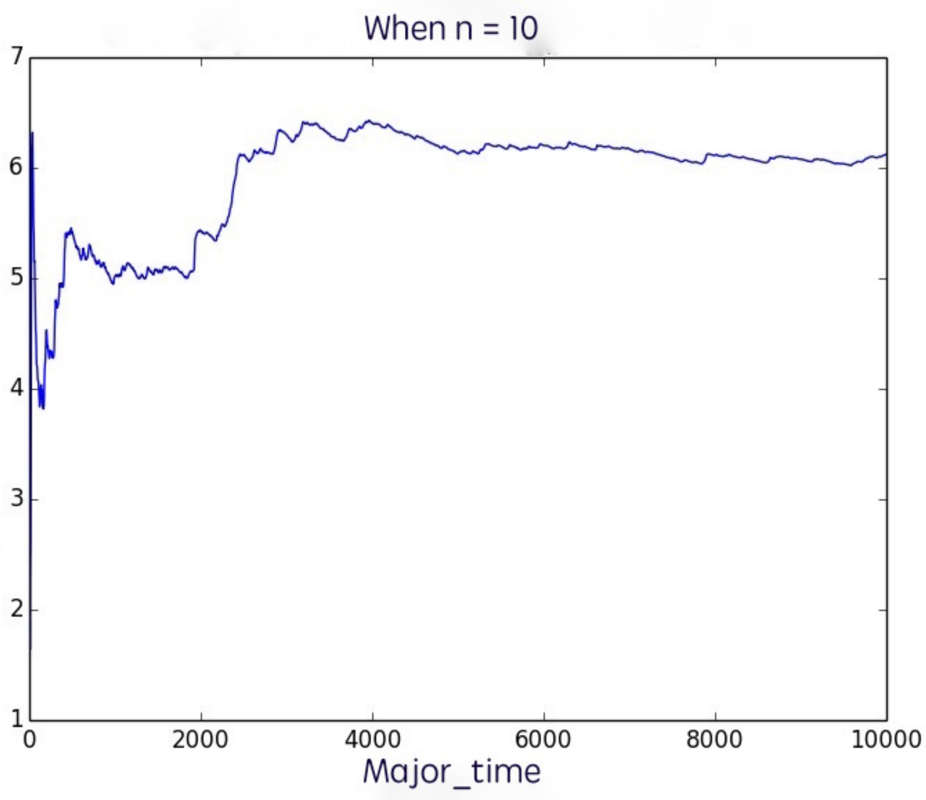
Since we did 3 independent experiments and want to 95% confidence interval, I use t(2,0.975)

From the t-distribution table, the value of t(9, 0.975) = 4.303,

therefore the 95% confidence interval:

[T-4.303*(S/((3)**(1.0/2)),T+4.303*(S/((3)**(1.0/2)))

=**[7.186535152, 8.586386088]**



T_o=($T^{o}_{tend=4500}$-$T^{o}_{tend=5500}$)/($N^{o}_{tend=4500}$-$N^{o}_{tend=5500}$)

T1=8.084020056

T2=6.078238181

T3=7.56384522

T=(T1+T2+T3)/3

=**9.30803834**

The sample mean of 3 replications = 9.30803834

S=(((T-T_1)**2+(T-T_2)**2+..+(T-T_o)**2)/2)**(1.0/2)

=**4.300093143**

The sample standard deviation of 3 replications is **4.300093143**

I want to compute the 95% confidence interval, a = 0.05

Since we did 3 independent experiments and want to 95% confidence interval, I use t(2,0.975)

From the t-distribution table, the value of t(9, 0.975) = 4.303,

therefore the 95% confidence interval:

[T-4.303*(S/((3)**(1.0/2)),T+4.303*(S/((3)**(1.0/2)))

=**[-1.374847355, 19.99092404]**

Summary of Statistical Analysis:

n	confident interval	mean value of confident interval
1~7	**bigger**	**bigger**
8	**[7.520889643, 9.428370556]**	**8.4746301**
9	**[4.656099931, 9.827969041]**	**7.242034486**
10	**[-1.374847355, 19.99092404]**	**9.30803834**

The minimum response time of 9 is the result.

Data Support:

o	n	tend	T	N	response	To	t	s	interval 1	interval 2
1	8	45000	319438.2923	35840	8.91289878	8.4953	8.4746	0.3839	7.52088964	9.428370556
2	8	45000	216561.6641	31893	6.790256925	8.0808				
3	8	45000	297467.653	35317	8.422789393	8.8478				
1	8	55000	387001.7539	43793	8.837068799					
2	8	55000	273054.4797	38884	7.022283707					
3	8	55000	367267.5973	43206	8.500384143					
o	n	tend	T	N	response					
1	9	45000	279332.4222	35620	7.842010729	8.084	7.8865	0.28174	7.18653515	8.586386088
2	9	45000	280361.8722	35741	7.742499361	8.0115				
3	9	45000	283746.2333	35530	7.986102823	7.5638				
1	9	55000	344206.6831	43645	7.88650895					
2	9	55000	344269.74	43718	7.78675913					
3	9	55000	342895.5029	43350	7.909930863					
o	n	tend	T	N	response					
1	10	45000	263244.6877	35647	7.38476415	14.07	9.308	4.30009	-1.3748474	19.99092404
2	10	45000	201596.1038	31678	6.363915139	5.7095				
3	10	45000	238376.5859	35018	6.807258721	8.1445				
1	10	55000	374778.9713	43574	8.600976989					
2	10	55000	242241.7307	38797	6.243826345					
3	10	55000	301642.9208	42786	7.050037883					