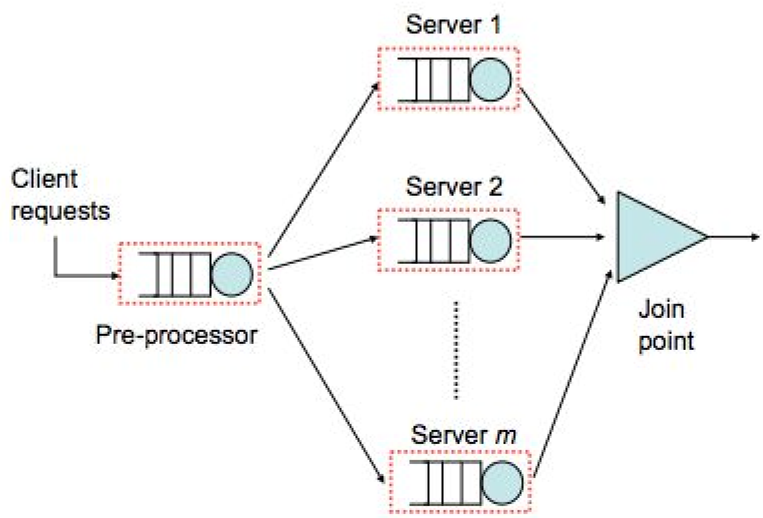
COMP9334 Tai Hang Fu, z3418854, thfu013

COMP9334 Capacity Planning of Computer System and Network

Introduction

1. Background

The simulation is known as Fork-Join System, which is a simulation model of a multiple-server system.



2. Parameter used

We have assumed the following as to investigate the performance of the system:

The inter-arrival time {a(1),a(2)...a(k)}of the **client requests**

For each a(k) = a(1)(k) + a(2)(k), where k = 1,2,3..; a(1)(k) is exponentially distributed with a mean arrival rate = 0.85 requests/s; a(2)(k) is uniformly distributed in the interval [0.05, 0.25]

The service time required by a **client requests**

Exponentially distributed with a mean service time: (n/10) s/ request where n is the number of servers selected

The service time required by each sub-task at each server

Independent
Independently distributed
Probability density function f(t(s)) of the service time t(s) at the server is according to the Pareto distribution:

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

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 $[\frac{t_m}{n^{1.65}}, \infty)$.

m(number of Servers) = 10
n(number of Servers used) <= m

3. Goal
Simulates the system for different choices of n(number of server selected) and generate required probability distribution such that it gives the smallest mean response time.

Uses statistically sound method to compare systems and find out if one system is better than other.

Program Simulation

The program is written on Matlab and the statistic analysis is also on Matlab.

The simulation program takes number of servers used, number of clients requests as input and output the mean response time. include pre-processor that process clients request with inter-arrival time a(k) = a(1)(k) + a(2)(k) as mentioned above and it takes service time, (n/10) s/request with Exponentially distributed. It then pass the sub tasks to the servers that required a service time according to the Pareto distribution.

Time to break and deliver sub tasks from pre-processor to the servers are neglected. Time to deliver task from servers to the joint point and the processing time on the joint point also neglected. It also takes the join point negligible time to assemble and deliver the final result.

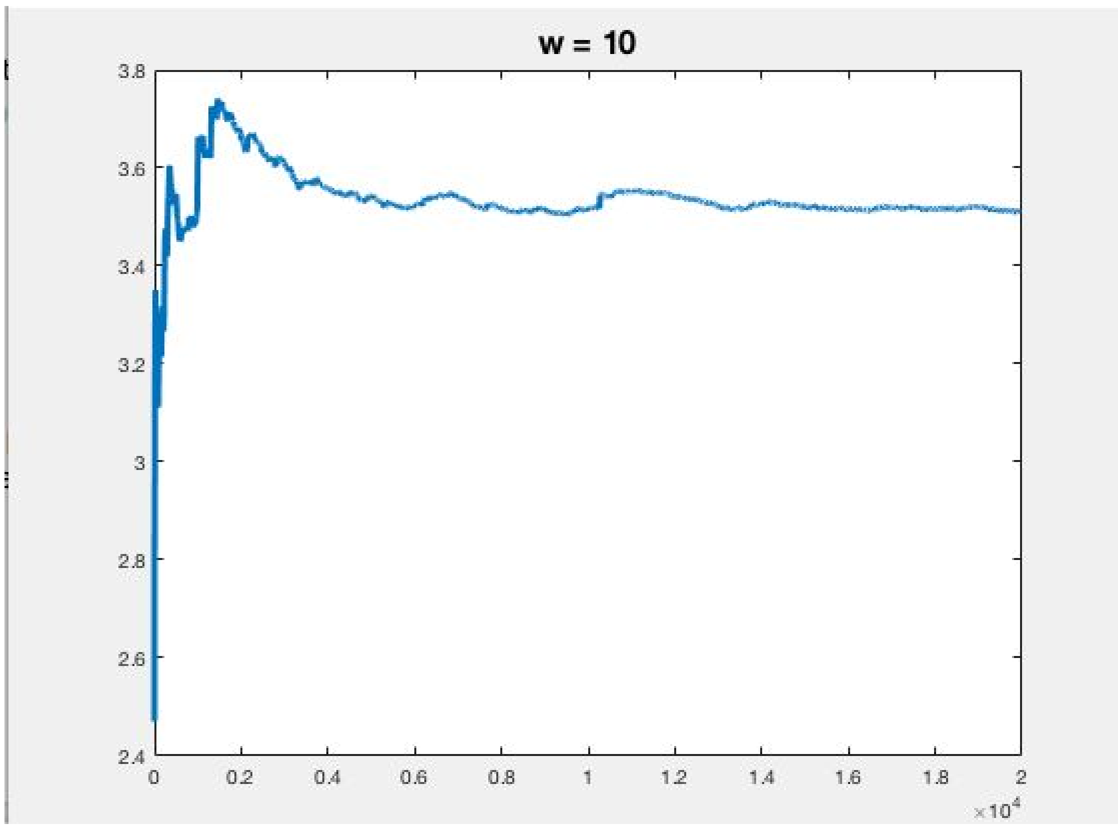
COMPARISON AND ANALYSIS

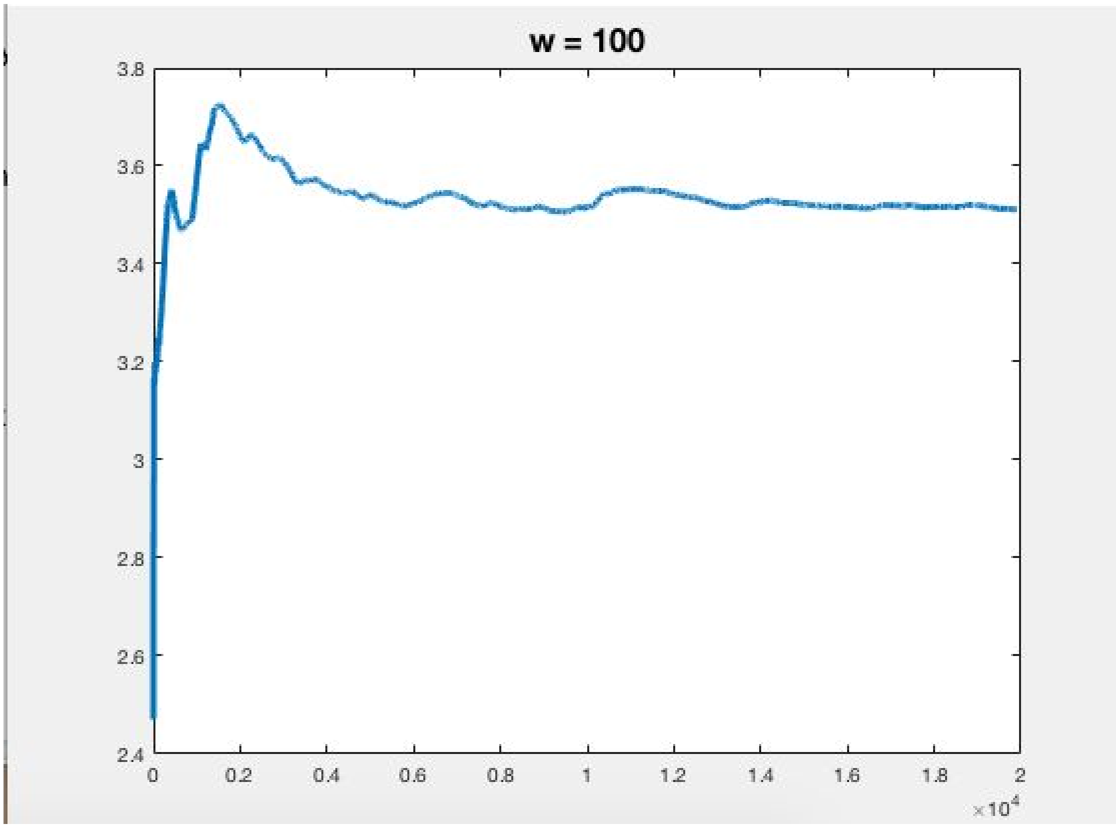
It is important to do sound statistical analysis on the simulation results.

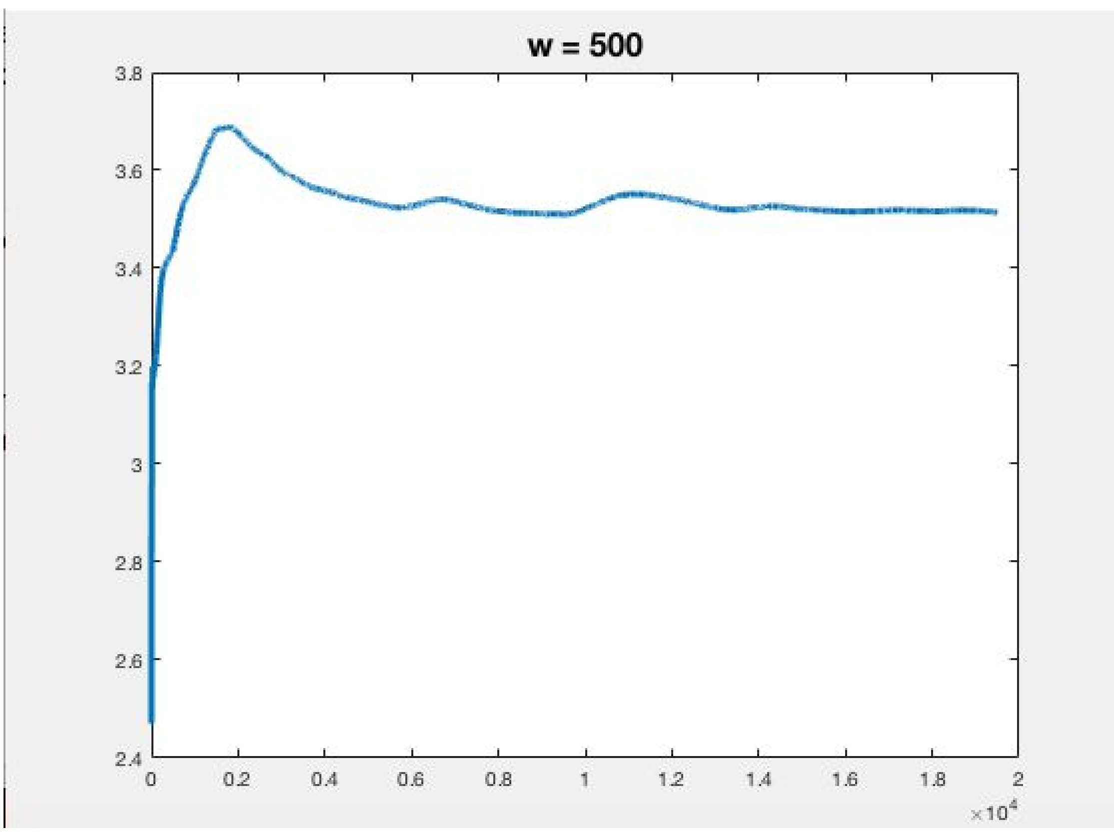
We are interested in the steady state value, so we avoid including the transient part of the data to compute the steady state value.

Under the n = 5(number of Servers used = 5), we used 20000 client requests to observe the response time (running **transient.m** on **save_arr**.mat)

Different values of w(e.g. 10,100..500), and seems that w=500 is able to get a pretty smooth graph applying **transient removal procedure** in Law and Kelton




w = 100
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^4


w = 500
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^4

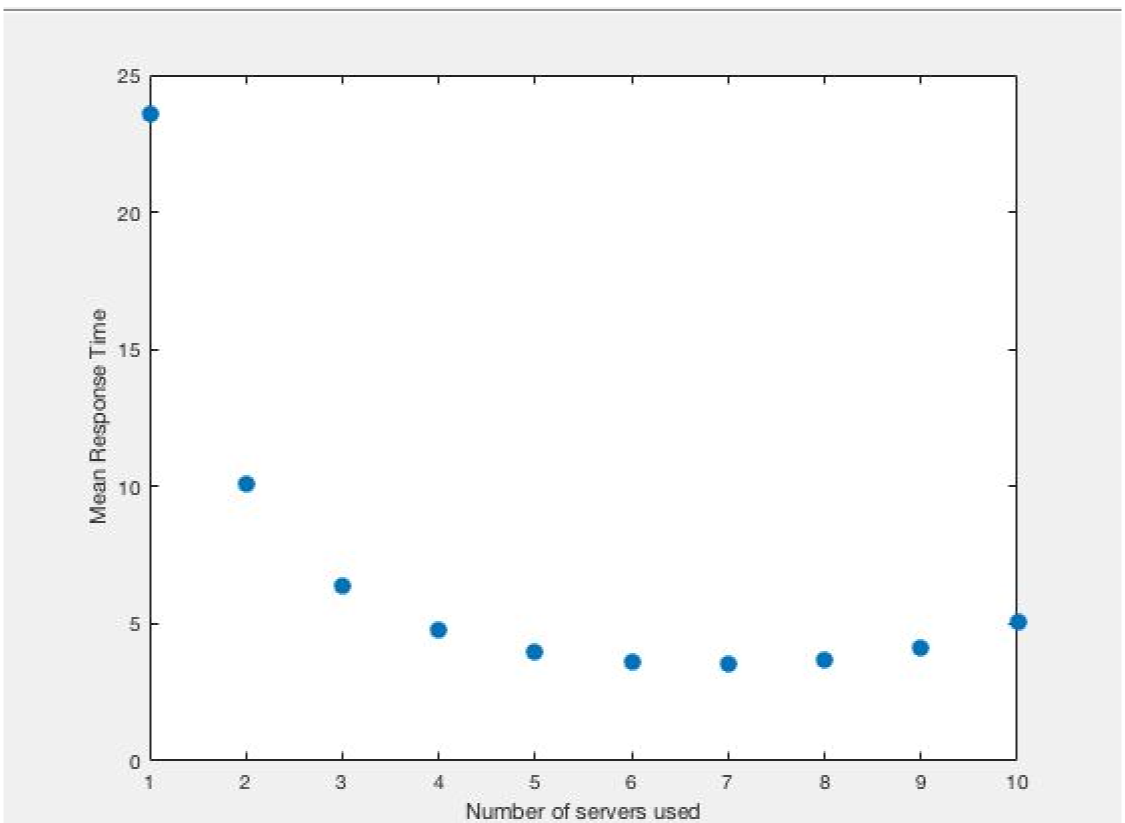
By visual inspection(on the graph where **w = 500**), we can observed the early part of the simulation displays nonsteady state while the later part of the simulation converges around the steady state value

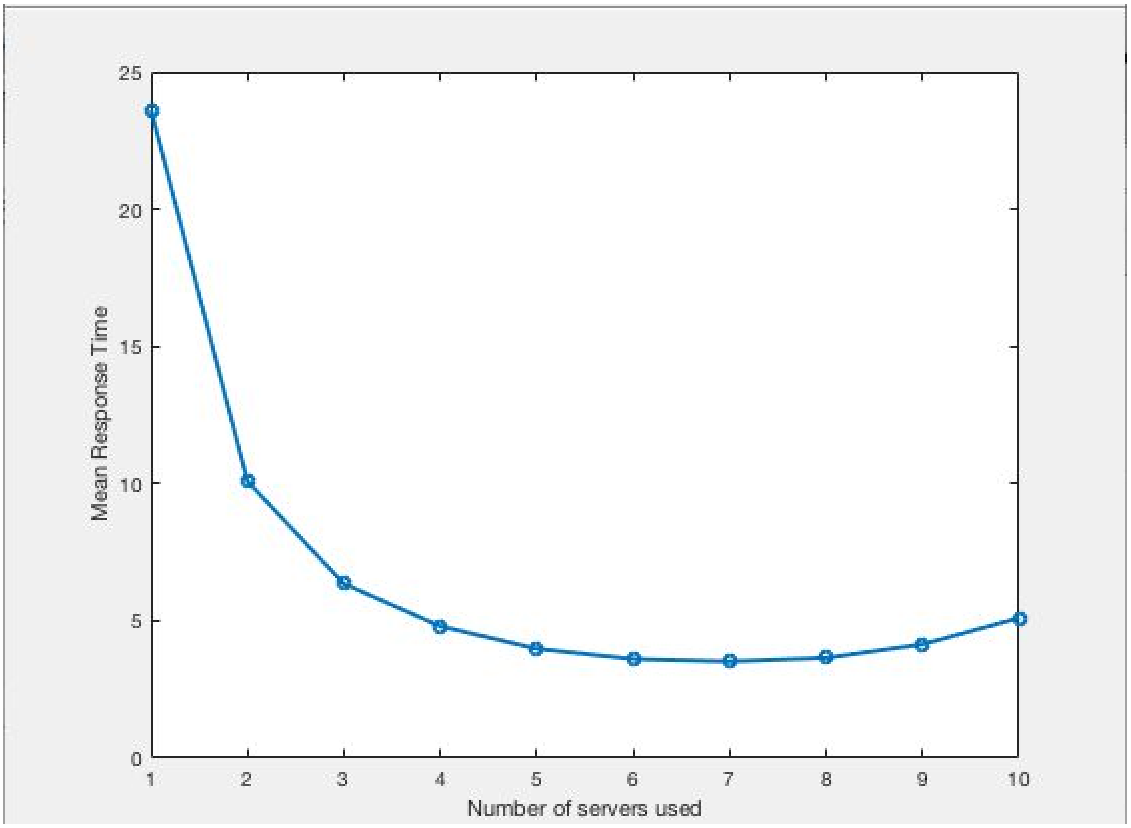
From the graph above(w=500), we learn that the response time does not settle to a constant value. We want to do is removing the transient part, so at last we cut off the early 4000 requests (20000 tasks as the total client requests).

1.Mean Response Time for n(number of servers used) = 1:10 with Sample size = 15

After running 15 independent replications(started from 5 times, then gradually increased) for each possible number of servers used (n), we can get an approximate mean response time of the system

(*All **saved_rand_setting** files**, load_rng_setting.m, table_generate, interval.m** have been used for the following graphs and table**)***





2. Table of all 15 Independent Replications records, Sample Mean of all Replications Mean Response Time and Sample Standard Deviation

Where n = number of servers used

	n = 1	n = 2	n = 3	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9	n = 10
1st Replication	23.5075116	10.02006229	6.345133853	4.744318709	3.939372213	3.556960943	3.563079719	3.688242527	4.208956199	5.039389367
2nd Replication	24.29419409	10.13729996	6.37027508	4.795643314	4.000948678	3.559113183	3.514165969	3.63867767	4.184466646	5.016833593
3rd Replication	23.429453	10.0810328	6.381515422	4.752641706	3.978697532	3.565457428	3.453611913	3.704270322	4.295848125	5.14375099
4th Replication	23.23220987	10.05503487	6.335883456	4.821952219	3.941305627	3.54866666	3.497456945	3.68396497	4.151522762	5.31142204
5th Replication	23.48917456	10.06184987	6.334337581	4.696426658	3.93117766	3.574071849	3.506630028	3.724367949	4.021383987	5.232456863

6th Replication	23.88339956	10.04158489	6.38908159	4.845866939	4.069585508	3.615682223	3.498706213	3.629019451	4.141396273	5.157280466
7th Replication	23.67279065	10.09041401	6.31394947	4.863918763	4.02241782	3.583148777	3.542193687	3.600511803	4.133728197	4.98014901
8th Replication	23.57988976	10.01558867	6.26528515	4.846986292	4.022027617	3.710420972	3.499225926	3.562764241	4.120454369	5.135499302
9th Replication	23.54494669	10.08448173	6.348260351	4.845384018	3.962664592	3.624626804	3.526622247	3.678726609	4.118300499	5.247886904
10th Replication	23.45308097	10.2548058	6.343909884	4.801408346	3.945608587	3.605114283	3.483552166	3.66218665	4.173674671	5.020858555
11th Replication	23.6001297	10.07491464	6.383614542	4.809489297	4.021183707	3.607020998	3.606770777	3.686546031	4.150862784	5.14124577
12th Replication	23.47461527	10.03428816	6.353909987	4.875552069	3.980573057	3.619097553	3.559497416	3.586366323	4.125734585	4.976857835
13th Replication	23.48917456	10.18023769	6.301805629	4.79990226	3.969849705	3.609719569	3.537950144	3.65311953	4.140273818	4.980166246
14th Replication	23.88339956	10.07660607	6.33693404	4.769820186	4.015172202	3.644990681	3.516249613	3.624951751	4.141007822	4.870645203
15th Replication	23.67279065	10.0546502	6.367193204	4.739301806	3.937629316	3.649871031	3.450273797	3.603261886	4.05208807	5.255510799
Sample Mean of All Mean Response Time	***23.61378403***	***10.08419011***	***6.344739283***	***4.800574172***	***3.982547588***	***3.604930864***	***3.517065771***	***3.648465181***	***4.14397992***	***5.1006353***
Sample Standard	***0.2519211022***	***0.06363597159***	***0.03320436979***	***0.05169874182***	***0.04121454065***	***0.04312361418***	***0.04131080119***	***0.0469531377***	***0.06271539057***	***0.1281165562***

Deviation										

By just looking at the graphs and table above,
we can observe the Mean response time start dramatically drops from n = 1,
and the Mean Response Time starts getting stable when Number of servers used is about 5,
and starts slightly increase after Number of servers used = 8.

ANALYSIS OF DIFFERENT SYSTEM

Student's t-distribution (or simply the t-distribution) is a part of continuous probability distributions that arise when estimating the mean of a normally distributed population in situations where the sample size is small(***usually <30***) and population standard deviation is unknown.

As to compute the CI for all replications in all cases (n=1:10),
We repeat the experiment 15 times using different sets of random numbers ***as mention before***.
For each independent experiment recorded the response time and looked for the Confidence Interval.
Confidence interval is computed using the Sample Mean, Sample Standard Deviation from the tables, as well as the formula below with looking up t-distribution table.

Student t-distribution table

TABLE A.4 Quantiles of the t Distribution

	p							
n	0.6000	0.7000	0.8000	0.9000	0.9500	0.9750	0.9950	0.9995
1	0.325	0.727	1.377	3.078	6.314	12.706	63.657	636.619
2	0.289	0.617	1.061	1.886	2.920	4.303	9.925	31.599
3	0.277	0.584	0.978	1.638	2.353	3.182	5.841	12.924
4	0.271	0.569	0.941	1.533	2.132	2.776	4.604	8.610
5	0.267	0.559	0.920	1.476	2.015	2.571	4.032	6.869
6	0.265	0.553	0.906	1.440	1.943	2.447	3.707	5.959
7	0.263	0.549	0.896	1.415	1.895	2.365	3.499	5.408
8	0.262	0.546	0.889	1.397	1.860	2.306	3.355	5.041
9	0.261	0.543	0.883	1.383	1.833	2.262	3.250	4.781
10	0.260	0.542	0.879	1.372	1.812	2.228	3.169	4.587
11	0.260	0.540	0.876	1.363	1.796	2.201	3.106	4.437
12	0.259	0.539	0.873	1.356	1.782	2.179	3.055	4.318
13	0.259	0.538	0.870	1.350	1.771	2.160	3.012	4.221
14	0.258	0.537	0.868	1.345	1.761	2.145	2.977	4.140
15	0.258	0.536	0.866	1.341	1.753	2.131	2.947	4.073
16	0.258	0.535	0.865	1.337	1.746	2.120	2.921	4.015
17	0.257	0.534	0.863	1.333	1.740	2.110	2.898	3.965
18	0.257	0.534	0.862	1.330	1.734	2.101	2.878	3.922
19	0.257	0.533	0.861	1.328	1.729	2.093	2.861	3.883
20	0.257	0.533	0.860	1.325	1.725	2.086	2.845	3.850
21	0.257	0.532	0.859	1.323	1.721	2.080	2.831	3.819
22	0.256	0.532	0.858	1.321	1.717	2.074	2.819	3.792
23	0.256	0.532	0.858	1.319	1.714	2.069	2.807	3.768
24	0.256	0.531	0.857	1.318	1.711	2.064	2.797	3.745
25	0.256	0.531	0.856	1.316	1.708	2.060	2.787	3.725
26	0.256	0.531	0.856	1.315	1.706	2.056	2.779	3.707
27	0.256	0.531	0.855	1.314	1.703	2.052	2.771	3.690
28	0.256	0.530	0.855	1.313	1.701	2.048	2.763	3.674
29	0.256	0.530	0.854	1.311	1.699	2.045	2.756	3.659
30	0.256	0.530	0.854	1.310	1.697	2.042	2.750	3.646
60	0.254	0.527	0.848	1.296	1.671	2.000	2.660	3.460
90	0.254	0.526	0.846	1.291	1.662	1.987	2.632	3.402
120	0.254	0.526	0.845	1.289	1.658	1.980	2.617	3.373

Confidence Interval

$$\hat{T} = \frac{\sum_{i=1}^{n} T(i)}{n} \qquad \hat{S} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{T} - T(i))^2}{n-1}}$$

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

As number of replications = 15(Sample Size = 15) and we want 95% Confidence Interval, from the table above, we know that **t(n-1,1-alpha/2)** = t(15-1,1-0.05/2) = t(14,0.975) ~= 2.145

	n = 1	n = 2	n = 3	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9	n = 10
Sample Mean of all replications	23.61378403	10.08419011	6.344739283	4.800574172	3.982547588	3.604930864	3.517065771	3.648465181	4.14397992	5.10066353

Mean Response Time										
Sample Standard Deviation	0.2519211022	0.06363597159	0.03320436979	0.05169874182	0.04121454065	0.04312361418	0.04131080119	0.0469531377	0.06271539057	0.1281165562
Lower CI	***23.46937823***	***10.0477128***	***6.325705928***	***4.770939504***	***3.958922657***	***3.580211616***	***3.493385661***	***3.621550781***	***4.108030307***	***5.027224768***
Upper CI	***23.75818983***	***10.12066742***	***6.363772637***	***4.83020884***	***4.006172519***	***3.629650111***	***3.54074588***	***3.675379581***	***4.179929533***	***5.174102291***

From the table above:

When number of servers used = **1**,
The 95% confidence interval =[***23.46937823, 23.75818983***]

When number of servers used = **2**,
The 95% confidence interval =[***10.0477128, 10.12066742***]

When number of servers used = **3**,
The 95% confidence interval =[***6.325705928, 6.363772637***]

When number of servers used = **4**,
The 95% confidence interval =[***4.770939504, 4.83020884***]

When number of servers used = **5**,
The 95% confidence interval =[***3.958922657, 4.006172519***]

When number of servers used = **6**,
The 95% confidence interval =[***3.580211616, 3.629650111***]

When number of servers used = **7**,
The 95% confidence interval =[***3.493385661, 3.54074588***]

When number of servers used = **8**,
The 95% confidence interval =[***3.621550781, 3.675379581***]

When number of servers used = **9**,

The 95% confidence interval =[***4.108030307, 4.179929533***]

When number of servers used = **10**,
The 95% confidence interval =[***5.027224768, 5.174102291***]

1. Approximate visual test on Confidence interval and Mean Response Time(*generated by CI_interval.m*)

<u>On the following Graphs, RED LINE represents the 95% Confidence Interval while the BLUE DOT within the red line represents the MEAN</u>

Obviously, the CI of **System n=1**(with Mean= 23.61378403 and CI=[***23.46937823, 23.75818983***]) DOES NOT Overlap any other Systems and The Mean and CI is much higher than all others(where n >1), thus we can conclude that we are confident that 95% of the chance True Mean of **System n=1** is much greater than any other Systems.

<u>And thus **System n = 1** is worse than **all other Systems**.</u>

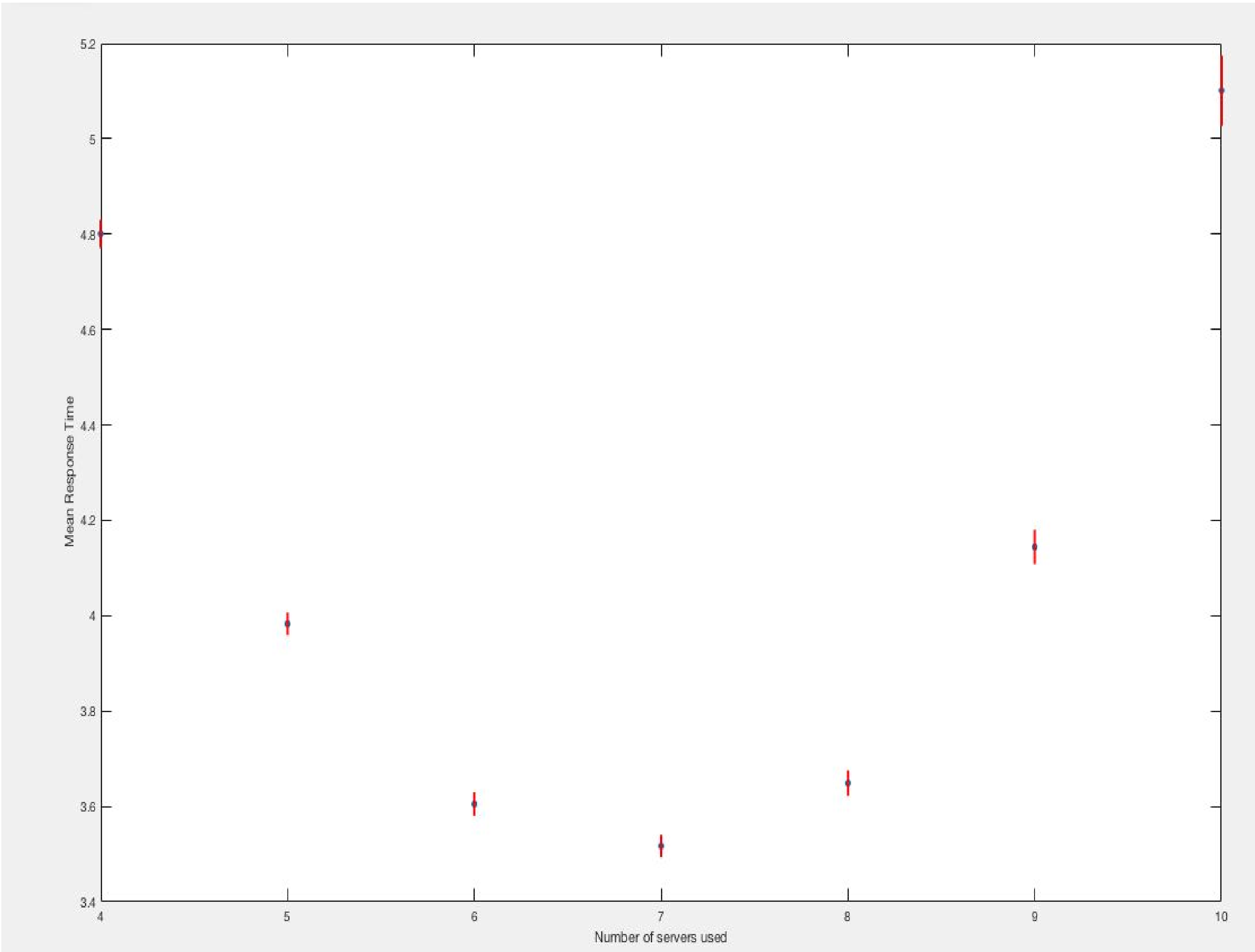
Similarly, the CI of **System n=2**(Mean = 10.08419011, CI=[***10.0477128, 10.12066742***]) DOES NOT Overlap any other Systems and The Mean and CI is much higher than all others(where n >2), thus we can conclude that we are confident that 95% of the chance True Mean of **System n=2** is much greater than any other Systems with n>2.

<u>And thus **System n = 2** is worse than all other **Systems with n>2**.</u>

The CI of **System n=3**(Mean = 6.344739283, CI=[***6.325705928, 6.363772637***]) DOES NOT Overlap any other Systems and The Mean and CI is much higher than all others(where n >3), thus we can conclude that we are confident that 95% of the chance True Mean of **System n=3** is much greater than any other Systems with n>3.

<u>And thus **System n = 3** is worse than all other **Systems with n>3.**</u>

Now Let us compare **System n=4** with all others (except n=1, n=2, n=3 since they are obviously performs worse than any others as mention above), By looking at the **Table** above and the **Graph(**using ***interval.m*****)** below:



The Left Most is the Server n =4, the Right Most is the Server n =10, where all others between are system n =5,6,7,8,9

Obviously, **System n=4** with Mean = 4.800574172, CI=[***4.770939504, 4.83020884***] , whose CI DOES NOT Overlap with with any others' Mean and CI.
And from the graph, we can observe that the Mean and CI for **System n=4** is much higher than **System with n=5,6,7,8,9**, thus we can conclude that we are confident that 95% of the chance True Mean of **System n=4** is much **Greater** than any other **Systems with n=5,6,7,8,9**.
And thus **System n = 4** is worse than all other **Systems with n=5,6,7,8,9**.

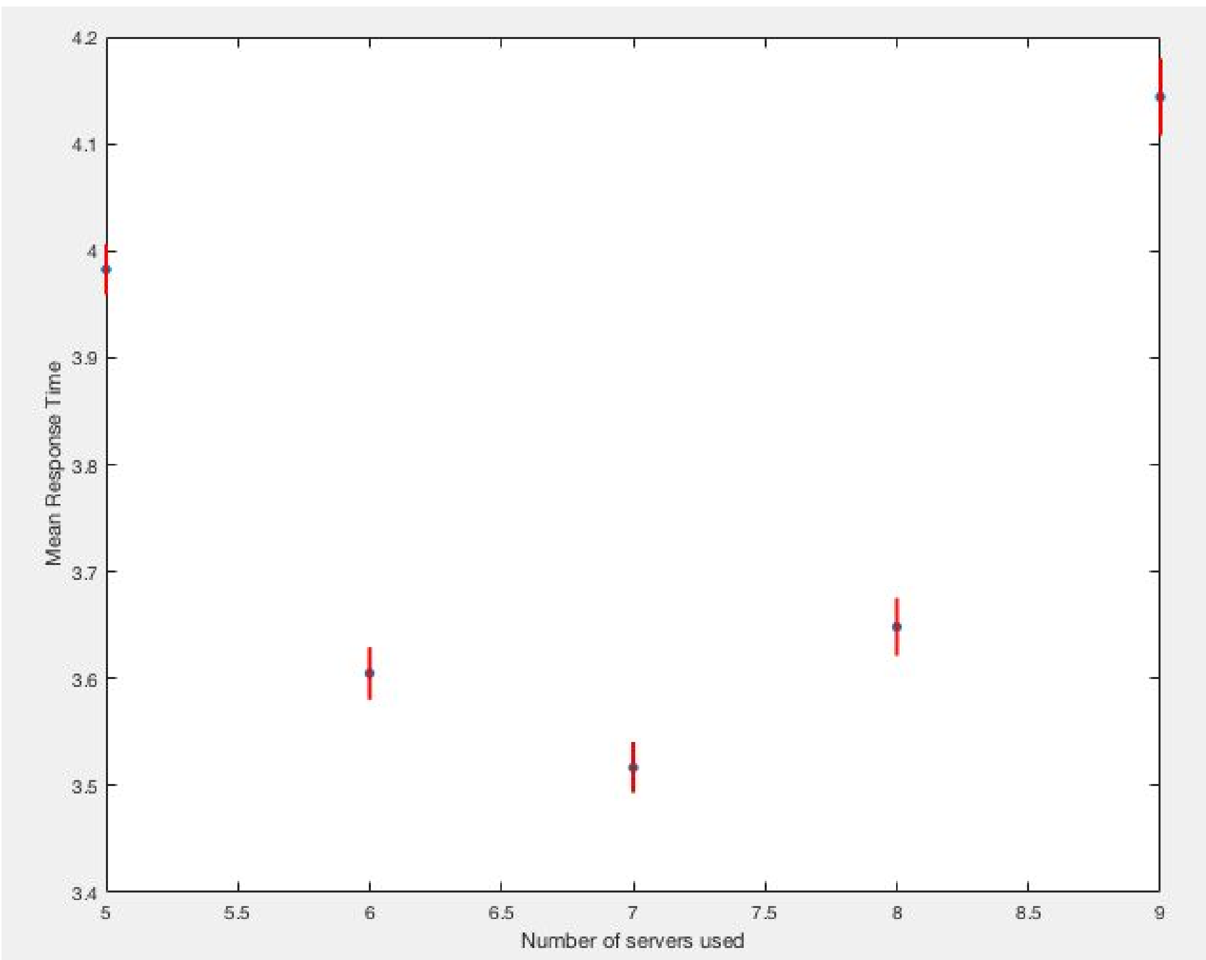
However, **System n = 4** has **Lower** Mean and CI than **System n =5 (**with Mean = 5.10066353, CI=[***5.027224768, 5.174102291***] **)**and the CIs DO NOT Overlap as in the graph, thus we can

conclude that we are confident that 95% of the chance True Mean of **System n=4** lower than any **Systems n=10**.
And thus **System n = 4** is better than **Systems with n=10.**

And thus **System n = 4** is better than **Systems with n=10,1,2,3**(as mentioned before), but worse than **Systems with n=5,6,7,8,9**

While **Systems with n=10** is better than **System with n=1,2,3** but worse than **all other Systems**

Let us compare and look closer to **System with n = 5,6,7,8,9**:



Obviously, **System n=5** with Mean = 3.982547588, CI=[***3.958922657, 4.006172519***]
, whose CI DOES NOT Overlap with with any others' Mean and CI.

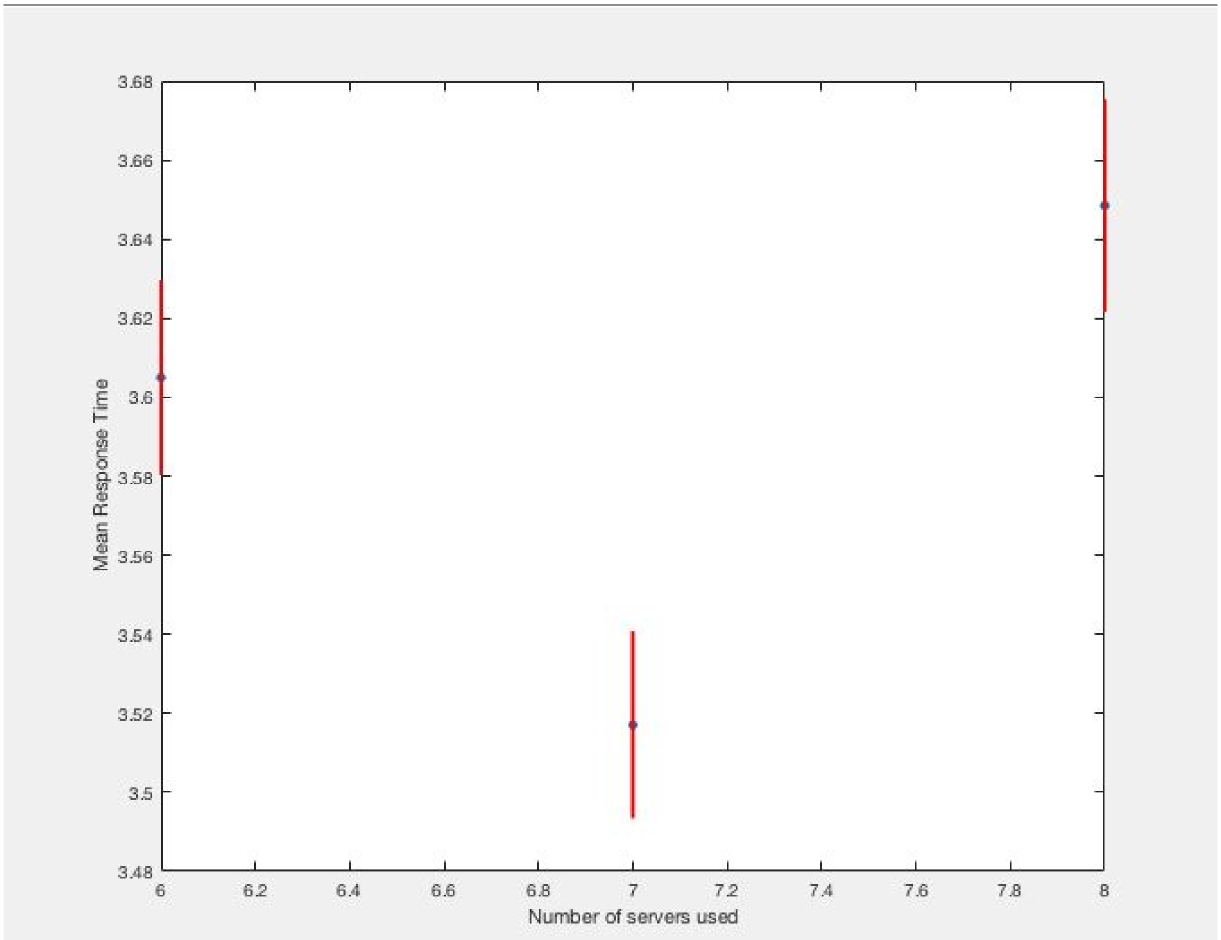
And from the graph, we can observe that the Mean and CI for **System n=5** is much higher than **System with n=6,7,8**, thus we can conclude that we are confident that 95% of the chance True Mean of **System n=5** is **much Greater** than any other **Systems with n=6,7,8**.
And thus **System n = 5** is worse than all other **Systems with n=6,7,8**.

However, **System n = 5** has **Lower** Mean and CI than **System n =9 (**with Mean = 4.14397992, CI=[***4.108030307, 4.179929533***] **)**and the CIs DO NOT Overlap as in the graph, thus we can conclude that we are confident that 95% of the chance True Mean of **System n=5 Lower** than any **Systems n=9**.
And thus **System n = 5** is better than **Systems with n=9.**

<u>And thus **System n = 5** is better than **Systems with n=9,1,2,3,4,10**(as mentioned before), but worse than **Systems with n=6,7,8**</u>

<u>While **Systems with n=9** is better than **System with n=1,2,3,4,10** but worse than **all other Systems**</u>

Let us "ZOOM IN" to **System n =6,7,8**, compare and look closer to **System with n = 5,6,7,8,9**:



The Mean and CI of System n=6,7,8 are respectively:

System n=6 with Mean = 3.604930864, CI=[***3.580211616, 3.629650111***]
System n=7 with Mean = 3.517065771, CI=[***3.493385661, 3.54074588***]
System n=8 with Mean = 3.648465181, CI=[***3.621550781, 3.675379581***]

Obviously, the CI of **System n=6** DOES NOT Overlap with **System n=7**.
And we can observe that the Mean and CI for **System n=6** is Greater than **System n=7**, thus we can conclude that we are confident that 95% of the chance True Mean of **System n=6** is **Greater** than any other **Systems n=7.**
And thus **System n = 6** is worse than **Systems n=7.**
Similarly, the CI of **System n=8** DOES NOT Overlap with **System n=7**.
And we can observe that the Mean and CI for **System n=8** is Greater than **System n=7**, thus we can conclude that we are confident that 95% of the chance True Mean of **System n=68** is **Greater** than any other **Systems n=7**
<u>And thus **System n = 8** is worse than **Systems n=7** and **System= 7** is the best System among System n=6,7,8 and also among all other Systems.</u>

However, we cannot tell whether **System n=6** is better than **System n=8** or vice versa since their Mean of any one is not in the CI of the others but their **CIs OVERLAP**.

System n=6 with Mean = **3.604930864**, CI=[***3.580211616, 3.629650111***]
System n=8 with Mean = **3.648465181**, CI=[***3.621550781, 3.675379581***]

We need a ***t-test*** on **System n=6** and **System n=8** to tell which one perform better.

2. Compare System n =6 and System n =8 with Paired t-test

Here the table of 2 System with 15 independent replications after removing transient:

	n = 6	n = 8	**System n=8 - System n=6**
1st Replication	3.556960943	3.688242527	0.131281584
2nd Replication	3.559113183	3.63867767	0.079564487
3rd Replication	3.565457428	3.704270322	0.138812894
4th Replication	3.54866666	3.68396497	0.13529831
5th Replication	3.574071849	3.724367949	0.1502961
6th Replication	3.615682223	3.629019451	0.013337228
7th Replication	3.583148777	3.600511803	0.017363026
8th Replication	3.710420972	3.562764241	-0.147656731
9th Replication	3.624626804	3.678726609	0.054099805
10th Replication	3.605114283	3.66218665	0.057072367
11th Replication	3.607020998	3.686546031	0.079525033
12th Replication	3.619097553	3.586366323	-0.03273123
13th Replication	3.609719569	3.65311953	0.043399961
14th Replication	3.644990681	3.624951751	-0.02003893
15th Replication	3.649871031	3.603261886	-0.046609145
Sample Mean			***0.04353431727***

Sample Standard Deviation			***0.08264354725***

Let us compare these 2 Systems. The mean response time of **System n=8** minus mean response time of System n=6, over 15 independent replications, are shown on the third column on the table above.

The sample mean and sample standard deviation are, respectively, 0.04353431727 and 0.08264354725.
The **95% confidence interval** is 0.04353431727±t(14,0.0975) * 0.08264354725/sqrt(15) = 0.04353431727± **2.145** * 0.08264354725/sqrt(15) = **[-0.0022, 0.0893]**

The interval **[-0.0022, 0.0893] include a positive and a negative values,** thus we cannot conclude whether System **n=8** is better or worse than **System n=6**.

3. Using common random numbers

Since we cannot conclude which Systems better, let construct a paired t test with Common Random Number Methods.

We use 15 different random number set to do 15 independent replications,
each system process 20000 client requests and cut the transient(3000).***(load the same rand_setting(rand_setting_6_1 to rand_setting_6_15), using common_random_number.m)***

	System n=6	System n=8	**System n=8 - System n=6**
1st Replication	3.556960943	3.640387245	0.08342630232
2nd Replication	3.559113183	3.64221895	0.08310576683
3rd Replication	3.565457428	3.605087126	0.03962969807
4th Replication	3.54866666	3.750588368	0.2019217081
5th Replication	3.574071849	3.626898749	0.05282690056
6th Replication	3.615682223	3.600726714	-0.01495550943
7th Replication	3.583148777	3.750289443	0.1671406657
8th Replication	3.710420972	3.751130889	0.04070991704
9th Replication	3.624626804	3.663339157	0.03871235301

10th Replication	3.605114283	3.66370904	0.05859475752
11th Replication	3.607020998	3.691459937	0.08443893931
12th Replication	3.619097553	3.608136732	-0.0109608209
13th Replication	3.609719569	3.574348341	-0.03537122785
14th Replication	3.644990681	3.711683827	0.06669314617
15th Replication	3.649871031	3.731202582	0.08133155058
Sample Mean			***0.06248294313***
Sample Standard Deviation			***0.06253642799***

Let us compare these 2 Systems. The mean response time of **System n=8** minus mean response time of System n=6, over 15 independent replications, are shown on the third column on the table above.

The sample mean and sample standard deviation are, respectively, ***0.06248294313*** and ***0.06253642799***.
The **95% confidence interval** is ***0.06248294313***±t(14,0.0975) ****0.06253642799***/sqrt(15) = ***0.06248294313***± **2.145** * ***0.06253642799***/sqrt(15) = **[0.0279, 0.0971]**

And finally, The interval **[0.0279, 0.0971]** can be conclude that System **n=8** is worse than **System n=6**.

In other words, we are **confident that there is 95%** chance **System n = 6 is better than System n = 8** .

CONCLUSION

To sum up, at this stage, we can conclude that, we are confident that 95% chance **System= 7** is the **Best System** among all other Systems.
While **System n = 6, System n =8, System n =7 perform quite closed to each other but System n = 6 is slightly better than System n = 8** .

System n=1 is the **worst system**, following System n=2, System n=3, System n=10, System n=4, System n=9 System n=5 are respectively the 2rd worst, the 3rd worst… systems.

In this project, we simulated a queueing system with different n(number of servers used where n<= m=10) and then computed the transient graph as to decide the transients to be removed. Meanwhile, we attempted sound statistics to observe and analyze Mean Response Time of each system(after removed transients) with 15 independent replications with different random set(n=10 * 15 different random sets for each n = 10*15 = **150 random sets**).

We have also computed 95 % confidence interval for each systems and had an visual inspection on the graph produced to learn whether one system is better than others.
At last we had paired t-test on different system but the results generated are not enough to make conclusion and thus we apply common random numbers and we finally obtains a conclusion.