David Bittle

Homework 8
Computer Performance Modeling

1. Suppose an interactive system is supporting 100 users with 15 second think times and a system throughput of 5 interactions/second. a. What is the response time of the system? b. Suppose that the service demands of the workload evolve over time so that system throughput drops to 50% of its former value (i.e., to 2.5 interactions/second). Assuming that there still are 100 users with 15 second think times, what would their response time be? c. How do you account for the fact that response time in (b) is more than twice as large as that in (a)?

A) (100/5) - 15 = 5 seconds.

B) (100/2.5) -15 = 25 seconds

2) You're going to develop a bounded model for the proxy and web server system. You should follow Algorithm 5.1 in the book "Quantitative System Performance" (PDF on the Moodle).

1. *Write down the visit count for the Proxy, Router, Web server A and B; these will be as a function of the probability of a hit in the proxy. Write this down in your PDF writeup. You need to properly account for the fact that you either use either web server A or B, but not both, averaged.*

Vp = 1 + (1*.3) = 1.3

Vr = .6 = .3*2

Va = .15 = .3 * .5

Vb = .15 = .3 * .5

2. *Write down the Demand (D_i) for each service center - this is based on the visits V_i and the service time S_i. The "delay center" for the transmission time should be included in the total demand D, but is not eligible for the maximum delay center D_{max} because there is no queuing for the transmission time, so it's not the bottleneck.*

print "proxy " + str(Vp * sProxy)

print "router " + str(Vr * sRouter)

print "A " + str(Va * sA)

print "B " + str(Vb * sB)

proxy 13.0

router 3.0

A 22.5

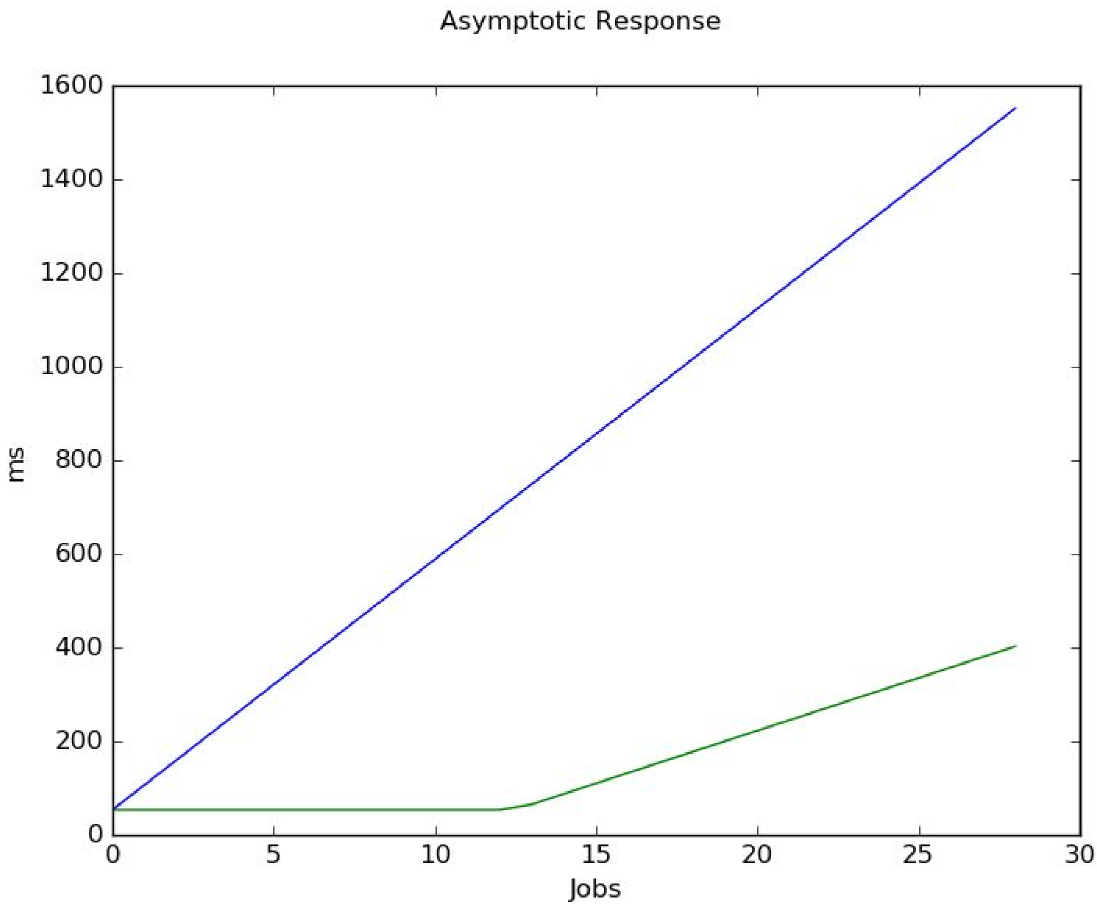
B 15.0

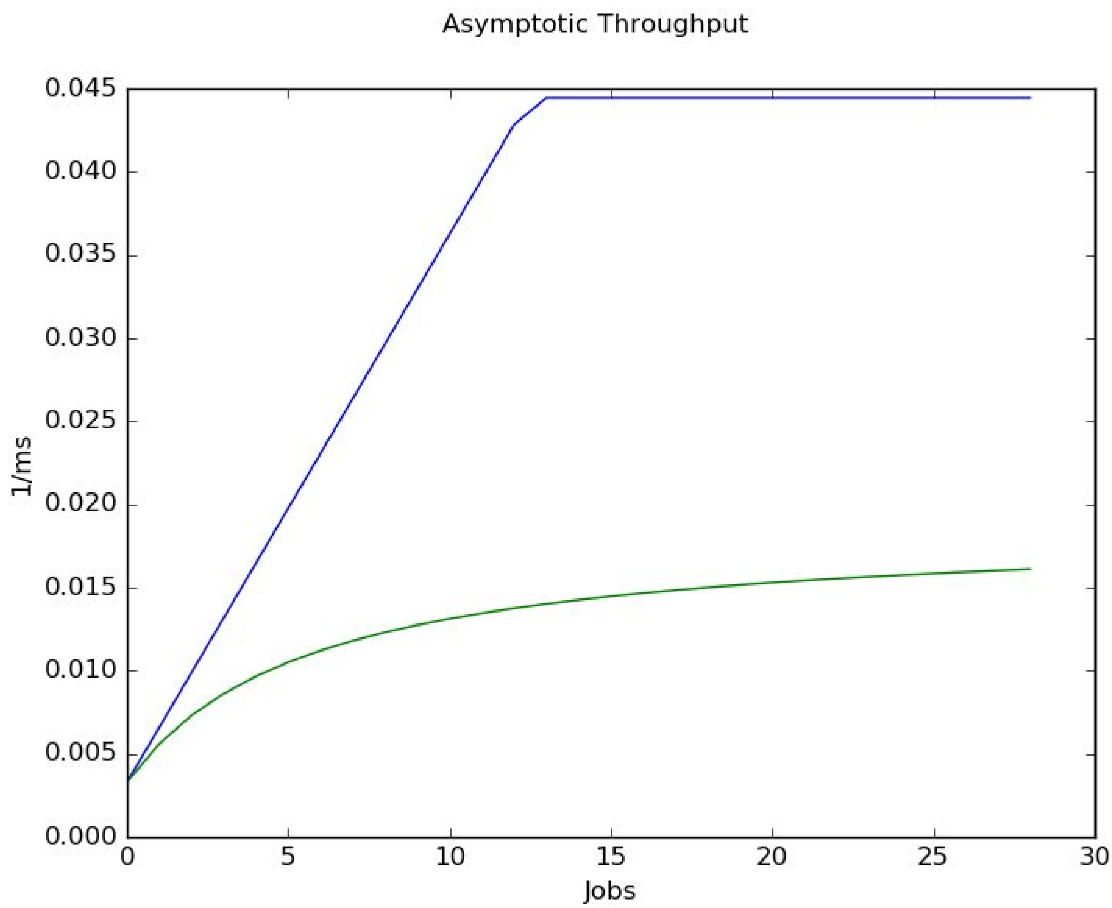
3. *Examine case study 5.3.1 for the process of calculating the job bounds, and calculate the upper bounds on throughput (X) and the lower bounds on response time (R). Write the bounds down and also graph those bounds (two lines for each).*

```
def AsymptoticlowboundThroughput(N,Dmax,D):

    low = N/(N*D+Z)

    return low


def AsymptoticlowboundResponse(N,Dmax,D):

    return max(N*Dmax-Z,D)


def AsymptoticupboundThroughput(N,Dmax,D):

    #1/Dmax

    up = N/(D+Z)

    return min(1/Dmax,up)


def AsymptoticupboundResponse(N,Dmax,D):

    return N*D
```

0.031 ms lower bound throughput

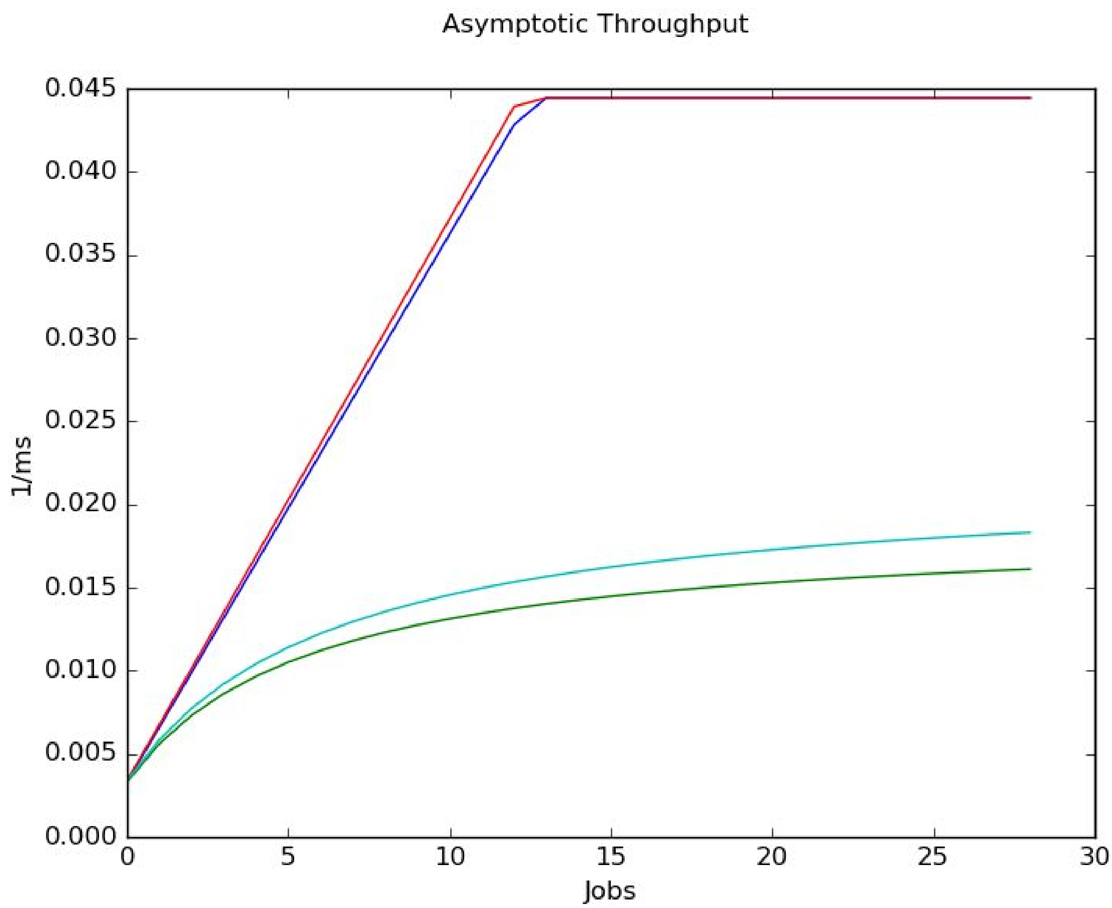
400 ms lower bound response time

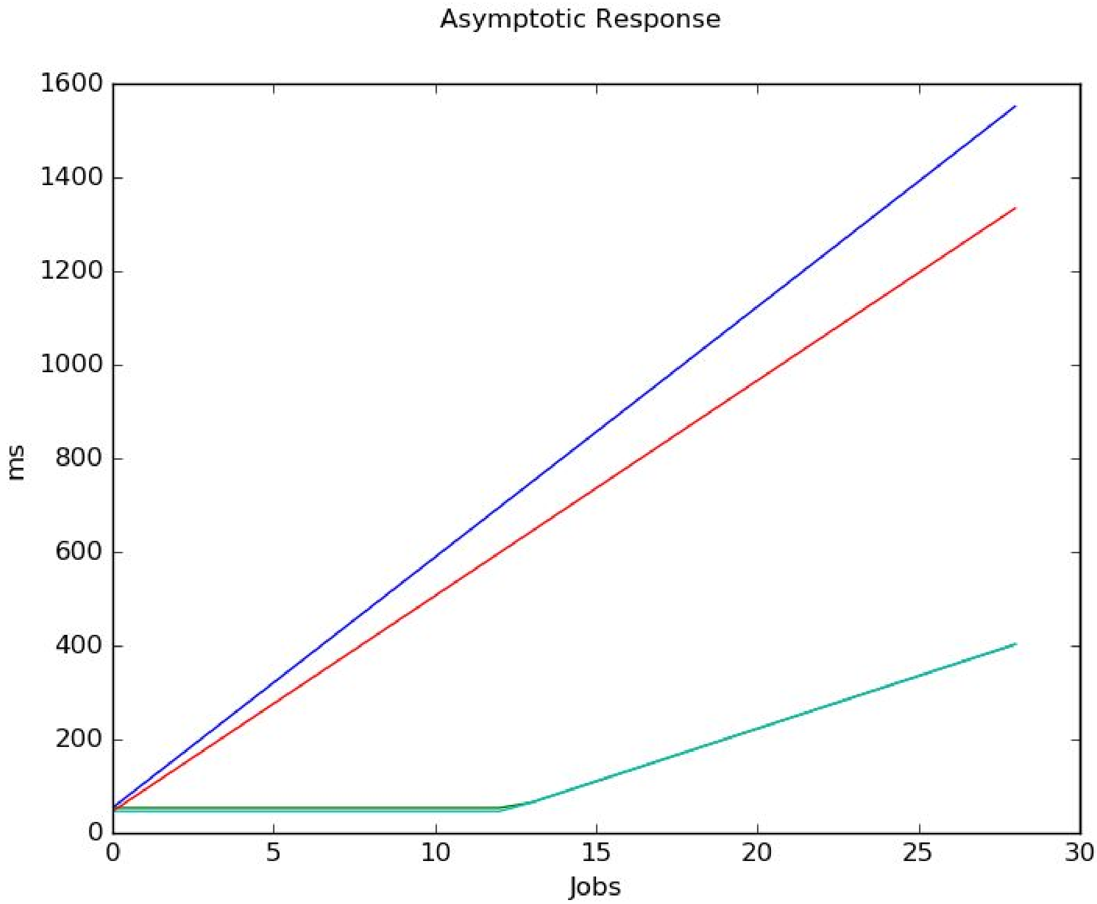




4. *Modify the model in a method similar to 5.3.3 of the QSP text. In your case, assume that the two origin servers are of equal performance. Write down the bounds equations and graph them again, overlaying them with the original bounds from part 3.*

I used the same functions as in the prior problem but changed the service time on A to be the same b.

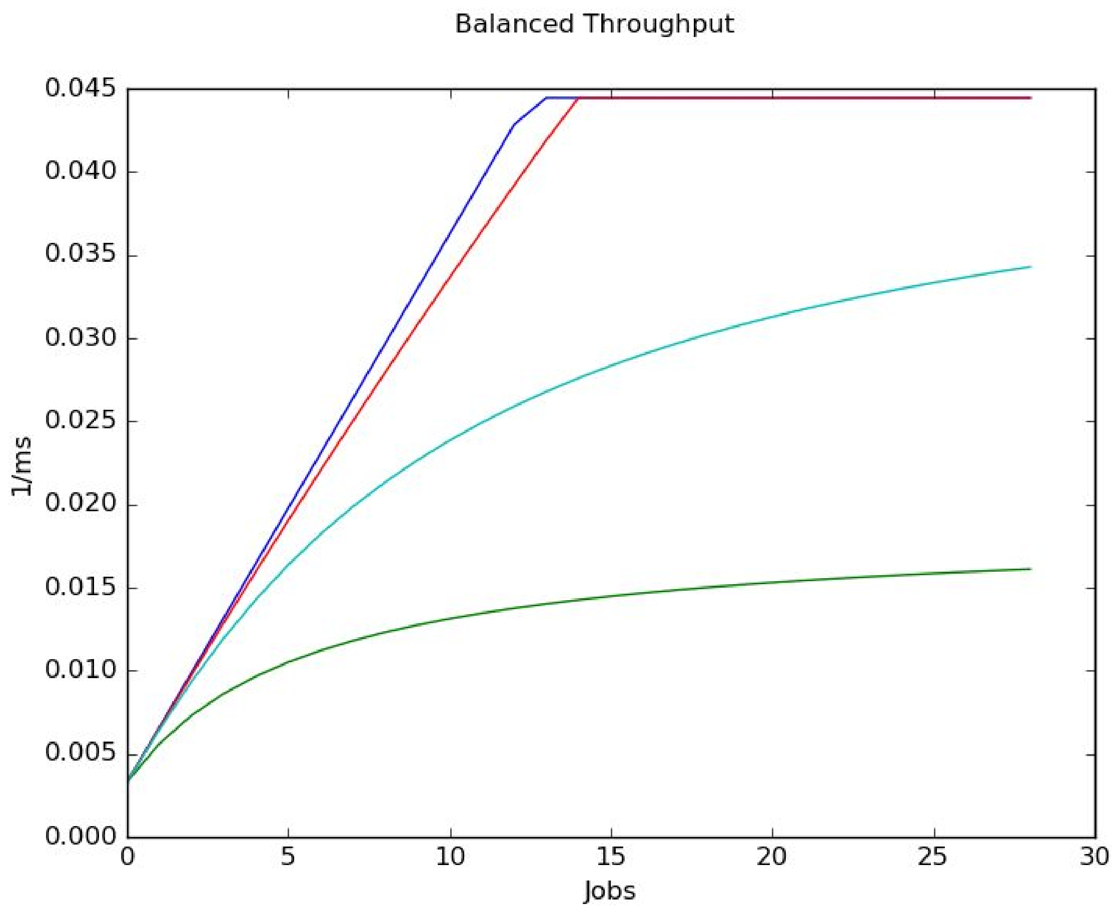

Asymptotic Throughput
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
1/ms
0
5
10
15
20
25
30
Jobs



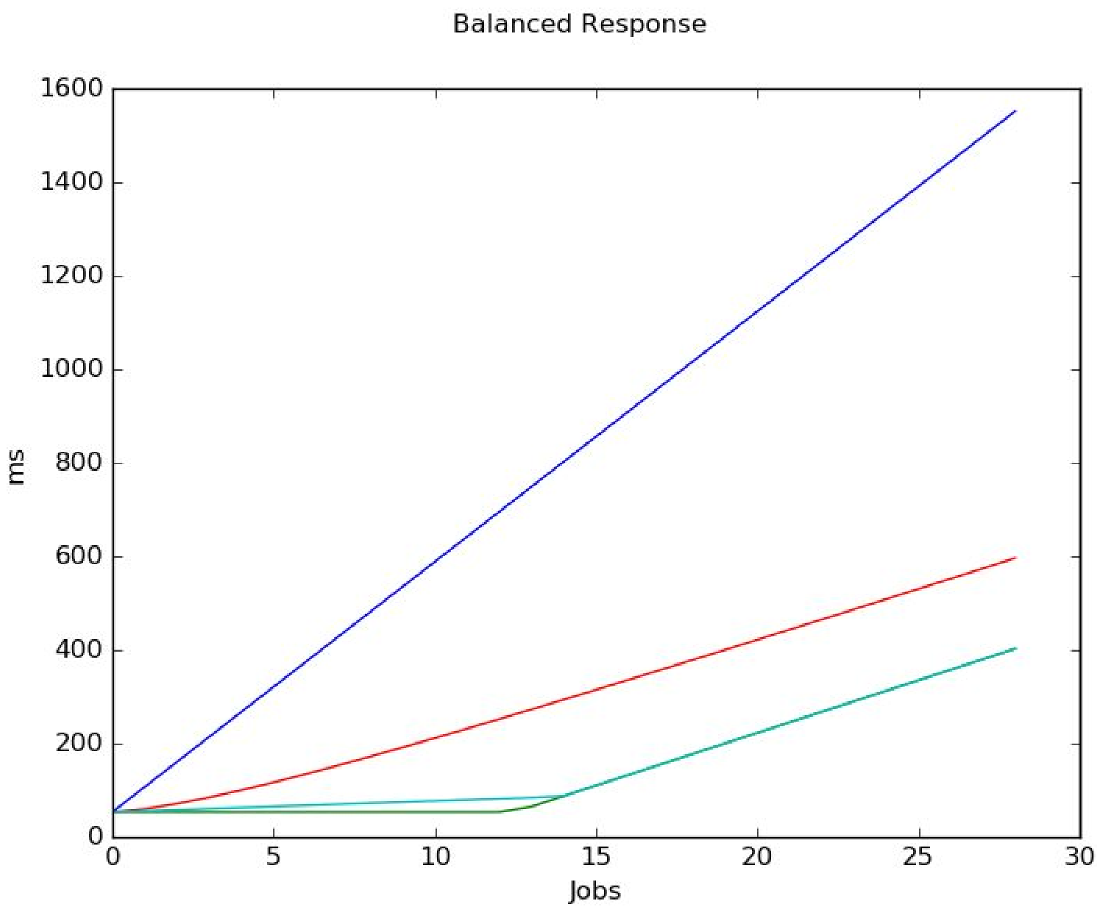
5. *Now, determine the lower bounds on throughput (X) and the upper bounds on response time (R) for the original system using the balanced job bounds equations summarized in Table 5.2. Produce a single graph with the upper and lower bounds of X and R.*

```
def balancedlowboundThroughput(N,Dmax,D):

        low = N/(D+Z+((N-1)*Dmax)/(1+Z/(N*D)))

        return low


def balancedlowboundResponse(N,Dmax,D):

        low = D + ((N-1)*Dave)/(1+Z/(D))
```

```
    return max(N*Dmax-Z,low)


def balancedupboundThroughput(N,Dmax,D):
    #1/Dmax
    up = N/(D+Z+((N-1)*Dave)/(1+Z/(D)))
    return min(1/Dmax,up)


def balancedupboundResponse(N,Dmax,D):
    low = D+ ((N-1)*Dmax)/(1+Z/(N*D))
    return low
```



Balanced Throughput
1/ms
Jobs
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25
30



3.

1. You will use the V_i and D_i from the prior problem.
2. Write down the equations for the Response for the individual components for N
3. customers in the system, using the appropriate service times (again, in the PDF document) and the Queue length for N-1 customers in the system

Rproxy = Dproxy * (1+Qproxy(N-1))
Qproxy(N) = X(N) * Rproxy(N)

Rrouter = Drouter * (1+Qrouter(N-1))
Qrouter(N) = X(N) * Rrouter(N)

Ra = Da * (1+Qa(N-1))
Qa(N) = X(N) * Ra(N)

Rb = Db * (1+Qb(N-1))
Qb(N) = X(N) * Rb(N)

Rwan = Dwan * (1+Qwan(N-1))
Qwan(N) = X(N) * Rwan(N)

4. Write down the equation for the System Response, which combines the visit ratios and the individual response times

systemresponsetime= N/X-Z

averagenumberinsystem= N - XZ

5. Write down the equation for the Throughput.

$$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$

6. Write down the equations for the Q (number in queue) for N customers in the system

$$Q_k(N) = X(N) R_k(N)$$

7. Now, wrap all that in into a lovely Excel spreadsheet and drag out the computation until at least N=30; cut and paste or do whatever it is that gets it into your single PDF document that is easy to grade. The example in Table 6.2 of the QSP book shows the structure to follow (although that example is "sideways", or use the PPT example from class.

	Response					Queue					
Jobs	Proxy	Router	A	B	Wan	Proxy	Router	A	B	Wan	X(N)
0						0	0	0	0	0	
1	13	3	22.5	15	21.6	0.039988	0.009228	0.069209	0.04614	0.066441	0.003076
2	13.51984	3.027684	24.05721	15.69209	21.6	0.082464	0.018467	0.146736	0.095714	0.131749	0.006099
3	14.07203	3.055402	25.80157	16.4357	21.6	0.127555	0.027695	0.233876	0.14898	0.195791	0.009064
4	14.65821	3.083086	27.76221	17.2347	21.6	0.17537	0.036886	0.332145	0.206195	0.258421	0.011964
5	15.27981	3.110658	29.97327	18.09292	21.6	0.225995	0.046008	0.443317	0.267602	0.319473	0.01479
6	15.93793	3.138024	32.47464	19.01403	21.6	0.279478	0.055027	0.569456	0.333419	0.378765	0.017535
7	16.63322	3.16508	35.31277	20.00128	21.6	0.335819	0.063902	0.712952	0.403819	0.436096	0.02019
8	17.36564	3.191706	38.54143	21.05728	21.6	0.394947	0.072589	0.876549	0.478907	0.49125	0.022743
9	18.13432	3.217767	42.22235	22.1836	21.6	0.45671	0.081039	1.063363	0.55869	0.543992	0.025185
10	18.93723	3.243117	46.42566	23.38035	21.6	0.520845	0.089198	1.276881	0.643048	0.594082	0.027504
11	19.77099	3.267594	51.22983	24.64572	21.6	0.586971	0.09701	1.520936	0.731694	0.641271	0.029688
12	20.63062	3.29103	56.72105	25.97541	21.6	0.654563	0.104417	1.79963	0.824141	0.685319	0.031728
13	21.50931	3.313251	62.99167	27.36211	21.6	0.722953	0.111362	2.117223	0.919672	0.726001	0.033611
14	22.39839	3.334087	70.13752	28.79508	21.6	0.791332	0.117793	2.47795	1.017327	0.763126	0.03533

15	23.28732	3.353379	78.25389	30.2599	21.6	0.858773	0.123663	2.885791	1.115903	0.796549	0.036877
16	24.16405	3.37099	87.43029	31.73855	21.6	0.924268	0.128939	3.344183	1.21399	0.826194	0.03825
17	25.01548	3.386818	97.74411	33.20985	21.6	0.98679	0.1336	3.855727	1.310034	0.852059	0.039447
18	25.82827	3.400801	109.2539	34.65051	21.6	1.045365	0.137643	4.421906	1.402434	0.874232	0.040474
19	26.58975	3.412929	121.9929	36.03651	21.6	1.099151	0.141082	5.04287	1.489656	0.892888	0.041337
20	27.28897	3.423245	135.9646	37.34484	21.6	1.147507	0.143948	5.717342	1.570359	0.908285	0.04205
21	27.9176	3.431845	151.1402	38.55539	21.6	1.190045	0.146289	6.44266	1.643502	0.920744	0.042627
22	28.47058	3.438868	167.4598	39.65253	21.6	1.226647	0.148163	7.214961	1.708418	0.93063	0.043085
23	28.94641	3.444488	184.8366	40.62628	21.6	1.257461	0.149632	8.029487	1.764846	0.938326	0.043441
24	29.34699	3.448896	203.1635	41.47269	21.6	1.282854	0.150763	8.880944	1.812908	0.944207	0.043713
25	29.6771	3.452288	222.3212	42.19362	21.6	1.303355	0.151617	9.763878	1.853054	0.948626	0.043918
26	29.94362	3.454851	242.1872	42.79581	21.6	1.319591	0.152252	10.67299	1.885976	0.951894	0.044069
27	30.15468	3.456757	262.6423	43.28964	21.6	1.332218	0.152718	11.6034	1.912514	0.954277	0.044179
28	30.31883	3.458153	283.5766	43.68771	21.6	1.341878	0.153054	12.55078	1.93357	0.955992	0.044259
29	30.44441	3.459162	304.8926	44.00354	21.6	1.349157	0.153294	13.51145	1.950036	0.957213	0.044315
30	30.53904	3.459883	326.5075	44.25053	21.6	1.354568	0.153464	14.48233	1.962745	0.958074	0.044355

8. Now, repeat the analysis for a system where the origin servers have balanced service times. If you structure your spreadsheet correctly, this should be just changing a single cell that defines the service time of the web servers.

	Response					Queue					
Jobs	Proxy	Router	A	B	Wan	Proxy	Router	A	B	Wan	X(N)
0						0	0	0	0	0	
1	13	3	22.5	15	21.6	0.039988	0.009228	0.069209	0.04614	0.066441	0.003076
2	13.51984	3.027684	24.05721	15.69209	21.6	0.082464	0.018467	0.146736	0.095714	0.131749	0.006099
3	14.07203	3.055402	25.80157	16.4357	21.6	0.127555	0.027695	0.233876	0.14898	0.195791	0.009064
4	14.65821	3.083086	27.76221	17.2347	21.6	0.17537	0.036886	0.332145	0.206195	0.258421	0.011964
5	15.27981	3.110658	29.97327	18.09292	21.6	0.225995	0.046008	0.443317	0.267602	0.319473	0.01479
6	15.93793	3.138024	32.47464	19.01403	21.6	0.279478	0.055027	0.569456	0.333419	0.378765	0.017535
7	16.63322	3.16508	35.31277	20.00128	21.6	0.335819	0.063902	0.712952	0.403819	0.436096	0.02019
8	17.36564	3.191706	38.54143	21.05728	21.6	0.394947	0.072589	0.876549	0.478907	0.49125	0.022743
9	18.13432	3.217767	42.22235	22.1836	21.6	0.45671	0.081039	1.063363	0.55869	0.543992	0.025185
10	18.93723	3.243117	46.42566	23.38035	21.6	0.520845	0.089198	1.276881	0.643048	0.594082	0.027504
11	19.77099	3.267594	51.22983	24.64572	21.6	0.586971	0.09701	1.520936	0.731694	0.641271	0.029688
12	20.63062	3.29103	56.72105	25.97541	21.6	0.654563	0.104417	1.79963	0.824141	0.685319	0.031728
13	21.50931	3.313251	62.99167	27.36211	21.6	0.722953	0.111362	2.117223	0.919672	0.726001	0.033611
14	22.39839	3.334087	70.13752	28.79508	21.6	0.791332	0.117793	2.47795	1.017327	0.763126	0.03533
15	23.28732	3.353379	78.25389	30.2599	21.6	0.858773	0.123663	2.885791	1.115903	0.796549	0.036877
16	24.164	3.3709	87.430	31.738	21.6	0.9242	0.1289	3.3441	1.2139	0.8261	0.0382

	05	9	29	55		68	39	83	9	94	5
17	25.01548	3.386818	97.74411	33.20985	21.6	0.98679	0.1336	3.855727	1.310034	0.852059	0.039447
18	25.82827	3.400801	109.2539	34.65051	21.6	1.045365	0.137643	4.421906	1.402434	0.874232	0.040474
19	26.58975	3.412929	121.9929	36.03651	21.6	1.099151	0.141082	5.04287	1.489656	0.892888	0.041337
20	27.28897	3.423245	135.9646	37.34484	21.6	1.147507	0.143948	5.717342	1.570359	0.908285	0.04205
21	27.9176	3.431845	151.1402	38.55539	21.6	1.190045	0.146289	6.44266	1.643502	0.920744	0.042627
22	28.47058	3.438868	167.4598	39.65253	21.6	1.226647	0.148163	7.214961	1.708418	0.93063	0.043085
23	28.94641	3.444488	184.8366	40.62628	21.6	1.257461	0.149632	8.029487	1.764846	0.938326	0.043441
24	29.34699	3.448896	203.1635	41.47269	21.6	1.282854	0.150763	8.880944	1.812908	0.944207	0.043713
25	29.6771	3.452288	222.3212	42.19362	21.6	1.303355	0.151617	9.763878	1.853054	0.948626	0.043918
26	29.94362	3.454851	242.1872	42.79581	21.6	1.319591	0.152252	10.67299	1.885976	0.951894	0.044069
27	30.15468	3.456757	262.6423	43.28964	21.6	1.332218	0.152718	11.6034	1.912514	0.954277	0.044179
28	30.31883	3.458153	283.5766	43.68771	21.6	1.341878	0.153054	12.55078	1.93357	0.955992	0.044259
29	30.44441	3.459162	304.8926	44.00354	21.6	1.349157	0.153294	13.51145	1.950036	0.957213	0.044315
30	30.53904	3.459883	326.5075	44.25053	21.6	1.354568	0.153464	14.48233	1.962745	0.958074	0.044355

9. Compare the results from the simulation model, MVA model and the balanced job bounds. Do the bounds correctly bound the MVA model? How different are the MVA results than the simulation?

The bounds look to have properly bound the MVA model. The MVA results are different than the simulation but not excessively so.