Carleton UNIVERSITY

Winter, 2022

SYSC 5101W: Design of High-Performance Software

Project Template 1

Submitted by: Aziz Al-Najjar, ID: 101244840

Muhammad Bin Salman, ID: 100995527

Contents

List of Figures

List of Tables

Project Objective:

In the first template of the project, we were provided with a UML overview of a pre-defined software program, along with a thorough description of the various tasks and execution cycles that the program undertakes. We were also given the different components that would be used in this program, such as the databases, disks, processors, etc., along with their service demand times and mean average of calls to each other.

In this project, we will use the LQN software to build a Layered Queuing Network to investigate the model's behavior, identify the model's Bottleneck, saturation points, and apply sensitivity tests. Afterward, we will try different techniques and utilize multiple changes to the LQN model to alleviate the model's Bottleneck. Lastly, we will study what Software Performance Engineering Principles could be applied to our model.

Model Description:

The provided model is that of a basic webserver in which the user has the option of requesting three different types of requests. The user's request will first enter a protocol stack in which it analyzes the request and determines the request type.

The main model **reduced** Execution Graph: Analyze and Process Request expansion:

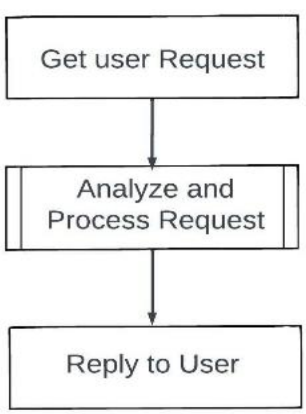


Figure 2 Main model reduced execution Graph

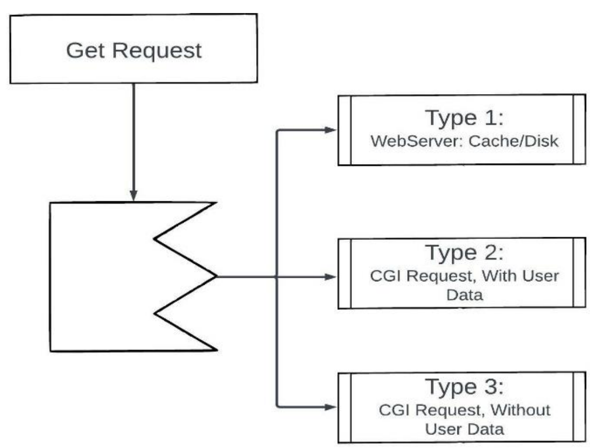


Figure 1 Analyze and Process Request expansion

The first type is a basic static page request. Here, the model will first check the webserver's cache memory and send the page to the user if present. Otherwise, there is a 30% probability that the page is not available at the cache and must be retrieved from the web server's disk. In this case, an average of 3 disk block read operations from WSDisk will occur, and a reply to the user with the requested page will be sent. This request type has the highest probability of 90%

Type 1 expansion:

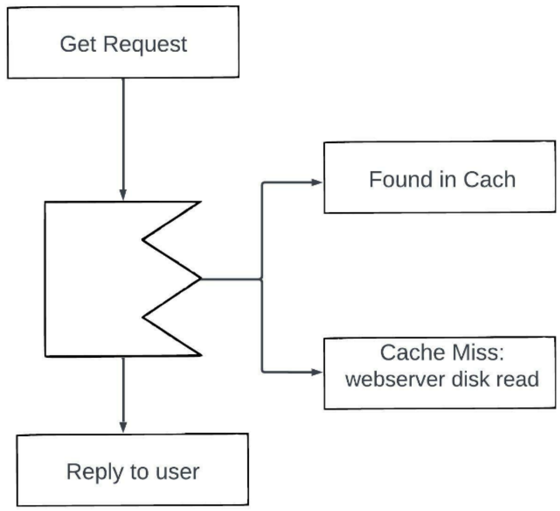


Figure 3 Type 1 expansion

Table 1. Type 1 request details:

Operation / Software requirement	**Task/ Entry**	**Computer requirements**	**Service/ Host demand (ms)**
Retrieve page from cache (63%)	Webserver/ Cache	WSP	1
Retrieve Page from Disk (27%)	Webserver/ Disk	WSP	1
Static web page retrieve	3 * WSDisk/ read	WSDisk/ WSP	3*(0.1+10) = 30.3
		Total	32.3

The second user request type is a CGI request with user data. This request type has three primary operations. Processing the user data, storing it in the database, and determining the next page to send. In other words, it will apply 1.7 database update operations. Each database update operation requires a 2 disk reads and 4 disk writes on Database Disk and send a reply to the user. This request type has a probability of 60%*10% = 6%

Type 2 expansion:

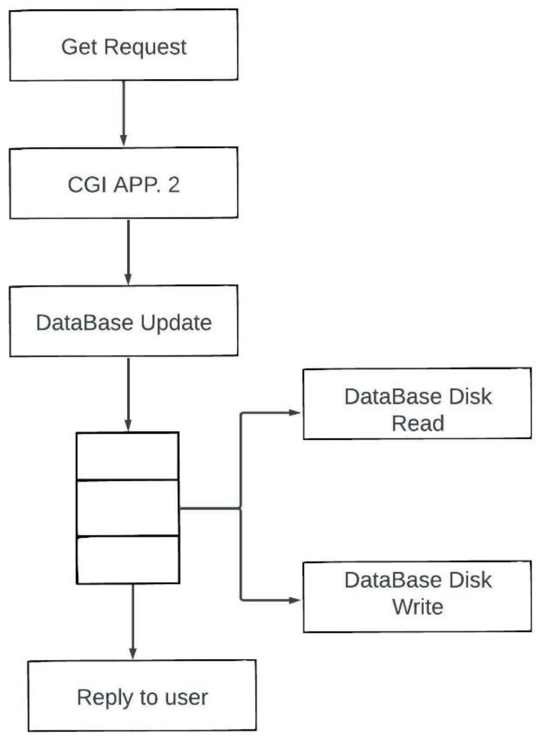


Figure 4 Type 2 expantion

Table 2 Type 2 request details:

Operation / Software requirement	**Task/ Entry**	**Computer requirements**	**Service/ Host demand (ms)**
Process user data	CGI/ Type 2	WSP	15
Database update	1.7* Database/ update	DBP	1.7*30 = 51
DBDisk read	2* DBDisk/ read	DBDisk/ DBP	2*(0.1+10) = 20.2
DBDisk write	4* DBDisk/ write	DBDisk/ DBP	4*(0.1+10) = 40.4
		Total	126.6

The third type is a CGI request without user data. It will start by constructing a dynamic page in which it will apply a 4.3 database read operations. Each database read operation requires 4 disk read operations on Database Disk. Afterward, it will send a static page with 8 embedded graphic objects. Lastly, it will apply protocol operations. The third type has a probability of 40%*10% = 4%.

Type 3 expansion:

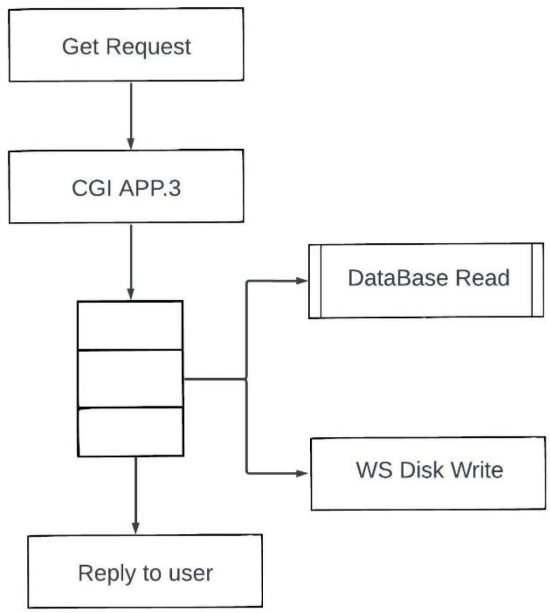


Figure 5 Type 3 expansion

Table 3 Type 3 request details:

Operation / Software requirement	**Task/ Entry**	**Computer requirements**	**Service/ Host demand (ms)**
Construct dynamic page	CGI/ Type 3	WSP	5
Database read	4.3*Database/ read	DBP	4.3*20 = 86
DBDisk read	4* DBDisk/ read	DBDisk/ DBP	4*(0.1+10) = 40.4
Send static page	CGI/ Type 3	WSP	0.8
Retrieve and send embedded object	8*CGI/ Type 3	WSP	8*1=8
Average operations on WSDisk	8*0.2 * WSDisk/ write	WSDisk/ WSP	8*0.2*10.1 = 16.16
Protocol operations	CGI/ Type 3	WSP	0.25*(16+3*8) = 10
		Total	166.36

Layered Queuing Network Implementation:

Base model code will be provided in the appendix.

Base Model Results:

In the base model of our code, saturation first occurs after 300 users, with the Protocol Stack being the initial Bottleneck. The graphs for the throughput, response time, and device utilization can be seen below. Please refer to Appendix to view the tabular results for response time, throughput, and device utilizations for the base model (thread=1).

The LQN base model can be seen below, with the Bottleneck at Protocol Stack in red:

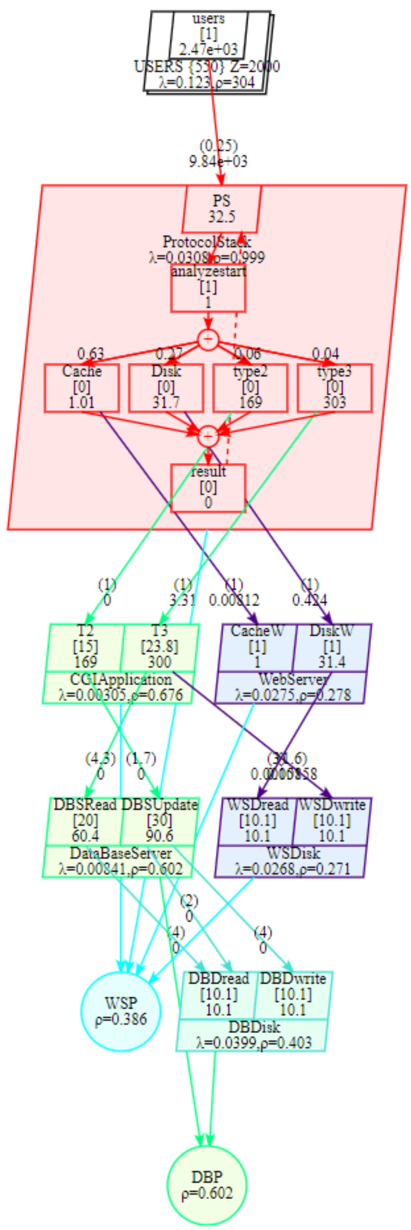


Figure 6 LQN Base Model

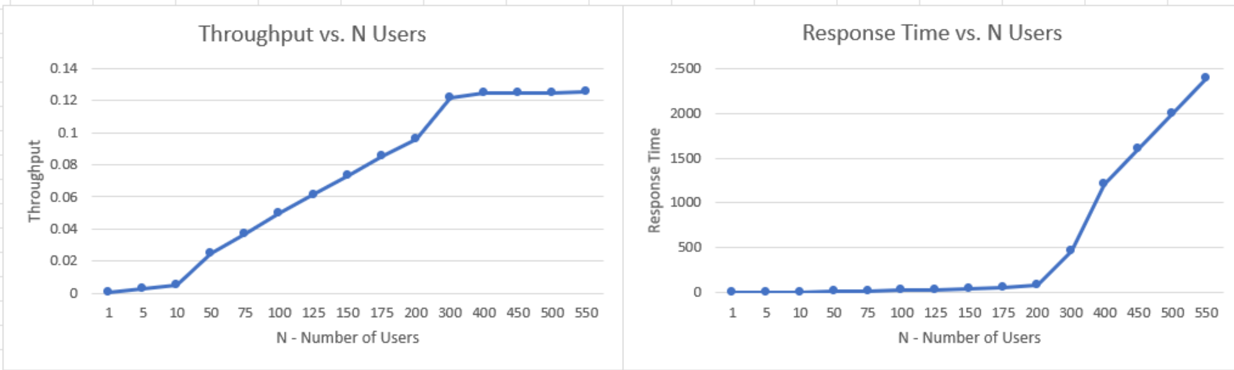


Figure 7 Base Model Throughput and Response Time vs N Users graph

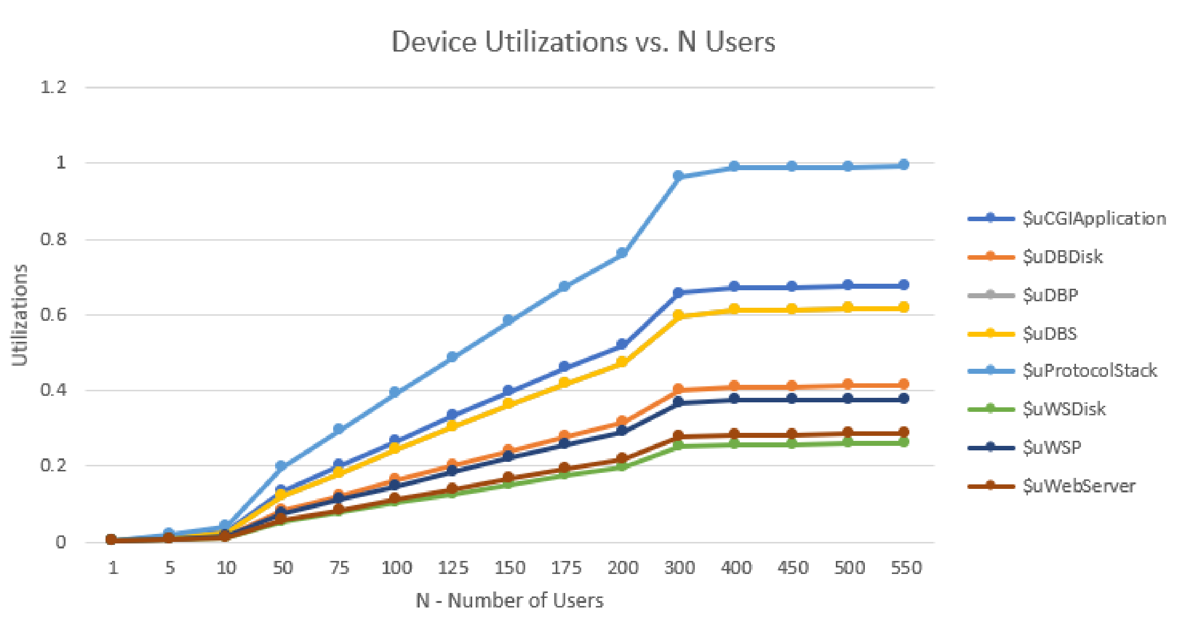


Figure 8 Base model Device Utilization vs. N Users

We can see from the LQN above and the device utilization graph that Protocol Stack is our bottleneck. This understandable as the system is designed to analyze all request in the protocol stack activity and again all replies will go through PS to the User. In next sections we will provide multiple solutions to problem.

Sensitivity tests:

We will try different Probability of Cache Miss (PCM) values as a sensitivity test and report how the performance metrics change.

For PCM = 0.5:

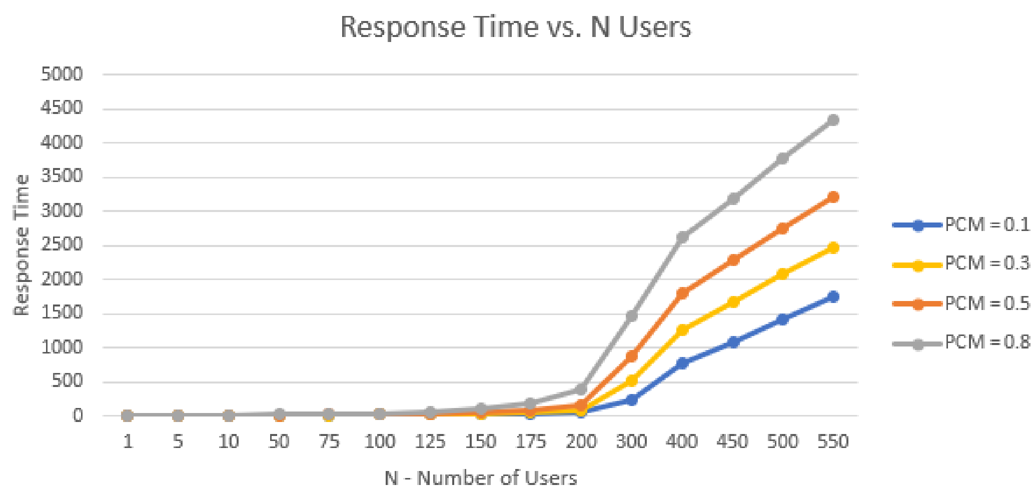


Figure 9 Different PCM Model Response Time vs N Users graph

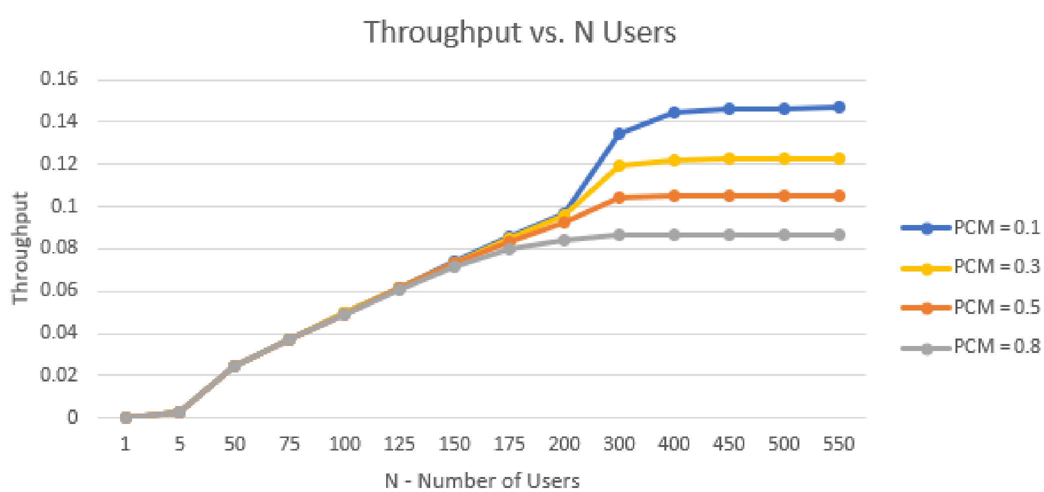


Figure 10 Different PCM Model Throughput vs N Users graph

As we can see from graphs above, as a result of increasing cache miss probability to 0.5 and 0.8, the response time increased and throughput saturation point decreased. This is expected as going through the disk and retrieving data blocks will require more resources and time. On the other, decreasing PCM to 0.1 gave higher throughput saturation and faster response time.

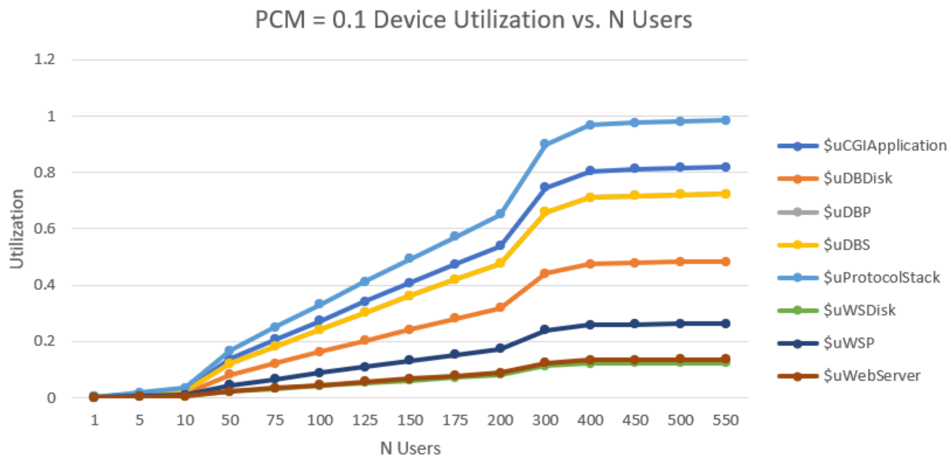


Figure 11 PCM = 0.1 Device Utilization vs. N Users

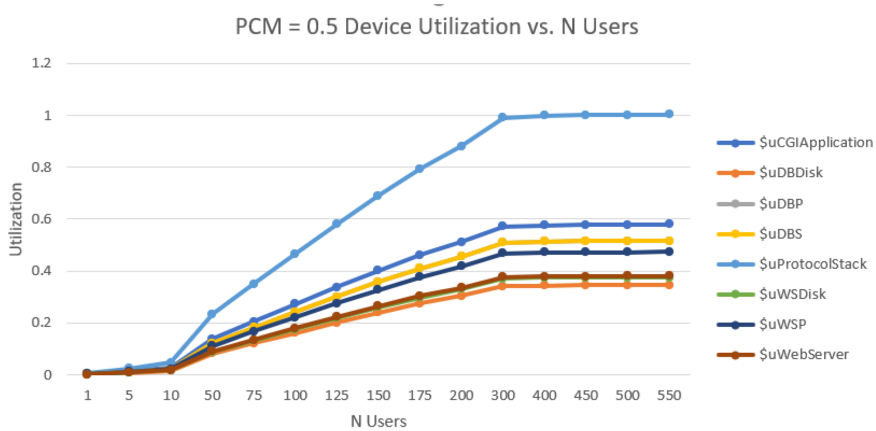


Figure 12 PCM = 0.5 Device Utilization vs. N Users

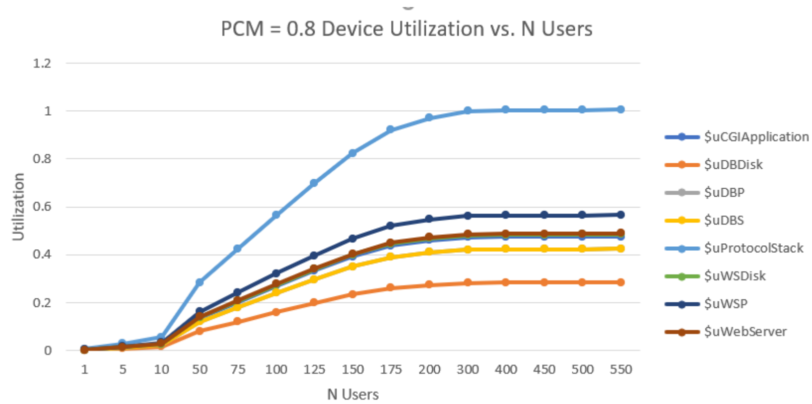


Figure 13 PCM = 0.8 Device Utilization vs. N Users

Similarly for devices utilization. Higher PCM results the devices to reach saturation with less number of users. Hence, we can see that getting more capacity for cache memory will directly help the overall performance of the system.

LQN Bottleneck Removal:

When we initially analyze the base model of our LQN, we realize the Bottleneck can be seen at the Protocol Stack activity. We begin by multi-threading with an array of various thread values: [1, 5, 8, 10]. A brief overview of the multi-threading sequence can be seen on the next page. By the end of the multi-threading sequence, the Bottleneck reaches the Database processor, while the other tasks, such as the Protocol Stack, are still very highly saturated as a result of the Bottleneck at the Database processor. The graphs for the throughput and response time across the various threads can be seen below. As it can be seen, the response time drastically decreases after moving from 1 thread to 5 threads, and then slightly decreases after increasing the threads from there on out. The same applies for the throughput, where we see a decent increase after adding onto the single thread, but the improvement saturates quickly between 5-10 threads. The tabular results at the different threads can be seen in Table 1 in Appendix A.

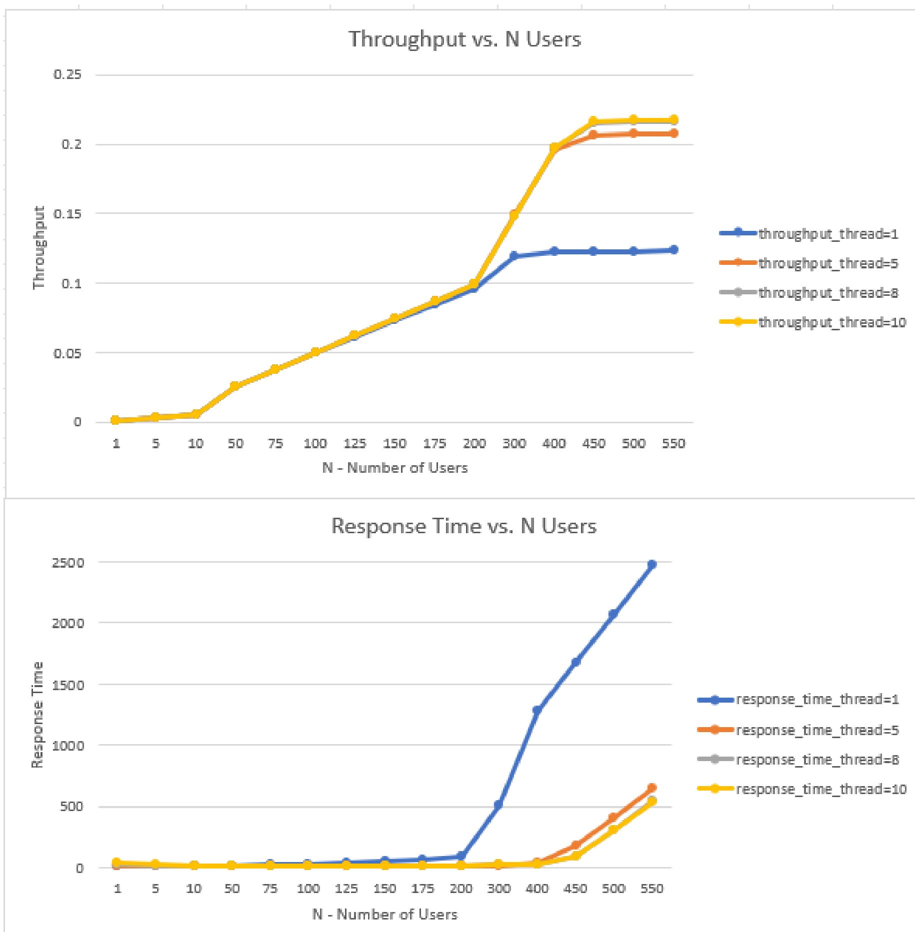
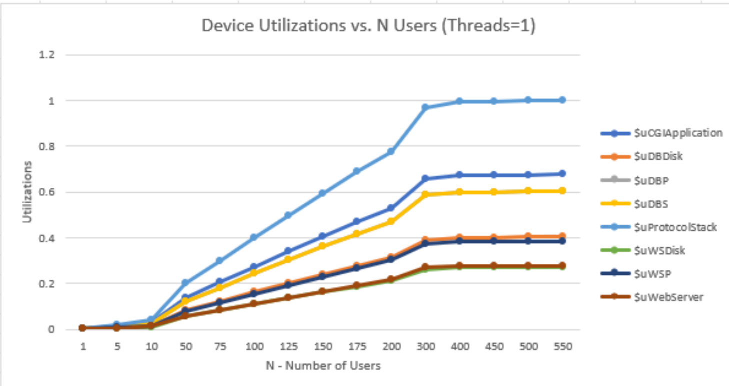
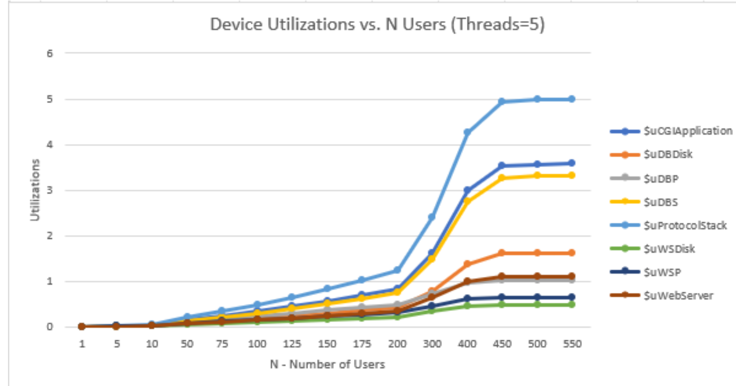


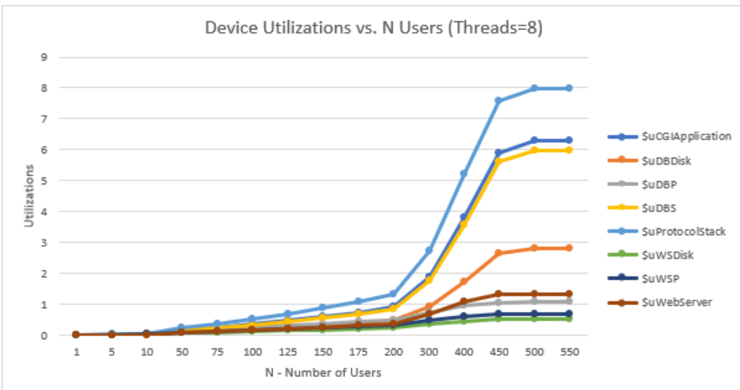
Figure 14 Throughput and Response Time Comparison after Multi-Threading



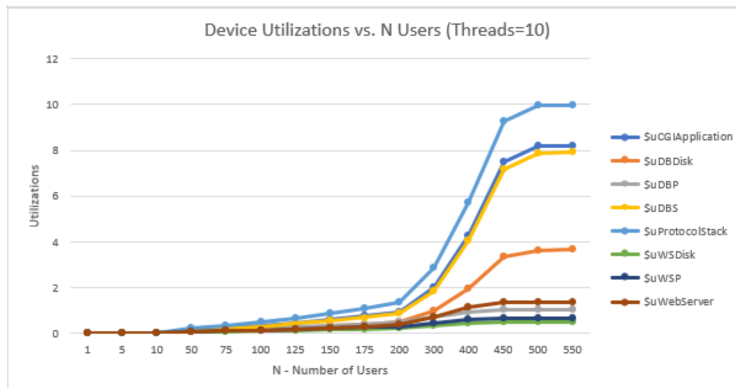
a): Device Utilizations (Threads =1)



b): Device Utilizations (Threads =5)

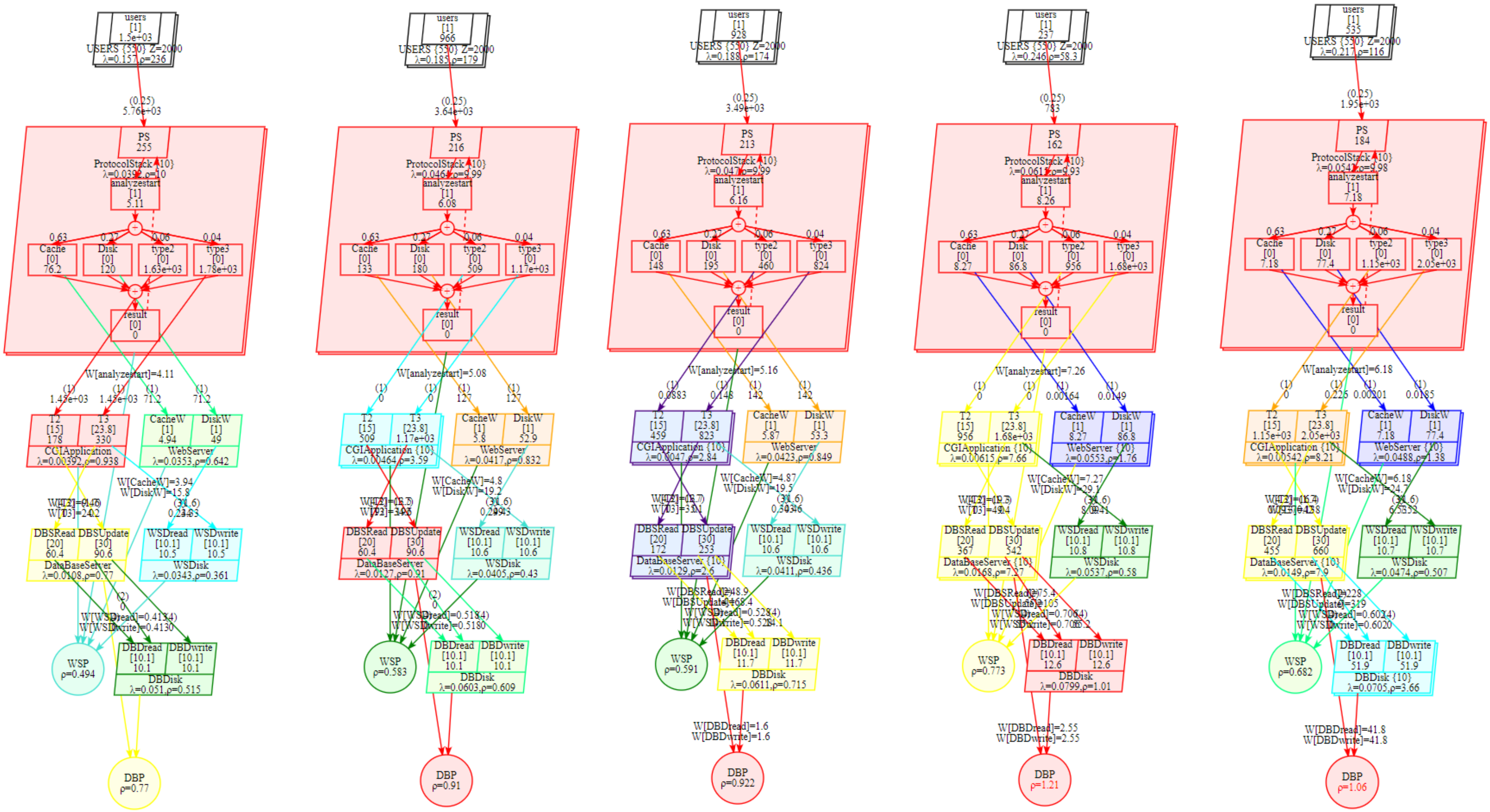


c): Device Utilizations (Threads =8)



d): Device Utilizations (Threads =10)

Figure 15 Multi-Threading Device Utilization



a): Multi-threading Protocol Stack to move bottleneck to CGI Application

b) Multi-threading CGI Application to move bottleneck to Database Server

c) Multi-threading Database server to move bottleneck to Web Server, based off database processor

d) Multi-threading webserver server to move bottleneck to Database disk, based off database processor

e) Multi-threading database disk, Database processor is now the bottleneck

Figure 16 multi-threading tasks

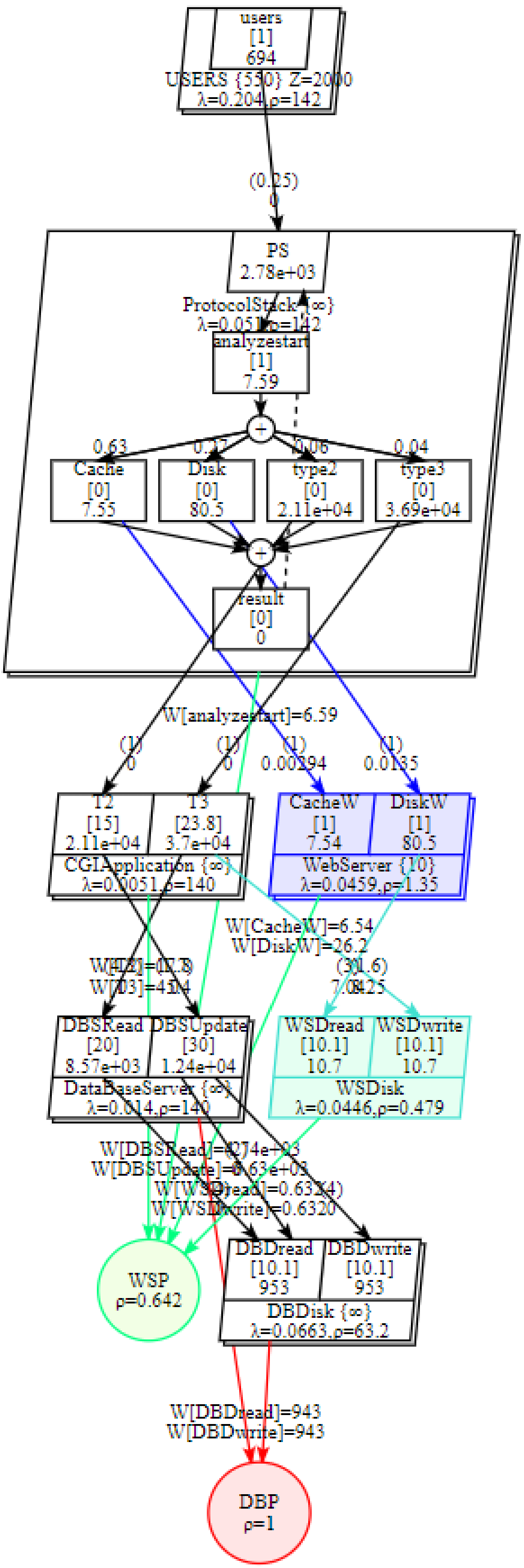


Figure 17: LQN with Infinite Processors

The other option is to make the tasks we previously multi-thraded into infinite servers, that would remove the chances of a bottleneck completely. This way, the only Bottleneck that will exist would be at the Database Processor. The issue with this approach is, we see a notable decrease in performance results, such as a decrease in throughput and an increase in response time.

Table 4 Infinite Server and Base Model Result Comparison

	Threads	Throughput	Response Time
Base Model Results	10	0.216993	534.75
Infinite Server Results	10	0.204127	694.384

To end, for the LQN analysis and bottleneck removal process, the best results in terms of throughput, response time, and device utilization are seen once we multi-thread the various tasks that are seen as bottlenecks. The code for multi-threading the various tasks, as well as more creating infinite servers can be seen in Appendix A.

Changes to the Model and Investigating Software Performance Engineering Principles:

A performance pattern is used to improve the overall design of the software by reducing resource use, both hardware and software. These performance patterns are based on software principles. To begin, we can use the Fast Path pattern, based on the Centering principle. The centering principle reduces the processing time and resources of the dominant workload. Additionally, the fast path pattern aims at reducing response time.

Fast Path Pattern:

In our case, the three types of entries are probability based, with the highest probability going to retrieving the webpage from the cache. Also, under the cache probability, there is a chance of a cache miss, in which the page must be retrieved from the disk, increasing service time, as well as spending time on the Webserver Disk read operation. The way the LQN is originally set up, each one of the four entries falls under the Protocol Stack task activity, with a given probability. Although this is still correct, we can streamline this by always going to the cache first, since it has the highest chance, over 50% in fact, of occurring.

What we do to tackle this is form a type1 activity under the Protocol Stack, which then breaks into two dummy parameters called Cache (prob.=0.7) and Disk (prob=0.3). Based on those additional probabilities, it will lead to the webserver and go to the real Cache or Disk. If it goes to the Disk, once again it will have the additional Disk Read operation.

Although this is a different alternative of doing the same Protocol Stack activity as in the base model, the results are nearly the same, as the same probabilities apply, and the actual entries of CacheW and DiskW are still utilized the same. The revised LQN model can be seen in the Figure 18.

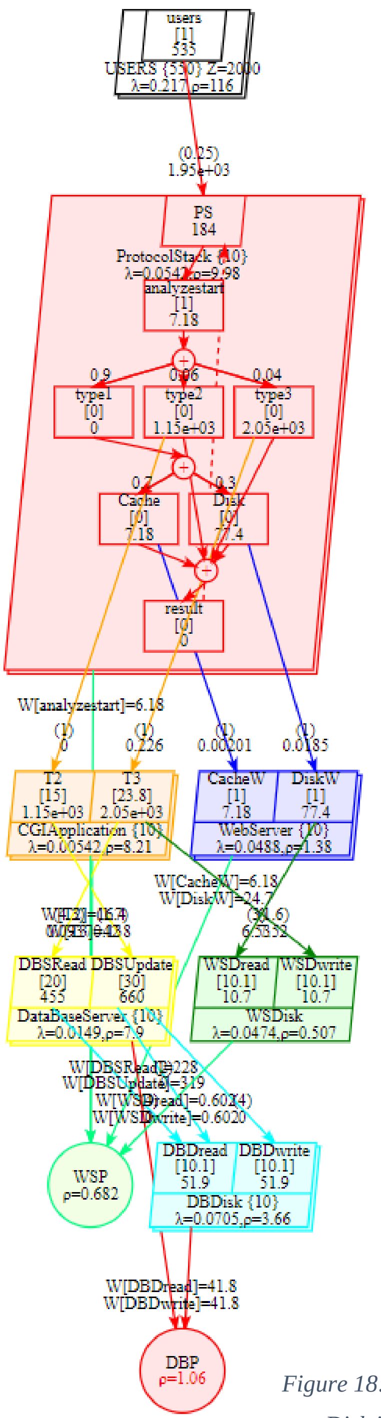


Figure 18: Dividing Cache and Disk in Protocol Stack

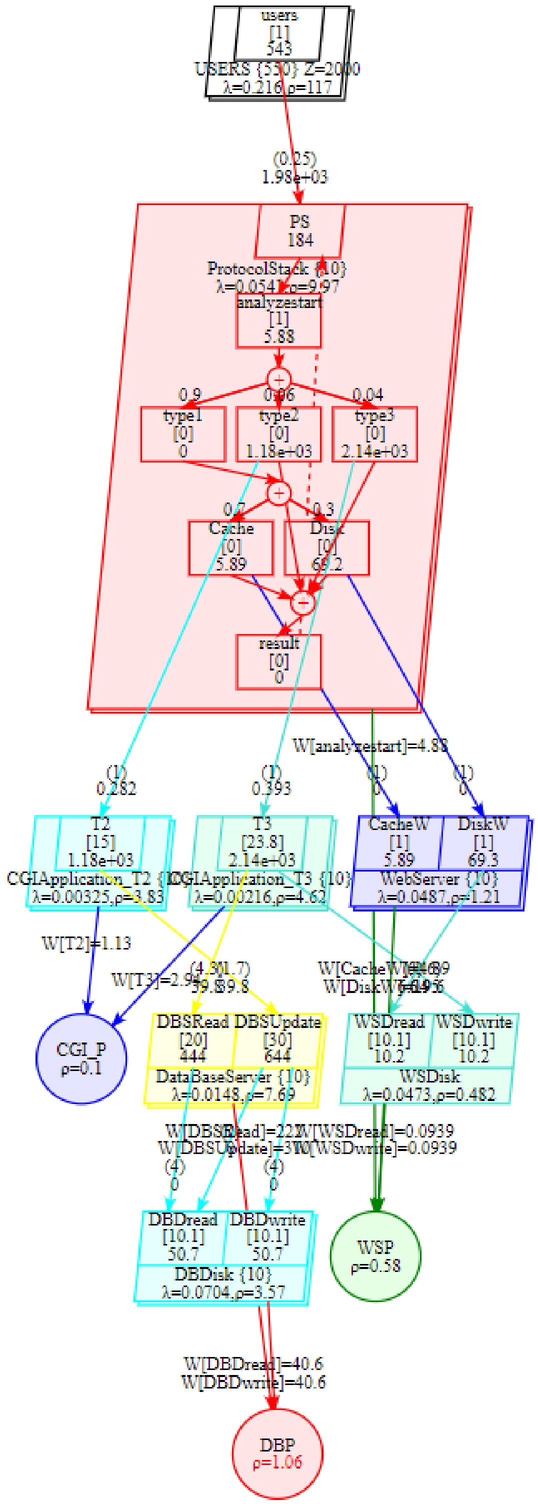


Figure 19: LQN Diagram After CGI Split

Flex Time Pattern:

Another software pattern that can be utilized to enhance our performance is called Flex Time. This pattern is based off the common notion to spread the demand of a given saturated objects over different times, without adding extra hardware. For example, in our case we understand that the fastest process is to retrieve the webpage off the cache. It would be advantageous if we could time the users based on their type of demand. For example, a surge of users would be better handled if they all required webpages that could be retrieved from the cache, and all users, if lower in quantity, that needed to access the rest of the three types of actions would come at another time. This way, we could avoid any bottlenecks that would occur because of users surging with high hardware demand requests. This pattern builds off the spread-the-load principle, which deals with spreading the load of a system based on time or resources.

Alternate Routes pattern:

An additional software pattern that also builds off the spread-the-load principle is the Alternate Routes pattern, in which we distribute the load for high-demand locations to different resources. What we can do in our case is split the CGI application tasks, T2 and T3, to two different tasks (create another CGI application secondary task to support T3), as well as add an additional CGI application allocated processor. In essence to follow this pattern, we can just keep adding resources, such as processors to divide the demand on the existing resources, however it soon becomes a question of resource management. An LQN diagram of this can be seen in the Figure 19. Also, a brief table below, Table 5, compares the results of this configuration comparted to the base model, where we see a slight decrease in improvement in throughput and response time. This is mainly because the CGI Application in general does not have a utilization in general, so spending extra resources on it may not be the best course of action, and additional resources will only end up slowing down the entire system, even if minimally.

Table 5 CGI Processor and Task Splitting Resulting

	Threads	Throughput	Response Time
Base Model Results	10	0.216993	534.75
CGI_Processor and Task Splitting Results	10	0.216284	542.872

Combining Fast Path and Alternate Routes:

The other course of action we have will essentially combine the Fast Path principle, and the Alternate Routes principle by mainly focusing on the tasks with the highest probability of occurring, so reading from the cache/disk, and providing them their own processor and splitting the cache and disk entries to independent tasks. The LQN diagram and results table for this comparison can once again be seen below, where we do see a slight improvement in results for throughput and response time.

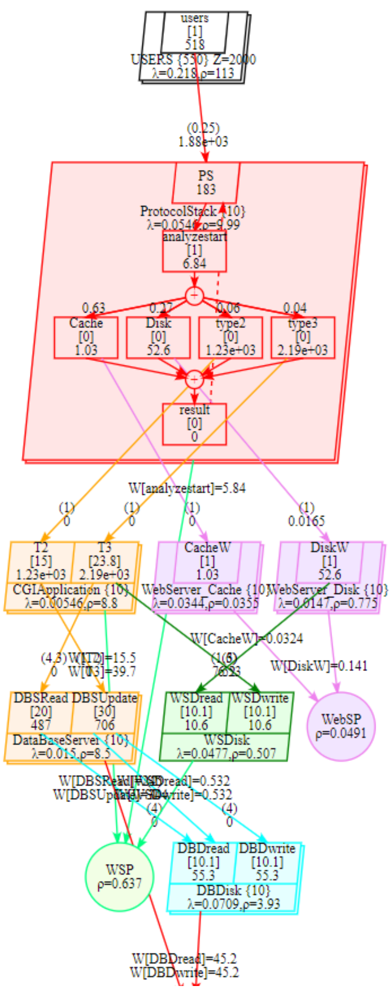


Figure 20: Dividing Tasks and Processors for Cache and Disk

Table 6 Cache/Disk Result Comparison to Base Model Results

	Threads	Throughput	Response Time
Base Model Results	10	0.216993	534.75
Cache/Disk Split	10	0.218456	517.731

Adding another processor for Protocol Stack:

Since the highest utilization of any task in our program is at the Protocol Stack, we can once again try adding another processor for it and observe the results. Nevertheless, although the results do seem slightly better, the best results we get are from the cache/disk split.

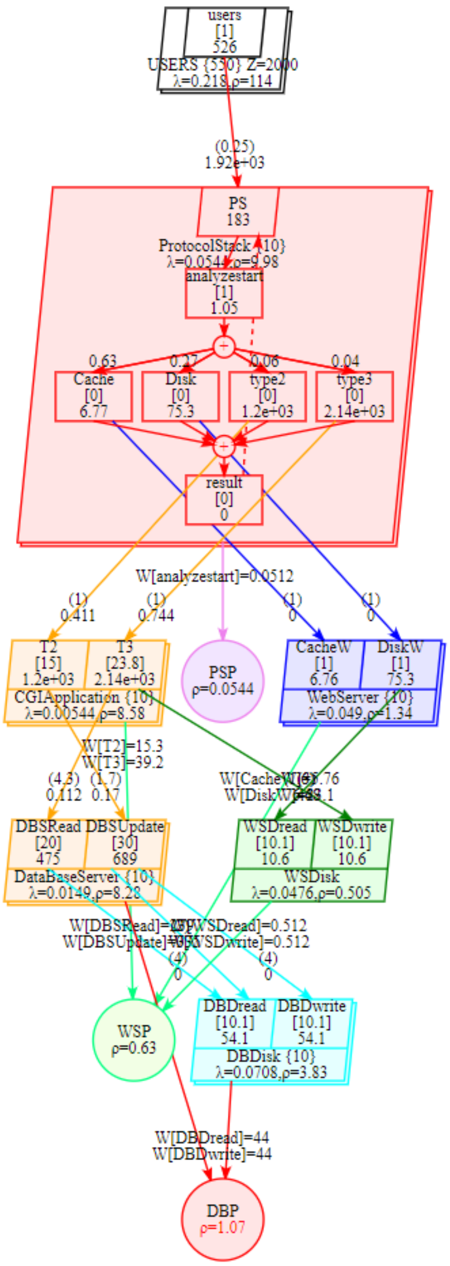


Figure 21: Adding Protocol Stack Processor

Table 7 Adding PS Processor Result Comparison to Base Model Results

	Threads	Throughput	Response Time
Base Model Results	10	0.216993	534.75
Protocl Stack Processor Results	10	0.217768	525.615

There are no major anti-patterns as such that exist by default in this software program and can only truly be understood once we assign software cases or test use cases to analyze the resulting data. The one anti-pattern that could be applicable is Traffic Jam, which as the name suggests is the inconsistency in response time, based on many different factors, such as One-lane bridge (processes waiting for other processes to finish before beginning execution, leading to longer queue times), or even a surge in user requests. Depending on what exactly the issue is, it may require including more resources such as extra processors, or even multi-threading like we did in our bottleneck removal. The tabulated results for the utilizations of the different devices, response times, and throughputs, for the different changes performed in the SPE analysis can be found in Appendix A.

Conclusion:

In the project report, we started by defining the model given in template one. The model required analyzing a request from the user and processes based on its type. We provided the reduced execution graphs for the model and each request type. Afterward, we implemented the model using LQN software, discussed the performance results, and identified the base model bottleneck as Protocol Stack activity. Next, we tested the model's sensitivity to different cache miss probabilities and found that best performance occurs when cache miss probability is minimized. Moreover, the bottleneck removal process took place. The best results occurred after multi-threading Protocol stack and other high utilization tasks. Finally, we applied multiple changes to the model based on the Software Performance Engineering Principles. We utilized Fast Path Pattern-based changes on the protocol stack, which slightly increased the performance. Also, we tried the Flex Time Pattern, and the Alternate Routes pattern changes, as well as combining the patterns. Lastly, we added a separate processor for Protocol Stack, which helped improve the results.

Appendix A:

Base Model Code:

```
$ConstructDynamic_SendStatic_Graphics = 23.8 # = 5 + 0.8 + K*1 + 0.25*(16+3*K), K = 8
#$threads=[1,5,8,10]
$N=[1,5,10,50,75,100,125,150,175,200,300,400,450,500,550]
G "Layers: 1, Customers: 1, Clients: 1, Tasks: 1, (Delay: 0), Processors: 1"          # Model comment
1e-03                                   # Convergence test value.
50                       # Maximum number of iterations.
10                       # Print intermediate results (see manual pages)
0.9                      # Model under-relaxation ( 0.0 < x <= 1.0)
-1
P 3
  p PC f i        # infinite processor for users
  p WSP f %u $uWSP      # processor for “ProtocolStack”, “WebServer”, "WSDisk" and “CGIApplication”
  p DBP f %u $uDBP  # processor for database server, and DBdisk
-1

T 7
  t USERS r users -1 PC z $ZU m $N %f $thru
  t ProtocolStack n PS -1 WSP %u $uProtocolStack
  t WebServer n CacheW DiskW -1 WSP %u $uWebServer
  t WSDisk n WSDread WSDwrite -1 WSP %u $uWSDisk
  t CGIApplication n T2 T3 -1 WSP %u $uCGIApplication
  t DataBaseServer n DBSUpdate DBSRead -1 DBP %u $uDBS
  t DBDisk n DBDread DBDwrite -1 DBP %u $uDBDisk
-1
E 0
# ---------- USERS ----------
  s users 1 -1 %s1 $rt
  #f users 1 -1   #testing, copying from slide 43
  y users PS 0.25 -1
# ---------- ProtocolStack ----------
  A PS analyzestart                       # Entry protocol analyze is defined using activties
# ---------- Webserver ----------
  s CacheW 1.0 -1
  s DiskW 1.0  -1
  y DiskW WSDread 3 -1
# ---------- WSDisk ----------
  s WSDread 10.1 -1
  s WSDwrite 10.1 -1
# ---------- CGIApplication ----------
  s T2 15 -1
  s T3 $ConstructDynamic_SendStatic_Graphics -1
  y T2 DBSUpdate 1.7 -1
  y T3 DBSRead 4.3 -1
  y T3 WSDwrite 1.6 -1  # 0.2*8
# ---------- DataBaseServer ----------
  s DBSUpdate 30 -1
  s DBSRead 20 -1
  y DBSUpdate DBDread 2 -1
  y DBSUpdate DBDwrite 4  -1
  y DBSRead DBDread 4 -1
# ---------- DBDisk ----------
  s DBDread 10.1 -1
  s DBDwrite 10.1 -1
-1
A ProtocolStack
  s analyzestart 1
  # Cache, Disk, type2 and type3  are just dummy entries to determine request type
  s Cache 0
  s Disk 0
  s type2 0
  s type3 0
  # Send the dummy entry to the related task entry
```

```
 y Cache CacheW 1
 y Disk DiskW 1
 y type2 T2 1
 y type3 T3 1
 # result for sending a reply to the protocol stack
 s result 0
: # separator
 # With PS = 0.9, PR = 0.6, and PCM = 0.3
 # type1 Cache prob = (1-PCM)*PS, type1 disk = PCM*PS
 # type2 prob = PR*(1-PS), and type3 = (1-PR)(1-PS)
 analyzestart -> (0.63)Cache + (0.27)Disk + (0.06)type2 + (0.04)type3;
 # Sending the results as a reply to protocol stack(PS)
 # as requierd by the LQN language
 Cache + Disk + type2 + type3 -> result;
 result[PS]
-1
R 0
$0=$N
#$threads
$thru $rt
$uCGIApplication $uDBDisk $uDBP $uDBS
$uProtocolStack $uWSDisk $uWSP  $uWebServer
-1
```

Adding threads:

```
#threads
t USERS r users -1 PC z $ZU m $N %f $thru
t ProtocolStack n PS -1 WSP i %u $uProtocolStack
t WebServer n CacheW DiskW -1 WSP m $threads %u $uWebServer
t WSDisk n WSDread WSDwrite -1 WSP %u $uWSDisk
t CGIApplication n T2 T3 -1 WSP m $threads %u $uCGIApplication
t DataBaseServer n DBSUpdate DBSRead -1 DBP m $threads %u $uDBS
t DBDisk n DBDread DBDwrite -1 DBP m $threads %u $uDBDisk
-1
```

Figure 22: Multi-threading Tasks [1,5,8,10]

Infinite Server on given Tasks:

```
t USERS r users -1 PC z $ZU m $N %f $thru
t ProtocolStack n PS -1 WSP i %u $uProtocolStack
t WebServer n CacheW DiskW -1 WSP m $threads %u $uWebServer
t WSDisk n WSDread WSDwrite -1 WSP %u $uWSDisk
t CGIApplication n T2 T3 -1 WSP i %u $uCGIApplication
t DataBaseServer n DBSUpdate DBSRead -1 DBP i %u $uDBS
t DBDisk n DBDread DBDwrite -1 DBP i %u $uDBDisk
-1
```

Figure 23 Infinite Servers on Given Tasks

Output tables:

Table 8: Base Model (Threads=1) and Multi-Threading Results for Response Time, Throughput, and Device Utilizations

N	$threads	$thru	$rt	$uCGIApplication	$uDBDisk	$uDBP	$uDBS	$uProtocolStack	$uWSDisk	$uWSP	$uWebServer
1	1	0.000498	9.05235	0.00275359	0.00163385	0.00244269	0.00244269	0.00400811	0.00109845	0.001565	0.00113002
5	1	0.002488	9.47431	0.0137652	0.00816754	0.0122109	0.0122109	0.0200396	0.0054911	0.007825	0.00564905
10	1	0.004975	10.0214	0.027523	0.0163306	0.024415	0.024415	0.0400757	0.0109792	0.015646	0.0112953
50	1	0.02481	15.3475	0.137216	0.0814248	0.121734	0.121734	0.200133	0.054742	0.078007	0.0563216
75	1	0.037132	19.8549	0.205291	0.121855	0.182177	0.182173	0.299789	0.0819188	0.116727	0.0842713
100	1	0.049365	25.7582	0.272778	0.161976	0.242156	0.242146	0.398897	0.108883	0.155138	0.111984
125	1	0.061463	33.7796	0.33939	0.201636	0.301443	0.301423	0.49708	0.135529	0.193087	0.139344
150	1	0.073343	45.2336	0.404642	0.240554	0.359618	0.359586	0.593669	0.161666	0.230302	0.166151
175	1	0.084867	62.1545	0.467742	0.278267	0.41599	0.415942	0.687531	0.18698	0.266336	0.192082
200	1	0.095703	89.9908	0.527663	0.313941	0.469346	0.469331	0.775481	0.210969	0.300481	0.216687
300	1	0.119438	512.721	0.656915	0.391319	0.585051	0.585071	0.969285	0.262922	0.37442	0.26982
400	1	0.122174	1275.72	0.67169	0.40012	0.598224	0.598276	0.991703	0.268846	0.382861	0.275895
450	1	0.122592	1672.65	0.673954	0.401476	0.600252	0.600306	0.99513	0.269756	0.384157	0.276826
500	1	0.122856	2072.01	0.675378	0.402323	0.60152	0.601578	0.997296	0.270326	0.38497	0.277411
550	1	0.123034	2472.76	0.676339	0.402896	0.602378	0.602438	0.998756	0.270712	0.385519	0.277807
1	5	0.000498	9.05997	0.00275367	0.00163385	0.00244269	0.00244269	0.00400821	0.00109845	0.001565	0.00113009
5	5	0.002489	9.06378	0.0138514	0.00822282	0.0122129	0.0123373	0.0201525	0.00549484	0.007827	0.00572747
10	5	0.004977	9.14066	0.0279932	0.0165533	0.0244233	0.0249352	0.0408006	0.0109942	0.015652	0.01161
50	5	0.024884	9.323	0.147564	0.087112	0.121705	0.135146	0.214553	0.0550933	0.078133	0.0642329
75	5	0.037314	9.95935	0.236258	0.13516	0.182186	0.214114	0.348146	0.0827039	0.117007	0.102173
100	5	0.049736	10.5974	0.332629	0.186772	0.242547	0.301806	0.49288	0.110474	0.155901	0.14423
125	5	0.062152	11.1887	0.439177	0.242061	0.302468	0.398942	0.653408	0.13826	0.194625	0.190266
150	5	0.074559	11.8096	0.557094	0.301506	0.362013	0.506726	0.831101	0.166101	0.233227	0.240394
175	5	0.086957	12.4667	0.687796	0.365516	0.421095	0.626446	1.02732	0.193988	0.271692	0.294655
200	5	0.099345	13.1615	0.832543	0.434561	0.479673	0.759341	1.24374	0.221912	0.310005	0.353094
300	5	0.148717	17.1575	1.60603	0.786536	0.716495	1.4688	2.38903	0.335186	0.463266	0.633508
400	5	0.195996	40.7731	2.98335	1.37014	0.954225	2.75036	4.26105	0.450286	0.613972	0.99976
450	5	0.206686	177.204	3.53055	1.60163	1.01394	3.27241	4.94144	0.478435	0.64996	1.10277
500	5	0.207434	410.516	3.56875	1.61775	1.01756	3.3088	4.9896	0.480208	0.652236	1.10944
550	5	0.207557	649.78	3.5742	1.61973	1.01791	3.31256	4.9979	0.480322	0.652392	1.10983
1	8	0.00049	23.9571	0.00530271	0.0016331	0.0024399	0.0026613	0.00967787	0.0010926	0.00156	0.00112882
5	8	0.00248	14.2542	0.0138514	0.0082228	0.0122129	0.0123373	0.0212309	0.0054937	0.00782	0.00572747
10	8	0.00498	9.14277	0.0280298	0.0165928	0.024424	0.025005	0.0408422	0.0109954	0.01565	0.0116389
50	8	0.02488	9.3473	0.148921	0.08834	0.121823	0.137337	0.216002	0.0551485	0.07817	0.0650126
75	8	0.03731	10.0593	0.241017	0.138179	0.182432	0.219722	0.353933	0.082836	0.1171	0.103986
100	8	0.04973	10.7918	0.342424	0.192638	0.242926	0.312912	0.504913	0.110686	0.15603	0.147504
125	8	0.06214	11.4756	0.456721	0.252178	0.303055	0.418415	0.674659	0.138593	0.19482	0.195508
150	8	0.07454	12.206	0.58579	0.317606	0.362845	0.538169	0.865365	0.16658	0.2335	0.248147
175	8	0.08693	12.9882	0.732245	0.389741	0.422158	0.674629	1.07986	0.194622	0.27203	0.306047
200	8	0.09931	13.8395	0.898355	0.469586	0.480981	0.830127	1.32113	0.222724	0.31042	0.369194
300	8	0.14858	19.1222	1.8957	0.916594	0.720005	1.74928	2.72871	0.337616	0.46463	0.683213
400	8	0.19718	28.5627	3.81209	1.73791	0.949505	3.57071	5.20373	0.45252	0.61453	1.09697
450	8	0.215	92.9767	5.90876	2.63899	1.05291	5.61349	7.58338	0.500955	0.67551	1.3044
500	8	0.21674	306.823	6.27475	2.79899	1.06276	5.97384	7.97771	0.505601	0.68132	1.32541
550	8	0.21655	539.842	6.28848	2.80513	1.06305	5.98809	7.98399	0.505637	0.68132	1.32568
1	10	0.00049	36.5401	0.00812405	0.0016329	0.0024376	0.002909	0.0154007	0.0010865	0.00155	0.00112317
5	10	0.00248	20.0506	0.0138514	0.0082228	0.0122129	0.0123373	0.0224502	0.0054924	0.00782	0.00572746
10	10	0.00498	9.14348	0.0280423	0.0166068	0.0244243	0.0250292	0.0408561	0.0109958	0.01565	0.0116486
50	10	0.02488	9.35547	0.149397	0.0887899	0.121872	0.138122	0.216493	0.0551688	0.07818	0.0652764
75	10	0.03731	10.0936	0.242729	0.139306	0.182538	0.221773	0.355941	0.0828863	0.11714	0.104604
100	10	0.04973	10.8597	0.346007	0.194868	0.243093	0.317046	0.509152	0.110766	0.15608	0.148623
125	10	0.06214	11.5778	0.463279	0.25611	0.303318	0.425821	0.68244	0.138718	0.1949	0.197484
150	10	0.07454	12.3526	0.596893	0.324019	0.363213	0.550504	0.87866	0.166756	0.23361	0.251597
175	10	0.08692	13.1944	0.749779	0.399685	0.422696	0.693983	1.10063	0.194875	0.2722	0.311048
200	10	0.0993	14.1124	0.925006	0.484401	0.481688	0.859422	1.35185	0.223059	0.31063	0.375991
300	10	0.14851	20.041	2.01832	0.975071	0.720783	1.87252	2.86854	0.338156	0.46479	0.70104
400	10	0.19705	29.864	4.26663	1.94357	0.948806	4.02728	5.69359	0.452933	0.61415	1.13263
450	10	0.21618	81.5511	7.49525	3.34299	1.05629	7.19204	9.24619	0.504179	0.67854	1.36414
500	10	0.21723	301.693	8.17933	3.64443	1.06461	7.86884	9.95228	0.507608	0.68272	1.38052
550	10	0.21699	534.75	8.21262	3.66049	1.06485	7.90329	9.98415	0.507302	0.68229	1.37901

Table 9: Infinite Server Model Results for Response Time, Throughput, and Device Utilizations

N	$threads	$thru	$rt	$uCGIApplication	$uDBDisk	$uDBP	$uDBS	$uProtocolStack	$uWSDisk	$uWSP	$uWebServer
1	1	0.0005	9.05235	0.00275359	0.00163	0.00244	0.00244	0.00400811	0.00109845	0.00157	0.00113002
5	1	0.00249	9.09863	0.0138511	0.00822	0.01221	0.01234	0.0202312	0.00549472	0.00783	0.00570634
10	1	0.00498	9.21943	0.0280413	0.01661	0.02442	0.02503	0.0412148	0.0109953	0.01565	0.0115531
50	1	0.02488	9.8229	0.150708	0.09036	0.12203	0.14077	0.229369	0.0551872	0.07818	0.0634776
75	1	0.03729	10.9647	0.248864	0.14371	0.18294	0.22955	0.392845	0.0829963	0.1172	0.100724
100	1	0.04969	12.2583	0.359845	0.2042	0.24376	0.33378	0.585916	0.110929	0.15614	0.14178
125	1	0.06207	13.6129	0.490057	0.27364	0.30447	0.45759	0.821489	0.138991	0.19499	0.18675
150	1	0.07443	15.192	0.644162	0.35444	0.36504	0.60617	1.11099	0.167176	0.23375	0.235733
175	1	0.08676	17.0579	0.824294	0.44937	0.42545	0.78251	1.46967	0.195475	0.27238	0.288814
200	1	0.09899	20.2223	1.09221	0.56552	0.4855	1.01607	1.97359	0.22423	0.31117	0.34614
300	1	0.14683	43.1696	3.02663	1.44523	0.7205	2.85597	6.21614	0.337934	0.46173	0.614097
400	1	0.18649	144.929	11.1829	5.0996	0.91552	10.8973	26.7714	0.435925	0.58656	0.888456
450	1	0.19599	296.02	19.6785	8.9432	0.96179	19.4246	57.7991	0.46001	0.61639	0.9587
500	1	0.19907	511.72	25.144	11.4439	0.9766	24.9602	101.658	0.467873	0.62604	0.981614
550	1	0.20003	749.598	27.6807	12.5821	0.98134	27.4966	149.739	0.470348	0.62906	0.988854
1	5	0.00047	148.419	0.0084769	0.00163	0.00244	0.00273	0.0381104	0.00106226	0.00151	0.00109435
5	5	0.00248	21.1314	0.0138514	0.00822	0.01221	0.01234	0.0215736	0.00549333	0.00782	0.00572751
10	5	0.00498	9.1435	0.0280423	0.01661	0.02442	0.02503	0.0408466	0.0109958	0.01565	0.0116379
50	5	0.02488	9.36522	0.150753	0.09038	0.12206	0.14081	0.21726	0.0552319	0.07823	0.0656863
75	5	0.03731	10.1797	0.249013	0.14378	0.18301	0.22967	0.361966	0.0830758	0.11729	0.105887
100	5	0.04972	11.0795	0.360204	0.20436	0.24391	0.33406	0.523624	0.11108	0.15632	0.151289
125	5	0.06213	11.941	0.490805	0.27397	0.30474	0.45818	0.710455	0.139244	0.1953	0.202137
150	5	0.07452	12.8981	0.645596	0.35508	0.36549	0.60735	0.927382	0.16757	0.23422	0.25868
175	5	0.08689	13.9727	0.826883	0.45057	0.42615	0.78474	1.18046	0.196057	0.27309	0.321182
200	5	0.09925	15.1465	1.04867	0.56596	0.48668	1.0061	1.47313	0.224707	0.31189	0.389922
300	5	0.14792	28.0435	3.10735	1.48112	0.72585	2.9329	4.03146	0.340273	0.46516	0.730521
400	5	0.19168	86.9229	14.8215	6.71888	0.94106	14.5035	16.3747	0.447678	0.60291	1.14767
450	5	0.20302	216.595	41.957	18.8877	0.9967	41.6183	43.6734	0.476125	0.63857	1.27615
500	5	0.204	451.105	90.0773	40.66	1.00123	89.7804	91.7583	0.478543	0.64159	1.28763
550	5	0.20413	694.379	140.027	63.2438	1.00136	139.72	141.667	0.478719	0.64181	1.28863
1	8	0.00037	689.934	0.0286994	0.00163	0.00243	0.00391	0.143407	0.00093619	0.00132	0.000965638
5	8	0.00248	18.0548	0.0139017	0.00822	0.01221	0.01234	0.0209553	0.00549399	0.00782	0.00572764
10	8	0.00498	9.16388	0.0280423	0.01661	0.02442	0.02503	0.040863	0.0109958	0.01565	0.0116459
50	8	0.02488	9.36684	0.150754	0.09038	0.12206	0.14081	0.217466	0.0552354	0.07823	0.06591
75	8	0.03731	10.1887	0.249015	0.14377	0.18301	0.22966	0.362492	0.0830807	0.1173	0.106431
100	8	0.04972	11.0942	0.360208	0.20436	0.24391	0.33405	0.524643	0.111088	0.15632	0.152334
125	8	0.06213	11.9623	0.490815	0.27396	0.30474	0.45817	0.712181	0.139258	0.19531	0.203897
150	8	0.07452	12.9268	0.645615	0.35507	0.36548	0.60733	0.930076	0.16759	0.23424	0.261409
175	8	0.08689	14.01	0.826915	0.45055	0.42614	0.78471	1.18443	0.196084	0.27312	0.32518
200	8	0.09924	15.1934	1.04871	0.56594	0.48667	1.00605	1.4787	0.224743	0.31192	0.39554
300	8	0.14791	28.1546	3.10715	1.48085	0.72581	2.93231	4.0476	0.34023	0.46513	0.747194
400	8	0.19166	87.0575	14.8111	6.71376	0.941	14.4921	16.3996	0.447557	0.60287	1.18436
450	8	0.20302	216.63	41.9209	18.8707	0.99668	41.5806	43.6799	0.475994	0.63857	1.32046
500	8	0.204	451.107	90.0356	40.6404	1.00123	89.7372	91.7587	0.478414	0.64159	1.33269
550	8	0.20413	694.382	139.985	63.2239	1.00136	139.677	141.667	0.478588	0.64181	1.33376
1	10	0.00037	689.762	0.0286923	0.00163	0.00243	0.00391	0.143374	0.00093623	0.00132	0.00096568
5	10	0.00248	18.0519	0.0139017	0.00822	0.01221	0.01234	0.0209551	0.00549399	0.00782	0.00572764
10	10	0.00498	9.16388	0.0280423	0.01661	0.02442	0.02503	0.0408654	0.0109958	0.01565	0.0116486
50	10	0.02488	9.36738	0.150754	0.09038	0.12206	0.14081	0.217536	0.0552365	0.07824	0.0659855
75	10	0.03731	10.1918	0.249015	0.14377	0.18301	0.22966	0.362671	0.0830823	0.1173	0.106616
100	10	0.04972	11.0993	0.36021	0.20436	0.24391	0.33405	0.524988	0.111091	0.15633	0.152691
125	10	0.06213	11.9696	0.490818	0.27396	0.30473	0.45817	0.712776	0.139263	0.19531	0.204502
150	10	0.07452	12.9367	0.645621	0.35507	0.36548	0.60733	0.931011	0.167597	0.23425	0.262356
175	10	0.08689	14.0229	0.826926	0.45055	0.42614	0.7847	1.18581	0.196094	0.27313	0.326579
200	10	0.09924	15.2098	1.04873	0.56593	0.48667	1.00603	1.48068	0.224756	0.31194	0.397525
300	10	0.14791	28.1963	3.10708	1.48075	0.7258	2.93209	4.05365	0.340215	0.46512	0.753335
400	10	0.19166	87.11	14.807	6.71176	0.94098	14.4876	16.4093	0.447517	0.60286	1.19854
450	10	0.20302	216.644	41.9065	18.864	0.99668	41.5656	43.6824	0.475952	0.63856	1.33784
500	10	0.20392	451.934	90.0772	40.6443	1.0011	89.7509	91.8714	0.478233	0.6414	1.34965
550	10	0.20413	694.384	139.967	63.2157	1.00136	139.658	141.667	0.478546	0.64181	1.35149

Table 10: Implementing Fast Path Principle Initially by Combining Cache/Disk

N	$threads	$thru	$rt	$uCGIApplication	$uDBDisk	$uDBP	$uDBS	$uProtocc	$uWSDisk	$uWSP	$uWebServer
1	1	0.00049778	8.91095	0.0026834	0.001634	0.00244	0.00244	0.00394	0.0010281	0.0014951	0.0011301
5	1	0.00248842	9.30976	0.0134143	0.0081682	0.01221	0.01221	0.01969	0.0051397	0.0074738	0.00564935
10	1	0.00497555	9.82623	0.0268218	0.0163322	0.02442	0.02442	0.03936	0.0102768	0.0149438	0.0112958
50	1	0.0248159	14.8374	0.133771	0.0814566	0.12178	0.12178	0.19636	0.0512549	0.0745314	0.056338
75	1	0.0371463	19.0478	0.200229	0.121929	0.18229	0.18229	0.29396	0.0767207	0.111561	0.0843279
100	1	0.0493944	24.522	0.266233	0.162129	0.24239	0.24239	0.39093	0.102015	0.148341	0.112127
125	1	0.0615189	31.899	0.331553	0.201921	0.30188	0.30188	0.48694	0.127051	0.184744	0.139639
150	1	0.0734475	42.2799	0.395799	0.241068	0.36041	0.3604	0.58143	0.151679	0.220553	0.166699
175	1	0.0850456	57.7365	0.45824	0.279122	0.4173	0.41729	0.67331	0.17562	0.255361	0.193
200	1	0.0960487	82.3277	0.517446	0.315211	0.47125	0.47125	0.76051	0.198323	0.28837	0.21794
300	1	0.121556	468.015	0.656388	0.399043	0.5966	0.59663	0.96187	0.251139	0.365224	0.276686
400	1	0.124571	1211.18	0.670928	0.408782	0.61114	0.61114	0.98653	0.257184	0.373946	0.282592
450	1	0.125007	1600.04	0.673111	0.410152	0.61319	0.61318	0.99009	0.258037	0.375187	0.283517
500	1	0.125174	1994.1	0.676083	0.411046	0.61453	0.61453	0.99054	0.25868	0.376175	0.285023
550	1	0.125465	2384.09	0.675708	0.411679	0.61548	0.61548	0.99362	0.259012	0.376608	0.284608
1	5	0.00049778	8.91197	0.0026834	0.001634	0.00244	0.00244	0.00394	0.0010281	0.0014951	0.0011301
5	5	0.0024889	8.92155	0.0134961	0.0082234	0.01221	0.01234	0.01979	0.0051434	0.0074752	0.0057237
10	5	0.0049776	8.99467	0.0272741	0.0165545	0.02443	0.02494	0.04006	0.0102914	0.0149492	0.0115942
50	5	0.0248858	9.16908	0.143643	0.0871306	0.12173	0.13518	0.2101	0.0515994	0.0746388	0.0638295
75	5	0.037317	9.7745	0.230058	0.135191	0.18222	0.21417	0.34068	0.0774767	0.111778	0.10126
100	5	0.0497409	10.3921	0.323943	0.186827	0.24261	0.3019	0.48191	0.103523	0.148944	0.142606
125	5	0.0621585	10.9625	0.427802	0.24215	0.30256	0.3991	0.63842	0.1296	0.185956	0.187731
150	5	0.0745678	11.5626	0.542828	0.30164	0.36215	0.50697	0.81158	0.155748	0.222859	0.236749
175	5	0.0869682	12.1982	0.670441	0.365708	0.42127	0.6268	1.00275	0.181957	0.259639	0.289705
200	5	0.0993589	12.8714	0.811905	0.43483	0.4799	0.75985	1.2136	0.20822	0.296284	0.346643
300	5	0.14875	16.7212	1.56979	0.787233	0.71686	1.47033	2.3323	0.314746	0.442731	0.61814
400	5	0.196357	37.0738	2.9358	1.37522	0.95582	2.76158	4.174	0.423758	0.587497	0.970486
450	5	0.208278	160.507	3.54722	1.63714	1.02202	3.35216	4.92821	0.453116	0.625527	1.07935
500	5	0.210638	373.653	3.60917	1.64709	1.02287	3.39779	4.99577	0.457523	0.631475	1.09638
550	5	0.209349	627.16	3.60137	1.66068	1.0271	3.40542	4.99742	0.455418	0.628521	1.0881
1	8	0.00049415	23.7816	0.00530313	0.0016332	0.00244	0.00267	0.00974	0.0010223	0.0014864	0.00112912
5	8	0.00248315	14.2562	0.0134961	0.0082234	0.01221	0.01234	0.0209	0.0051422	0.0074726	0.0057237
10	8	0.0049776	8.99668	0.0273097	0.016594	0.02443	0.02501	0.0401	0.0102926	0.0149494	0.0116222
50	8	0.0248854	9.19245	0.144965	0.0883552	0.12184	0.13736	0.21149	0.0516497	0.0746696	0.064582
75	8	0.0373151	9.87162	0.234743	0.138206	0.18246	0.21977	0.34633	0.0776003	0.111858	0.103012
100	8	0.0497361	10.5822	0.333604	0.192685	0.24298	0.31299	0.49368	0.103721	0.149055	0.145769
125	8	0.0621496	11.2438	0.445132	0.252255	0.30313	0.41855	0.65925	0.129912	0.186126	0.192793
150	8	0.0745532	11.9518	0.571208	0.317724	0.36295	0.53838	0.84519	0.156198	0.223097	0.244214
175	8	0.0869458	12.7111	0.71443	0.389936	0.42232	0.67499	1.054	0.182563	0.259944	0.300353
200	8	0.0993258	13.535	0.87703	0.469857	0.48118	0.83064	1.28899	0.208993	0.296646	0.361547
300	8	0.148608	18.6886	1.85724	0.917344	0.72032	1.75087	2.66428	0.317122	0.444017	0.663962
400	8	0.197302	27.2929	3.7066	1.72508	0.94802	3.54245	5.03299	0.425638	0.587592	1.05929
450	8	0.215943	83.9387	5.88714	2.66556	1.05566	5.66993	7.50967	0.472564	0.647376	1.26186
500	8	0.21847	288.719	6.31718	2.84219	1.06423	6.08609	7.99237	0.477451	0.653601	1.28389
550	8	0.217329	530.659	6.3429	2.86366	1.06639	6.11916	7.98814	0.476838	0.65267	1.28135
1	10	0.00049108	36.4781	0.00817921	0.0016331	0.00244	0.00292	0.01557	0.0010163	0.0014777	0.00112338
5	10	0.0024767	20.1661	0.0134961	0.0082234	0.01221	0.01234	0.02215	0.0051409	0.0074696	0.00572369
10	10	0.0049776	8.99735	0.0273218	0.016608	0.02443	0.02503	0.04011	0.010293	0.0149495	0.0116315
50	10	0.0248853	9.20031	0.14543	0.0888038	0.12189	0.13815	0.21196	0.0516681	0.0746819	0.0648364
75	10	0.0373145	9.90491	0.23643	0.139331	0.18257	0.22182	0.34829	0.0776472	0.111892	0.10361
100	10	0.0497344	10.6486	0.33714	0.194912	0.24314	0.31712	0.49783	0.103796	0.149103	0.146849
125	10	0.0621465	11.344	0.451604	0.256186	0.30339	0.42596	0.66675	0.130031	0.1862	0.194571
150	10	0.0745479	12.0934	0.582144	0.324136	0.36331	0.55072	0.85794	0.156365	0.223196	0.247291
175	10	0.0869371	12.9096	0.731696	0.399862	0.42283	0.69431	1.0742	0.182794	0.260079	0.305046
200	10	0.0993125	13.801	0.903327	0.484658	0.48187	0.85991	1.31901	0.209304	0.296828	0.368009
300	10	0.148542	19.5926	1.97888	0.975756	0.72103	1.874	2.80181	0.317632	0.444137	0.680705
400	10	0.197118	29.2025	4.20648	1.94556	0.94907	4.03257	5.5771	0.426066	0.587073	1.09239
450	10	0.21626	80.7447	7.54063	3.39784	1.05787	7.31555	9.22456	0.474528	0.648655	1.31246
500	10	0.217368	300.169	8.24324	3.70951	1.06633	8.01426	9.94554	0.477894	0.652825	1.32866
550	10	0.217354	530.533	8.27781	3.72489	1.0665	8.04741	9.98437	0.477766	0.652656	1.32803

Table 11: Implementing Alternate Routes Principle by Splitting CGI Application

N	$threads	$thru	$rt	$uCGIApplication_T2	$uCGIApplication_T3	$uCGI_P	$uDBDisk	$uDBP	$uDBS	$uProtocolStack	$uWSDisk	$uWSP	$uWebServer
1	1	0.0005	9.05235	0.00126204	0.00149185	0.000230456	0.00163385	0.00244	0.00244	0.00401092	0.00109845	0.00133	0.00113001
5	1	0.00249	9.47396	0.00631061	0.00746114	0.00115195	0.00816729	0.01221	0.01221	0.0201078	0.00549076	0.00667	0.00564863
10	1	0.00498	10.0229	0.0126203	0.0149249	0.00230276	0.0163289	0.02441	0.02441	0.0403468	0.0109766	0.01334	0.0112921
50	1	0.02481	15.353	0.0626891	0.0742807	0.011402	0.0811852	0.12137	0.12137	0.206615	0.0544209	0.06616	0.0559278
75	1	0.03713	20.0027	0.0932222	0.110597	0.0169237	0.121279	0.18128	0.1812	0.314216	0.08096	0.09847	0.083038
100	1	0.04936	26.1245	0.122719	0.145775	0.0222384	0.1607	0.24014	0.23995	0.42434	0.10669	0.12984	0.109163
125	1	0.06142	35.8074	0.152266	0.180993	0.0275694	0.200299	0.29937	0.29922	0.532673	0.132437	0.16119	0.135359
150	1	0.07325	48.9815	0.179525	0.213638	0.0324564	0.238103	0.35583	0.35561	0.644046	0.156382	0.19047	0.159439
175	1	0.08466	69.514	0.204727	0.243898	0.0369612	0.273961	0.40938	0.4091	0.754264	0.178633	0.21776	0.181691
200	1	0.09529	103.872	0.22706	0.270775	0.0409435	0.306432	0.45792	0.45763	0.859206	0.198405	0.24212	0.201428
300	1	0.11607	607.431	0.267309	0.319276	0.0481038	0.364761	0.54555	0.546	1.06731	0.233589	0.28598	0.237103
400	1	0.11829	1413.72	0.271363	0.324172	0.0488216	0.370796	0.55462	0.55516	1.08999	0.237144	0.29044	0.240695
450	1	0.11865	1829.12	0.272012	0.324958	0.0489366	0.371798	0.55612	0.55667	1.0937	0.237723	0.29116	0.241271
500	1	0.11887	2246.6	0.272424	0.325457	0.0490094	0.37244	0.55708	0.55763	1.09607	0.238092	0.29162	0.241636
550	1	0.11903	2665.11	0.272712	0.325806	0.0490605	0.372893	0.55776	0.55831	1.09774	0.238351	0.29195	0.241893
1	5	0.0005	9.18123	0.00126194	0.00149173	0.000230457	0.00163385	0.00244	0.00244	0.00400829	0.00109845	0.00133	0.00113009
5	5	0.00249	9.06284	0.00634084	0.00749818	0.00115226	0.0082229	0.01221	0.01234	0.0201349	0.00549261	0.00667	0.00572115
10	5	0.00498	9.13112	0.0128019	0.015142	0.00230435	0.0165537	0.02442	0.02494	0.040722	0.0109853	0.01335	0.011584
50	5	0.02488	9.29679	0.066938	0.0794114	0.0115058	0.0871925	0.12182	0.13528	0.212598	0.0548856	0.06665	0.0635427
75	5	0.03732	9.86166	0.106785	0.126719	0.0172299	0.135318	0.18238	0.21438	0.343184	0.0822227	0.09982	0.100497
100	5	0.04974	10.4487	0.149806	0.177904	0.0229605	0.187082	0.24291	0.30235	0.483647	0.109615	0.13302	0.141081
125	5	0.06216	10.9856	0.197179	0.234308	0.0286671	0.242618	0.30307	0.39993	0.638408	0.136913	0.16609	0.185086
150	5	0.07457	11.5496	0.249457	0.296582	0.0343577	0.302432	0.36294	0.50841	0.808756	0.164155	0.19907	0.232563
175	5	0.08697	12.1462	0.307304	0.365505	0.04003	0.36698	0.42244	0.62917	0.995964	0.191331	0.23195	0.283489
200	5	0.09936	12.7776	0.371321	0.441784	0.0456822	0.436779	0.48155	0.76354	1.20162	0.21843	0.26472	0.337841
300	5	0.14877	16.4622	0.716144	0.851852	0.0682593	0.791259	0.71879	1.47829	2.29794	0.326284	0.39491	0.590684
400	5	0.19656	34.8795	1.35042	1.61646	0.0905171	1.37825	0.95663	2.76843	4.11784	0.434511	0.52445	0.910818
450	5	0.20799	163.608	1.66376	2.0124	0.0962381	1.62585	1.01956	3.32728	4.94518	0.462357	0.55747	1.00484
500	5	0.20824	401.152	1.68114	2.03572	0.0964177	1.63687	1.02186	3.35185	4.98892	0.463231	0.55848	1.00787
550	5	0.20857	636.887	1.68521	2.04122	0.0964895	1.6386	1.02204	3.35828	5.00155	0.463624	0.55898	1.00923
1	8	0.00049	24.0762	0.00247242	0.00286412	0.000229176	0.00163079	0.00244	0.00266	0.00970966	0.00109237	0.00133	0.00112461
5	8	0.00248	14.3135	0.00634081	0.00749814	0.00115226	0.00822112	0.01221	0.01234	0.0212277	0.00549141	0.00667	0.00572114
10	8	0.00498	9.13301	0.0128162	0.0151593	0.00230439	0.0165931	0.02442	0.02501	0.0407564	0.0109857	0.01335	0.0116104
50	8	0.02488	9.31839	0.0674913	0.0800736	0.0115104	0.0883986	0.1219	0.13744	0.213842	0.0549127	0.06668	0.0642392
75	8	0.03731	9.95119	0.108825	0.129118	0.0172423	0.138301	0.18258	0.21993	0.348425	0.0822949	0.09989	0.102103
100	8	0.04974	10.6264	0.154042	0.182877	0.0229775	0.192884	0.2432	0.31334	0.494623	0.109718	0.13311	0.143943
125	8	0.06215	11.249	0.20482	0.24327	0.028693	0.25264	0.30351	0.41924	0.657875	0.137071	0.16623	0.189615
150	8	0.07455	11.9147	0.262028	0.311318	0.0343936	0.318403	0.36356	0.53962	0.840272	0.164377	0.19927	0.239185
175	8	0.08695	12.6284	0.326856	0.388482	0.0400736	0.391065	0.42323	0.67709	1.04448	0.191608	0.23219	0.293178
200	8	0.09933	13.4078	0.400387	0.476005	0.0457337	0.471683	0.48251	0.83411	1.2734	0.218767	0.265	0.351485
300	8	0.14863	18.4102	0.846181	1.00482	0.0685199	0.923498	0.7231	1.7638	2.61715	0.328222	0.39676	0.633643
400	8	0.19749	25.3414	1.64531	1.95167	0.089786	1.69524	0.94211	3.47778	4.80453	0.431365	0.52095	0.975787
450	8	0.21516	91.4605	2.77476	3.33073	0.099552	2.66159	1.05442	5.66452	7.5703	0.478746	0.57665	1.16029
500	8	0.21608	313.977	2.93686	3.54802	0.100072	2.76531	1.06073	5.89777	7.96329	0.48129	0.57962	1.17073
550	8	0.21602	546.021	2.94619	3.56069	0.100108	2.77124	1.06118	5.91042	7.98188	0.481422	0.57975	1.17138
1	10	0.00049	36.8801	0.00374512	0.00430567	0.00022791	0.00162833	0.00243	0.00289	0.0152328	0.00108642	0.00132	0.00111852
5	10	0.00248	19.8876	0.00634077	0.0074981	0.00115226	0.00821919	0.01221	0.01233	0.022401	0.00549011	0.00667	0.00572113
10	10	0.00498	9.13362	0.0128211	0.0151652	0.0023044	0.0166071	0.02442	0.02503	0.0407678	0.0109858	0.01335	0.0116193
50	10	0.02488	9.32564	0.0676861	0.0803066	0.0115123	0.0888393	0.12194	0.13821	0.214263	0.0549232	0.06669	0.0644742
75	10	0.03731	9.98191	0.109561	0.129983	0.0172476	0.139412	0.18266	0.22195	0.350247	0.0823244	0.09992	0.102649
100	10	0.04973	10.6886	0.155596	0.184701	0.0229848	0.195085	0.24332	0.31742	0.498498	0.10976	0.13315	0.144917
125	10	0.06215	11.343	0.207677	0.246636	0.0287039	0.256522	0.30371	0.42655	0.665039	0.137135	0.16629	0.191336
150	10	0.07455	12.0503	0.266877	0.317074	0.0344079	0.324744	0.36384	0.55183	0.852583	0.164463	0.19934	0.242211
175	10	0.08694	12.8204	0.334558	0.397612	0.0400948	0.400906	0.42364	0.69625	1.0638	0.191734	0.2323	0.297529
200	10	0.09932	13.6623	0.41216	0.489945	0.0457614	0.486377	0.48304	0.86317	1.30207	0.218932	0.26515	0.357339
300	10	0.14856	19.3063	0.901491	1.06983	0.0685291	0.982347	0.72357	1.88717	2.75328	0.328341	0.39681	0.648288
400	10	0.19715	28.8595	1.937	2.29322	0.0906408	1.98271	0.95491	4.11647	5.48715	0.435343	0.52511	1.01982
450	10	0.21612	82.0646	3.52102	4.21025	0.0998514	3.37121	1.05691	7.2557	9.25452	0.480306	0.57846	1.2048
500	10	0.21646	309.855	3.81814	4.60919	0.100199	3.55895	1.06196	7.67778	9.9588	0.48205	0.58042	1.21226
550	10	0.21628	542.872	3.82803	4.62272	0.100233	3.5661	1.06249	7.69155	9.9747	0.48214	0.58048	1.21278

Table 12: Implementing Cache Disk Split

N	$threads	$thru	$rt	$uCGIApplication	$uDBDisk	$uDBP	$uDBS	$uProtocolStack	$uWSDisk	$uWSP	$uWebSP	$uWebServer_Cache	$uWebServer_Disk
1	1	0.0005	9.05235	0.00275359	0.00163	0.00244	0.00244	0.00400811	0.0010985	0.00145	0.00011199	7.84E-05	0.00105162
5	1	0.00249	9.47431	0.0137652	0.00817	0.01221	0.01221	0.0200396	0.0054911	0.00727	0.00055985	0.000391892	0.00525715
10	1	0.00498	10.0214	0.027523	0.01633	0.02442	0.02442	0.0400757	0.0109792	0.01453	0.00111937	0.000783562	0.0105117
50	1	0.02481	15.3475	0.137216	0.08142	0.12173	0.12173	0.200133	0.054742	0.07243	0.00558	0.003906	0.0524156
75	1	0.03713	19.8549	0.205292	0.12186	0.18218	0.18217	0.299786	0.081919	0.10838	0.00834773	0.00584341	0.0784278
100	1	0.04936	25.7581	0.272781	0.16198	0.24216	0.24215	0.398885	0.108885	0.14405	0.0110912	0.00776384	0.104221
125	1	0.06146	33.7792	0.339402	0.20164	0.30145	0.30143	0.497048	0.135534	0.17929	0.013799	0.00965928	0.129686
150	1	0.07334	45.232	0.404673	0.24057	0.35964	0.35961	0.593596	0.161678	0.21387	0.0164515	0.011516	0.15464
175	1	0.08487	62.1483	0.467812	0.2783	0.41604	0.41599	0.68739	0.187007	0.24736	0.0190169	0.0133119	0.178783
200	1	0.09568	90.3849	0.527611	0.3139	0.46929	0.46928	0.775173	0.210949	0.279	0.0214475	0.0150132	0.201636
300	1	0.11945	512.423	0.65719	0.39145	0.58526	0.58529	0.969021	0.263029	0.34785	0.0267111	0.0186978	0.251192
400	1	0.12219	1275.24	0.672027	0.40029	0.59848	0.59855	0.991395	0.268978	0.35573	0.0273131	0.0191192	0.256863
450	1	0.12261	1672.09	0.674299	0.40165	0.60052	0.60058	0.994817	0.269892	0.35693	0.0274054	0.0191838	0.257732
500	1	0.12287	2071.39	0.675729	0.4025	0.60179	0.60186	0.996976	0.270465	0.35769	0.0274634	0.0192244	0.258278
550	1	0.12305	2472.07	0.676694	0.40307	0.60265	0.60272	0.998432	0.270852	0.3582	0.0275025	0.0192518	0.258647
1	5	0.0005	9.05958	0.00275367	0.00163	0.00244	0.00244	0.00400821	0.0010985	0.00145	0.00011199	7.84E-05	0.0010517
5	5	0.00249	9.0637	0.0138514	0.00822	0.01221	0.01234	0.0201052	0.0054947	0.00727	0.00055996	0.000392082	0.00528309
10	5	0.00498	9.11939	0.0279936	0.01655	0.02442	0.02494	0.0406005	0.0109938	0.01453	0.00111985	0.000784339	0.0106161
50	5	0.02488	9.27937	0.147677	0.08718	0.1218	0.13526	0.209611	0.0551214	0.0726	0.00559461	0.00392739	0.055034
75	5	0.03732	9.74184	0.236441	0.13528	0.18233	0.21431	0.336356	0.0827337	0.10873	0.00837797	0.00588966	0.0842941
100	5	0.04974	10.2678	0.332988	0.187	0.24282	0.30221	0.471487	0.11053	0.14492	0.0111669	0.00786144	0.114848
125	5	0.06217	10.7437	0.439792	0.24245	0.3029	0.39964	0.619438	0.138348	0.18098	0.0139452	0.00983139	0.146523
150	5	0.07458	11.2469	0.55807	0.30213	0.36265	0.50785	0.781483	0.166231	0.21695	0.016717	0.0118023	0.179353
175	5	0.08699	11.7824	0.68926	0.36645	0.42198	0.62816	0.958909	0.194166	0.25282	0.0194812	0.0137733	0.213318
200	5	0.09938	12.3527	0.834651	0.43592	0.48086	0.76184	1.15329	0.222145	0.28858	0.0222369	0.015744	0.248402
300	5	0.14883	15.6206	1.60893	0.78897	0.71782	1.47348	2.18347	0.334504	0.43053	0.0332203	0.0236528	0.400656
400	5	0.19764	23.8137	2.81288	1.29216	0.93246	2.5861	3.64978	0.446023	0.56848	0.0437923	0.0313546	0.576544
450	5	0.21304	112.235	3.85494	1.74255	1.04413	3.59047	4.85782	0.490892	0.62166	0.0479025	0.0343712	0.654501
500	5	0.21824	291.037	4.01254	1.76664	1.04609	3.70401	4.99562	0.501823	0.63532	0.0489827	0.0351665	0.675425
550	5	0.21491	559.19	3.97642	1.79493	1.05386	3.71027	4.99754	0.495432	0.62711	0.0483158	0.0346753	0.662654
1	8	0.00049	23.1384	0.005609	0.00163	0.00244	0.00269	0.0102169	0.0010919	0.00144	0.0001113	7.79E-05	0.0010498
5	8	0.00248	14.8794	0.0138514	0.00822	0.01221	0.01234	0.0213147	0.0054934	0.00726	0.00055996	0.000392082	0.00528308
10	8	0.00498	9.12148	0.02803	0.01659	0.02442	0.02501	0.0406242	0.0109949	0.01453	0.00111986	0.000784432	0.010625
50	8	0.02488	9.29948	0.148993	0.08839	0.12189	0.13742	0.210577	0.0551599	0.07262	0.00559585	0.00393033	0.0552727
75	8	0.03732	9.82043	0.241138	0.13827	0.18254	0.21988	0.340964	0.0828406	0.10879	0.00838268	0.00589765	0.0848649
100	8	0.04974	10.429	0.342665	0.19282	0.24313	0.31323	0.481344	0.110696	0.145	0.0111727	0.00787389	0.115856
125	8	0.06216	10.9852	0.457144	0.2525	0.30338	0.41899	0.637166	0.138608	0.18109	0.0139541	0.00985068	0.148123
150	8	0.07457	11.5849	0.586479	0.31813	0.36333	0.53911	0.810513	0.166601	0.21711	0.016729	0.0118296	0.181715
175	8	0.08697	12.2328	0.73331	0.39055	0.42283	0.6761	1.00409	0.194649	0.253	0.0194949	0.0138087	0.217195
200	8	0.09935	12.947	0.899947	0.47079	0.48189	0.83234	1.22077	0.222755	0.28879	0.0222521	0.0157884	0.254416
300	8	0.14869	17.545	1.89928	0.91962	0.7213	1.75553	2.49225	0.336996	0.43206	0.0333054	0.0237919	0.424641
400	8	0.1975	25.2604	3.78023	1.73239	0.94943	3.55814	4.70895	0.450838	0.57164	0.0440623	0.0316872	0.630602
450	8	0.2175	68.8707	6.20032	2.77064	1.06221	5.90958	7.33427	0.503074	0.63369	0.0488263	0.0352192	0.739204
500	8	0.22139	258.538	6.81263	3.02324	1.07341	6.50068	8.00267	0.510337	0.6425	0.0494468	0.0356824	0.754937
550	8	0.21929	508.192	6.83513	3.05026	1.07575	6.53866	7.98927	0.508623	0.64011	0.0493233	0.0355888	0.751294
1	10	0.00049	36.161	0.00864994	0.00163	0.00244	0.00296	0.0163456	0.0010854	0.00144	0.00011062	7.74E-05	0.00104391
5	10	0.00248	21.1366	0.0138514	0.00822	0.01221	0.01234	0.0226316	0.005492	0.00726	0.00055996	0.000392082	0.00528307
10	10	0.00498	9.12218	0.0280424	0.01661	0.02442	0.02503	0.0406321	0.0109952	0.01453	0.00111987	0.000784464	0.010628
50	10	0.02488	9.30625	0.149453	0.08883	0.12192	0.13819	0.210906	0.0551738	0.07262	0.00559635	0.00393142	0.0553537
75	10	0.03732	9.8475	0.242825	0.13939	0.18263	0.22191	0.342574	0.0828813	0.10882	0.00838472	0.00590074	0.085062
100	10	0.04974	10.4857	0.346199	0.19503	0.24327	0.31733	0.484848	0.110758	0.14503	0.0111753	0.00787868	0.116203
125	10	0.06215	11.072	0.463622	0.2564	0.3036	0.42634	0.643759	0.138705	0.18114	0.0139579	0.009858	0.148858
150	10	0.07456	11.7115	0.59746	0.3245	0.36364	0.55136	0.822023	0.166736	0.21717	0.0167338	0.0118397	0.183322
175	10	0.08696	12.4144	0.750672	0.40043	0.42329	0.69534	1.02233	0.194844	0.2531	0.0195023	0.0138232	0.219572
200	10	0.09934	13.1894	0.926361	0.48552	0.48248	0.86149	1.24809	0.223011	0.28891	0.0222616	0.0158072	0.257709
300	10	0.14863	18.4099	2.02092	0.97811	0.72197	1.87856	2.6236	0.337418	0.43213	0.0333107	0.0238229	0.433764
400	10	0.1973	27.3429	4.28322	1.95287	0.9501	4.04989	5.23586	0.451227	0.57126	0.0440081	0.0316963	0.650127
450	10	0.21778	66.1858	7.77507	3.47101	1.06138	7.47882	8.94952	0.504007	0.63395	0.0488449	0.0352929	0.768031
500	10	0.22249	247.277	8.7648	3.87465	1.06916	8.43471	10.0296	0.511175	0.64285	0.0494452	0.0357435	0.784733
550	10	0.21846	517.731	8.79731	3.92556	1.0713	8.49703	9.98828	0.507208	0.6375	0.0491185	0.0354971	0.775404

Table 13: Implementing Protocol Stack Independent Processor

N	$threads	$thru	$rt	$uCGIApplication	$uDBDisk	$uDBP	$uDBS	$uProtocolStack	$uPSP	$uWSDisk	$uWSP	$uWebServer
1	1	0.0005	9.05235	0.00275359	0.00163	0.00244	0.00244	0.00400811	0.00012	0.0011	0.00144	0.00113002
5	1	0.00249	9.47431	0.0137652	0.00817	0.01221	0.01221	0.0200396	0.00062	0.00549	0.0072	0.00564905
10	1	0.00498	10.0214	0.027523	0.01633	0.02442	0.02442	0.0400757	0.00124	0.01098	0.0144	0.0112953
50	1	0.02481	15.3475	0.137216	0.08142	0.12173	0.12173	0.200133	0.0062	0.05474	0.0718	0.0563216
75	1	0.03713	19.8548	0.205291	0.12186	0.18218	0.18217	0.299789	0.00928	0.08192	0.10744	0.0842713
100	1	0.04936	25.7578	0.272778	0.16198	0.24216	0.24215	0.398896	0.01234	0.10888	0.1428	0.111985
125	1	0.06146	33.7783	0.339391	0.20164	0.30144	0.30142	0.497078	0.01537	0.13553	0.17772	0.139345
150	1	0.07334	45.2299	0.404644	0.24056	0.35962	0.35959	0.593665	0.01834	0.16167	0.21197	0.166153
175	1	0.08487	62.0168	0.467778	0.27829	0.41602	0.41597	0.687573	0.02122	0.18699	0.24514	0.192097
200	1	0.09567	90.7734	0.526676	0.31357	0.46875	0.46869	0.775655	0.02392	0.21066	0.27612	0.216309
300	1	0.11946	512.324	0.657032	0.39138	0.58514	0.58517	0.969429	0.02986	0.26297	0.34462	0.269868
400	1	0.1222	1274.97	0.671859	0.40021	0.59835	0.59841	0.991927	0.03055	0.26891	0.3524	0.275964
450	1	0.12262	1671.78	0.67413	0.40156	0.60039	0.60045	0.995363	0.03066	0.26982	0.35359	0.276897
500	1	0.12289	2071.01	0.67556	0.40241	0.60166	0.60172	0.997538	0.03072	0.27039	0.35434	0.277485
550	1	0.12307	2471.65	0.676525	0.40299	0.60252	0.60259	0.999004	0.03077	0.27078	0.35485	0.277882
1	5	0.0005	9.06	0.00275367	0.00163	0.00244	0.00244	0.00400821	0.00012	0.0011	0.00144	0.00113009
5	5	0.00249	9.05279	0.0138513	0.00822	0.01221	0.01234	0.0201216	0.00062	0.00549	0.0072	0.00572672
10	5	0.00498	9.11702	0.0279929	0.01655	0.02442	0.02494	0.040676	0.00124	0.01099	0.01441	0.0116068
50	5	0.02489	9.20769	0.147559	0.08712	0.12172	0.13516	0.21137	0.00622	0.05507	0.07192	0.0641333
75	5	0.03732	9.77205	0.236238	0.13518	0.18221	0.21415	0.340843	0.00933	0.08265	0.1077	0.101908
100	5	0.04974	10.3408	0.332589	0.18682	0.2426	0.30188	0.479678	0.01244	0.11038	0.1435	0.143694
125	5	0.06216	10.8605	0.439109	0.24214	0.30255	0.39908	0.632455	0.01554	0.13811	0.17914	0.189323
150	5	0.07457	11.4078	0.55699	0.30163	0.36214	0.50696	0.800462	0.01864	0.16588	0.21467	0.238885
175	5	0.08698	11.9881	0.687651	0.36571	0.42128	0.62681	0.984986	0.02174	0.19367	0.25007	0.29239
200	5	0.09937	12.6036	0.832356	0.43485	0.47992	0.75989	1.18762	0.02484	0.22149	0.28533	0.349852
300	5	0.14879	16.1791	1.6053	0.78732	0.71693	1.47041	2.25605	0.0372	0.33398	0.42631	0.623046
400	5	0.19747	25.5705	2.79164	1.28284	0.92925	2.56478	3.78569	0.04937	0.44346	0.56045	0.960465
450	5	0.21089	133.719	3.73454	1.69157	1.03382	3.47548	4.89717	0.05272	0.48512	0.61024	1.10802
500	5	0.21189	359.681	3.81157	1.72487	1.04035	3.55005	4.98546	0.05297	0.48785	0.61349	1.1181
550	5	0.21246	588.725	3.82521	1.72704	1.04054	3.56041	4.99885	0.05312	0.48895	0.61485	1.12231
1	8	0.00049	24.5439	0.0054913	0.00163	0.00244	0.00268	0.0100641	0.00012	0.00109	0.00143	0.00112776
5	8	0.00248	14.6274	0.0138513	0.00822	0.01221	0.01234	0.0212805	0.00062	0.00549	0.0072	0.00572671
10	8	0.00498	9.11704	0.0280295	0.01659	0.02442	0.02501	0.0407057	0.00124	0.01099	0.01441	0.0116354
50	8	0.02489	9.22074	0.148911	0.08835	0.12184	0.13735	0.212504	0.00622	0.05512	0.07196	0.0648983
75	8	0.03732	9.85324	0.240986	0.1382	0.18246	0.21976	0.345888	0.00933	0.08278	0.10779	0.10368
100	8	0.04974	10.5089	0.342362	0.19268	0.24297	0.31299	0.490342	0.01243	0.11058	0.14363	0.146879
125	8	0.06215	11.1132	0.456614	0.25226	0.30313	0.41856	0.651483	0.01554	0.13841	0.17934	0.194401
150	8	0.07456	11.7614	0.585624	0.31774	0.36296	0.53841	0.831411	0.01864	0.16632	0.21494	0.246361
175	8	0.08696	12.4585	0.73201	0.38995	0.42233	0.67502	1.03283	0.02174	0.19425	0.25041	0.303345
200	8	0.09934	13.2221	0.898048	0.4699	0.48121	0.83073	1.25866	0.02484	0.22223	0.28574	0.365299
300	8	0.14865	18.0992	1.89501	0.9177	0.72047	1.7516	2.57802	0.03716	0.33626	0.42775	0.670418
400	8	0.19741	26.1622	3.76764	1.72685	0.94835	3.54641	4.85977	0.04935	0.44979	0.56589	1.06665
450	8	0.21658	77.801	6.1052	2.73158	1.05972	5.8191	7.44083	0.05415	0.50038	0.62594	1.2756
500	8	0.21823	291.093	6.60528	2.94942	1.07095	6.31237	7.95858	0.05456	0.50537	0.63181	1.29739
550	8	0.21896	511.817	6.63463	2.96483	1.07213	6.33509	8.00965	0.05474	0.50581	0.6323	1.29918
1	10	0.00049	37.5829	0.0084597	0.00163	0.00244	0.00294	0.0160511	0.00012	0.00109	0.00142	0.00112159
5	10	0.00248	20.7315	0.0138513	0.00822	0.01221	0.01234	0.0225655	0.00062	0.00549	0.0072	0.00572671
10	10	0.00498	9.11705	0.0280419	0.01661	0.02442	0.02503	0.0407157	0.00124	0.01099	0.01441	0.011645
50	10	0.02489	9.22513	0.149385	0.0888	0.12188	0.13814	0.212889	0.00622	0.05514	0.07197	0.0651569
75	10	0.03732	9.88117	0.242695	0.13933	0.18256	0.22181	0.347645	0.00933	0.08282	0.10782	0.104284
100	10	0.04974	10.5679	0.345936	0.19491	0.24314	0.31712	0.494117	0.01243	0.11065	0.14368	0.147966
125	10	0.06215	11.2038	0.463156	0.25619	0.30339	0.42596	0.658513	0.01554	0.13853	0.17942	0.196319
150	10	0.07456	11.8934	0.596703	0.32415	0.36333	0.55075	0.843568	0.01864	0.16648	0.21505	0.249709
175	10	0.08695	12.647	0.74951	0.3999	0.42286	0.69438	1.05199	0.02174	0.19449	0.25057	0.308181
200	10	0.09933	13.4734	0.924652	0.48473	0.48191	0.86004	1.28718	0.02483	0.22254	0.28595	0.371847
300	10	0.14859	18.9856	2.01767	0.97629	0.72123	1.87508	2.7123	0.03715	0.33673	0.42793	0.687339
400	10	0.19721	28.2319	4.2697	1.94797	0.94945	4.03788	5.39698	0.0493	0.4501	0.56545	1.09998
450	10	0.21699	73.7903	7.72649	3.44993	1.06001	7.43205	9.10999	0.05425	0.50181	0.62678	1.32481
500	10	0.22072	265.362	8.54573	3.78557	1.06678	8.23325	10.012	0.05518	0.50771	0.63366	1.35184
550	10	0.21777	525.615	8.58155	3.83078	1.0692	8.28259	9.9811	0.05444	0.50495	0.63041	1.33916