The Effect of Workload Groupings on Distributed Transaction Capacity Models

Dr. Tim R. Norton
Simalytic Solutions, LLC
Colorado Springs, CO
email: tim.norton@simalytic.com
CMG 2002, Session 324

Abstract

Modeling transaction workload behavior with a queuing network model becomes much more difficult when the application consists of multiple workloads. One of the goals of this type of model is to predict how the application will behave when moved to a distributed environment. Add to the situation the goal of predicting the necessary capacity to support the application in the distributed environment and the model quickly becomes very complex. One way to reduce the complexity is to reduce the number of workloads being modeled, but the question then becomes how to combine the transactions into workloads. This paper is a case study of a hypothetical modeling situation where the transactions are grouped differently to determine if such workload groupings have a significant effect on the outcome of a capacity model and whether the effect is the same in a distributed environment. These hypothetical transactions were grouped into several different groups of workloads and each of the groups was studied to see how the groupings effected the response times after the transaction arrival rates were increased. This analysis was done on both a single system scenario and a networked systems scenario.

Keywords: workload, model, transaction, queuing, performance, response time, distributed, servers

Introduction

The behavior of a single transaction workload executing on a single system can be described with reasonable accuracy using any of several simple queuing network models. The modeling assumptions that the transactions are very similar and that the workload is homogeneous do not generally reduce the accuracy of the model if the transactions truly belong to a single workload. However, this task becomes much more difficult when the application consists of several workloads and one of the goals of the model is to predict how the application will behave when moved to a distributed environment. The assumption of homogeneity for each of the workloads becomes much more important and has more influence on the results. Add to the situation the goal of predicting the necessary capacity to support the application in the distributed environment and the model quickly becomes very complex.

Workloads can be constructed by grouping transactions together based on their function within the application. For example, the transactions that access a database might be divided into three groups: order entry, customer service and maintenance. But this increased business homogeneity may come at the expense of resource homogeneity and reduce the accuracy of the models. On the other hand, grouping transactions into workloads based on strict resource homogeneity may produce workloads that model well but do not bear the slightest resemblance to the real application. Grouping the transactions together by both application function and homogeneity may dramatically complicate the model by greatly increasing the number of workloads.

This paper is a case study of how these different workload groupings effect the results of such a modeling study, independent of the modeling tools and techniques used. Throughout this paper the term workload is assumed to mean a homogeneous grouping of transactions based on some objective criteria. It is the objective criteria that provide the homogeneous nature for the workload. If the criteria are function related, such as the transaction name or result, then it is a business workload. If the criteria are usage related, such as CPU time, disk accesses or server usage, then it is a resource workload. The most desirable situation is when the business workloads map directly to the resource workloads.

The goals of the modeling study were twofold. First, to investigate the effects of different groupings of transactions into workloads. Second, to investigate the effects of moving workloads from a single system with slow resources to a distributed environment with faster resources and network delays.

It was assumed that the workloads would grow over time and the results of the study needed to include the projected response times for both environments. In addition, the time period when any bottlenecks develop (e.g., saturated servers) should also be identified.

Overview

The modeling study was broken into several parts to insure the desired results could be achieved in a reasonable time. The transaction data was grouped several different ways, each referred to as a Work Group (WG). The Work Groups provide a higher level of abstraction when looking at the transaction data and therefore necessarily offer less homogeneity than the more detailed workloads. The criteria used for combining workloads into Work Groups can be complementary, contrary or indifferent to the criteria used for grouping transaction data into workloads. Unfortunately, in real world situations, there are usually components of all three of these forces. As stated above, the first objective of this study is to explore just how important it is to understand the relationship between workloads and Work Groups. If the results of the modeling experiments are indifferent to the various combinations of these groupings, then the model creation process can be greatly simplified by aggregating different transaction types indiscriminately. If the results of the modeling experiments are sensitive to the different combinations, then the degree of sensitivity can be used as guide to the importance of workload groupings.

	Tran Vol	CPU	Disk 1	Disk 2	Disk 3	Disk 4	Network
A??	High						
B??	Med						
C??	Low						
?1?		Low	High	High			
??2		Med	Med	Med			
?3?		High	Low	Low			
??a					Low	Low	None
??b					High	Low	Med
??c					High	High	High

Table 1 Transaction Profiles

Hypothetical Transaction Data

Data from hypothetical transactions was used to produce an exaggerated distinction between the different transactions. This was done to insure that the results from the study would represent a worse than normal situation.

Table 1 shows the transaction details for this study. The detailed transaction data was created using "resource units" for each transaction and these were multiplied by the service times for each server. This allows for greater flexibility in generation the numbers and creating additional scenarios as desired. Volume or resource usage was controlled by the transaction names. The blank cells indicate that that part of the name was not used to define the resource usage for the given resource. For example, transaction A1a exhibits high volume, low usage of the CPU, Disk 3 and Disk 4, high usage of Disk 1 and Disk 2, and no Network usage. On the other extreme, transaction C3c exhibits low volume, high usage of the CPU, Disk 3 and Disk 4, low usage of Disk 1 and Disk 2, and high Network usage.

Work Groups

Every transaction is in each Work Group. The only difference between the Work Groups is how the transactions are grouped together into workloads. Each of the Work Groups has either three or nine workloads. Although each Work Group has workloads with a different transaction mix, every transaction is assigned to a workload in each of the Work Groups. In Work Group 1 (WG1) the first letter of the transaction name is used for the grouping, so the workloads are sensitive only to the transaction volumes, which is similar to business groupings using only transaction volumes. Work Group 4 (WG4) uses only the last letter in the transaction name and focuses on the usage of disks 3 and 4 and the network while Work Group 5 (WG5) uses only the middle letter in the transaction name and focuses on the usage of the CPU and disks 1 and 2. Work Group 2 (WG2) and Work

	Workloads in Each Work Group								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
WG1	A??	B??	C??						
WG2	A1?	A2?	A3?	B1?	B2?	B3?	C1?	C2?	C3?
WG3	A?a	A?b	A?c	B?a	B?b	B?c	C?a	C?b	C?c
WG4	??a	??b	??c						
WG5	?1?	?2?	?3?						

Table 2 Transaction Names by Work Group

Group 3 (WG3) are more complex groupings to have better insight into the impact of each specific resource but that additional complexity requires the Work Groups to have more workloads.
Because the data was randomly generated for the study there was no business relationship between the transactions. Therefore, the names of the transactions were used to group the transactions into the different workloads. Table 2 shows which transaction names were included in each of the workloads for the different Work Groups.

Two Scenarios

Two different scenarios were modeled in the study. The first had only servers representing the CPU and four disk components of a computer system. Because there were no network delays, this scenario is used to represent the single computer system.

Server	Scenario #1	Scenario #2
CPU	0.0200	0.0010
Disk 1	0.0120	0.0120
Disk 2	0.0230	0.0100
Disk 3	0.0350	0.0030
Disk 4	0.0410	0.0040
Network	0.0000	0.2800

Table 3 Service Times

The other scenario reduced the service times for the CPU and three of the disks, but added a network server. The network was modeled as a load independent server with a queue to more accurately represent the idea that there are multiple systems in the distributed environment (see Figure 1).

The Scenario Service Times

The service time for each of the scenarios is shown in Table 3. Disk 1 was kept at the same service time to represent the disk local to the workstation generating the transactions.

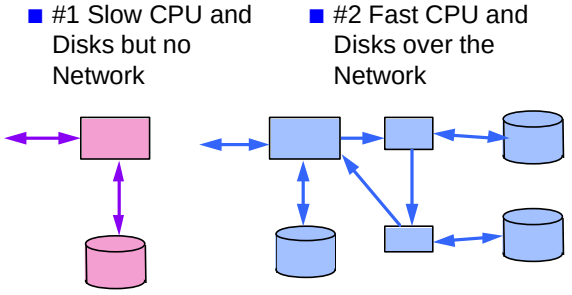

Figure 1 The Scenarios

The Model Used

An analytic queuing theory model was chosen because it could provide very rapid results for a large number of cases. In addition, workload analysis is much more critical when using queuing models because of the required aggregation of transactions into groups that are assumed to be homogeneous. A simulation model could simulate every individual transaction and avoid the problem but at the cost of greatly increased model execution time, resource usage and complexity.

Open Queuing Network Model

The model used was an open queuing network model to allow the number of transactions in the system to rise as the arrival rate begins to exceed the server's ability to process the transactions. This condition will continue until the server becomes totally saturated at 100% busy and the model will no longer product results. The time period when these happen is identified as when the system becomes bottlenecked.

Multiple classes

Queuing network models use the term *class* to describe work with similar attributes and features. The modeling tool supports multiple classes and each workload was modeled as a different class. There is a direct correspondence between classes and workloads. (Menascé 1994, 89-90)

Load independent servers

Although the modeling tool could model load dependent servers, this study uses only load independent servers. This is generally accepted for modeling CPU and disk devices because these servers do not change service times under load. The network is less well represented by this type of server, but load dependent nature of the network itself is assumed to be a minor factor in this study.

Program OPENQN.EXE

The modeling tool used for this study was the program OPENQN.EXE that is included with the book *Capacity Planning and Performance Modeling: from Mainframes to Client-Server Systems* (Menascé 1994) (refer to Appendix A - The Modeling Tool: OPENQN.EXE for sample input and output files). Details of the modeling technique used by OPENQN, complete with source code, are included in the book and will not be presented here. (Menascé 1994)

Baseline

The baseline was created from the original transaction data before any growth was applied to the arrival rates (see Figure 2). Because the data was not from an actual system, the baseline could not be calibrated to insure that it is a valid model. Therefore, the baseline is assumed to be correct and all of the results are relative to the baseline. Although not acceptable for real-life modeling situations, this approach is adequate for this study because it is focused on the relative changes between models rather than absolute accuracy.

Growth

The growth was applied in four new periods that could represent any time-period meaningful to the application. Each transaction had different growth applied and the workload growth depends on the mix of transactions in that workload.

The overall growth for the different workloads in the different Work Groups ranged from almost zero to over 50%. Table 4 shows the growth by workload for each Work Group expressed as the ratio of the fourth period to the baseline (P4/B). The actual growth applied to the transactions was weighted toward the transactions with names starting with "A" followed by "B" and "C" grew the least. Table 4 clearly shows that those work groups having workloads that grouped all of the "A" transactions together had the highest growth whereas the work groups that used a different grouping had a much more even growth across all of the workloads in those work groups.

	WG1 X??	WG2 X9?	WG3 X?x	WG4 ??x	WG5 ?9?
Workload 1	1.65	1.66	1.64	1.41	1.43
Workload 2	1.12	1.64	1.63	1.34	1.38
Workload 3	1.02	1.67	1.68	1.48	1.43
Workload 4		1.11	1.11		
Workload 5		1.12	1.14		
Workload 6		1.12	1.08		
Workload 7		1.02	1.02		
Workload 8		1.01	1.01		
Workload 9		1.02	1.03		

Table 4 Workload Growth by Work Group

Results

Two conclusions can be drawn from the results of this study, the first is related to the effects of queuing in a distributed environment and the second related to the workload groupings.

Queuing Effects

The queuing effects are more pronounced with slower local servers than with faster distributed servers. Even with a very large network service time, the faster CPU and disks on the network were able to absorb the higher arrival rates for the growth of the different workloads. While this was only a secondary objective of this study, it is still gratifying to see additional empirical evidence that a large server in a distributed implementation can compensate for the additional network delays.

The "knee of the curve" is very evident in the growth response time charts for scenario #2 (see

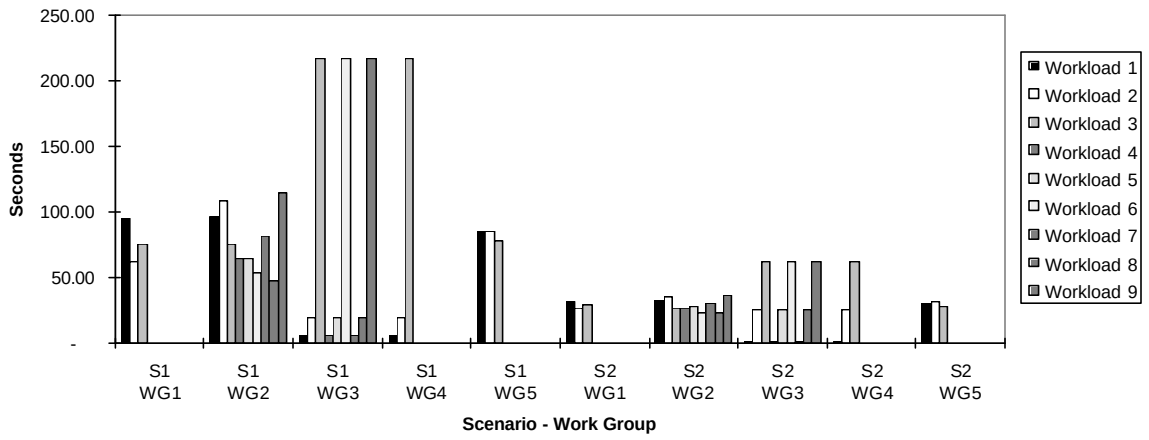

Figure 2 Baseline Response Times

Appendix B - Response Time Results After Growth). The growth curves for scenario #1 also exhibit this, but only the chart for WG5 actually shows it. This is because a device saturated at a lower arrival rate than represented by period 2 and results could not be generated for the other Work Groups. WG5 shows the extremely high response times associated with a server approaching saturation. The response times for scenario #2 show that eventually the system will reach a point where it will saturate.

A key assumption of this study was that it was reasonable to reduce the overall CPU service time for all servers across the network for all transactions. This is the main difference between the scenarios and it is the type of assumption referred to as a 'simplifying assumption' (because it makes creating models easier). This is often the type of business assumption that is made because there is not adequate measurement data to truly understand what would be more accurate (and there is seldom enough time to collect the necessary data). However, if this assumption holds and adding additional CPU's would further reduce the overall CPU service time, then this modeling study shows that additional servers can be added to the environment to reduce bottlenecks as the arrival rate increases.

Work Groups

Grouping has a major impact on how the workloads behave. When the workloads in a Work Group contain transactions that are either directly or indirectly sensitive to the bottleneck resource, then there is a high degree of variability in the response times. Work Group 3 (Figure 7) and Work Group 4 (Figure 8) show this characteristic. The workload response times are either 'good' or 'bad' but not in-between. When workloads are grouped in this way, the groupings are more related to how the resources are used. The transactions most sensitive to the resource are most impacted when queues develop for that resource, causing a wider difference between the high and low resource users.

If the transactions are indifferent to that resource (which means there are more sensitive to one of the other resources), then there is less difference in the responsiveness of the workloads until the knee of the curve, after which there is a more uniform distribution of results. Work Group 1 (Figure 5) and Work Group 2 (Figure 6) show this characteristic. The workload response times range from 'good' or 'bad' fairly evenly. Because each transaction reacts differently to the building queues at the bottleneck resource, the response times don't show the marked bifurcation seen in the other groupings.

A difficulty arises from the fact that transaction workloads are seldom neat. What makes sense from the business standpoint may not make sense from a resource usage standpoint and vice versa. In addition, one transaction may be very sensitive to queuing at one resource but not at another while a different transaction is just the opposite. This relationship can be seen as relatively straightforward when there are only two resources and two types of transactions. Unfortunately, modern distributed environments are seldom so simple and the interdependencies may mean that the only way to get accurate results is to treat each transaction type as a unique workload! The analysis in this overly simple study was difficult enough using nine workloads for WG2 and WG3. Making every transaction type a workload would have meant dealing with 27 different workloads; clearly an unworkable situation.

This study also points out one of the most difficult problems with workload analysis: the groups must reflect growth as well as initial resource usage. If all of the workloads in a Work Group do not have the same growth sensitivity then the Work Group will grow at an incorrect rate (e.g. WG2 and WG3). Just because two transactions have similar resource usage and volume in the baseline measurements does not mean that the business drivers for them will cause even growth. Grouping them into the same workload is making the assumption that they have the same growth projections because growth in a model is applied to workloads, not transactions. The question then is whether it is 'better' to have a model that validates well (the baseline response times match the actual response times quite closely) or one the predicts well (the future response times show the business impact of growth in each transaction type). This may not be an easy dilemma to resolve.

Conclusion

"Different behavior in different environments produces different results" is an obvious statement but it shows the need for better understanding about how to group work and what to use for the Work Group drivers. Measuring individual transactions is only the first step in understanding how a total system of interconnected servers and applications behave. Changing the assumptions about the relationships between individual transactions can have a profound impact on the outcome of the analysis stage of a performance study. Grouping transac-

tions by current resource usage may lead to incorrect predictions about future business impacts. While this study just begins to explore this complex issue, it is clear that all performance modeling activities should include questioning the assumptions about workload groupings and experimentation to determine how sensitive the model is to different groupings.

References

Menascé, D., V. Almeida, and L. Dowdy. 1994. Capacity Planning and Performance Modeling: from mainframes to client-server systems. Englewood Cliffs, New Jersey: Prentice Hall.

Appendix A - The Modeling Tool: OPENQN.EXE

```
6  3  devices(CPU,D1,D2,D3,D4,Network)
workloads(W1,W2,W3)
  0.047 0.021 0.013  Vector_N
  0 0   Device 1 type (LI): CPU
  0 0   Device 2 type (LI): Disk1
  0 0   Device 3 type (LI): Disk2
  0 0   Device 1 type (LI): Disk3
  0 0   Device 1 type (LI): Disk4
  0 0   Device 1 type (LI): Network
>>>> Service Demand Matrix
  0.774      0.729     0.872
  0.486      0.506     0.382
  0.991      1.013     0.899
  5.301      5.355     5.549
 12.227      7.403     9.424
  0          0         0
```

Figure 3: Sample OPENQN.EXE input file

```
OpenQN - (c) Copr. 1994 D. Menasce', V. Almeida,
and L. Dowdy.
            All Rights Reserved.
This program comes with the book 'Capacity Planning
and
Performance Modeling: from mainframes to client-
server systems'
by Menasce, Almeida, and Dowdy, published by
Prentice Hall.

>>>> Class  1 Throughput:   0.047000
>>>> Class  2 Throughput:   0.021000
>>>> Class  3 Throughput:   0.013000

>>>> Utilization of Device  1 :      6.302 %
>>>> Utilization of Device  2 :      3.843 %
>>>> Utilization of Device  3 :      7.954 %
>>>> Utilization of Device  4 :     43.374 %
>>>> Utilization of Device  5 :     85.264 %
>>>> Utilization of Device  6 :      0.000 %

Class  1 metrics:

>>>> Device Residence Times:

     Device  1 :   0.826061
     Device  2 :   0.505426
     Device  3 :   1.076632
     Device  4 :   9.361408
     Device  5 :  82.975922
     Device  6 :   0.000000
>>>> Class  1 Response Time........:  94.745448
>>>> Class  1 Avg. Number in System:   4.453036

Class  2 metrics:

>>>> Device Residence Times:
     Device  1 :   0.778034
     Device  2 :   0.526225
     Device  3 :   1.100533
     Device  4 :   9.456770
     Device  5 :  50.238877
     Device  6 :   0.000000
>>>> Class  2 Response Time........:  62.100439
>>>> Class  2 Avg. Number in System:   1.304109

Class  3 metrics:

>>>> Device Residence Times:
     Device  1 :   0.930653
     Device  2 :   0.397269
     Device  3 :   0.976682
     Device  4 :   9.799368
     Device  5 :  63.953962
     Device  6 :   0.000000

>>>> Class  3 Response Time........:  76.057933
>>>> Class  3 Avg. Number in System:   0.988753

>>>> Press Enter
```

Figure 4: Sample OPENQN.EXE output file

Appendix B - Response Time Results After Growth

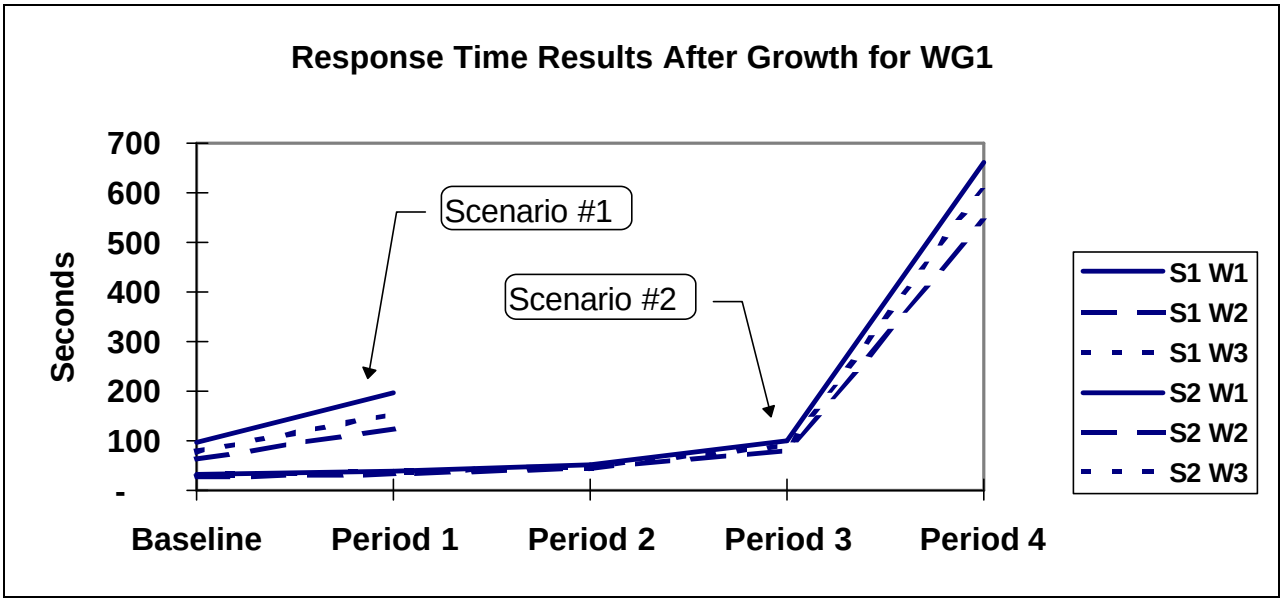

Figure 5 WG1 Results

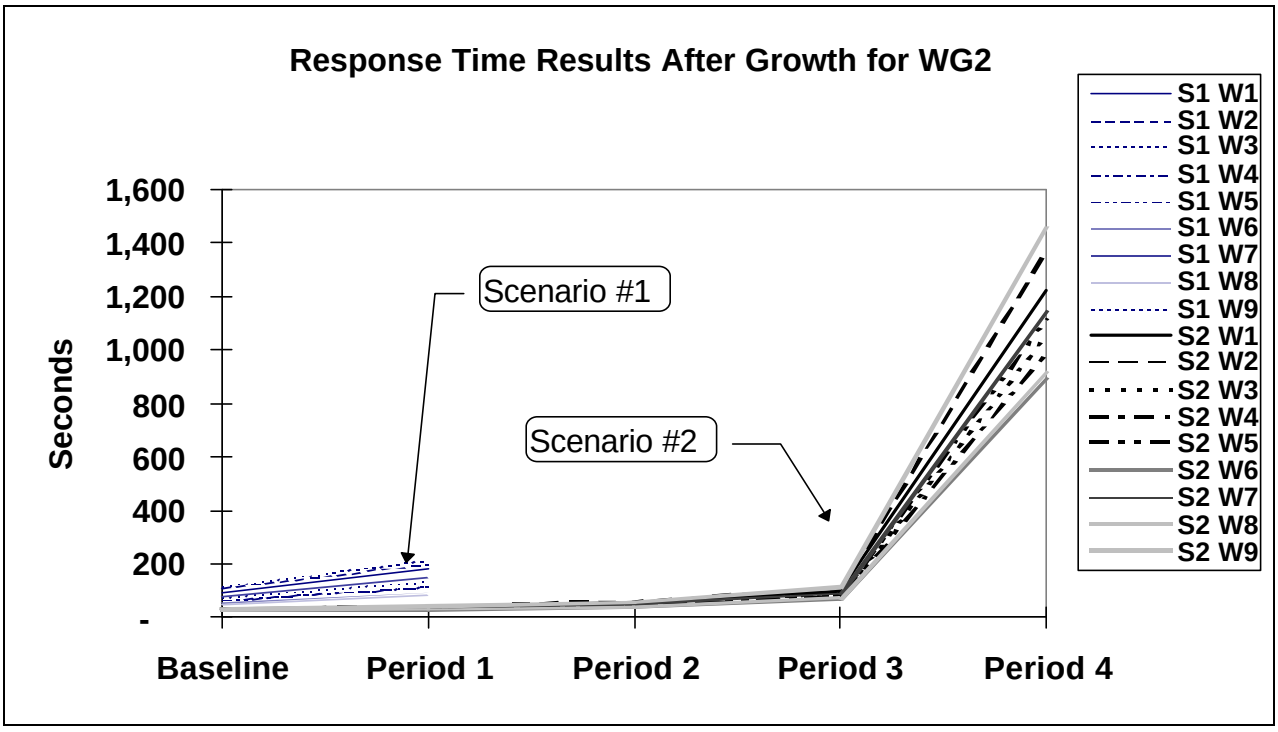

Figure 6 WG2 Results

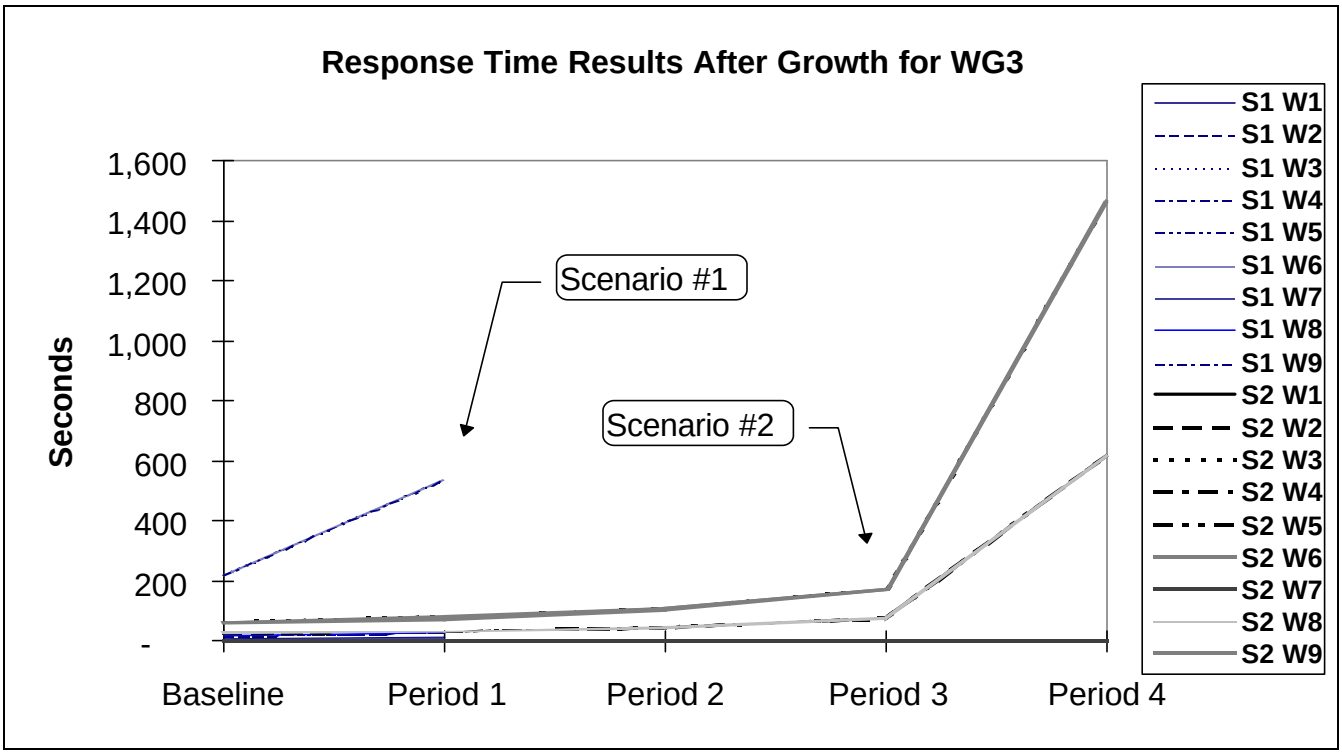

Figure 7 WG3 Results

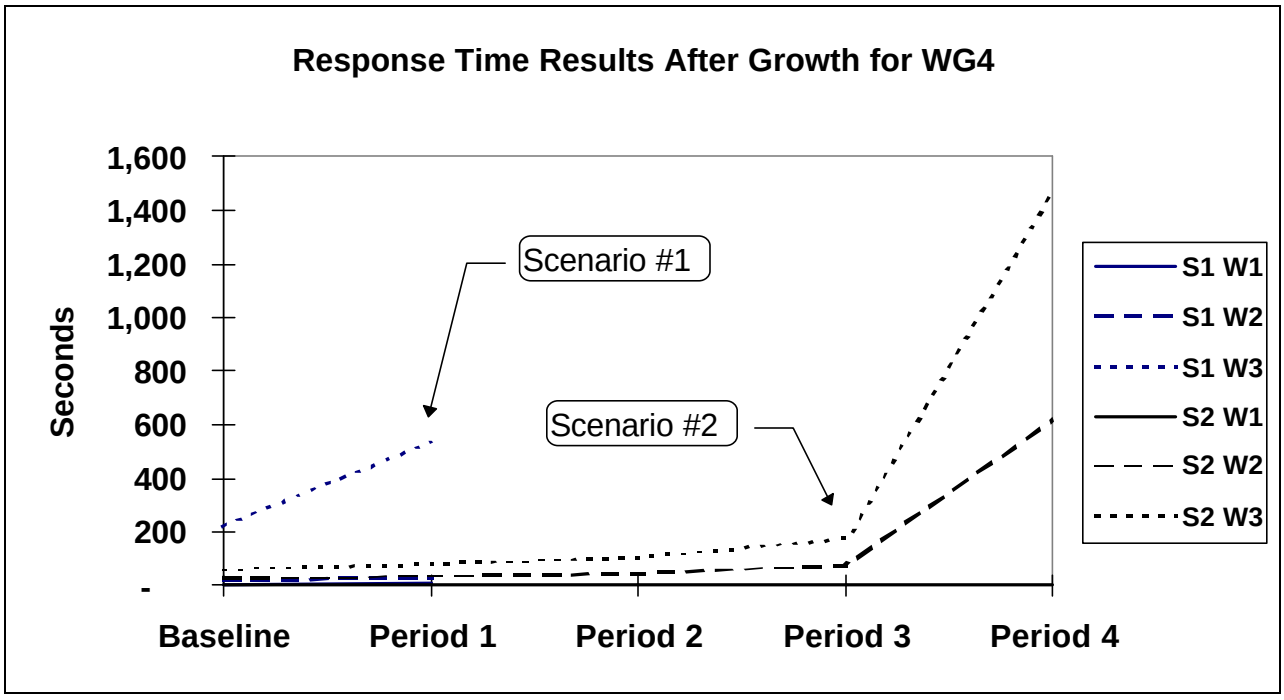

Figure 8 WG4 Results

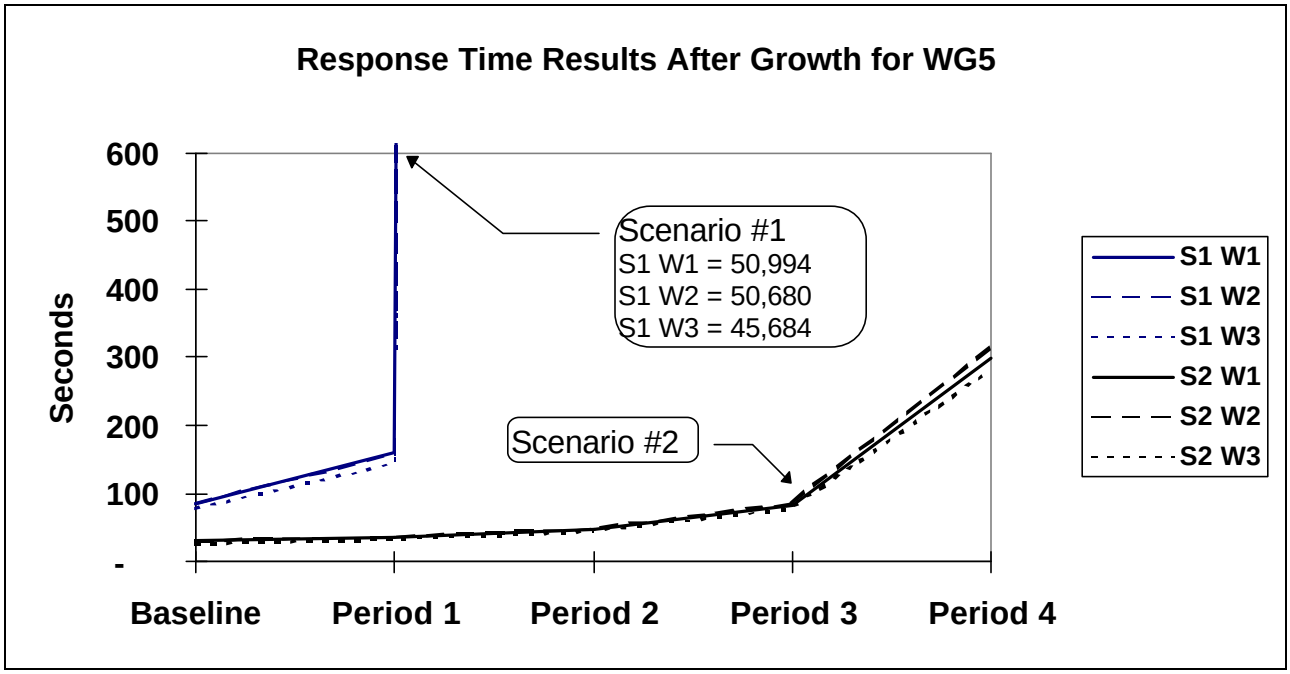

Figure 9 WG5 Results