MobileClean Technologies

SUDS (A new service built on SOAP/HTTP)

Performance Characterization Report

Bernie Velivis

Performax Inc.
2 Colburn Lane, Hollis, N.H. 03049
Bernie.Velivis@iperformax.com
(603) 860-7900

May 19, 2006
Version 1.0

This is a sample report is based on a recent Performax performance characterization project. The client's identity, product name, and performance statistics have been altered to protect privacy. Performance statistics may appear inconstant to the trained observer but this was inevitable once the numbers were changed to obscure confidential data.

Table of Contents

Executive Summary

This report documents the performance characteristics of the SUDS, a fictitious fee based service for managing delivery vehicle routing. It includes results and analysis of load tests executed to determine application stability, scalability, failover characteristics, and capacity planning guidance.

Stability Under Expected Load

A load test was run to simulate a full day of activity where 200 subscribers from 4 time zones exercised the system for 13 hours. The application provided consistently good performance and showed no signs of memory leaks or instability.

Scalability

The initial load on the production servers is expected to be 50 subscribers growing to 200 in the near future. Scalability tests indicate that if hardware is reallocated according to our recommendations[1] then the production cluster is capable of supporting 270 active subscribers and would provide slightly degraded performance in the event of loosing any one server.

The scalability of the cluster is limited by the capacity of the database tier. Performance models estimate that the current database server, a ProLiant DL360, should scale to 1,800 active subscribers. Scaling to this level would require a detailed IO analysis, and potentially more disk IO bandwidth and a version of SQL server that can map more then 2GB of memory as well as additional server capacity at the application, map/geocode, and mobile communications tiers. Guidance on server selection appears in the Capacity Planning section of this document.

Failover Behavior

While the full cluster is capable of providing consistently good performance to 270 active subscribers, the worst case scenario of losing one application server would allow the cluster to provide slightly degraded performance to approximately 175 subscribers.

One benefit of the application's design is that relatively few requests are in flight at any given time, which minimizes the impact of server or service failing. Failover tests were run with a load of 200 subscribers. A series of server failures were initiated and application errors and response times were monitored to measure message loss, duration of the recovery, and application performance during the loss and restoration of servers. The production cluster performed very well during failover tests. Under the worst case scenario where a database server failed, the system was unavailable for 71 seconds during the failover. Other tests where individual map, web service, or mobile communication servers were rebooted showed minimal message loss and minor impact to the performance of transactions being serviced by the surviving servers. Detailed results can be found in the Failover Characteristics section of this document.

Efficiency

SUDS is very efficient at client to server network utilization and CPU utilization at the database tier. This should be viewed as validation of the decision to use SOAP over HTTP to make infrequent yet resource intensive requests to the application servers. Single user tests were run to facilitate future regression tests.

[1] The application tier has the greatest CPU demand. We recommend recasting the map servers as applications servers and vice versa. This will increase overall capacity by 63%.

Project Methodology and Approach

The goals of the SUDS performance characterization project were to:

1. Measure production cluster responsiveness and stability under normal operating conditions.
2. Determine how capacity varied with respect to the number and speed of CPUs in order to build a capacity planning model.
3. Discover how response times vary with respect to load to identify bottlenecks and set expectations on service levels.
4. Explore the cluster's failover behavior
5. Document single user response time for all transactions to form a baseline for subsequent regression analysis.

Workload Definition

The performance characterization process starts with defining a workload. This definition includes a list of transactions, details on how the transaction will be emulated, arrival rate of each transaction, and data demographics which affect the scope and performance of the transaction. Errors or omissions in the workload definition can diminish the return on investment of performance characterization projects. The workload definition for this project was created with input from both marketing and engineering and was the result of a comprehensive process facilitated by Performax.

SUDS subscribers will have two distinct user types. Dispatchers are typically busy during delivery hours. Planners tend to perform most of their work directly following normal delivery hours. The two user types exercise different functions within the application and have unique work habits, requiring a separate workload description for each. The two workloads were codified in OpenSTA scripts and mixed in different ways according to the goals of the various test scenarios.

Dispatcher Workload Description and UI Narrative

Transaction	*UI Narrative*	*Rate / Hour / Subscriber*
Dispatcher Transactions		
Acknowledge Exception	▪ Select active exceptions tab ▪ Click '+' to open route ▪ RightMouse first exception and select acknowledge ▪ Click OK to confirm acknowledgement and record response time ACKEXCEPTION	13
Summary View	▪ Select summary tab ▪ Menu ACTION->Refresh and record response time SUMMARYVIEW	8.5
Show Exception	▪ Select active exceptions tab ▪ Click '+' to open route ▪ RightMouse on first exception ▪ Select SHOW and record response time SHOWEXCEPTION ▪ Timer MAPSERVER records map/geocode server response time	7
Map View	▪ Select all routes ▪ Click map tab and record response time MAPROUTEDISPATCHER ▪ Press ZOOM icon ▪ Timer MAPSERVER records map/geocode server response time	8.5

Planner user type Workload Description and UI Narrative

Transaction	*UI Narrative*	*Rate / Hour / Subscriber*
Planners Transactions		
Order Import	▪ Menu Action->ImportUnassignedOrders ▪ Specify order input file containing 300 orders ▪ Press IMPORT button and record response time IMPORT300ORDERS	1
Route Unassigned Orders Method 1	▪ Note: User configuration "routing technique" set to Standard ▪ Menu Action - > Route All Unassigned Orders ▪ Specify date and route set in dialog box ▪ Click OK and record response time ROUTEUNASSIGNEDSTD	1
Route Unassigned Orders Method 2	▪ Note: User configuration "routing technique" set to Dynamic ▪ Menu Action - > Route All Unassigned Orders ▪ Click OK and record response time ROUTEDYNAMIC	1
Route to Route Move	▪ Select routes tab ▪ RightMouse TESTROUTE1, select 'explore left' and record response time EXPLORELEFT ▪ RightMouse TESTROUTE2, select 'explore right' and record response time EXPLORERIGHT ▪ Drag and drop the last stop from one pane to the other (alternating with each iteration) ▪ When prompted to confirm, click YES and record response time RTE2RTEMOVE	5
Route Re-sequence	▪ Select routes tab ▪ RightMouse TESTROUTE1, select 'explore left' and record response time EXPLORELEFT ▪ Click on first stop in left route and then drag and drop on bottom of left pane to re-sequence to the last route ▪ When prompted to confirm, click YES and record response time RESEQROUTE	5
Map Route	▪ Select all routes ▪ Click map tab ▪ Record response time MAPROUTESPLANNER USER TYPE ▪ Timer MAPSERVER records map/geocode server response time	5
Location Add	▪ Menu List location ▪ Click FIND button and record response time FINDLOCATIONS ▪ Select an unedited line and modify the description ▪ Click SAVE and record response time EDITLOCATIONSAVE ▪ NOTE: The script 0's out the lat/long in the call to the web service to force the geo-coder to recalculate the lat/long.	2

Java Phone Workload

Another source of load comes from Java Phones carried in the delivery vehicles. As vehicles leave the Laundromat there is a burst of message traffic related to starting the route. As the vehicle progresses along the route, the phone will send various messages that correspond to planned pickups or pre-defined exceptions such as being off the planned route or being stuck in traffic. The message rate and load on the servers is at its peak when the route is started.

An emulator was developed to simulate the Java phone workload. During the full day test, the emulator simulated message traffic associated with route startup, followed by periodic messages related to stops and exceptions. During tests where load was applied in stages, only the initial startup messages were emulated. This was necessary since each load step lasted 20 minutes which was smaller than the time between typical route events. The net effect is that workload used to size the mobile communications servers is conservative, but not so much as to be considered pathological. Ten java phones were simulated for each active subscriber.

Workload Mix

To measure cluster performance characteristics under normal operating conditions, a full work day was simulated where 200 users from 4 time zones exercised the system for 13 hours. Dispatcher user type scripts were run for the first 8 hours of the simulated day after which Planner user type scripts were run for two hours. The java phone emulator was used to simulate realistic message traffic throughout the simulated day. A description of the test and results can be found in the Full Day Simulation.

All other tests, with the exception of single user testing, were run with a mix of 75% Dispatchers and 25% Planners. The goal was to load the system to simulate peak demand conditions. The java phone emulator was used to simulate message traffic from a vehicle just starting its route and there were 10 java phones simulated for every active subscriber.

Single user tests consisted of 50 iterations of each transaction. The java phone emulator was inactive during single user tests.

Testing Methodology

Performax uses a unique combination of trend analysis and modeling to measure and predict the capacity of servers based on the speed and number of CPUs. Our modeling methodology is described in the capacity planning section of this document.

The trend analysis technique is straightforward. Once the workload has been defined and codified in scripts, a load is applied in a step wise fashion to various hardware configurations so that trends in response time, throughput, and hardware resource utilization can be measured and analyzed. Each logical tier of the architecture was exercised with one, two, and (in some cases) 4 CPUs to determine how capacity changed as a function of the number of CPUs. In all cases there was sufficient memory, disk, and network capacity such that CPU was the primary bottleneck.

To allow the system to settle into a steady state and achieve consistent and repeatable results, the load tests had to run for at least two hours. Each test was logically partitioned into four 30 minute periods. Each period consisted of a ramp-up lasting 10 minutes where 25% of the maximum workload was added followed by a 20 minute data collection window. An average value was produced for each test metric (i.e. response times, throughput, and operating performance statistics by server) over each of the twenty minute intervals. Data collected during the ramp up period was discarded. For example, a 200 user test would be executed as follows:

Restore database and restart database service

From 1 to 10 minutes, log in users 1-50
From 11 to 30 minutes, observe behavior of 50 users working in parallel
From 31 to 40 minutes, log in users 51-100
From 41 to 60 minutes, observe behavior of 100 users working in parallel
From 61 to 70 minutes, log in users 101-150
From 71 to 90 minutes, observe behavior of 150 users working in parallel
From 91 to 100 minutes, log in users 150-200
From 101 to 120 minutes, observe behavior of 200 users working in parallel

The same technique was used by the java phone emulator.

OpenSTA and Load Testing SUDS

OpenSTA (www.opensta.org) was selected as the load testing tool for this project after a proof of concept was executed to determine its feasibility. OpenSTA has native support for XML documents which makes it a good candidate for emulating SOAP/HTTP clients. However, a basic design feature of OpenSTA is that to create scripts, it monitors the HTTP traffic between client and server by using a gateway which the HTTP client is directed to use. This gateway can only monitor and reverse engineer messages sent in clear text. The gateway has a special mode which allows it to handle HTTPS. It does so by having the client speak to the gateway in clear unencrypted text. A special character "{" is inserted at the start of the URL which instructs the gateway to communicate with the server using HTTPS.

SUDS communications present two challenges to OpenSTA. The first is that the application receives very large XML messages from the servers. To compensate, BZIP2/BASE64 is used to compress large portions of the SOAP payload. OpenSTA did not have the capability to decompress the encoded messages. A decision was made to extend OpenSTA so that it could encode and decode compressed messages coming from the server. Messages sent from the client to the server were not compressed as it would have added considerable time to the scripting with no significant return on investment.

Another complication was that SOAP messages were encrypted for security reasons. This made it impossible to modify the content of the messages captured by the OpenSTA gateway. The decision was made to disable security for all recording and testing and to conduct a series of manual tests with and without encryption to determine what corrections would be made to the capacity planning recommendations. Testing revealed that the CPU overhead increased by 7% on the application and map/geocode servers when with encryption enabled. The capacity planning heuristics and related server capacity planning tables take this into account. All graphs and reports of CPU utilization from tests run during this project are presented here unaltered and should be viewed with the knowledge that CPU utilization for the application server and map servers would be 7% higher in the real world.

Modeling

Tests are designed to capture information necessary to build a simple model that predicts capacity as a function of the speed and number of processors. The assumption is that there is adequate memory, disk bandwidth, and network bandwidth such that the processor is the limiting factor to capacity. This was the case in all tests executed during this project.

The model is based on an industry standard CPU benchmark run on most popular servers. The organization that defines the tests, ensures the integrity of testing, and distributes results is www.spec.org. The benchmark statistics used is from the Spec Integer 2000 benchmark and the particular metric is Spec Integer 2000 base mark. The reason for using this metric are;

1. We have found a very high correlation with CPU bound commercial application capacity and this metric.
2. As new servers are introduced (sometimes even before they are commercially available), manufactures will test and publish results from this benchmark allowing us to update capacity planning recommendations for SUDS without any additional testing.

Modeling heuristics and formulas are discussed at length in the modeling methodology section of this document.

Test results

Production Cluster Hardware

All tests were conducted on the SUDS production cluster. It consists of the following ProLiant/Intel servers running Windows 2003/SP1:

- Two DB Servers: DL360 G3, dual 3.0 GHz / 2MB L2 cache, Xeon (clustered, one active, one standby), 4 GB ram, small HP fibre channel disk array dual ported between servers
- Two Map/Geocode servers: DL360 G3, dual 3.0 GHz / 2MB L2 cache, Xeon, 2 GB ram, locally attached SCSI array.
- Two Application (Web Service) Servers: DL360 G4p quad 2.8GHz (dual-core) Xeon 2MB L2 (total of 4 CPUs per server), 4 GB ram, locally attached SCSI array.
- Two Mobile Communication Servers: ProLiant DL360 G3 single 3.0GHz Xeon, 2MB L2 cache, 2 GB ram, locally attached SCSI array.
- One OpenSTA server: ProLiant DL360, single 3Ghz CPU, 2GB ram, Win 2003/SP1, OpenSTA 1.4.1 (in production, this server will function as a matrix build server)

Network: Dual Gb Ethernet LAN used for all tests except the Single user test which, in addition to the gigabit Ethernet, was load tested remotely over a 1.6 Mb cable modem which approximates T1 performance.

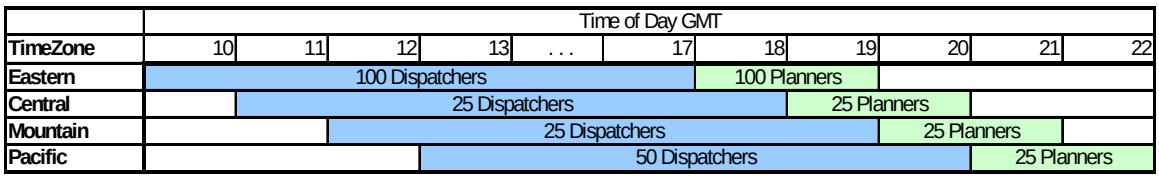
Full Day Simulation

A load test was run to simulate a full day of activity where 200 subscribers from 4 time zones exercised the system for 13 hours. The application provided consistently good performance and showed no signs of memory leaks or instability. There was one error logged over the 13 hour period where a call to the application server timed out (OpenSTA timeout was set to 2 minutes).

Workload assumptions:

- A total of 200 subscribers[2] across 4 time zones were simulated (see workload description section of this document for workload details)
- The java phone simulator was programmed to simulate 2000 trucks starting their routes between 5 and 8 am Eastern Time with start times biased towards the top of the hour and on the half hour.
- Subscribers distributed across time zones as 50% Eastern, 12.5% Central, 12.5% Mountain, and 25% Pacific.
- Subscribers began their work day at 8am local and performed Dispatcher functions for 8 hours followed by Planner user type functions for two hours.

The workload over time can be visualized as;

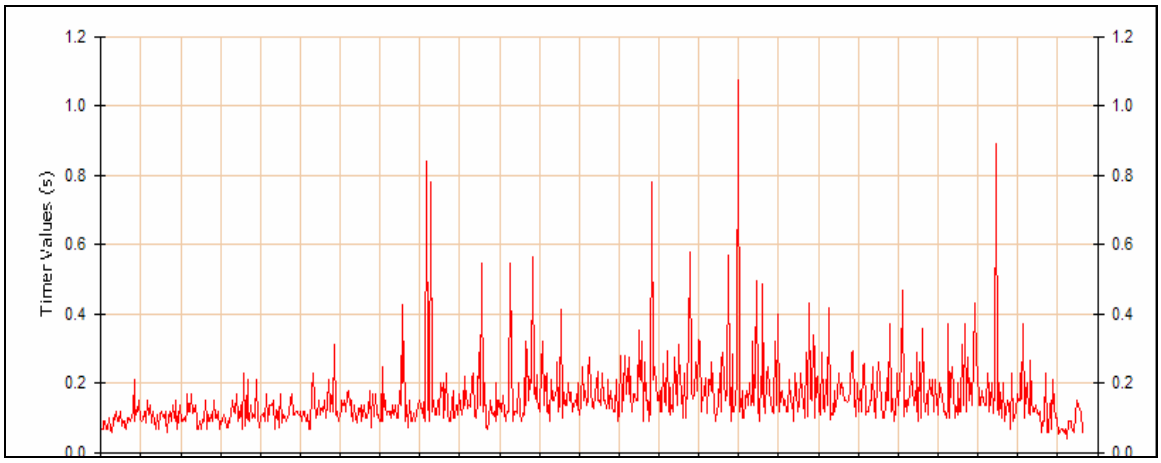


[2] See workload description section to see tasks performed and work rate

There are four response times of particular interest to view over time when analyzing the stability and responsiveness of the cluster. Each occurs frequently and is sensitive to the performance of a different server. “Acknowledge Exception” is sensitive to application (web services) server performance and is executed frequently by Dispatchers. “Import orders” is sensitive to application server performance and is executed frequently by Planners. “Mapserver” times calls to the map/geocode servers and is executed frequently by both Planners and Dispatchers. PHONE is the timer name used to record response times measured by the Java phone simulator and is sensitive to the performance of the mobile communication servers.

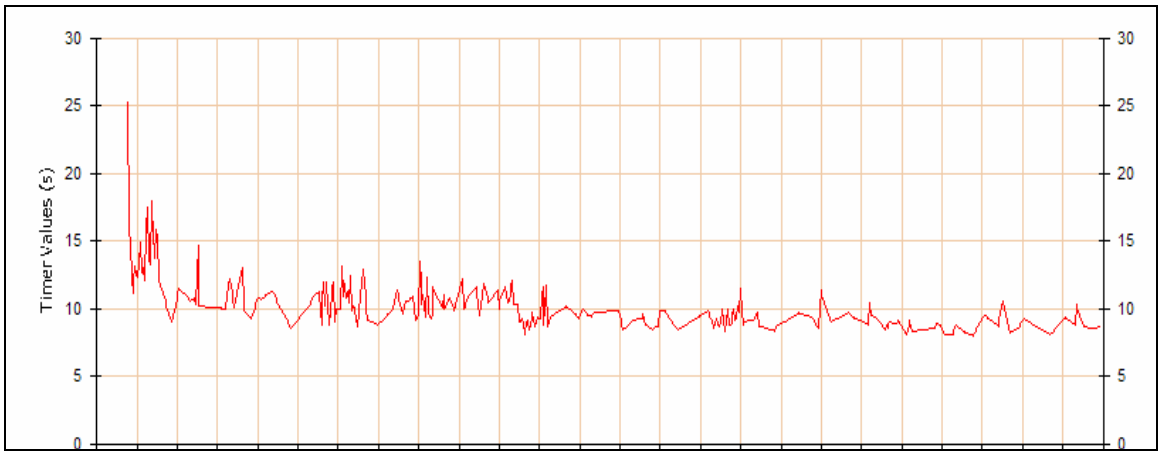
The graphs on the following pages depict services times over the course of the day. They are included here to illustrate the stability of the system over time and spikes, if any, which occur during the day.

Acknowledge Exception Response Times between 10 and 21 GMT



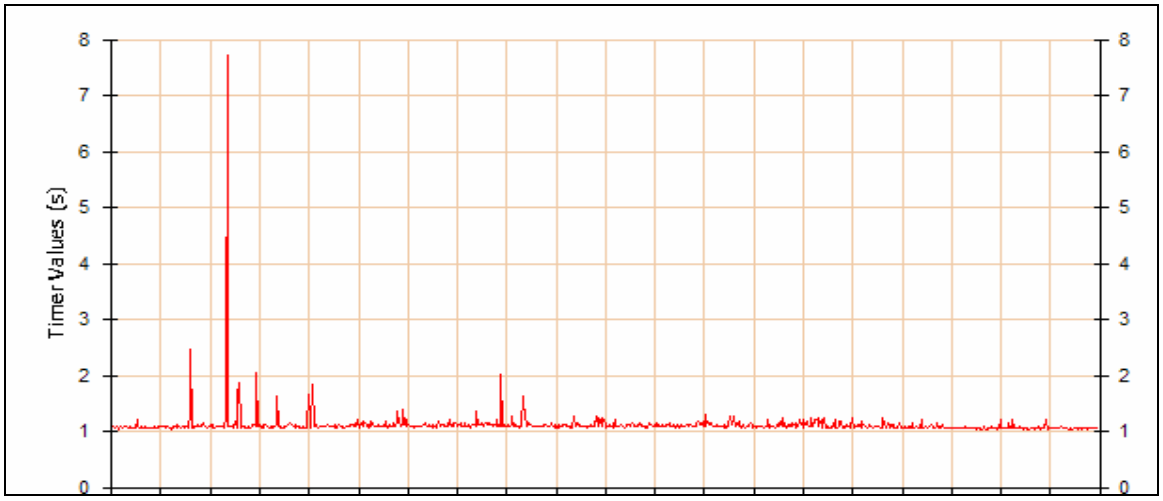
Import Orders Response Time between 18 and 23 GMT

As Planners joined the simulation there was a spike in the number of concurrent file imports. This is reflected in longer response times at the beginning of the Planner user type work window.



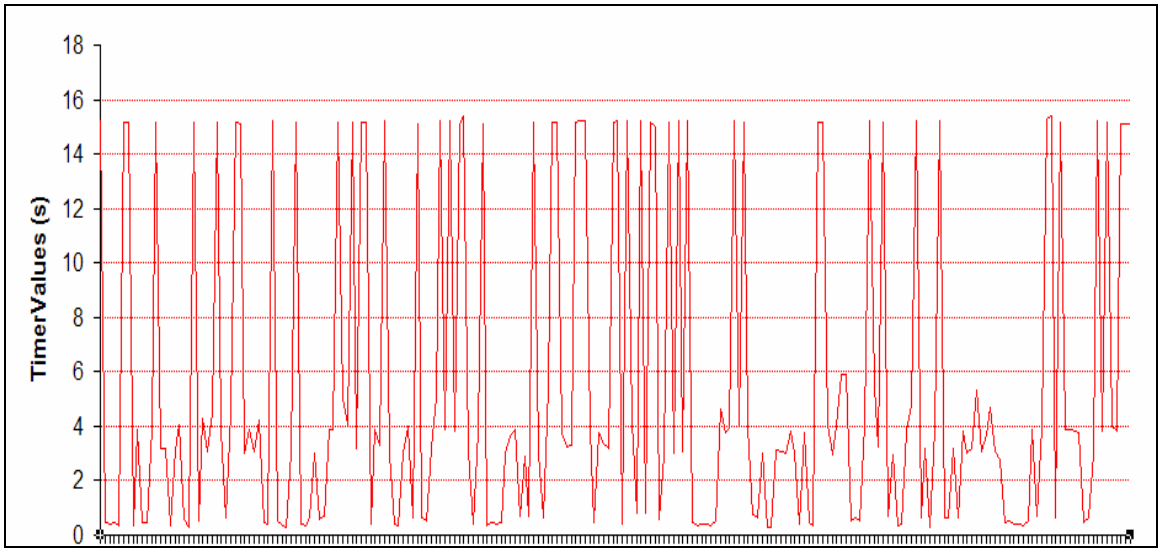
Map/geocode Server Response Time between 10 and 23 GMT

Map Server response times are very consistent at 1.1 seconds. The server gets busy occasionally while managing the large working set of the map/geocode service, which is shown by the spikes in response time.



Java Phone Response Time between 10 and 23 GMT

The mobile communication servers responded in less than 16 seconds in all cases. This test was inadvertently run with one of the mobile comm. servers disabled. With both mobile comm. servers active, more of the response times would have been in the 2 to 4 second range.

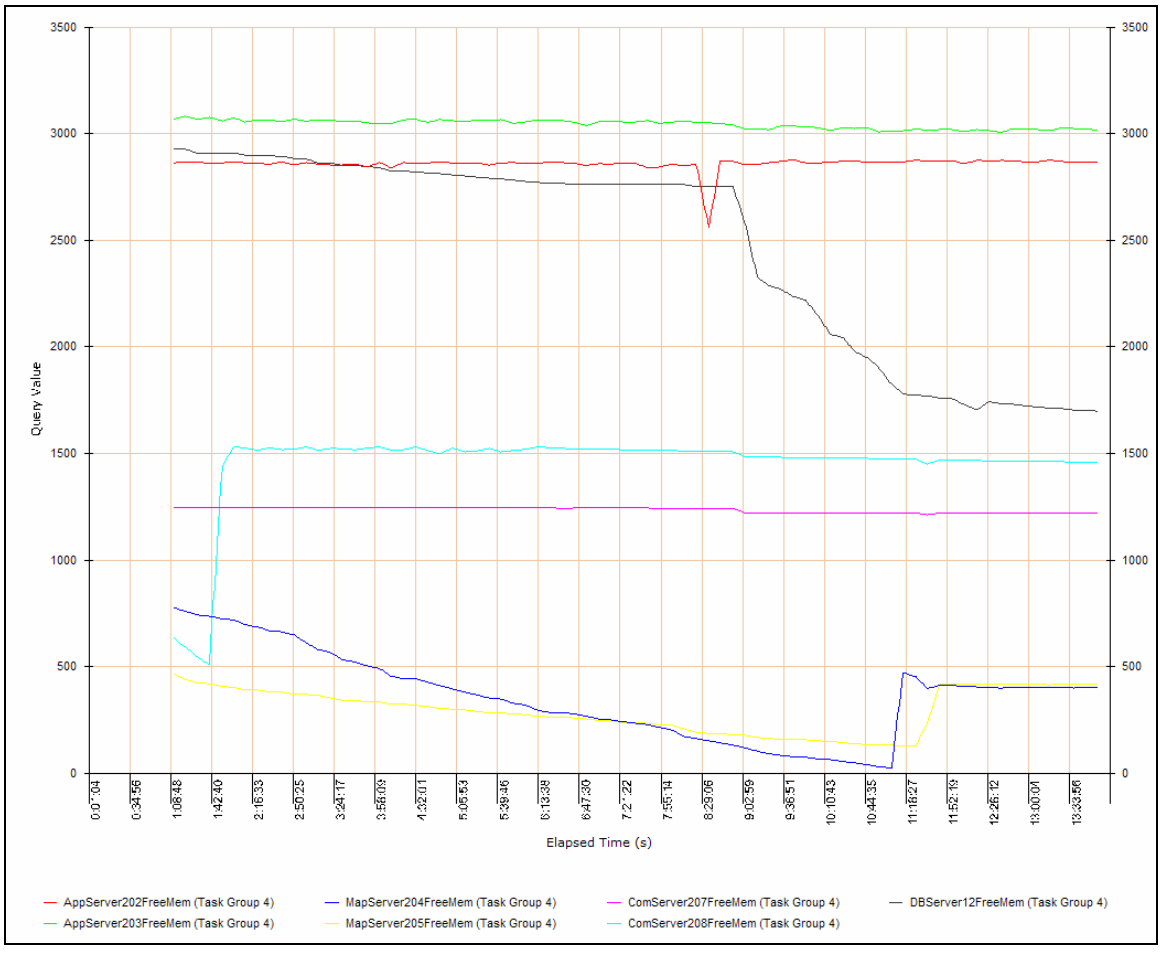


Production Server Memory Utilization between 10 and 23 GMT

Key points;

- Application server memory was essentially unchanged during the day.
- DB server memory increased substantially then stabilized. It is likely that more memory will be needed should the cluster need to scale above 500 users. This could imply a move to a version of SQL server capable of supporting more then 2 GB of memory.
- Map/geocode servers are very memory intensive. The operating system began paging heavily around 10.5 hours into the simulation. While response times did not suffer (see map server response time graph on the preceding page), it may be worthwhile to increase the memory on the map servers by 1 GB.
- Mobile Comm. server memory utilization was unchanged over the course of the day. One of the comm. servers (172.20.179.207) was inadvertently disabled during the simulation. There were no significant effects of running with one comm. server, as will be discussed further in the failover characteristics section of this document.

Graph of free memory (MB) on the y axis vs. elapsed time for all servers.

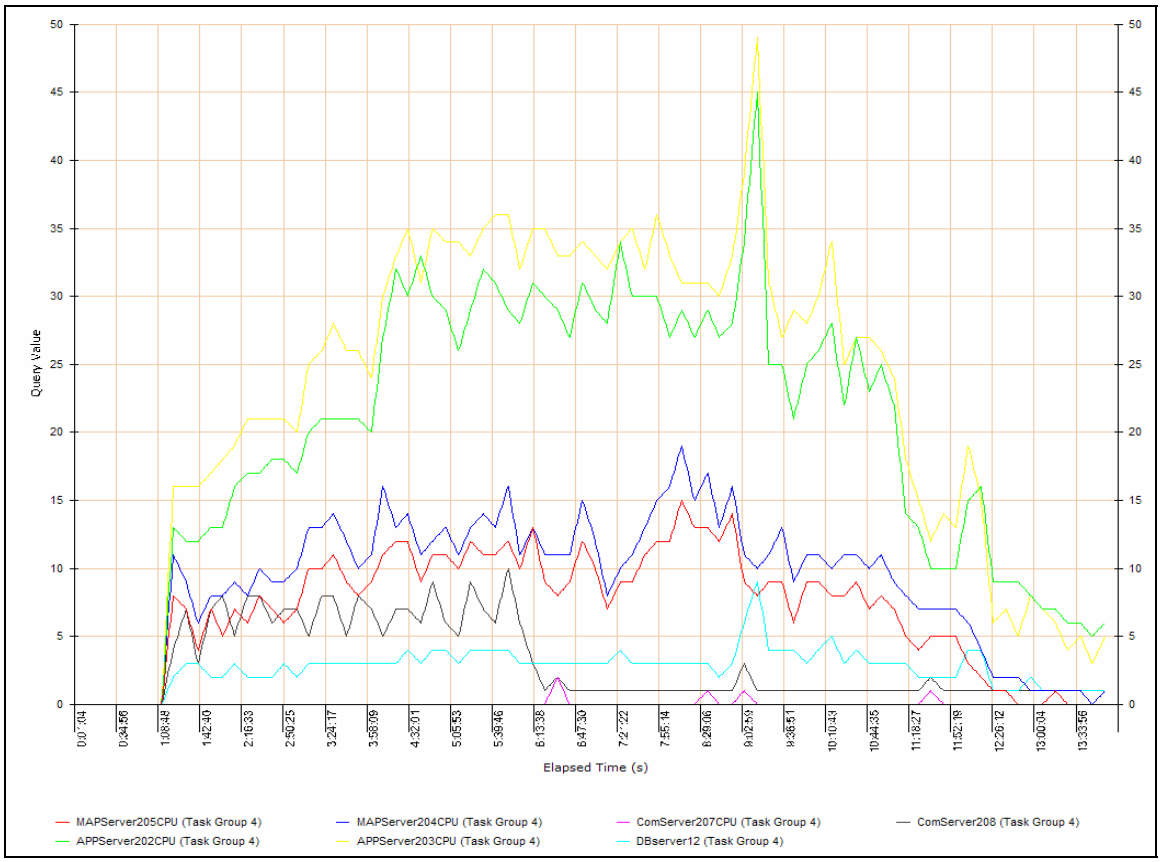


Production Server CPU Utilization between 10 and 23 GMT

The following graph plots average CPU busy on the y axis vs. elapsed time for each server in the cluster.

Key points

- The application server tier is the most CPU intensive. The spike in application server CPU utilization (9 hours into the test) is due to Planners coming on line and performing order imports at the start of their shift.
- All servers in the cluster were operating below our capacity planning recommendation of average CPU <= 50%.
- One of the comm. servers (172.20.179.207) was inadvertently disabled during the simulation. There were no significant effects of running with one comm. server as will be discussed further in the fail over section of this document.
- OpenSTA was able to properly emulate the majority of application behavior. It was not able however to emulate SOAP security encryption. Separate tests determined the CPU cost increased by 7% when security encryption was enabled. The values plotted below for application and map servers are 7% lower then what would be expected from real world measurements of an identical load test performed with real clients.



Production Cluster Response Time Summary between 10 and 23 GMT

This table shows a summary of response time statistics gathered during the full day simulation. See the workload definition for a description of the workload and the various timer names. The columns Avg, Min, and Max are response time statistics in seconds. "95th" is the 95th percentile in seconds (i.e. 19 out of 20 response times are less than or equal to this value). StdDev is the standard deviation between response times. Count is the number of samples collected during the load test.

Full day load test response time statistics

Timer Name	Average	Count	StdDev	Min	95th	Max
ACKEXCEPTION	0.08	12,922	0.1	0.01	0.15	1.76
EDITLOCATIONSAVE	1.23	738	0.3	1.12	1.39	3.76
EXPLORELEFT	0.16	1,940	0.1	0.10	0.29	0.65
EXPLORERIGHT	0.17	1,938	0.1	0.11	0.10	0.79
EXPLOREROUTE	0.17	1,876	0.1	0.10	0.29	1.09
FINDLOCATIONS	0.99	748	0.2	0.76	1.39	2.20
IMPORT300ORDERS	10.8	388	2.4	8.04	14.5	27.3
MAPROUTESDISPATCHER	5.89	13,126	7.8	2.59	24.7	126
MAPROUTESPLANNER	4.91	1,948	4.8	2.35	17.8	35.0
MAPSERVER	1.06	38,654	0.0	1.03	1.09	3.21
RESEQROUTE	0.92	1,846	0.6	0.45	2.20	3.54
ROUTEDYNAMIC	9.15	42	2.4	6.20	13.09	13.5
ROUTEUNASSIGNED	10.62	346	2.2	8.00	15.2	23.0
RTE2RTEMOVE	1.32	1,938	0.6	0.73	2.67	5.10
SHOWEXCEPTION	4.46	11,792	2.0	3.32	6.15	104
SUMMARYVIEW	1.54	12,700	0.3	1.18	2.09	4.95

Key Points:

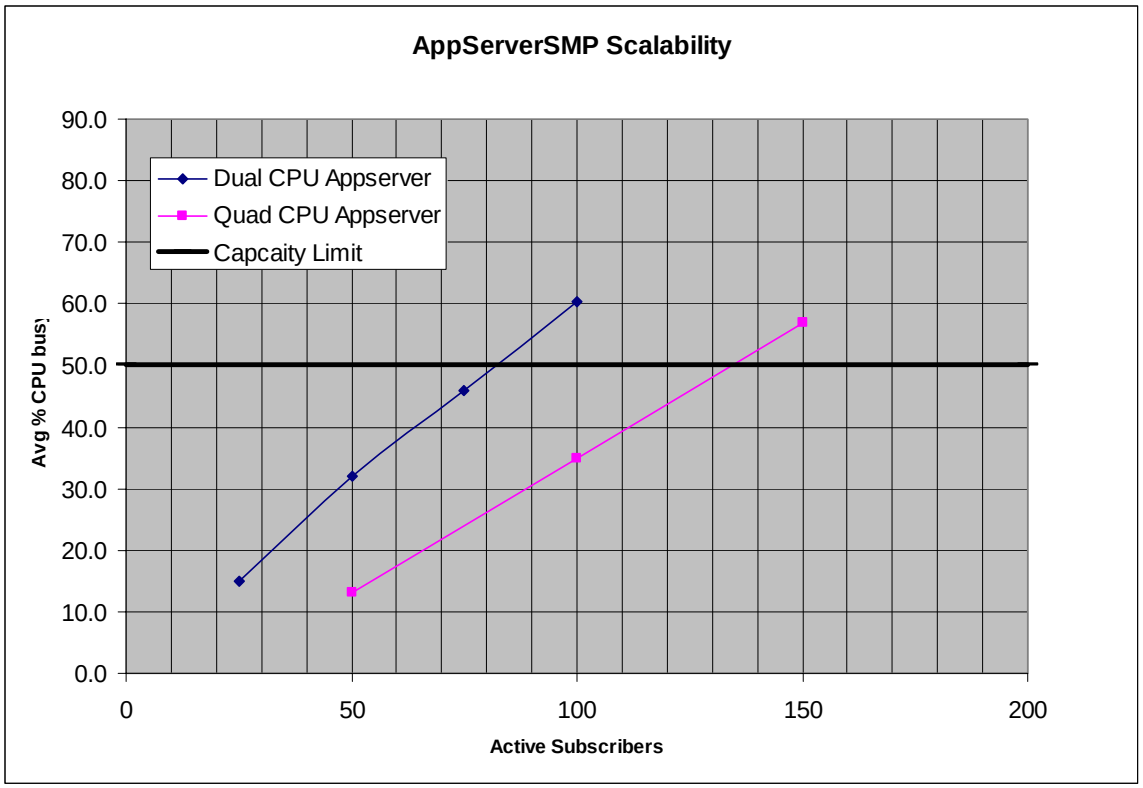
- Response times were gathered over a one gigabit Ethernet. Customer experience will vary with available network bandwidth and will likely be slightly worse then response times portrayed above when accessing the application via a T1, DSL, or cable modem. Dialup connections will perform significantly worse. See the Single User test results for an example of how the user experience varies with network speed.
- All servers in the cluster were operating below recommended capacity limits
- One of the comm. servers (172.20.179.207) was inadvertently disabled during the simulation. There were no significant effects as will be reinforced in the failover section of this document.
- 98% of all transactions completed in 8 seconds or less
- 99% of all transactions completed in 21 seconds or less

SMP Scalability Tests

Application Server CPU Utilization vs. Load

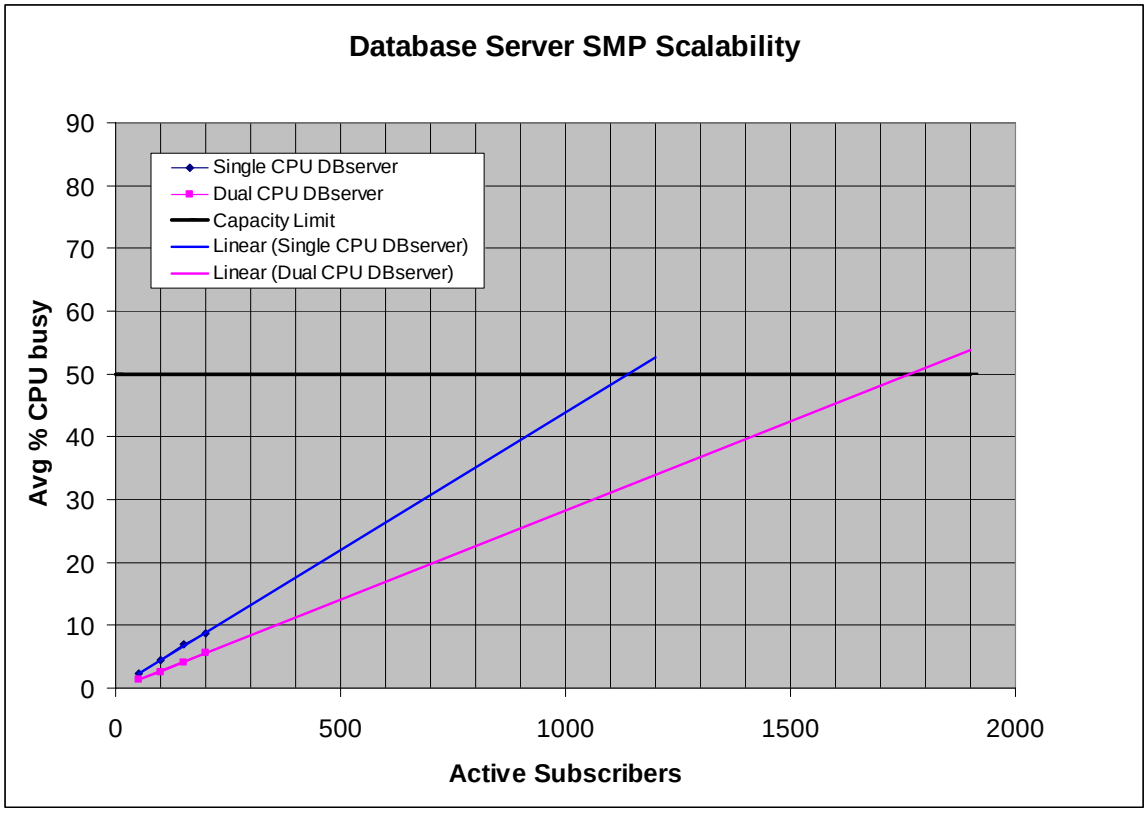
Analysis of the SMP scalability tests indicate that the capacity of the current hardware will be limited by the application servers. Original plans for the configuration had two dual processor DL360s as application servers and two quad processor DL380s as map/geocode servers. By recasting the DL380s as application servers the DL360s as map/geocode servers, cluster capacity was increased from 176 subscribers to 272 subscribers.

The following graph plots the application (web services) CPU utilization vs. active subscribers on a DL380.



Database Server CPU Utilization vs. Load

The following graph plots the database CPU utilization vs. active subscribers on a DL360.



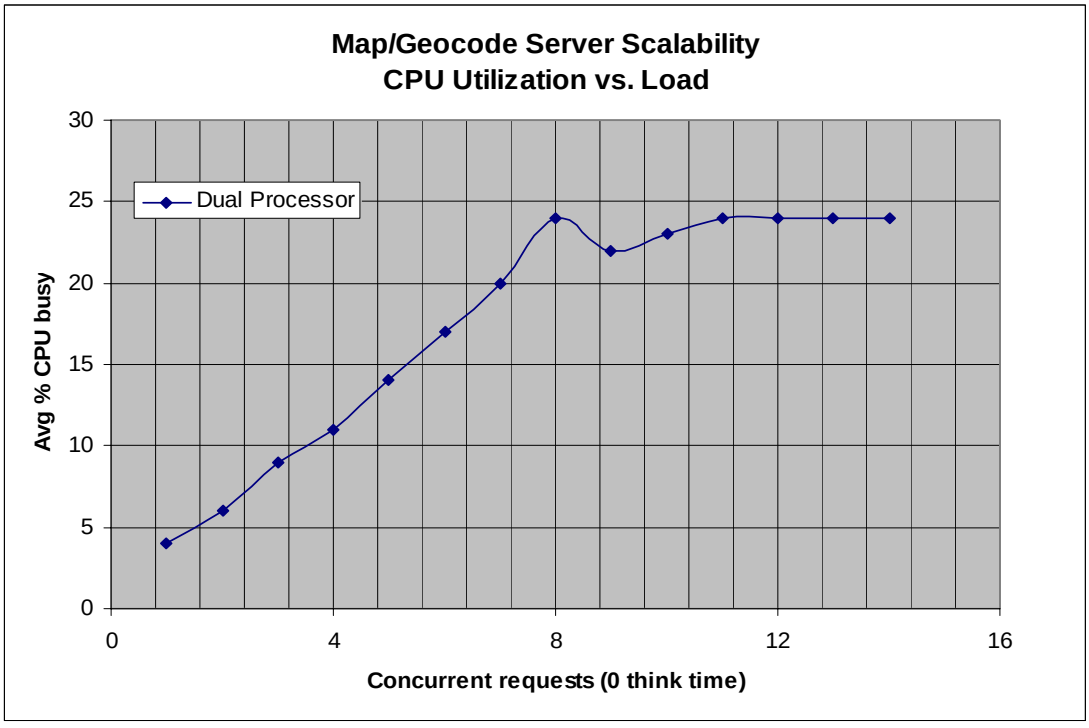
Key points:

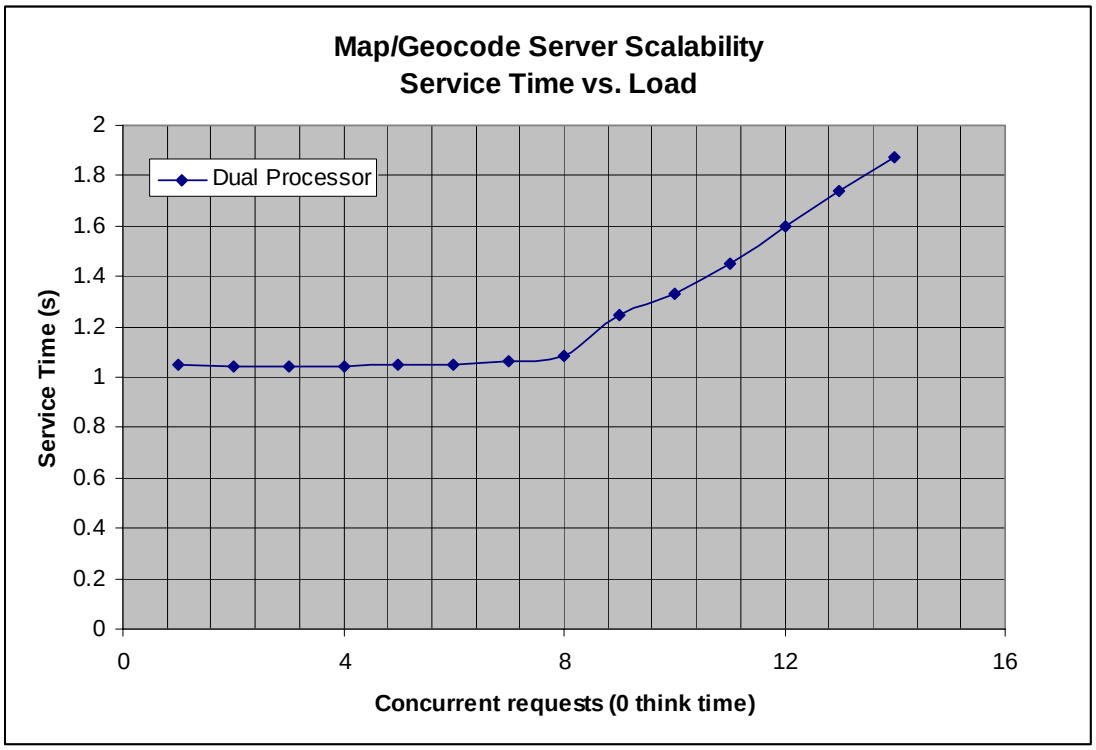
- The database was very lightly loaded. Extrapolating from such low utilization is potentially error prone. The projections have been verified up to 360 users during the cluster test, and no signs of blocking or IO performance problems were encountered. If loads are expected to increase beyond 500 active subscribers, the database server should be monitored for bottlenecks that may limit capacity.
- Data and log files were on the same device (S:). Best practices for performance and reliability indicate the log(s) should be on a separate device (perhaps T:)
- For future reference, the database size was 534 GB and table row counts were:
    - Table1 10,000,000 rows
    - Table2 137,500,000 rows
    - Table3 306,465,000 rows
    - Table4 1,007,111,000 rows
    - Table5 156,151,000 rows
    - Table6 6,000,000 rows
    - Table for two, 154,947,000 rows

Map/Geocode Server Scalability

The map/geocode service contains critical path logic that is not thread safe. As a result, portions of the code are single threaded. The consequences are:

- Memory utilization is very high on the Map/geocode server. The operating system was fairly busy paging during the full day test. While no instability was observed, an additional gigabyte of memory for the map/geocode servers should be considered as cheap insurance.
- No more then 8 concurrent requests can be serviced. At higher loads requests are queued and response times increase linearly (i.e. 8 requests are serviced in 1.1 seconds each, with 16 concurrent requests, average response time is 2.2 seconds)
- A single 3 GHz CPU is sufficient to handle 8 concurrent requests.
- Map/geocode servers will scale linearly with the speed of the CPU but poorly with the number of CPUs per server. Capacity planning models will therefore be limited to single CPU servers for the map/geocode server.
- To keep this in perspective, each map/geocode server is capable of supporting 335 active subscribers. Considering capacity far exceeds expected demand and the price/performance of current servers should additional capacity be required, remedial action beyond purchasing additional memory may not be cost effective.





Mobile Communication Server Scalability

The mobile communication servers are implemented on single CPU servers, and as such, no SMP tests were possible. Sizing data will be limited to single CPU server for the mobile communication server.

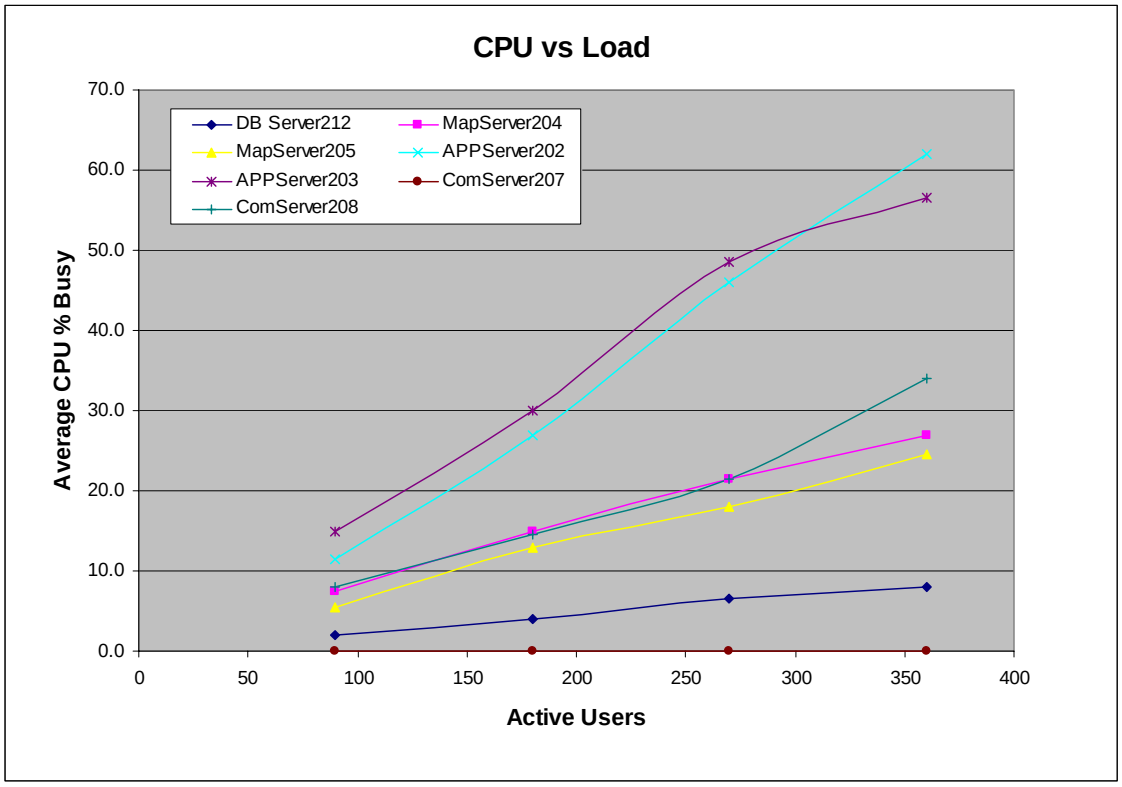
Cluster Capacity Test

The current production cluster will provide consistently good response times up to 270 active subscribers. The following graphs illustrate how average response times for Planner user type and Dispatcher activities vary with respect to load. Data was captured on the production cluster during a load test designed to demonstrate performance characteristics of the cluster when running at 33%, 67%, 100% and 133% of recommended capacity, which corresponds to 90, 180, 270, and 360 active subscribers.

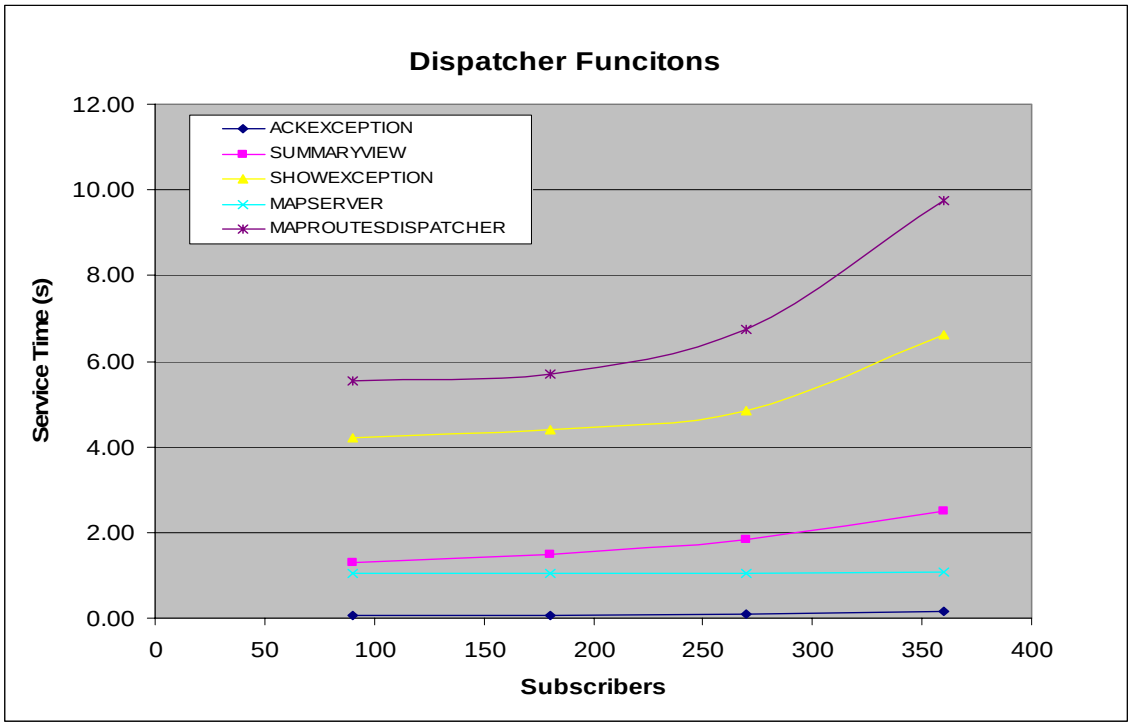
Key Points:

- Workload mix was 75% Dispatchers and 25% Planners.
- Application server CPU reached 50% at approximately 270 users.
- When the application server is operating at 50% CPU or less, large increases in load produce small increases in response time.
- Between 50% and 65%, small increases in load results in moderate increases in response time.
- Above 65% CPU utilization (not plotted), small increases in load will result in large increases in response time.
- Java phone transactions averaged 4 seconds at each of the load points even with one of the mobile communication servers inadvertently disabled during the test.

Graph: CPU vs. Subscribers



Graph: Dispatcher Response Times vs. Subscribers



Graph: Planner user type Functions Response Time vs. Subscribers

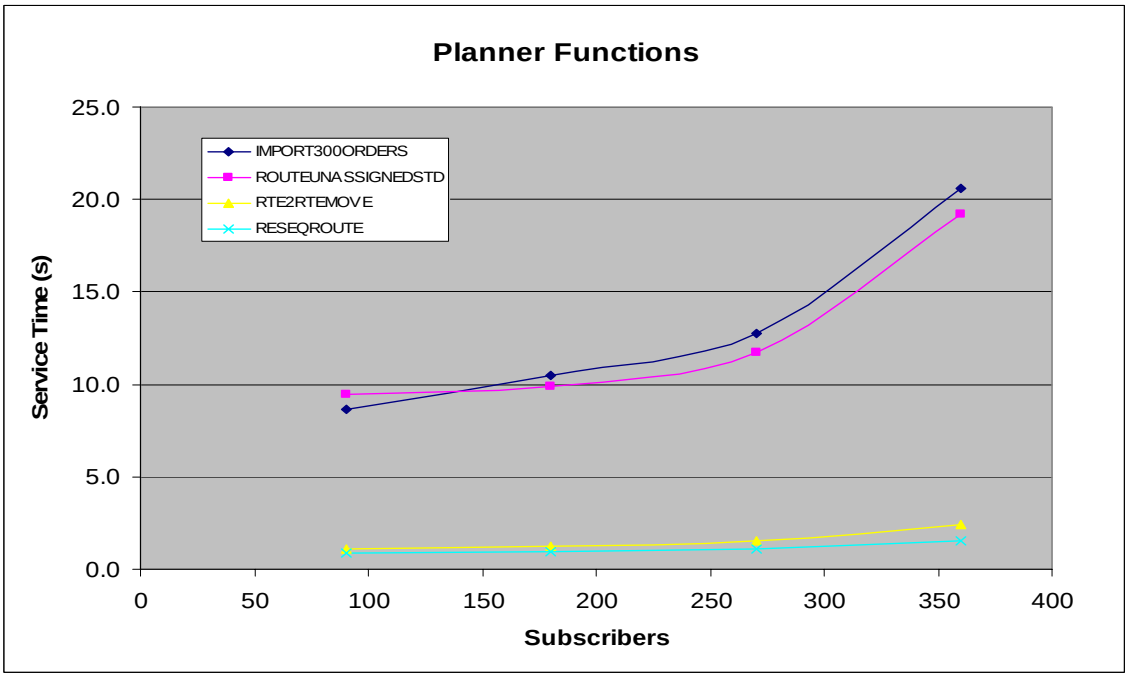


Table: Response Time Table From Maximum Demonstrated Load Test

Response times in the table were gathered from production cluster. See the workload description section for a description of timer names. The columns Avg, Min, and Max are response time statistics in seconds. StdDev is the standard deviation between response times. Cnt is the number of samples observed in over 20 minutes.

Response times measure the elapsed time for all server and network activity required to complete a particular function within the application. Response times reported in this document do not include the time required by the browser to execute java script or render the page. It is conceivable that customers using slow clients or connected over slow (less than T1 speed) or congested networks could experience significantly worse performance.

		Service Times							Service Times					
Timer Name	Users	Avg	Cnt	Dev	Min	Max	Timer Name	Users	Avg	Cnt	Dev	Min	Max
ACKEXCEPTION	90	0.05	362	0.0	0.0	0.2	MAPROUTESPLANNER	90	4.9	14	4.5	2.6	20.6
	180	0.06	588	0.0	0.0	0.4		180	6.1	71	5.0	2.5	20.7
	270	0.09	897	0.1	0.0	0.8		270	6.1	105	6.1	2.5	37.8
	360	0.17	1387	0.2	0.0	2.3		360	8.8	200	9.3	2.8	57.0
EDITLOCATIONSAVE	90	1.1	9	0.0	1.1	1.2	MAPSERVER	90	1.0	584	0.0	1.0	1.1
	180	1.2	25	0.2	1.1	1.9		180	1.1	1137	0.0	1.0	1.2
	270	1.5	35	0.6	1.1	3.8		270	1.1	1734	0.0	1.0	1.2
	360	2.4	53	1.1	1.1	4.0		360	1.1	2638	0.4	1.0	21.1
EXPLORELEFT	90	0.1	18	0.0	0.1	0.3	PHONE	90	4.0	164	5.3	0.3	15.3
	180	0.2	66	0.1	0.1	0.4		180	4.1	1074	5.7	0.3	15.4
	270	0.3	100	0.2	0.1	1.2		270	4.0	1622	5.5	0.3	15.5
	360	0.4	182	0.3	0.1	2.2		360	4.1	3506	5.9	0.2	101.8
EXPLORERIGHT	90	0.1	18	0.0	0.1	0.2	RESEQROUTE	90	0.9	13	0.6	0.5	2.1
	180	0.2	66	0.1	0.1	0.6		180	0.9	52	0.5	0.5	2.3
	270	0.3	100	0.2	0.1	1.0		270	1.1	109	0.7	0.5	3.3
	360	0.4	182	0.3	0.1	2.4		360	1.5	143	1.0	0.5	6.4
EXPLOREROUTE	90	0.2	13	0.0	0.1	0.3	ROUTEUNASSIGNEDSTD	90	9.4	1	0.0	9.4	9.4
	180	0.2	52	0.1	0.1	0.6		180	9.9	6	0.8	9.0	11.0
	270	0.2	109	0.1	0.1	0.9		270	11.7	9	2.0	10.1	16.6
	360	0.4	143	0.3	0.1	2.3		360	19.2	24	7.1	11.4	44.1
FINDLOCATIONS	90	1.0	9	0.1	0.8	1.2	RTE2RTEMOVE	90	1.1	18	0.4	0.8	2.3
	180	1.0	25	0.2	0.9	1.4		180	1.3	66	0.7	0.8	3.9
	270	1.2	35	0.3	0.9	2.3		270	1.6	99	0.8	0.8	3.8
	360	1.6	53	0.5	0.9	2.8		360	2.5	182	1.5	0.8	10.3
IMPORT300ORDERS	90	8.6	1	0.0	8.6	8.6	SHOWEXCEPTION	90	4.2	178	0.7	3.4	8.0
	180	10.5	5	1.1	9.5	11.8		180	4.4	327	0.9	3.4	9.9
	270	12.8	9	2.6	9.9	17.4		270	4.8	530	1.5	3.4	18.6
	360	20.6	24	6.4	13.3	37.6		360	6.6	788	2.8	3.6	20.6
MAPROUTESDISPATCHER	90	5.5	214	6.3	2.7	33.2	SUMMARYVIEW	90	1.3	226	0.1	1.2	1.8
	180	5.7	412	6.7	2.7	44.2		180	1.5	433	0.2	1.2	2.8
	270	6.7	570	8.0	2.7	49.5		270	1.8	575	0.5	1.2	4.3
	360	9.7	861	13.1	2.8	117.4		360	2.5	992	0.9	1.2	7.2

Failover Performance Characteristics

Failover tests were run with a load of 200 subscribers. A series of server failures were initiated and application errors and response times were monitored to measure message loss, duration of the recovery, and application performance during the loss and restoration of servers. The production cluster performed very well during failover tests. Under the worst case scenario where a database server failed, the system was unavailable for 71 seconds while the standby server was activated. Other tests were individual map, web service, or mobile communication servers were rebooted showed minimal message loss and minor impact to the performance of transactions being serviced by the surviving servers. The following table summarizes the impact of each server failing.

	Message Loss	
Server Failed	**Loss Window (seconds)**	**Count**
Application	2	7
Database	71	80
Mobile Comm.	N/A	N/A
Map/Geocode	1	1

Application (Web Service) Server Failover Test

Event Log

11:45 Begin ramping up to 200 active subscribers over a 15 minute period

12:00 200 Subscribers logged in and working according to work load specifications

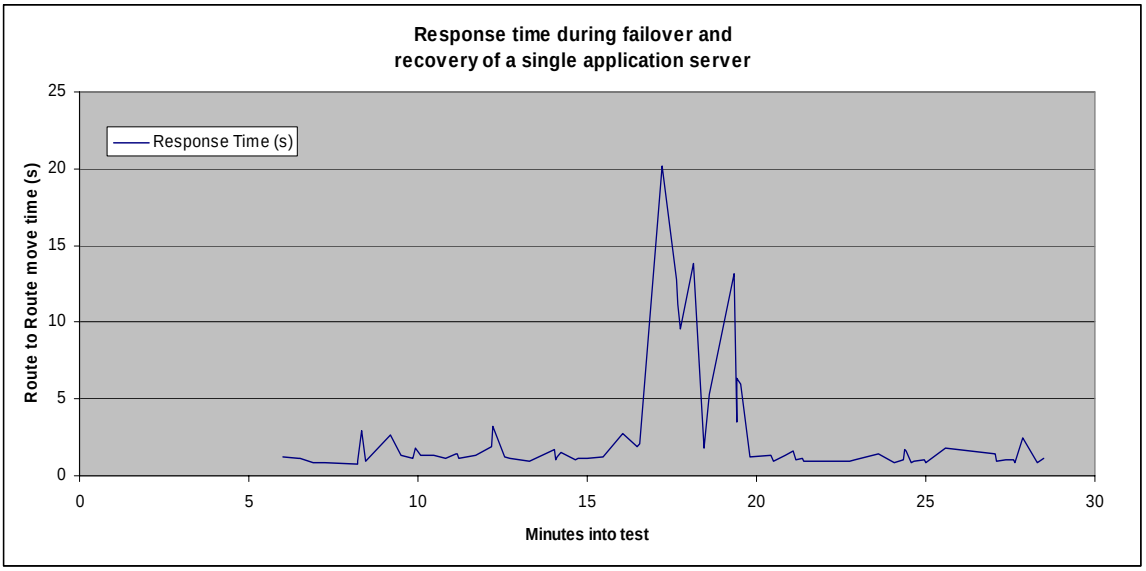
12:02 Shutdown one application server

12:02:50 – 12:02:52 Seven calls to GetManifest failed with http status code 503

12:02:52 Cluster now functioning on one application server, response times elevated

12:06 The failed application server reboot completes, server rejoins cluster, response times return to normal

The following graph shows the time to execute a route to route move before, during (between 17 and 20 minutes into the test), and after the failover event.



Map/Geocode server failover test

Event Log

11:45 Begin ramping up to 200 active subscribers over a 15 minute period

12:00 200 subscribers logged in and working according to work load specifications

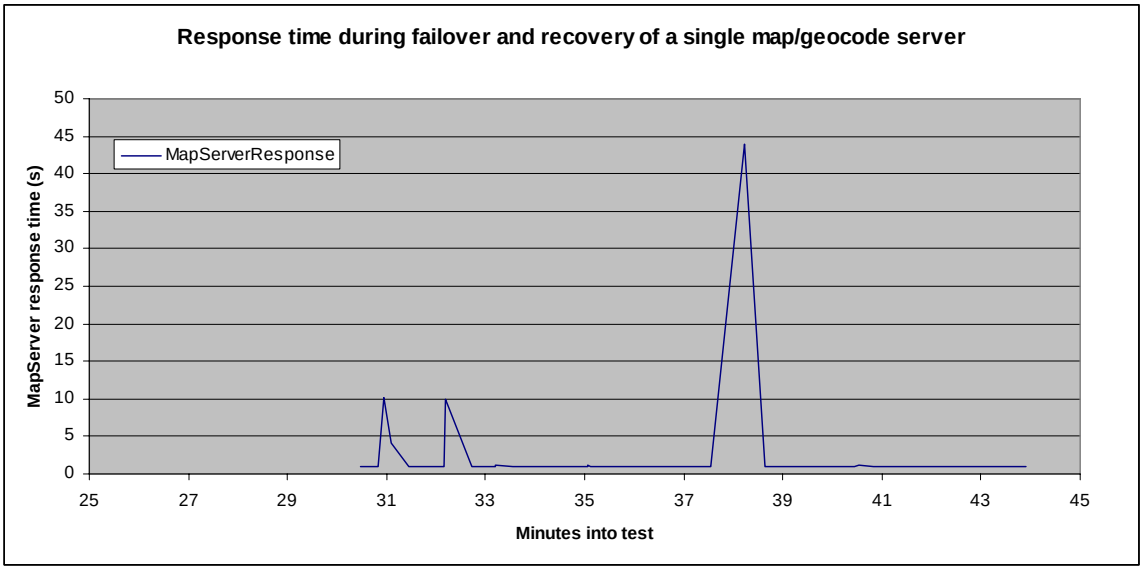
12:16 Shutdown one map/geocode server by killing the process then rebooting the server

12:16:35 One call to the map server failed with error 1236 (Network connection aborted by local system)

12:16 - 12:22 cluster now functioning on one application server, response times spike then return to normal

12:22 Failed map server reboot complete, server rejoins cluster, response times spike as first request to restored server incurs startup overhead.

The following graph shows map server response time before, during (from 30 and 38 minutes into the test), and after the failover event.



Database server failover test

Event Log

11:45 Begin ramping up to 200 active subscribers over a 15 minute period

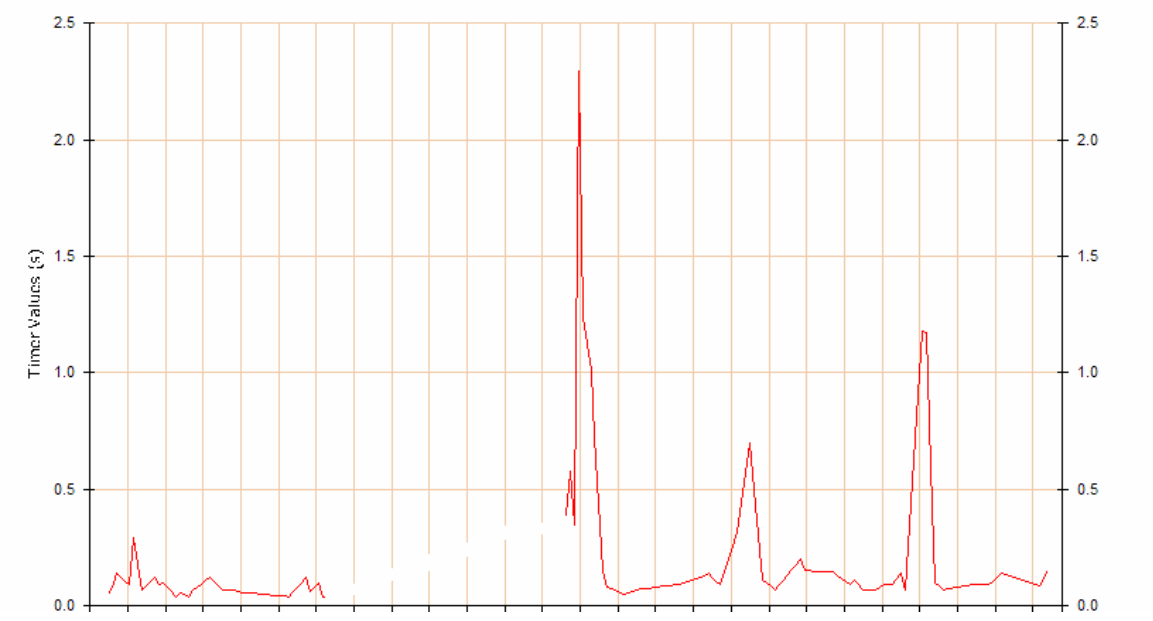
12:00 200 subscribers logged in and working according to work load specifications

12:31 Shutdown database server with restart

12:31 – 12:32: 71 calls to application and map servers fail with HTTP code 500. As all messages failed, no response times recorded during the 71 second outage.

12:32 Cluster is fully functional again, response times spike as database cache is filled on the replacement server.

The following graph shows response time of the Acknowledge exception even before during and after the failover of the of the database cluster. The period where no data is plotted is approximately 71 seconds long. The graph shows activity starting approximately one minute before the failure through approximately 2 minutes after the failure.



Mobile Communication Server Failover test

Java phones (and the java phone emulator) are programmed to retry messages indefinitely should contact be lost with the mobile communication server(s). No failover test was run as we would only be validating the correctness of the emulator. Java phone response times would increase by the length of time it takes to restore the service. During the full day test, one of the mobile communication servers was inadvertently disabled. Results indicate a single communication server can easily handle all traffic should the other server fail.

Single user tests

Single user statistics are very useful for regression analysis. These were obtained by serially executing each of the multi-user scripts 50 times with wait times disabled. Response times measure time spent on the servers and network but not include time for the browser to render the output or execute java script. Customers running slow clients or accessing the application over a dialup connection will experience slower response times. To help understand the effect network bandwidth has on the user experience, the test was run once over the gigabit Ethernet and repeated over a 1.6 Mb cable modem connection (from our office located in Hollis, NH) which approximates T1 performance.

The columns Avg, Min, and Max are response time statistics in seconds. StdDev is the standard deviation between response times. Count is the number of iterations a particular function was executed. Elapsed seconds is the computed as avg. response time multiplied by count. The cable modem test includes a column "increase vs. Gb" which is the difference between cable modem and gigabit Ethernet response times. "% of load" is a good indicator of the relative cost in servicing the transaction.

Single User Regression Test data (Gb Ethernet)

Timer Name	Average	Count	StdDev	Min	Max	Elapsed Secs	% of Load
ACKEXCEPTION	0.0	50	0.0	0.0	0.1	1.7	0.09%
EDITLOCATIONSAVE	1.1	50	0.0	1.1	1.3	56.5	2.91%
EXPLORELEFT	0.1	50	0.0	0.1	0.2	5.7	0.29%
EXPLORERIGHT	0.1	50	0.0	0.1	0.1	5.3	0.27%
EXPLOREROUTE	0.1	50	0.0	0.1	0.1	5.0	0.26%
FINDLOCATIONS	0.8	54	0.0	0.7	0.9	42.1	2.17%
IMPORT300ORDERS	7.6	50	0.7	7.3	12.3	380.1	19.60%
MAPROUTESDISPATCHER	6.1	50	6.3	3.1	24.0	304.4	15.70%
MAPROUTESPLANNER	4.2	50	2.7	2.9	14.3	209.1	10.79%
MAPSERVER	1.1	200	0.0	1.0	1.1	210.4	10.85%
RESEQROUTE	1.0	50	0.5	0.4	1.6	48.7	2.51%
ROUTEDYNAMIC	5.7	25	0.1	5.6	6.1	142.7	7.36%
ROUTEUNASSIGNEDSTD	7.7	25	1.0	6.9	12.1	191.8	9.89%
RTE2RTEMOVE	1.2	50	0.6	0.6	1.9	61.1	3.15%
SHOWEXCEPTION	4.4	50	0.5	4.0	6.5	220.0	11.35%
SUMMARYVIEW	1.1	50	0.3	1.0	2.9	54.4	2.80%
				Grand Total		**1939.0**	

Single User Regression Test data (Cable modem 1.6 Mb)

Timer Name	Average	Count	StdDev	Min	Max	Increase vs. Gb
ACKEXCEPTION	0.1	50.0	0.0	0.1	0.3	0.1
EDITLOCATIONSAVE	1.3	50.0	0.1	1.3	1.6	0.2
EXPLORELEFT	0.4	50.0	0.0	0.3	0.5	0.3
EXPLORERIGHT	0.3	50.0	0.0	0.2	0.3	0.2
EXPLOREROUTE	0.4	50.0	0.0	0.3	0.4	0.3
FINDLOCATIONS	1.2	54.0	0.1	0.9	1.5	0.4
IMPORT300ORDERS	15.2	50.0	2.1	13.9	16.2	7.6
MAPROUTESDISPATCHER	6.1	50.0	7.4	3.1	33.5	0.0
MAPROUTESPLANNER	4.9	50.0	2.1	3.4	16.7	0.7
MAPSERVER	1.3	200.0	0.1	1.2	1.8	0.2
RESEQROUTE	1.4	50.0	0.6	0.8	2.1	0.4
ROUTEDYNAMIC	6.3	25.0	0.1	6.2	6.6	0.6
ROUTEUNASSIGNEDSTD	7.9	25.0	0.9	7.1	12.2	0.2
RTE2RTEMOVE	1.8	50.0	0.6	1.1	3.5	0.6
SHOWEXCEPTION	4.4	50.0	0.6	3.9	7.4	0.0
SUMMARYVIEW	1.6	50.0	0.2	1.5	3.2	0.5

Capacity Planning

The SUDS application is implemented with a four-tier architecture. The four tiers are application (web services), map/geocode, mobile communication, and database. In order for the production cluster to provide optimal performance, each tier must be configured with adequate capacity. Performax recommends configuring servers so that the average CPU utilization is 50% or less. The capacity planning tables list each of the current HP ProLiant servers and the number of active subscribers (performing transactions in accordance with the workload specification) the server can support at 50% CPU utilization vs. the number of CPUs.

Database Server Capacity Chart

		Subscribers vs. Number of CPUs		
Database Server	**SpecInt 2000 Base**	**1**	**2**	**4**
ProLiant DL360 G3 (3.0GHz/2MB L2 cache, Intel Xeon)	1463	1075	**1834**	
ProLiant DL360 G3 (3.2GHz/2MB L3 cache, Intel Xeon)	1491	1096	1869	
ProLiant DL360 G3(3.06GHz, Intel Xeon)	1028	755	1288	
ProLiant DL360 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1245	915	1560	
ProLiant DL360 G4 (3.4GHz, Intel Xeon)	1367	1004	1713	
ProLiant DL360 G4 (3.6GHz, Intel Xeon)	1429	1050	1791	
ProLiant DL360 G4p (3.6GHz, Intel Xeon)	1675	1231	2099	
ProLiant DL360 G4p (3.8GHz, Intel Xeon)	1799	1322	2255	
ProLiant DL360 G4p (2.8GHz, Intel dual-core Xeon, 2x2MB L2)	1377	1012	1726	2829
ProLiant DL380 G3 (3.2GHz, Intel Xeon)	1484	1090	1860	
ProLiant DL380 G3(3.06GHz, Intel Xeon)	1028	755	1288	
ProLiant DL380 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1239	910	1553	
ProLiant DL380 G4 (3.4GHz, Intel Xeon)	1433	1053	1796	
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1653	1215	2072	
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1517	1115	1901	
ProLiant DL380 G4 (3.8GHz, Intel Xeon)	1797	1320	2252	
ProLiant DL380 G5 (3.73GHz, Intel Xeon)	1771	1301	2220	3639
ProLiant DL385 (AMD Opteron (TM) 252)	1558	1145	1953	
ProLiant DL385 (AMD Opteron (TM) 254)	1696	1246	2126	
ProLiant DL385 (AMD Opteron (TM) 275)	1380	1014	1729	2835
ProLiant DL385 (AMD Opteron (TM) 280)	1500	1102	1880	3082
ProLiant DL385 (AMD Opteron (TM) 280)	1596	1173	2000	3279
ProLiant DL385 (AMD Opteron (TM) 285)	1604	1179	2010	3296
ProLiant DL560 (2.8GHz, Intel Xeon MP)	1196	879	1499	
ProLiant DL580 G2 (3.0GHz, Intel Xeon MP)	1455	1069	1823	
ProLiant DL580 G2(2.8GHz, Intel Xeon MP)	1190	874	1491	
ProLiant DL580 G3 (3.0GHz, Dual-Core Intel(R) Xeon(R) processor)	1345	988	1686	3695
ProLiant DL580 G3 (3.333GHz, Intel Xeon)	1497	1100	1876	
ProLiant DL580 G3 (3.66GHz, Intel Xeon)	1388	1020	1740	

The production cluster consists of 2 database severs of the type highlighted but only one is active at any given time. The other is in standby in the case of a failover.

Legend

- Number of CPUs configured in the system or number of CPUs this tier can take advantage of.
- Specint 2000. Used with "# CPUs" to estimate system capacity.
- Subscribers determined by the load that would cause the processor(s) to average 50% CPU busy.

Application (Web Service) Server Capacity Chart

		Subscribers vs. Number of CPUs		
Server	SpecInt 2000 Base	1	2	4
ProLiant DL360 G3 (3.0GHz/2MB L2 cache, Intel Xeon)	1463	52	88	
ProLiant DL360 G3 (3.2GHz/2MB L3 cache, Intel Xeon)	1491	53	90	
ProLiant DL360 G3(3.06GHz, Intel Xeon)	1028	36	62	
ProLiant DL360 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1245	44	75	
ProLiant DL360 G4 (3.4GHz, Intel Xeon)	1367	48	82	
ProLiant DL360 G4 (3.6GHz, Intel Xeon)	1429	50	86	
ProLiant DL360 G4p (3.6GHz, Intel Xeon)	1675	59	101	
ProLiant DL360 G4p (3.8GHz, Intel Xeon)	1799	63	108	
ProLiant DL360 G4p (2.8GHz, Intel dual-core Xeon, 2 MB L2)	1377	49	83	**136**
ProLiant DL380 G3 (3.2GHz, Intel Xeon)	1484	52	89	
ProLiant DL380 G3(3.06GHz, Intel Xeon)	1028	36	62	
ProLiant DL380 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1239	44	75	
ProLiant DL380 G4 (3.4GHz, Intel Xeon)	1433	51	86	
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1653	58	99	
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1517	53	91	
ProLiant DL380 G4 (3.8GHz, Intel Xeon)	1797	63	108	
ProLiant DL380 G5 (3.73GHz, Intel Xeon)	1771	62	107	175
ProLiant DL385 (AMD Opteron (TM) 252)	1558	55	94	
ProLiant DL385 (AMD Opteron (TM) 254)	1696	60	102	
ProLiant DL385 (AMD Opteron (TM) 275)	1380	49	83	136
ProLiant DL385 (AMD Opteron (TM) 280)	1500	53	90	148
ProLiant DL385 (AMD Opteron (TM) 280)	1596	56	96	157
ProLiant DL385 (AMD Opteron (TM) 285)	1604	57	96	158
ProLiant DL560 (2.8GHz, Intel Xeon MP)	1196	42	72	
ProLiant DL580 G2 (3.0GHz, Intel Xeon MP)	1455	51	88	
ProLiant DL580 G2(2.8GHz, Intel Xeon MP)	1190	42	72	
ProLiant DL580 G3 (3.0GHz, Dual-Core Intel(R) Xeon(R) p	1345	47	81	177
ProLiant DL580 G3 (3.333GHz, Intel Xeon)	1497	53	90	
ProLiant DL580 G3 (3.66GHz, Intel Xeon)	1388	49	83	

The production cluster consists of 2 application severs of the type highlighted.

Legend

- Number of CPUs configured in the system or number of CPUs this tier can take advantage of.
- Specint 2000. Used with "# CPUs" to estimate system capacity.
- Subscribers determined by the load that would cause the processor(s) to average 50% CPU busy.

Map/Geocode Server Capacity Chart

		Subscribers vs. Number of CPUs
Server	**SpecInt2000Base**	**1**
ProLiant DL360 G3 (3.0GHz/2MB L2 cache, Intel Xeon)	1463	**335**
ProLiant DL360 G3 (3.2GHz/2MB L3 cache, Intel Xeon)	1491	341
ProLiant DL360 G3(3.06GHz, Intel Xeon)	1028	235
ProLiant DL360 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1245	285
ProLiant DL360 G4 (3.4GHz, Intel Xeon)	1367	313
ProLiant DL360 G4 (3.6GHz, Intel Xeon)	1429	327
ProLiant DL360 G4p (3.6GHz, Intel Xeon)	1675	383
ProLiant DL360 G4p (3.8GHz, Intel Xeon)	1799	412
ProLiant DL360 G4p (2.8GHz, Intel dual-core Xeon, 2x2MB L2)	1377	315
ProLiant DL380 G3 (3.2GHz, Intel Xeon)	1484	340
ProLiant DL380 G3(3.06GHz, Intel Xeon)	1028	235
ProLiant DL380 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1239	284
ProLiant DL380 G4 (3.4GHz, Intel Xeon)	1433	328
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1653	378
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1517	347
ProLiant DL380 G4 (3.8GHz, Intel Xeon)	1797	411
ProLiant DL380 G5 (3.73GHz, Intel Xeon)	1771	405
ProLiant DL385 (AMD Opteron (TM) 252)	1558	357
ProLiant DL385 (AMD Opteron (TM) 254)	1696	388
ProLiant DL385 (AMD Opteron (TM) 275)	1380	316
ProLiant DL385 (AMD Opteron (TM) 280)	1500	343
ProLiant DL385 (AMD Opteron (TM) 280)	1596	365
ProLiant DL385 (AMD Opteron (TM) 285)	1604	367
ProLiant DL560 (2.8GHz, Intel Xeon MP)	1196	274
ProLiant DL580 G2 (3.0GHz, Intel Xeon MP)	1455	333
ProLiant DL580 G2(2.8GHz, Intel Xeon MP)	1190	272
ProLiant DL580 G3 (3.0GHz, Dual-Core Intel(R) Xeon(R) processor)	1345	308
ProLiant DL580 G3 (3.333GHz, Intel Xeon)	1497	343
ProLiant DL580 G3 (3.66GHz, Intel Xeon)	1388	318

The production cluster consists of 2 map/geocode severs of the type highlighted. The single threaded nature of the map/geocode service negates any benefit of multiprocessor servers hence only single CPU capacity values are provided

Legend

- Number of CPUs configured in the system or number of CPUs this tier can take advantage of.
- Specint 2000. Used with "# CPUs" to estimate system capacity.
- Subscribers determined by the load that would cause the processor(s) to average 50% CPU busy. (8 threads are sufficient to reach 50% CPU utilization on a 3 Ghz processor. Faster processors will require more threads and therefore more memory.)

Mobile Communication Server Capacity Chart

		Subscribers vs. Number of CPUs
Server	**SpecInt2000Base**	**1**
ProLiant DL360 G3 (3.0GHz/2MB L2 cache, Intel Xeon)	1463	**625**
ProLiant DL360 G3 (3.2GHz/2MB L3 cache, Intel Xeon)	1491	637
ProLiant DL360 G3(3.06GHz, Intel Xeon)	1028	439
ProLiant DL360 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1245	532
ProLiant DL360 G4 (3.4GHz, Intel Xeon)	1367	584
ProLiant DL360 G4 (3.6GHz, Intel Xeon)	1429	610
ProLiant DL360 G4p (3.6GHz, Intel Xeon)	1675	716
ProLiant DL360 G4p (3.8GHz, Intel Xeon)	1799	769
ProLiant DL360 G4p (2.8GHz, Intel dual-core Xeon, 2x2MB L2)	1377	588
ProLiant DL380 G3 (3.2GHz, Intel Xeon)	1484	634
ProLiant DL380 G3(3.06GHz, Intel Xeon)	1028	439
ProLiant DL380 G3(3.2GHz/1MB L3 cache, Intel Xeon)	1239	529
ProLiant DL380 G4 (3.4GHz, Intel Xeon)	1433	612
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1653	706
ProLiant DL380 G4 (3.6GHz, Intel Xeon)	1517	648
ProLiant DL380 G4 (3.8GHz, Intel Xeon)	1797	768
ProLiant DL380 G5 (3.73GHz, Intel Xeon)	1771	757
ProLiant DL385 (AMD Opteron (TM) 252)	1558	666
ProLiant DL385 (AMD Opteron (TM) 254)	1696	725
ProLiant DL385 (AMD Opteron (TM) 275)	1380	590
ProLiant DL385 (AMD Opteron (TM) 280)	1500	641
ProLiant DL385 (AMD Opteron (TM) 280)	1596	682
ProLiant DL385 (AMD Opteron (TM) 285)	1604	685
ProLiant DL560 (2.8GHz, Intel Xeon MP)	1196	511
ProLiant DL580 G2 (3.0GHz, Intel Xeon MP)	1455	622
ProLiant DL580 G2(2.8GHz, Intel Xeon MP)	1190	508
ProLiant DL580 G3 (3.0GHz, Dual-Core Intel(R) Xeon(R) processor)	1345	575
ProLiant DL580 G3 (3.333GHz, Intel Xeon)	1497	640
ProLiant DL580 G3 (3.66GHz, Intel Xeon)	1388	593

The production cluster consists of 2 database severs of the type highlighted. The mobile communications servers are implemented on single processor servers hence no SMP testing was possible and results will be limited to single CPU servers.

Legend

- Number of CPUs configured in the system or number of CPUs this tier can take advantage of.
- Specint 2000. Used with "# CPUs" to estimate system capacity.
- Subscribers determined by the load that would cause the processor(s) to average 50% CPU busy.

Methodology Used for Calculating Positioning Tables

This section removed for sample report.