

Optimizing server performance: Handling the Curves like a Pro

by Carol Zimmet and Susan Florio Barber

[Editor's note: This article resides in "Iris Today", the technical Webzine located on the <http://www.notes.net> Web site produced by Iris Associates, the developers of Domino and Notes.]

You're cruising down the road, the wind rushes in through an open window, and your foot lowers on the accelerator. You're thinking that no other car performs as well as yours at this speed -- but suddenly, you see a sharp curve! Your heart drops into your stomach and you immediately let up on the gas while turning your steering wheel as fast as you can. Whew! You made it -- your car really does perform like a pro.

Your Domino server is a lot like your car. You can judge its performance based on some of the same criteria. You can look at speed, or capacity, or cost effectiveness as individual criteria, or you can look at the big picture. What you really want from your Domino server, as from your car, is for it to handle all the curves that Domino server administration, or the road, throws at you.

This article gives you a look at Domino's performance on various system configurations while executing various workloads. To do this, we examined the results of six different evaluation scenarios and saw how each configuration measured up under a particular workload. These evaluation scenarios helped us draw conclusions about the following aspects of performance:

- The amount of processor power the server used
- The amount of memory the server used
- The probe response time (end user response time)
- The disk input and output (I/O) overall performance
- The pages-per-second performance
- The Notesmark (Domino transactions) performance

By evaluating the "curves" of the graphs provided with each evaluation scenario, you can get an overall picture of Domino performance. You, or your vendor, can evaluate how a system configuration performs under a given workload, as well as determine how much of its capacity is used as a result of this workload.

For more background information about how we conduct performance analyses here at Lotus/Iris or an introduction to the tools we use, see "[Optimizing server performance: Port encryption & Buffer Pools](#)" or "[Optimizing Server Performance: CPU scalability](#)." To read more recommendations for improving server performance, see "[The top ten ways you can improve server performance](#)."

Overall test methodology

This section outlines the overall test methodology we used for all of our test scenarios. It includes the system configurations for each system, detailed information about the workloads, and information about our main capacity planning tool, Server.Planner.

System configurations

The system configurations were selected based on a tradeoff between price and performance. Also, the systems represent a few different, but typical, customer production environments. This sample selection occurred by adjusting the memory, processor speeds, and number of processors, to name a few of the configuration options. Also, it shows newly released systems for the tests in order to bring attention to them. Individuals who perform the tests follow the agreements established within the NotesBench Consortium. An additional set of agreements were made as to which datasets to include within the [Domino Server.Planner](#) dataset, so as to realistically recommend what end users can use in their environment.

To run all four tests, we set up five systems -- each with a slightly different configuration, as follows:

System 1

- System: IBM 704
- CPU: One 200MHz Pentium Pro
- Hard Drives: 10 Spindles (4GB disk)
- Raid Level: RAID5
- Network: TCP/IP
- Memory: 512MB
- OS: Windows NT 4.0
- Domino: Release 4.6x

System 2

- System: IBM Netfinity 3500
- CPU: One 233MHz Pentium II
- Hard Drives: 2 Spindles (4.51GB disk)
- Raid Level: RAID5
- Network: TCP/IP
- Memory: 320MB
- OS: Windows NT 4.0
- Domino: Release 4.6x

System 3

- System: IBM 325
- CPU: One 233MHz Pentium II
- Hard Drives: 9 Spindles (4.51GB disk)
- Network: TCP/IP
- Memory: 384MB
- OS: Windows NT 4.0
- Domino: Release 4.6x

System 4

- System: IBM 330
- CPU: One 300MHz Pentium II
- Hard Drives: 10 Spindles (4.51GB disk)
- Raid Level: RAID5
- Network: TCP/IP
- Memory: 512MB
- OS: Windows NT 4.0
- Domino: Release 4.6

System 5

- System: IBM Netfinity 7000
- CPU: Two 200MHz Pentium Pro
- Hard Drives: 10 Spindles (4.5GB disk)
- Raid Level: RAID5
- Network: TCP/IP
- Memory: 1280MB
- OS: Windows NT 4.0
- Domino: Release 4.6

About the workloads

The workloads come from the NotesBench benchmarking environment, defined by the [NotesBench Consortium](#). The NotesBench Consortium is an independent, non-profit organization dedicated to providing Domino and Notes performance information to customers. The various workloads executed are

effectively held constant, as the number of users and the type of workload varies. Using these workloads also helped us capture and analyze the behind-the-scenes performance characteristics of various NotesBench workload runs.

The following is a list of each workload and its characteristics:

- **Mail:** This workload models a server for Notes mail users at sites that rely only on mail for communication.
- **Mail and Shared Database (sometimes called MailDB):** This workload models a server for active users who are only performing mail and simple shared database operations.
- **Shared Database (sometimes called DiscDB):** This workload models a server for active users who are only performing heavy shared database operations.
- **Groupware:** This workload is a capacity test for Notes users that process large amounts of information. This workload models sites that use the most resource-intensive features of Notes. It can be used to establish a worst-case, lower boundary on the maximum number of users a server can support. The Groupware test includes mail and shared database activity plus mail, with messages equal to 532KB, and users that replicate with the system under test.

This set of workloads is interesting to test because it highlights different user behaviors and gives an indication of the resource impact as the different workloads run on the server.

Server.Planner guidelines

Domino Server.Planner is a capacity-planning tool that suggests server configurations by weighing user requirements against the benchmark data. To learn more about Domino Server.Planner, read the Notes.net article ["Simulating your workload environment with Domino.Server.Planner"](#). The data in our tests generally falls within the guidelines for submission of test results for Server.Planner data. The NotesBench Consortium established these guidelines for submission of vendor data sets.

The original set of guidelines established by the NotesBench Consortium for submission of IBM datasets outline that:

- Submitted results should use 50 to 70 percent of the CPU
- The Server.Planner probe response time should not exceed 5 seconds (this is the end user response time)
- Maximum capacity for configurations have already been established by a separate NotesBench workload evaluation effort
- Auditors (such as KMDS Technical Associates) who are familiar with the NotesBench tool and Consortium agreements must review the test results

About our evaluation scenarios

Vendors ran the tests using the NotesBench tool with a specific workload and then stored the results in Server.Planner. While the test was in progress, the NT Performance Monitor (PerfMon), a performance monitoring tool that runs on Windows NT, also monitored the Domino server. Then, they took the test results stored in Server.Planner and the results from Perfmon, and made sure that they met the Server.Planner capacity planning acceptance criteria. We analyzed the performance and capacity results further by looking at the graphs included in this article.

In general, we analyzed the data from the tests by looking first at the amount of the CPU used, then at the amount of memory used, and lastly at the disk behavior (if that was possible). There are some variations in the workloads we used when we analyzed system performance. We ran at least two configurations for

each workload. We gathered limited Perfmon statistics and there were minor variations in the objects collected. As our testing process evolves, we may reevaluate which performance metrics we collect.

We realized in analyzing these tests that it is important to clearly identify within the charts what the average values are for the data collected and what the approximate ceilings are for data results. By *ceiling*, we mean the highest amount or maximum value of, for example, memory you can use and still make practical use of the system. Identifying the system's performance in relation to the ceiling was also important because it helped us identify the growth potential of the system, and it can be an indication of the end user's response time. In addition, when we found the ceiling in a test scenario, we realized that often reaching or exceeding a recommended maximum value is really only a concern when you look at the entire system. This meant that we needed to test and evaluate other areas of performance before making any recommendations or determining that there really was a bottleneck. This has become particularly apparent in the analysis of the pages-per-second metric discussed later in the article.

Evaluation 1: Evaluating the amount of the CPU used in different workloads

In our first evaluation scenario, we looked at how much of the CPU the system used at each of the workloads. We used the Windows NT Perfmon utility to capture the information. When we configured multiple CPUs, we used the System Object metric, otherwise known as the % Total Processor Time (this measures processor utilization for all the processors combined). The NT PerfMon utility defines this metric as follows, "The % Total Processor Time is the average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%. If all processors are 50% busy, this is 50%. And, if 1/4th of the processors are 100% busy, this is 25%. It can be viewed as the fraction of the time spent doing useful work."

You can see the results of this test in the sidebar, "[Amount of CPU used test results.](#)"

Analyzing the tests

Based on the information in the charts and from other test data, we made the following observations:

- The maximum amount of the processor used for a production environment is in the 75 percent range. This is consistent with the NotesBench Consortium recommendation of about 70 percent.
- The average amount of the processor used across all the workloads is almost the same and is in the 30 percent range. This can be misleading because the range for the maximum number of users captured changes for each of the workloads, but the data included falls within the guidelines which require a maximum at 75 percent. When we took the data that was up to 75 percent and averaged it across the datapoints, the chart average was 30 percent. This illustrates that the user profile for the individual workload impacts the amount of the processor used based on the number of users that can execute that workload with an associated response time. The aggregate workloads used up a similar amount of CPU (by definition of the test) but the user counts are different because of the specifics of the individual user activity. You should consider this when you assess what your users are doing.

There are several different ways to analyze and compare the data from these tests, here are some of the ways:

- You can compare the results based on a given user count value. For example, you can see that at 1200 Mail users, the CPUs are 40 percent (20 percent times two processors, in absolute terms) used on system 7000. While for system 325 to achieve the same number of users, it uses 65 percent of the CPU. When you compare system 7000 to system 325, the amount of the processor used (40 percent versus 65 percent) is an over 50 percent gain in processor requirements.
- You can compare the results based on how much of the CPU the system used. With the Groupware workload, 375 users running on system 7000 use 45 percent of the CPU. Looking at system 704, 45 percent of the CPU can only handle 250 users. To understand the increase in capacity: $(375-250)/250$ is a 50 percent increase in the capacity of system 7000 over system 704. There is an additional contribution to performance as a result of the additional processor in system 7000.

System 7000 was a more capable configuration, if you look at the results from the point of view of how much of the CPU was used. There was a decrease in the amount of the CPU used with system 7000 as opposed to other system configurations. Also, for the Groupware workload, after 150 users, performance was appreciably better for system 7000. The same type of differentiating behavior appeared with some of the other workloads: Mail at 5-600 users and Mail and Shared Database at 600 users. The only workload that did not have better performance for system 7000 was the Shared Database workload.

Similarly, the results in the charts emphasize that the amount of memory available on the system has a large impact upon Notes. Additional memory and processors had a positive performance impact on the Groupware, Mail, and Mail and Shared Database workloads. This did not consistently hold true for the Shared Database (DiscDB) workload. In this workload, the disk I/O was the determining factor for system performance (as there was no major performance difference between system 330 and system 7000). When reviewing results for the Shared Discussion Database workload, it appears that either the disk I/O or network interface is the bottleneck, as variations on the amount of memory or number of processors didn't have a significant impact on the performance of these machines.

When adding on the additional memory and processor for system 7000, the following processor improvements were generally demonstrated:

- Groupware workload: 20 percent less CPU
- Mail workload: 15 to 20 percent less CPU
- Shared Mail and Database workload: 15 to 30 percent less CPU
- Database workload: 0 to 10 percent less CPU

These statistics show, that on average, system 7000 performed 20 percent better with the Groupware workload than the other systems we evaluated. We attribute this to the additional processors and memory.

Evaluation 2: Evaluating the amount of the memory used in different workloads

In our second evaluation scenario, we looked at how much memory the system used at each of the workloads. We used the Windows NT Perfmon utility to capture the information.

To calculate the memory used, we subtracted the total amount of memory available on the system from the memory object, Available Bytes Counter. NT's PerfMon Utility defines the Available Bytes Counter as, "the size of the virtual memory currently on the Zeroed, Free, and Standby lists. Zeroed and Free memory is ready for use, with Zeroed memory cleared to zeros. Standby memory is memory removed from a process's Working Set but still available." Therefore, the Memory Used metric represents the space required by the Domino Server as well as the underlying platform operating system.

You can see the results of this test in the sidebar, "[Amount of memory used test results](#)."

Analyzing the tests

Based on the information in the charts and from other test data, we made the following observations:

- In general, you should leave no less than 4MB of memory free on any system for the Domino server.
- The average memory used is higher for Mail workloads and Mail and Shared Database workloads, but also keep in mind that those two workloads had the highest user counts (getting up to 1800 users and thus having more databases and file locks open).
- In general, system 7000 uses more memory for the same workloads. It appears that if more memory is available, the internal calculation logic takes advantage of the additional space.
- We reached the maximum amount of memory used in a few of the configurations (these configurations had lower amounts of memory). If additional datapoints existed after this user count, no additional memory was available to process the additional users.

Based on the information in the charts, you can start to put some of the metrics together to form guidelines for system requirements.

Evaluation 3: Evaluating the probe response time for different workloads

In our third evaluation, we looked at the probe response time for each of the workloads. To test this, we used the Server.Planner process, where a "probe" task executes every minute, opening and closing the shared discussion database. This metric simulates and is the best indicator of end user response time. The probe results represent the "worst case" analysis for the Groupware, Shared Mail and Database, and Shared Database workloads, as the shared database that the workload interacts with is also opened by the probe task. The probe task for the Mail workload doesn't report the same level of contention on the shared database because the database is not used as part of the workload. However, the probe task still gives an indication of overall Domino server performance.

You can see the results of this test in the sidebar, "[Probe response time test results](#)."

Analyzing the tests

Based on the information in the charts and from other test data, we divided the probe response time values into a rating system. We used the classifications used by Server.Planner and we considered:

- A one second response time as fast
- A one to three second response time as medium
- A three to five second response time as slow

End users find the probe response time tests interesting and they benefit the most from looking at this information. This is because it shows them what they can expect when they use the system.

When we reviewed the average probe responses represented on each chart, we noticed that the Groupware workload had the highest average probe response. This is why the Groupware workload rates as our most intensive workload. It's interesting to evaluate the Probe results for the Groupware workload by comparing the results for system 7000 and system 704. At the user count values of approximately 325 (for example, more than 300 and less than 350) and below, system 704 had the best system performance. This led us to conclude that the overhead of the extra processor and memory found in system 7000 could have an impact until there are more than 325 users. However, even with this impact the response time is still acceptable. This performance behavior did not occur with any other workload or for any other system.

It is interesting to note the small slope of change for system 704. This occurs, for example, between 250 and 300 users, as compared to the steep slope of change for 300 to 350 users. This type of behavior is important to keep in mind whenever you make a generalization or prediction about system performance. This type of behavior caused us to make sure we knew the upper and lower boundaries of a benchmark workload and limited the estimated values to those that were between the known values.

When we reviewed the results for the Mail and Mail and Shared Database workloads, we found a consistent sub-second probe response range for the various user count data points. This showed that there was a low rate of change. These workloads also showed us that even at the highest user counts the system was not strained in terms of end user response time.

Analysis of the probe response is a good way to establish equitable relationships between the different workloads, and it is a good way to understand how certain types of users perform. Server.Planner's internal decision making logic follows this premise. As a result you should use the probe response time, not the Notesmark (TPM) rate, for evaluation and comparison purposes.

Evaluation 4: Evaluating the disk I/O of different workloads

In our fourth evaluation, we analyzed the disk I/O for each of the workloads. We used the Windows NT Perfmon utility to perform the tests. We analyzed the logical disk object, average disk queue length, and

percent of disk time. Average disk queue length is defined by NT's PerfMon Utility as follows, "Avg. Disk Queue Length is the average number of both read and write requests that were queued for the selected disk during the sample interval." Percent Disk Time is defined from the same source as follows, "Disk Time is the percentage of elapsed time that the selected disk drive is busy servicing read or write requests."

You can see the results of this test in the sidebar, "[Disk I/O test results.](#)"

Analyzing the tests

Based on the information in the charts and from other test data, we made the following observations:

- The information was very consistent across each workload, even though the Y axis was different in each chart.
- Since we did not reach the more dangerous levels of a disk I/O queue that was greater than two, *this configuration* was not I/O bound, meaning that the server and the disk I/O system could handle the amount of data in the queue. Unfortunately, this metric was not gathered for the earlier workloads for analysis. It only existed for system 7000.

Evaluation 5: Evaluating the paging per second of different workloads

The fifth metric we looked at was the paging per second for each of the workloads. We used the Windows NT Perfmon utility to perform the tests. We used the Memory Object, Pages per Second Counter, to measure performance. We also determined the paging per second by looking at the number of applications and processes contending for the memory allocated for paging. The size of the page file is influenced by the amount of total memory available and the amount of memory required by the application and processes. The metric pages per second is defined by NT's PerfMon Utility as follows, "Pages/ sec is the number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of the Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This value also includes the pages to or from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.

You can see the results of this test in the sidebar, "[Paging per second test results.](#)"

At the beginning of the article we planned to evaluate four metrics in depth. We chose to include this metric and the ones that follow to show examples of specific information that we gathered and analyzed, which was enlightening. These metrics were not our highest priority in our evaluations, but you should consider this information when looking at server performance.

Analyzing the tests

The basic guideline for satisfactory performance is that the pages per second should not exceed 15 to 20. The graph above shows that for mail intensive workloads (Mail and Shared Database) the pages per second exceeded the guideline. However, if you look at the probe response time and the amount of the processor used, (this information is not included in this article for the Mail and Shared Database workload) you can see that these systems fall into the acceptable range. We found similar results when we evaluated the Mail workload.

This test clearly shows that focusing solely on the test results in one area will not give you a clear indication of the overall performance of your system. When you look at the pages per second, you also need to look at the amount of memory that is available and whether or not the memory increases or decreases over time. In this case, since the amount of memory didn't decrease, the Domino server is most likely performing a lot of disk I/O. After digging deeper into this issue, we found that this is a data intensive activity, the data is not in the cache, and results in a high paging rate.

Evaluation 6: Evaluating the Notesmark of different workloads

In our final evaluation, we looked at the Notesmark for each of the workloads. Notesmark is the number of transactions executed by the Domino Server.

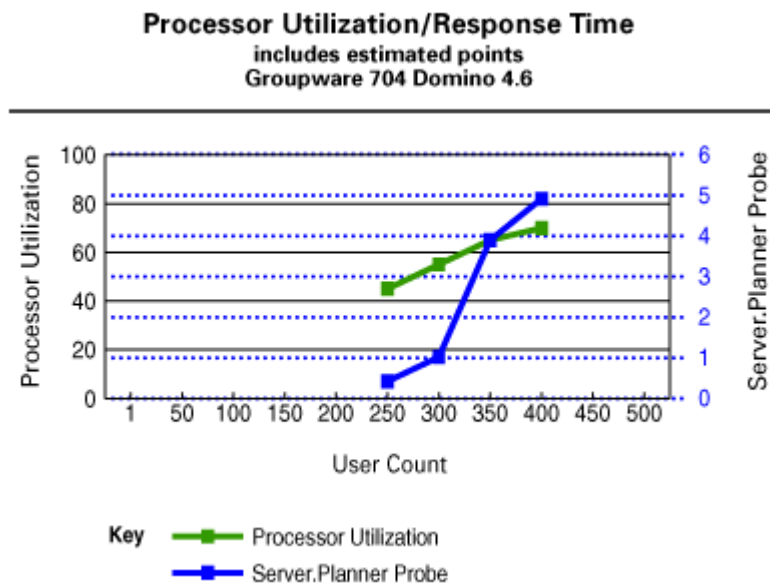
You can see the results of this test in the sidebar, "[Notesmark test results.](#)"

Analyzing the tests

When you review the Notesmark values, remember that all Domino server transactions are not equal transactions. As a result, you should *not* compare average Notesmark results across workloads. The interesting point you should notice with these tests is that the transaction rate is very consistent across the different workloads. This means that the same server execution results were consistent. Each configuration performed the same amount of work at each load level. The Domino server did not bottleneck itself. This same behavior occurred across all workloads, for all system configurations, and for all user count ranges.

Combining some of our evaluations

Notice that in the following graphic, as the amount of the processor used increases with the increasing user load, it affects response time. The response time is the Server.Planner Probe response time.



The graph above represents, for system 704, a combination of the amount of the processor used evaluation and the probe response time evaluation.

Overall conclusions based on all the tests

When we looked at all the machine configurations, their performance, and the capacity used, we found that the additional resources available on the machines (memory in particular) brought additional benefits. We are more proficient in analyzing the amount of the processor used and the amount of memory used, and we are now focusing more on the amount of disk I/O used (performance and configuration). We also found that in presenting the general methodology for analyzing platform specific statistics, Domino statistics as well as NotesBench and Server.Planner metrics, we gained a greater understanding of the similarities and differences of the resource requirements of the various workloads and the associated system performance and capacity requirements under varying workloads.

The test results led us to draw some more particular conclusions. We wanted to understand the impact of the fact that system 7000 had both more memory and processor capacity than the other systems. We wanted to understand this impact. This is consistent with our observations that "balanced" systems include more (although slower) processors, but are coupled with larger level two caches (the system 7000 had 1MB of cache versus 384MB). The additional CPUs, amount of level two cache, and memory provided better capacity and response time at specific user loads.

We also looked at the rate at which the server executed Notesmarks (Domino server transactions). We learned that, as the number of users increased, each configuration performed the same amount of work in terms of server transactions at each load level. This was a good indication that within a specific workload, you can compare Notesmarks to ensure that each user performs the same amount of work. When we saw the average number of transactions per minute TPM/user value decrease, we knew that we exceeded the practical user load range for a particular configuration (normally, at this same point, response time would change dramatically).

In addition, the test results showed us that we needed to identify the upper bound of each workload in order to come up with the options we had for dealing with that upper threshold. These options included addressing a bottleneck situation (and taking corrective action), defining valid operating characteristics for production environments, and continuing to test other workloads to further identify if there was an issue to address. Even though different workloads have different execution profiles, analyzing these can help you decide which platforms perform best (this analysis isn't covered in this article). You can also see how the variations in system configurations affect the end user's response time.

Through these benchmarking statistics, you can understand the benefits and the tradeoffs for the specific workloads. For example, you can understand the improvements in the amount of the CPU used when you compare a single CPU configuration with 384MB to 512MB of memory to a two CPU configuration with 1380MB of memory by looking at the following improvements in the amount of the CPU used:

- Groupware workload: 20 percent
- Mail workload: 15 to 20 percent
- Shared Mail and Database workload: 15 to 30 percent
- Database workload: 0 to 10 percent

From these tests, we've learned that the standard set of performance guidelines developed to apply to a generic system configuration and set of applications does not apply to more robust applications, such as Domino. This is particularly apparent with the Pages per second metric. The industry standard guideline for reviewing pages per second is within the 15 to 20 range. When meeting that range, it is important to keep delving deeper by looking at other metrics to see if there really is a problem. For example, the value for pages per second may only be a warning flag. The pages per second evaluation shows that mail-related workloads demonstrate a higher than average pages per second value. This is attributed to the many, and small number of writes the system performs to complete the task. Reviewing the amount of the processor used and the probe (end-user) response time reveals that there are no performance problems. This is important to keep in mind when reviewing the performance characteristics of a production mail server.

There are several ways you can interpret the data in the tests. For example, when we reviewed the rate at which the amount of the CPU used changed as the user count increased, we saw that when the number of users were equal to 200, the amount of the CPU used averaged to .025 per user. The results when there were 500 users were .05 per user. This was a 50 percent increase in the capacity requirements when we increased the number of users from 200 to 500. You can apply this type of analysis and calculation to any of the other datapoints and charts.

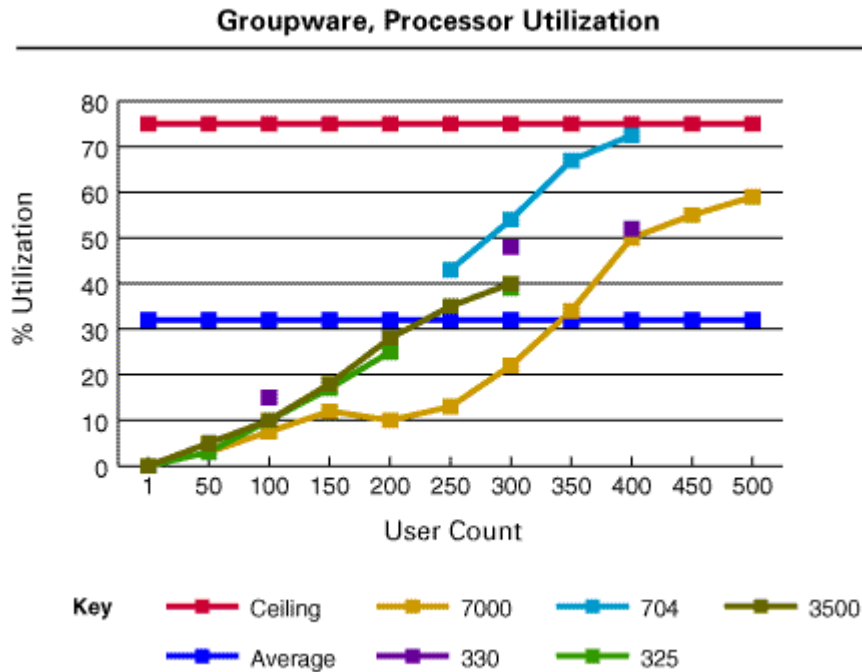
The tests in this article show that you can't judge Domino performance based on a limited number of tests. You need to look at all the test scenarios that include varying workloads, numbers of users, and system configurations. Then you also have to vary the ways that you analyze and interpret the data. If you do this,

or request that your vendors do this, following the examples provided in this article, you will see the big picture of Domino performance.

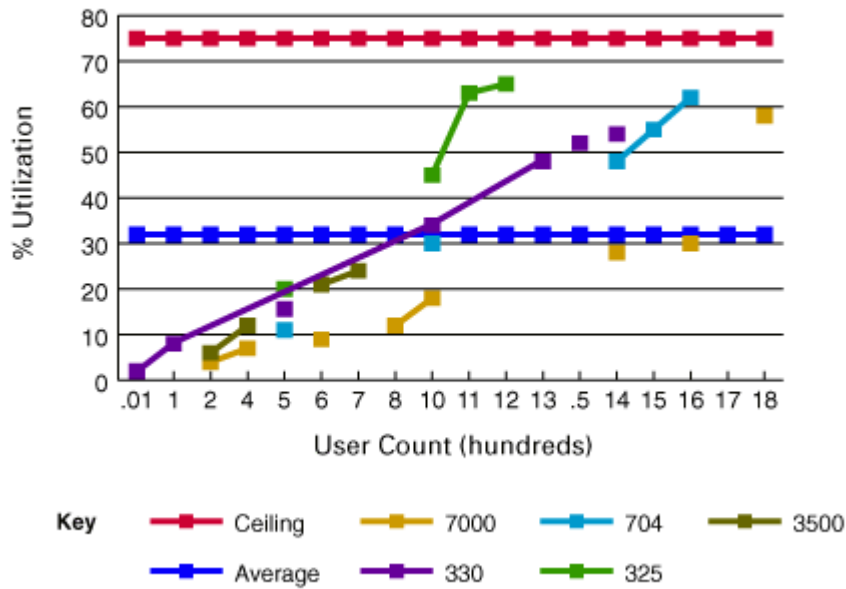
Amount of CPU used -- test results (sidebar)

The information in the charts below represents a subset of the entire information we collected. In this article, for each test we tried to present both ends of the spectrum, with the Mail workload as the light end and the Groupware workload as the heavier end.

Notice that the X-axis in each chart shows a different number of users that fit within the recommended guidelines.



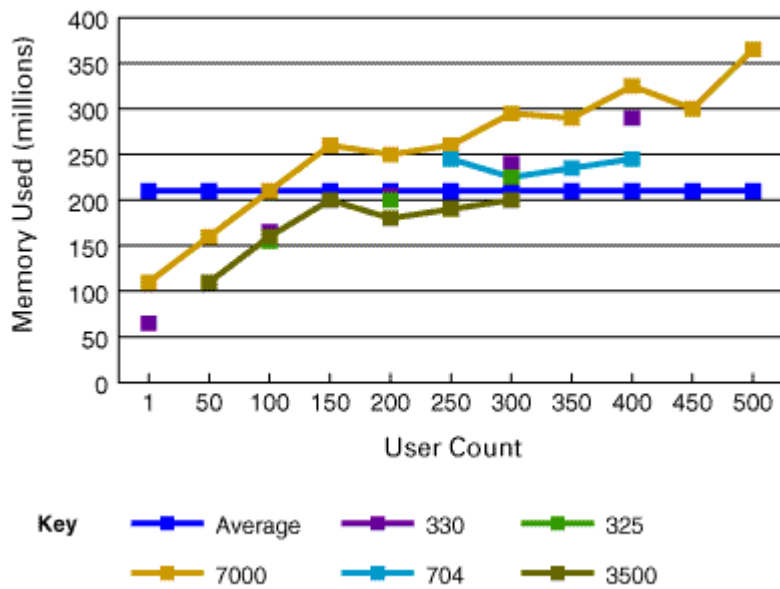
Mail, Processor Utilization



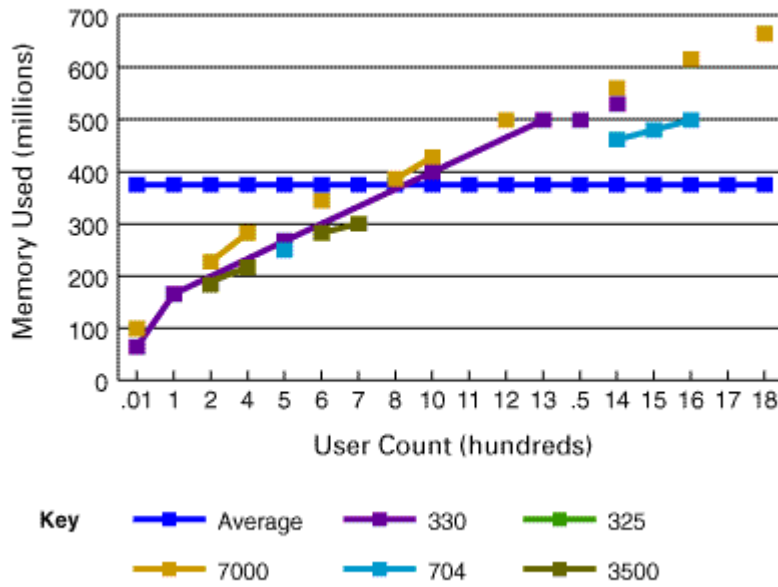
Amount of memory used -- test results (sidebar)

The information in the charts below represents a subset of the entire information we collected. In this article, for each test we tried to present both ends of the spectrum, with the Mail workload as the light end and the Groupware workload as the heavier end.

Groupware, Memory Used



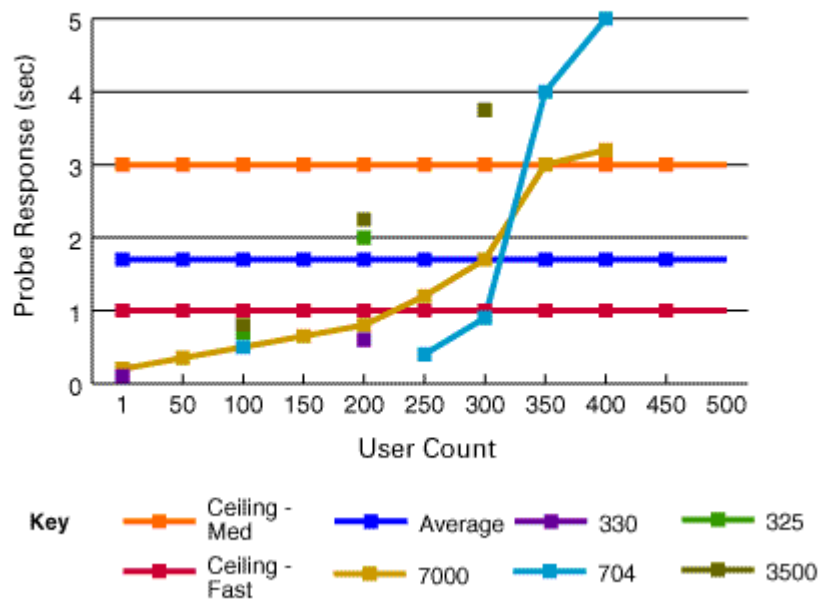
Mail: Memory Used



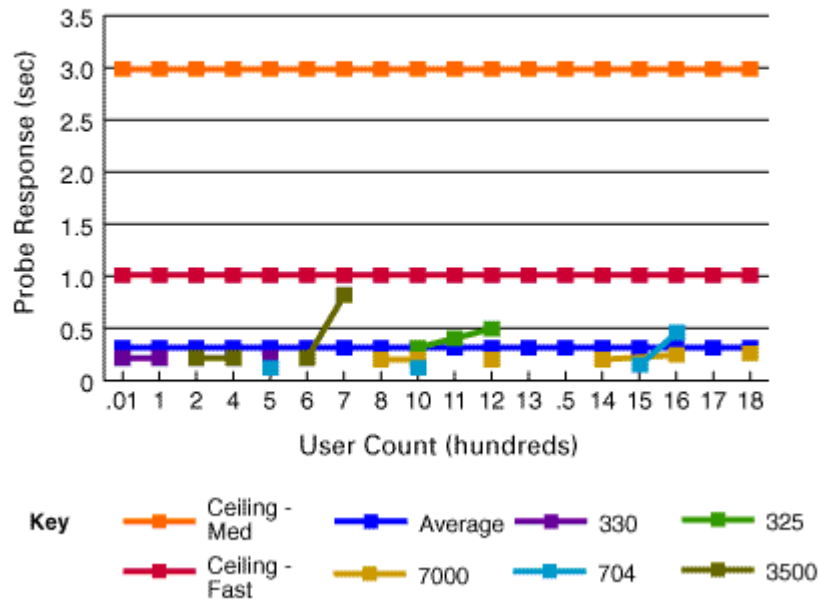
Probe response time - test results (sidebar)

The information in the charts below represents a subset of the entire information we collected. In this article, for each test we tried to present both ends of the spectrum, with the Mail workload as the light end and the Groupware workload as the heavier end.

Groupware: Probe Response



Mail: Probe Response

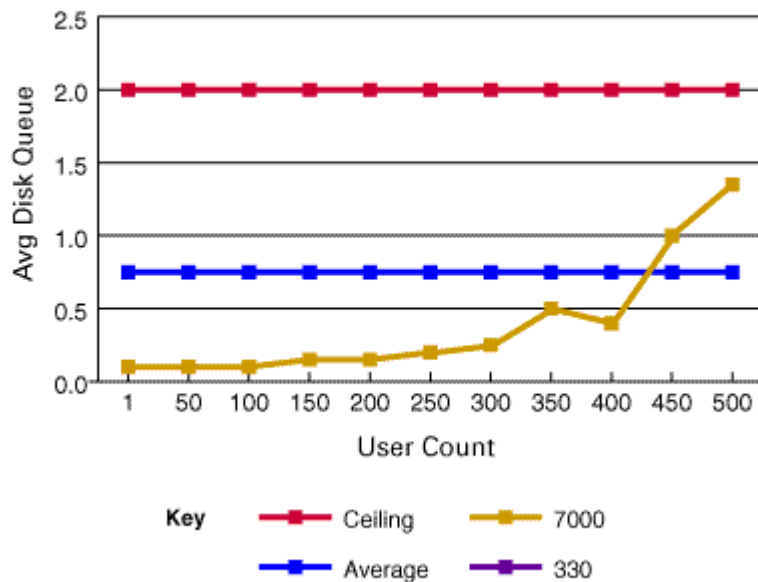


Disk I/O - test results (sidebar)

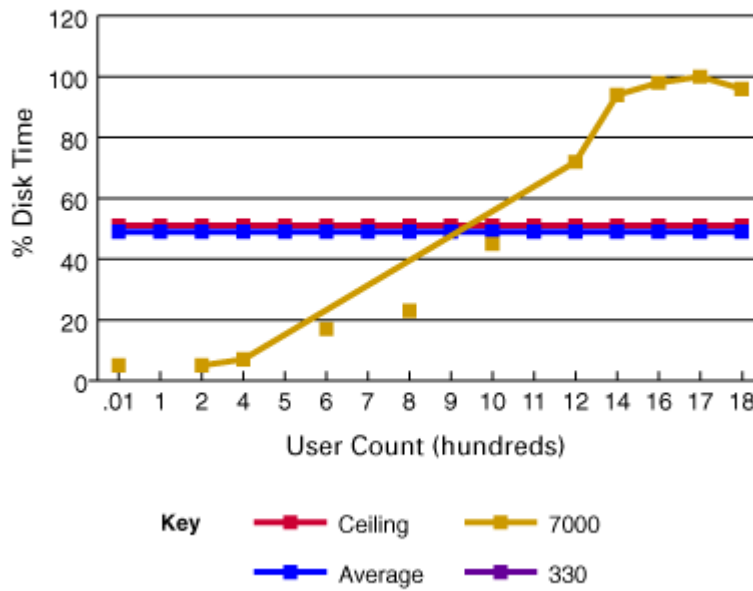
The information in the charts below represents a subset of the entire information we collected. In this article, for each test we tried to present both ends of the spectrum, with the Mail workload as the light end and the Groupware workload as the heavier end.

Notice that two different metrics, as well as workloads, appear in the graphs below:

Groupware: Average Disk Queue



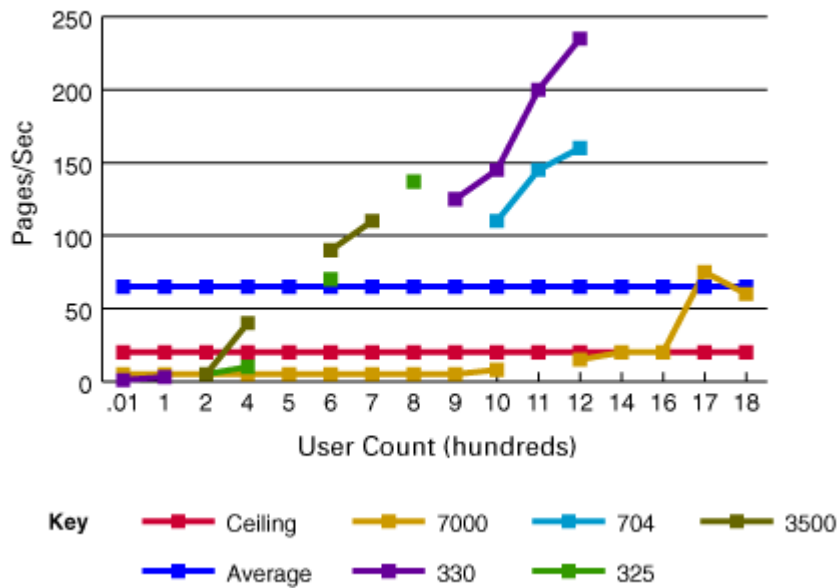
Mail & Shared Database: % Disk Time



Paging per second - test results (sidebar)

The following graph shows the pages per second as the user count increases.

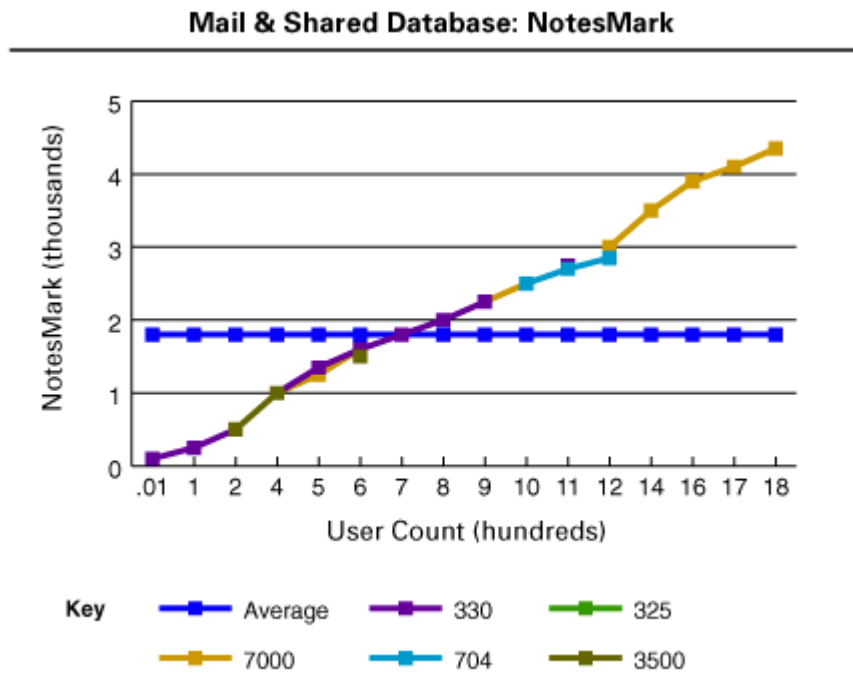
Mail & Shared Database: Pages/Sec



Notesmark - test results (sidebar)

The Notesmark test results

The following graph shows the Notesmark as the user count increases.



ABOUT CAROL

Carol Zimmet started working at Iris in 1994. She is responsible for evaluating performance and performance tool development on the server team. Carol continues to search for the one-step solution to everyone's performance problems. Carol enjoys bicycling with her kids in the trailer and playing racquetball. She has a longing to return to stained glass!

Copyright 1998 Iris Associates, Inc. all rights reserved.