Introduction to Domino Performance Tuning

by
Lou
Bradbard

Level: Beginner
Works with: Domino 5.0
Updated: 09/01/2000

There probably isn't a Domino administrator out there who doesn't wish his or her Domino servers were running more efficiently or who can't imagine performance improvements. After all, getting the most from your systems—even as the demands on those systems constantly change and evolve—is a critical part of your job.

This article introduces you to the basics of Notes/Domino performance analysis concepts and tools. At the end of this article, you will find links to more advanced articles and a reading list of performance-related online resources. This article assumes a general knowledge of computer systems and of Domino server administration.

What is performance tuning?

Performance tuning has one overall goal: To get the most from your hardware. But how can you make sure that you are getting maximum utilization of your Domino server's resources?

Performance tuning involves identifying bottlenecks at the operating system and application levels. It includes analyzing and tuning your system hardware and software for optimal performance of all major components including memory, CPUs, disk drives, and the network. For example:

- Processors affect the Indexer speed, Replicator speed, number of maximum possible database transactions, and number of add-ins that can run in parallel.
- Disk access rates and configurations affect the speed of database and view opening and of opening and navigating a collection (view index).
- Memory affects the maximum possible simultaneous Notes client connections (sessions), size of caches, and server add-in task performance (because of less paging to disk).

How to approach performance tuning

You should start with a top-down approach to performance tuning. Before tuning your Domino servers for performance, you have to make sure that your network components are operating efficiently. Network bottlenecks can come from many sources and can often be tricky to isolate. Routers, gateways, firewalls, and network collisions could all be a source of performance bottlenecks. You may also need to use a sniffer device to look for problems with network traffic.

You should then make sure that your Domino systems are using the network efficiently, Check the Network Utilization, Total Bytes sent/second, and on Windows under the Processor Object, check the Percent DPC (Deferred Process Calls) Time and DPCs Queued/Second. These will give you an idea of how much processor time is spent servicing the network and if the DPC queue is growing.

Once the network is tuned, you are ready to look at the performance of your Domino servers. How involved or complex Domino server performance tuning is depends on the size and complexity of your Domino network. If you have just one Domino server with one disk drive and one central processing unit (CPU), your performance tuning options are fairly basic:

- Design applications, forms, views, and pages for maximum performance

- Use redirection URLs
- Use faster alternatives to formulas, such as computed subforms and JavaScript
- Turn off unneeded Domino server tasks
- Optimize LotusScript and Java code

See this Lotus White Paper, **Maximizing Application and Server Performance in Domino**, for examples of these performance tweaks.

Performance tuning options on Domino servers with more than one CPU and multiple hard drives is obviously more involved. All of the above techniques apply to such systems, but in addition, you need to make sure that the Domino workload is efficiently distributed and balanced among the physical disk drives. Because the Domino server can be very disk intensive, configuration of disk drives, and arrays of hard drives called RAID drives, can greatly enhance server performance.

One of the keys to Domino server performance on anything but the most simple system can be summed up as follows:

Spread out the work over the hard disks and processors

Performance tuning of these systems involves setting up the disk drive partitions, and arranging the system, program, and data directories to distribute the workload across the system as evenly as possible. (See the *Iris Today* article **Optimizing server performance: I/O subsystems**.) These systems often require some experimenting with and adjustment of hardware and software before you reach optimal performance. Performance testing tools let you experiment with different system configurations, such as RAID array stripe size and hard disk file format, to see which combinations result in better performance.

Domino performance testing tools

Many performance bottlenecks will only reveal themselves when the server is under high load, so server performance tools are used to simulate a load of mail messages or database activity. Both the Server.Load and NotesBench performance tools that are developed with Domino run on Notes client systems and use workload scripts to control simulated user activity against a server.

NotesBench and Server.Load are both built using the same server performance testing engine that has been expanded over the years to include Internet mail, discussion databases, and calendar and scheduling. Server.Load was created as a simpler, more flexible alternative to NotesBench.

Server.Load

Server.Load uses a graphical user interface (GUI) to control workload operation. Server.Load is best for small and mid-sized evaluations and is included on the Domino server CD ROM. It:

- Collects 150 performance metrics from the system under test.
- Runs built-in and customized workload scripts.

NotesBench

NotesBench is a powerful command line based tool available only to members of the NotesBench Consortium and others who are individuals trained in performance analysis. It requires more Domino system knowledge and setup time than Server.Load and is not designed for small-scale evaluations. It:

- Generates a number of users-per-second system rating with a response time.
- Runs a standard set of built-in workloads.

- Gets all workload control information from the NOTES.INI file.
- Synchronizes multiple test drivers in a parent/child relationship.

Which is right for you?
Both NotesBench and Server.Load can be used to run workloads that will expose the bottlenecks in your server. However, the fastest way to get started with Domino performance analysis is to use Server.Load. You already have a copy of the program and documentation on your Domino server CD!

If you are planning a large-scale deployment, we recommend you enlist the services of a **NotesBench Consortium** member. They have experience configuring and running NotesBench and can customize a NotesBench run on their platforms that will conform to your requirements.

The NotesMark rating, analogous to a horsepower rating, allows you to compare Domino scalability on different hardware platforms. Using NotesBench to get a NotesMark rating requires more setup and a stricter set of guidelines than running workloads with Server.Load to expose system bottlenecks.

Understanding the role of benchmark servers

A benchmark server is a Domino server set up for testing with Server.Load, NotesBench, or third party tools. This server is created in an isolated Notes domain to contain the heavy load placed on the testing systems.

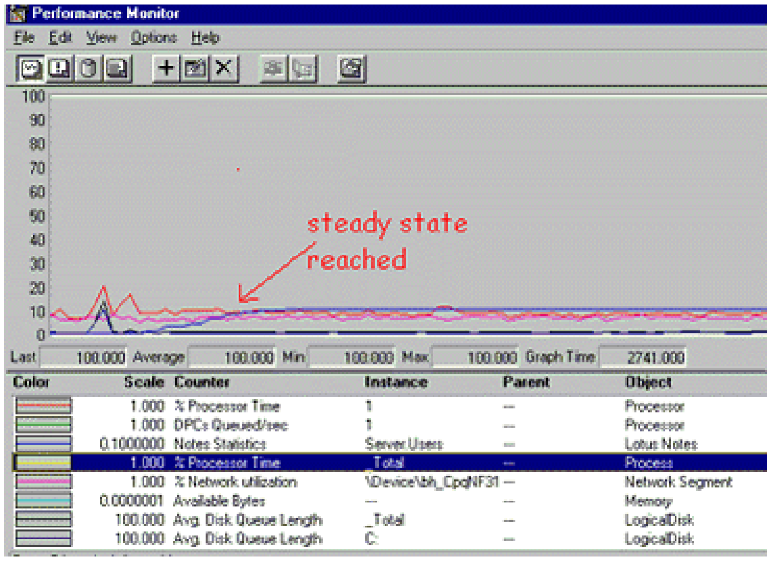
Hundreds, or even thousands, of mail files are created along with related Person documents by setup scripts and agents provided with the performance testing tools. This lets you better understand how your system will behave under high load, to get an idea of system limitations, and to head off problems in production servers before those servers are overloaded.

After the tests are run and the information is analyzed, the benchmark server can be converted back to a production server or can become a testing server with another focus.

What is steady state?

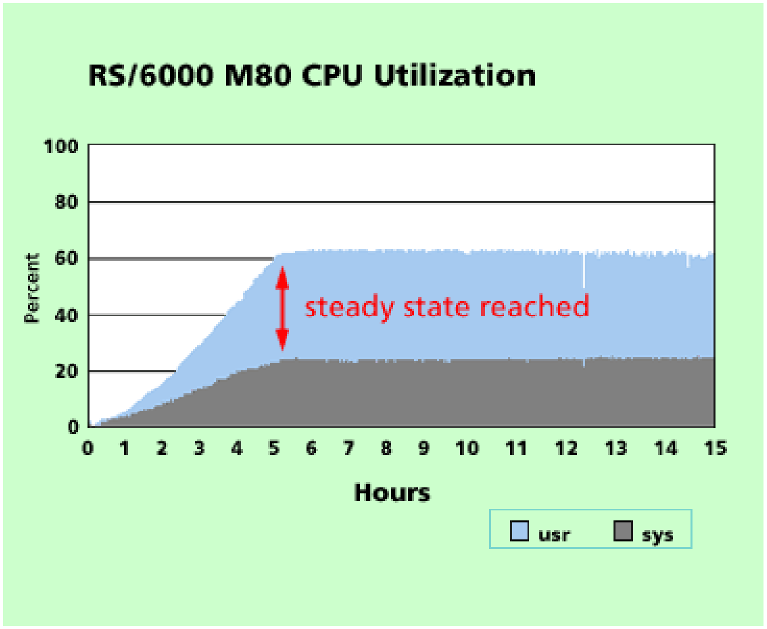
Steady state means the system under test has reached equilibrium and that the system could sustain the test load for as long as needed because memory, CPU, and disk I/O (input/output) are not increasing or are near critical limits.

The graphs below show servers running a Server.Load workload overnight that has reached steady state. The first graph is for a Windows NT system, and the second is for an RS/6000 system. Note how activity for the major system components has leveled off after the initial ramp-up. If activity of any system component is increasing throughout the test run, the system has not reached steady state.



Last 100.000 Average 100.000 Min 100.000 Max 100.000 Graph Time 2741.000

Color	Scale	Counter	Instance	Parent	Object
▭	1.000	% Processor Time	1	—	Processor
▭	1.000	DPCs Queued/sec	1	—	Processor
▭	0.1000000	Notes Statistics	Server Users	—	Lotus Notes
▭	1.000	% Processor Time	_Total	—	Process
▭	1.000	% Network utilization	\Device\bh_CpqNF31	—	Network Segment
▭	0.0000001	Available Bytes	—	—	Memory
▭	100.000	Avg. Disk Queue Length	_Total	—	LogicalDisk
▭	100.000	Avg. Disk Queue Length	C:	—	LogicalDisk

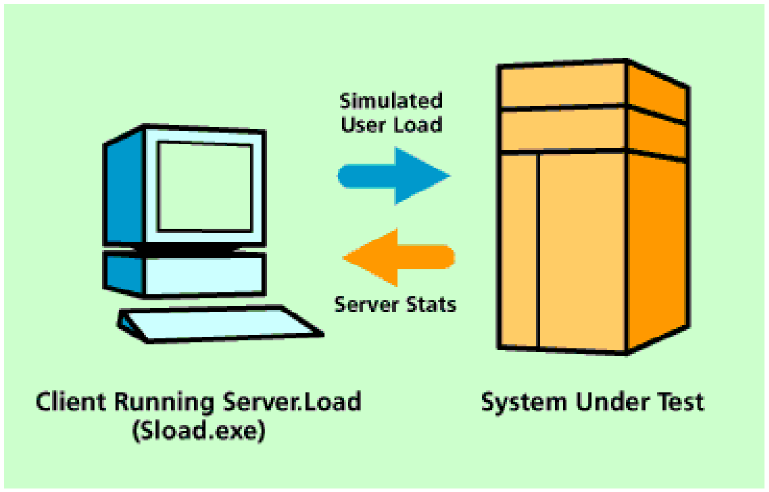


Collecting performance data with system monitoring tools

The operating systems that Domino runs on keep track of everything that is going on inside the system. Programs called monitors gather the requested data from the statistic counters and record it to a text file or display it as a graph. These data will show which system components are at or near maximum capacity and allow you to measure how adjustments to the system have affected performance. Microsoft-based platforms collect data with the Perfmon utility. UNIX platforms use command line utilities like IOSTAT and VMSTAT. Other tools and methods are available on the other Domino server platforms.

How Domino performance testing works

The illustration below shows how the data flows between test drivers and the system under test. The blue arrow represents the requests that simulate users on the server. The orange arrow represents the statistics retrieved from the server that are captured in the Server.Load metrics file. Server.Load also captures the text from the Script Output Monitor window into a results file, which can be searched for timeouts and error messages.



To run performance tests and collect data with monitoring tools:
1. Install Server.Load on the test clients.
2. Set up performance tests using special workloads and agents to create mail files and Person documents for test users.
3. Run the performance workload to create a heavy workload on the system under test until it reaches steady state.
4. View and graph the server statistics collected from Perfmon on Windows server platforms and the Platform Stats on Windows and Solaris platforms in the Statrep.nsf database.
5. Collect the response time data from the test driver client metrics file.
6. See the section below to learn how to interpret the collected data.

Analyzing test results: The most likely suspects

When looking for data bottlenecks, you should start at the central system components. Just as traffic backs up at some intersections only in heavy traffic, data can back up in your system in critical areas only when under heavy load. Performance testing clogs the weak points in your system by creating an e-mail "rush hour." Then the operating system performance statistics are used like traffic webcams to spy on the data flowing through the system.

The key thing to remember is that every subsystem is associated with a queue, and it is the queue length that needs to be monitored. In the case of the CPU, it is the System Processor Queue Length, for Disk it is the Average Queue Length, and so on. You can look for bottlenecks in a logical order.

The server CPU

The first place to check for bottlenecks is the server CPU. Even systems with infinite disk space and RAM are limited by the amount of data that can be pumped through the CPU. If the percent busy is close to or at 100 percent, then the CPU could be limiting the system. However, you can be at 100 percent CPU, and the CPU does not need to be the bottleneck if the System's Processor Queue Length does not stay beyond a value of 2 (on Windows NT) for a sustained period. Note that you can have a server hit 100 percent CPU, but it should not be pegged at 100 percent CPU for longer than one minute.

Ideally, a server should have some room for growth and once it is pushed to 80 percent processor time, you should not add any more registered users or server tasks.

Available memory

The next stat to check is available memory. If available RAM is approaching zero at any time during load, your system performance is limited by RAM. Look at the physical RAM and memory committed bytes.

Disk input and output
Next look at the disk input and output. This is where it gets interesting! If you have plenty of RAM and CPU cycles available, but the system is still taking a long time to open databases, look at the input/output of the hard disks on Windows NT by checking the Drive Queue Lengths and the Average Disk Reads/Writes per second, Percent Disk Time, and Percent Disk Bytes transfer time. On UNIX, look at the Logical Disk Percent Time and Logical Disk Service Time.

Where to go from here

The following list includes articles and other resources that explain how to use these tools to uncover performance bottlenecks in Domino servers and how to use these techniques and metrics for performance tuning and capacity planning.

Introductory performance articles
Top 10 ways to improve your Domino server performance
If you only read one more article on Domino performance it should be this excellent summary!

Command Performance and **Command Performance 2**
In these interviews, learn about the Iris Domino Peformance team, our mission, and a take a peek at the future of Domino performance.

Optimizing server performance: Handling the curves like a pro
This article gives you a look at Domino's performance on various system configurations while executing various test workloads. You can examine the results of six different evaluation scenarios to see how each configuration measured up.

Optimizing Domino server performance
Optimizing server performance: Port encryption and Buffer Pool settings
This article introduces you to the tools to use, and shows the impact of how port encryption and NSF_BUFFER_POOL_SIZE settings affect server performance, through an analysis of several different test scenarios.

Optimizing server performance: HTTP Threads settings
HTTP threads are like the fuel in your carburetor—not too much or too little, or performance is affected. Make sure your Domino Web server is operating at peak efficiency.

Optimizing server performance: I/O subsystems
Learn how to increase performance using transaction logging, multiple mail.boxes, and multiple disk drive controllers.

Notes.net Exposed: Improving Web site performance
Track the performance of Notes.net in this real-world example of a high-volume Web site. Learn how we improved performance with upgrades and system configuration changes.

Notes from Support: Does more memory really help?
This article provides an overview of how memory usage affects Domino server performance. It illustrates how server memory shortage can slow your server with excessive disk I/O operations.

Optimizing server performance: Transaction logging
This article presents an outline of transaction logging performance benefits and details of RAID drive configurations and the R5Mail test workload.

Optimizing server performance: Semaphores (Part 1)
The first of two parts, this article describes why semaphore timeouts occur and how to troubleshoot the reasons behind them.

Optimizing server performance: Semaphores (Part 2)
This article builds on the advice from the first article and presents additional troubleshooting techniques, this time based on real-life experiences.

Maximizing Application and Server Performance in Domino
This white paper focuses on what you can do to maximize the performance of Domino in terms of Web applications from the application development perspective, and overall server performance from the server administration perspective.

Performance Considerations for Domino Applications
This IBM Redbook focuses on the application aspects of performance but does also briefly touch on some general server tuning topics.

Optimizing Domino server cluster performance
Optimizing server performance: Domino clusters (Part 1)
This article first introduces you to Domino clusters, focusing on those aspects that relate to performance, and then discusses performance testing of Domino R4.6 clusters.

Optimizing server performance: Domino clusters (Part 2)
This article expands the discussion to performance tests of Domino R5 clusters, including data on the Internet Cluster Manager and cluster replication.

ABOUT THE AUTHOR
Lou Bradbard started working at Iris in 1995. He has previously done quality assurance testing for the Domino server and programmability teams. He joined the Performance team in November 1999 and is currently involved with testing of the various performance tools developed by the group. He is not ashamed of his love for the Apple Macintosh.