

Domino on Solaris: Common Tuning Tips

Thirteenth Edition, January 2009

Introduction

This paper suggests adjustments that may improve the performance of your IBM® Lotus® Domino® server working with the Sun™ Solaris™ operating system. Some adjustments are simple, others are relatively drastic. Some are all but mandatory, others are optional or even experimental. Some are known to have large effects, others make small differences. Most important, all these suggestions come from people who have never seen your systems and the loads they handle; in the end, you must rely on your own judgment and observations to choose the adjustments that are best for your servers.

The suggestions in this thirteenth edition of “Tuning Tips” are current as of January 2009. They have been tested with Domino 6.5, Domino 7, Domino 8, and Domino 8.5 on Solaris 9 and Solaris 10 with most of the current processors in the UltraSPARC® family. (Check the Domino release notes for information on which combinations of Domino and Solaris versions are supported by IBM, and the Solaris release notes for information on which processors each Solaris version supports.) Except as noted, the adjustments apply to all Domino and Solaris versions; suggestions that apply only to particular versions mention them explicitly.

Please heed these guidelines as you work to optimize performance:

- **Work methodically.** Insofar as possible, make just one adjustment at a time. Measure the system's performance before and after each change, and rescind any change that doesn't produce a measurable improvement. Keep a log or diary of the changes made and the results observed.
- **Solve problems.** Some adjustments are suggested as remedies for specific problems; if you don't have the problem, don't make the adjustment. Making changes for their own sake is seldom helpful.
- **Adjust gradually.** When adjusting a quantitative parameter, it is usually better to make several stepwise changes in succession than to make a drastic change all at once. Different systems face different circumstances, and you may leap right past your system's best setting if you change a value too abruptly.
- **Less is more.** At each major system change—hardware upgrade, software upgrade, major new application deployment, or whatever—review all previous adjustments to see whether they still apply. Don't perpetuate an old adjustment simply because “we've always done it that way;” a tuning that helped Domino R4.5 on Solaris 2.6 may be inappropriate for Domino 8.5 on Solaris 10.
- **Stay informed.** Read the Domino and Solaris release notes whenever you upgrade your system. The release notes often provide updated information about particular adjustments.

Monitoring Performance

Making adjustments without measuring their effects is silly: if you don't measure the system's behavior before and after a change, you won't know whether the change was a good idea, a bad idea, or merely irrelevant. This section describes some of the tools and utilities you can use to monitor your system's performance.

Short-term system monitoring

Solaris offers several tools for taking “snapshots” of system behavior. Although you can capture their output in files for later analysis, these tools are primarily intended for monitoring system behavior in real time:

- The `iostat -x nn` command reports disk performance statistics at `nn`-second intervals. Watch the `%b` column to see how much of the time each disk is active, and for disks active more than about 20% of the time pay attention to the service time as reported in the `svct` column. Other columns report the I/O operation rates,

the amount of data transferred, and so on. Use `man iostat` for more information.

- The `vmstat nn` command summarizes virtual memory activity and some CPU statistics at `nn`-second intervals. Monitor the `sr` column to keep track of the page scan rate and take action if it's consistently greater than zero (isolated “spikes” don't mean much, but steady scanning even at a low rate is a sign of trouble). Watch the `us`, `sy`, and `id` columns to see how heavily the CPUs are being used; remember that you need plenty of CPU power in reserve to handle sudden bursts of activity. Also keep track of the `r` column to see how many threads are contending for CPU time, and reduce the number of threads Domino uses if `r` remains too high. “Too high” depends on the number and type of CPUs in the system. Count four for each UltraSPARC III chip, eight for each UltraSPARC IV or IV+ chip, eight for each SPARC64 V1 chip, eight for each active core of an UltraSPARC T1 chip, and twelve for each active core of an UltraSPARC T2 chip; the total is the approximate threshold above which `r` would be considered “too high.” Use `man vmstat` for more information.

- The `mpstat nn` command gives a detailed look at CPU statistics, while the `netstat -i nn` command summarizes network traffic. We have found these reports less useful than the first two, but they can be helpful in particular situations. Use `man mpstat` and `man netstat` for more information.

Long-term system monitoring

It is important not only to “spot-check” system performance with the tools mentioned above, but also to collect longer-term performance histories so you can detect trends. If nothing else, a baseline record of a system performing well may help you figure out what has changed if the system starts behaving poorly. We recommend you enable the system activity reporting package by doing the following:

- Edit the file `/etc/init.d/perf` and remove the `#` comment characters from the lines near the end of the file.
- Run the command `crontab -e sys` and remove the `#` comment characters from the lines with the `sa1` and `sa2` commands. You may also wish to adjust how often the commands run and at what times of day, depending on your site's activity profile. Use `man crontab` for an explanation of the format of this file.

This causes the system to store performance data in files in the `/var/adm/sa/` directory, where they are retained (by default) for one month. You can then use the `sar` command to examine the statistics for time periods of interest. Use `man sar` and `man -s1m sar` for more information on storing and examining performance data.

Server activity monitoring

The tools described above monitor performance from the standpoint of how the system responds to the load Domino generates. It is also helpful to view things from Domino's perspective, using Domino's own capabilities to keep track of the demands the users place on the server. We confess that we are not very familiar with Domino's tools in this area; in NotesBench work the imposed load is already well-understood, and we're disinclined to “waste” resources on “extraneous” server tasks instead of on the benchmark workload itself. Hence, this list is sketchy at best:

- Add the line `Server_Show_Performance=1` to the `notes.ini` file to get a minute-by-minute summary of the number of client sessions active and the number of transactions performed.
- Add the line `Platform_Statistics_Enabled=1` to the `notes.ini` file. Thereafter, you can use the `show stat platform` console command to display the server's view of selected system performance

statistics. In Domino 7 and later you can use Domino Domain Monitoring (DDM) to set thresholds for important statistics and create alerts if they are exceeded.

- Consider enabling Domino's Activity Logging capabilities to collect detailed records of various server activities. See the Help facility of the Domino Administrator client for information on Activity Logging and Activity Analysis.

Obsolete Adjustments

Some adjustments that were once mandatory or at least highly recommended have become obsolete with advances in Domino and Solaris. We suggest that you remove these tunings if they are present on your system.

File descriptor limits

It is no longer necessary to adjust `rlim_fd_max` in the `/etc/system` file, as it used to be in Solaris 8 and earlier releases. Some versions of the Domino release notes still instruct you to set this parameter, but that recommendation is based on outdated information.

File system daemon adjustments

We used to recommend adjusting the `autoup` and `tune_t_fsflushr` parameters in `/etc/system`, but advances in Solaris have made this unnecessary for nearly all systems dedicated to Domino.

Page daemon I/O throttle

We used to recommend adjusting the `maxpgio` parameter in `/etc/system`, but the file system improvements in Solaris 9 and 10 have made this unnecessary.

Adjustments for All Servers

The adjustments in this section are “all but mandatory” for the systems they apply to. Unlike most of the other adjustments in this paper, these should almost certainly be applied together rather than separately.

Upgrade to Solaris 10

If you have a choice of which operating system to use for your Domino server, we strongly recommend using Solaris 10. At this writing Domino makes only limited use of the features unique to the Solaris 10 release, but it nonetheless benefits from the performance improvements to many Solaris components, particularly the TCP/IP network stack and the Unix File System (UFS). You will also find that Solaris 10 requires less tuning than earlier versions, since it is more able to adjust itself to changes in load.

If upgrading to Solaris 10 is not feasible at this time (for example, if you use a Domino version earlier than 6.5), you should at the very least run Solaris 9. This release has its own set of performance enhancements, including a file system improvement that is of great help to Domino when performing maintenance activities like compacting large NSF databases. All current Solaris 9 releases have this improvement, and older versions can obtain it in the form of a patch. To verify that your Solaris 9 has the patch, run

```
showrev -p | grep "Patch: 112233"
```

and check that version 112233-10 or later of the patch is present. If it is missing or outdated, obtain the current version from <http://sunsolve.sun.com/>.

Limit disk read-ahead

Modern disk subsystems can perform larger data transfers than their predecessors, and Solaris can take advantage of this to read more data than Domino asks for in hopes of having the next piece of a file already in memory as soon as Domino asks for it. Unfortunately, Domino doesn't always use the "anticipated" data but instead next asks for data from an entirely different region of the file, so the effort spent reading the extra data may be wasted. In extreme circumstances and with modern disk systems Solaris can be fooled into reading fifty or sixty times as much data as is actually needed, wasting more than 98% of the I/O effort.

To prevent this pathological behavior, build or tune the file systems that hold NSF databases so that Solaris won't try to read more than about 64KB at a time from them. The instructions that follow show how to do this for the default Unix File System (UFS); if you are using an alternative file system, consult its documentation.

If you are in a position to build or rebuild the file systems, we suggest using the command

```
newfs -i 200000 -c 256 -C 8 -m 1 /dev/rdisk/...
```

The important option is **-C 8**, which limits the amount of read-ahead to no more than eight pages of 8KB each. The other options are less important, and may even cause **newfs** to issue warning messages for some disk sizes; these warnings can be ignored since **newfs** will adjust the values to suit the actual disk.

If the file systems already exist and it is not practical to back them up, rebuild them, and restore their contents, you can get most of the benefit by using the command

```
tunefs -a 8 /mount_point
```

specifying the path name of the directory where the file system is mounted. Then unmount and remount the file system for the new setting to take effect.

Increase message queue size

If you use Solaris 9 with Domino 6.5 or earlier, raise the system's limit on inter-process messages to at least 1024 by appending the following line to **/etc/system** and rebooting Solaris:

```
set msgsys:msginfo_msgtql=1024
```

This value is sufficient for three or possibly four Domino partitions, but if you plan to run more you should raise the setting to about 300 times the partition count.

This setting is not required for Domino 7 or later or for Solaris 10.

Limit Domino's memory consumption

If you run more than one Domino partition on a single Solaris system, you must ensure that no partition uses more than its fair share of system memory. In each partition's **notes.ini** file, set the **PercentAvailSysResources** parameter to control the fraction of memory (and some other resources) the partition can use. The total of all partitions' settings should not exceed 100; you may wish to set the total lower if the system also runs applications not related to Domino, even if running only one Domino partition. Use the **show stat database** console command and monitor the **Database.Database.BufferPool.PerCentReadsInBuffer** statistic; if it stays below about 85% you should allocate more memory to the partition.

Since **PercentAvailSysResources** can only be adjusted in 1% increments, it may be too coarse an adjustment for systems with large amounts of memory: one percent of 128GB is more than 1300MB. For finer control, you can use the **NSF_Buffer_Pool_Size_MB** parameter in **notes.ini** to specify how many

megabytes Domino should assign to the largest of its memory pools. Use `show stat database` to see whether you've allocated enough, and monitor the `Database.Database.BufferPool.Peak.Megabytes` value to see whether you've allocated more than Domino actually uses.

CAUTION: Use *either* `PercentAvailSysResources` or `NSF_Buffer_Pool_Size_MB`, but do not use both unless specifically instructed to do so by IBM Support.

Adjustments for Most Servers

The adjustments in this section apply to most systems, but are less important than the “all but mandatory” adjustments of the previous section.

Reduce file system housekeeping

UFS (Unix File System) volumes keep track of the time each file is accessed. Domino maintains its own access times for the notes inside each NSF database, so the I/O traffic UFS uses to maintain its time stamps is wasted effort. On a file system dedicated to Domino, you can tell UFS not to bother updating the access times by adding the `noatime` parameter to the file system's entry in `/etc/vfstab`. For example:

```
#device          device          mount FS   fsck mount  mount
#to mount        to fsck         point type pass at boot options
/dev/dsk/c0t5d0s6 /dev/rdisk/c0t5d0s6 /data0 ufs 1   yes   noatime
```

The setting takes effect the next time the file system is mounted; you can unmount and remount it to apply the setting immediately.

Improve file system caching

Domino makes intensive use of the file system, and can benefit if the file system's buffer cache is permitted to use more of the machine's physical memory. Increase the number of memory pages the file system can map into its address space by adding this line to `/etc/system` and rebooting Solaris:

```
set segmap_percent=40
```

The optimal value for this parameter is difficult to determine. The suggested value of 40 seems to be safe for most systems and represents a noticeable improvement over the default value of 12. You may wish to experiment with higher values (some memory-rich systems in lab environments have done well with settings as high as 65), but be prepared to lower the setting if memory shortages occur. If the `sr` (scan rate) statistic output by the `vmstat` utility is consistently non-zero, reduce `segmap_percent` and reboot.

Adjustments for Particular Problems

The adjustments below are intended for servers that are demonstrating specific problems. Please read the descriptions carefully. If the description matches your situation, consider making the adjustment. If your server does not exhibit the described symptoms do not make the adjustments since they may actually harm performance.

Mail waiting to be delivered?

If Domino runs normally but the `Mail.Waiting` statistic reported by the `show stat mail` console command remains high relative to total mail volume, try increasing the number of mailboxes the server uses for

incoming messages. Open or create the server's Configuration document and select the Router/SMTP tab and then the Basics sub-tab. On this form, set the "Number of mail boxes" field to 2. Restart Domino and issue the **show server** console command to verify the setting, and gradually increase the number of mailboxes if the problem persists. We have tested with as many as ten mailboxes, but saw little or no improvement beyond four.

Low cache hit percentage?

If the console command **show stat database** displays cache hit rates lower than 85-90%, set the **notes.ini** parameter **NSF_DBCache_MaxEntries** parameter to slightly more than the actual number of open mail databases or concurrent users the Domino partition supports.

If you do not set **NSF_DBCache_MaxEntries** explicitly, Domino calculates a default value that you can examine with the **show stat database.dbcache** console command.

Long service times on busy disks?

Domino's responsiveness depends strongly on the performance of the disk subsystem. Use the **iostat -x nn** command to monitor how busy the disks are and how rapidly they complete I/O requests (the **%b** and **svc_t** columns, respectively). Service times are unimportant for disks that are less than about 20% busy, but for busier disks the service times can have a substantial influence on Domino's performance. The threshold of "good" as opposed to "slow" service time depends on what the disk is used for:

- Transaction logs should have service times not much slower than 5 milliseconds. Every update to a logged database or view must first be written to the transaction log, so the log disks are on the critical path for many Domino activities.
- Disks containing the server directories should show service times of 15 milliseconds or less. The server directories contain crucial databases like **names.nsf** and **mail.box**, which are involved in many of Domino's activities; if access to these databases is slow, the server as a whole will suffer.
- Disks devoted to mail databases can tolerate service times of up to 30 milliseconds or so, especially if the databases are spread across many such disks.

Ideally, you should use the fastest disks available for the transaction logs, dedicating a separate disk (or RAID-1 mirrored pair) to each partition. Put the server directories on the next-fastest disks (or RAID-10 striped mirrors). The users' mail databases can go on larger, slower disks (or controller-based RAID-5 stripes).

For file systems containing server directories or NSF databases, be sure to tune for limited read-ahead as described on page . You should also try to distribute the load as evenly as possible across the available disks by moving heavily-used databases off of overburdened disks and onto disks that are less heavily loaded. Correcting an imbalance is usually more effective than trying to tweak the performance of overloaded disks.

Too many router connections?

If the server console displays many "Router: No messages transferred to ..." messages and the **show stat mail** command reports relatively few messages waiting for delivery, your server may be wasting effort by running more transfer threads than it needs. Open or create a Configuration document for the sending partition and select the Router/SMTP tab, then the Restrictions and Controls sub-tab, and finally the Transfer Threads sub-sub-tab. Try reducing the "Maximum Concurrent Transfer Threads" value to 1 or 2, causing the router to connect to remote servers less frequently and deliver more messages at each

connection.

Use this value as a starting point only. We have found that Domino's default is too high for NotesBench work, but NotesBench is not always a good imitation of real-world conditions. The best settings for your server will depend on your mail routing topology and the mail patterns of your users.

Use the `show stat mail` and `tell router show` console commands to monitor the router's performance, and be alert for signs of trouble. It may turn out that you need to increase the number of transfer threads or even (as a last resort) the number of concurrent transfer threads if mail backlogs develop.

NOTE: We used to suggest lowering the total number of transfer threads, in addition to reducing the number of concurrent threads. We now feel that the concurrency adjustment alone is sufficient, and suggest leaving the total number of transfer threads blank unless you have some other reason to adjust it.

Adjustments for Systems Serving Web Clients

These settings arise from our work with the NotesBench R6iNotes workload, which simulates the activity of Domino Web Access clients accessing the server via HTTP. Consider these adjustments for partitions that server mostly Web clients as opposed to Notes, IMAP, or POP clients.

Lack of agent concurrency?

If response times as perceived by browser clients is slow for applications that use agents to deliver content, response may be improved by allowing the agents to run concurrently instead of sequentially. To enable this, open the Server document and select the Internet Protocols tab and the Domino Web Engine sub-tab, and set the "Run Web agents concurrently?" field to Enabled.

NOTE: Adding `DominoAsynchronizeAgents=1` to the server's `notes.ini` file has the same effect, but IBM Support recommends editing the Server document instead. Use one method or the other, but not both.

Lack of server thread concurrency?

If the Web server response is slow, try increasing the number of HTTP threads that handle the incoming requests. Open the Server document, select the HTTP tab, and raise the "Number of Active Threads" setting from the default value of 40 to 90 or a bit more. (It is not advisable to exceed 100, because of the amount of memory required for each thread and because of the risk of increasing contention between the threads.)

Failure to connect to HTTP server?

If browsers are timing out when attempting to connect to Domino, the queues involved in accepting new connections may be too small. Use the `netstat -s` command and examine the `tcpListenDrop`, `tcpListenDropQ0`, and `tcpHalfOpenDrop` statistics. If the reported values increase steadily as time passes, consider adjusting the queue lengths. This involves one Domino setting and two Solaris settings.

To change the Domino setting, shut down the partition, add the line

```
listenbacklog 2048
```

to its `httpd.cnf` file, and restart Domino.

To change the Solaris settings, execute these two commands as the root user:

```
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q 2048  
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q0 2048
```

To have these commands executed automatically each time Solaris boots, place them in a file called `/etc/init.d/network-tuning` and create a link to the file by executing

```
ln /etc/init.d/network-tuning /etc/rc2.d/S99network-tuning
```

Use `netstat -s` as before to monitor the effect of the changes. You may need to raise the values higher than 2048 (under laboratory conditions some systems have shown improvement with values as high as 4096), but do not increase them without limit. Each increase ties up more memory, and it is counterproductive to “accept” connections faster than Domino can service them.

Advanced and Experimental Settings

The adjustments in this section are suggested only for experienced Solaris and Domino administrators. Their benefits will not be applicable to all systems, and may carry some risk. Be careful out there.

Tuning TCP buffering

If the server's clients experience irregular network response times even when the server's load is relatively steady, the system may benefit from increasing the size of the queue that sits between the network hardware and the TCP/IP protocol driver. Try adding the following line to `/etc/system` and rebooting Solaris:

```
set sq_max_size=50
```

A queue capacity of 50 packets allows high volumes of network traffic without overflowing the queue. Higher values may be useful on systems with large amounts of memory, but please consult

```
http://sunsolve.sun.com/pub-dgi/retrieve.pl?doc=finfodoc%2F71160
```

before raising the setting beyond 50.

System V shared memory

Domino 7 and later versions support an alternative means of sharing memory between a server's various processes. Instead of using the familiar `/tmp/.NOTESMEM...` files, Domino can use System V shared memory segments. This method can yield improved performance, particularly on newer processors like the UltraSPARC T1 and T2, but requires considerably more effort to manage.

The mechanics of sizing, setting up, and monitoring System V shared memory are too involved for this short paper, but are covered in the latest IBM Redbook for Domino on Solaris; see the References below.

CPU scheduling control

In laboratory settings, Domino runs more efficiently when assigned to the FX *scheduling class*, which can be done in several ways. The simplest is to use the `prctl` command as in these examples (insert the command you ordinarily use to start Domino, like `/opt/lotus/bin/server`):

```
prctl -e -c FX server_start_command
```

If you always start Domino through a shell script, you can insert a `prctl` command in the script at any point before Domino itself actually starts:

```
prctl -s -c FX $$
```

Both the examples above run Domino in the FX priority class at priority zero, the lowest priority of all. It may seem odd to give Domino such a low priority, but if there is “nothing else” running on the system then Domino's only competition for the CPU is Domino itself. However, if the system is shared with other activities or if some

Domino partitions run in FX while others run in IA or TS, the zero-priority Domino threads may be starved for CPU cycles.

To avoid CPU starvation, you can specify a non-zero priority in the `prctl` command. For Solaris 9 this requires root privileges, so this approach is most suitable for use in a script that will be run by the root user. In that script, at some point before switching to the Domino user account and launching the server itself, add:

```
prctl -s -c FX -m 45 -p 45 $$
```

This sets Domino's priority to 45, moderately high in the allowable range of 0 through 60.

CAUTION: Do *not* run Domino at the maximum fixed priority of 60. If you do and if a Domino thread gets into an infinite loop, it will monopolize the CPU and you will have great difficulty logging in and using `nsd` to kill it. Leave yourself some “priority headroom” for emergencies.

On Solaris 10 it is possible to give the Domino user account the `priv_proc_prctl` privilege, allowing it to use the `prctl` command as above without needing to be root. One time only, the root user can run

```
usermod -K defaultpriv=basic,priv_proc_prctl username
```

This grants `username` the right to use the `prctl` command with a non-zero priority, thus allowing a non-root user to start and restart Domino at will without pestering the system administrators. The privilege can also be managed through the Sun Management Console, or can be assigned to a role instead of to a specific user account.

For more information on the `prctl` command, use `man -s1 prctl`. Background information on Solaris 10 privileges can be found with `man -s5 privileges`, while `man -s1m usermod` and `man -s1m smc` discuss tools to administer them.

References

- This “Tuning Tips” document is updated from time to time. You can always find the latest version in the Technical Documentation section of <http://www.sun.com/lotus/>.
- *Domino Administrator's Guide*, Lotus Development Corporation
- *Domino 7 for Sun Solaris 10*, IBM Technical Support Organization (part of the Redbook library). Despite the “Domino 7” in the title, much of the information in this book applies to Domino 8 and 8.5 as well. See <http://www.redbooks.ibm.com/abstracts/sg247162.html>
- *Tuning Domino for a High-Throughput Processor*, from *The View* volume 14 issue 2. A more detailed look at some adjustments of special importance for the UltraSPARC T1 and T2 processors. See <http://www.lotususergroup.org/submissions.nsf/ContentSpotlight/12882DEE6046D6CD862574150018D5C3/?OpenDocument>
- *Solaris Internals*, Jim Mauro and Richard McDougall. The descriptions in this book give more understanding about the significance of the statistics reported by `vmstat` and the like, and help demystify the effects of various Solaris parameters.

Sun Microsystems, Inc. Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com

©2009 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Solaris, and UltraSPARC are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Lotus, Domino, and Notes are trademarks or registered trademarks of IBM Corporation and/or Lotus Development Corporation.