I/O subsystems

by
Razeyah
Stephen

Level: Beginner
Works with: Domino 5.0
Updated: 08/02/99

Inside this article:

Test methodology and test data
Scenario 1: Windows NT
Scenario 2: AIX
Scenario 3: Sun Solaris
Recommended I/O configurations

Related links:

Configuring RAID levels for I/O performance sidebar
Optimizing server performance: Transaction logging
NotesBench Web site
Domino Performance Zone

Get the PDF:

You can say "No" to I/O bottlenecks with Domino R5.

When you upgrade databases and database servers to R5, you'll gain a number of key benefits in the input/output (I/O) subsystem. Some of the I/O optimizations include transaction logging, new justification for distributing multiple mail files across multiple logical disk units, multiple MAIL.BOX databases, and the new R5 database format. These I/O optimizations can result in greatly increased performance.

In this article, we'll look at the performance of Domino R5 on various system configurations, and pay special attention to the I/O subsystems. We'll then review the I/O lessons that we learned with each configuration. Finally, we'll give you some recommendations for I/O subsystem configurations, so you can take full advantage of the performance enhancements in Domino R5.

About the I/O improvements in Domino R5

Domino R5 includes the following I/O improvements that can lead to increased performance:

- Transaction logging writes all changes to a database *sequentially* to a log file, and then commits those changes to disk at a later time. The *sequential* writing results in much faster I/Os. (Note: You should place the transaction log files on a *separate* disk drive. For more information on transaction logging, see "**Optimizing server performance: Transaction logging**.")
- There are better reasons for you to now use file links to distribute multiple mail files (such as, razeyah.nsf) across multiple logical disk units. This means that you don't need to use a single, monolithic file. Instead, Domino can deliver mail concurrently, resulting in concurrent disk I/Os and increased I/O performance. (Note: You should distribute the logical disk units over multiple hardware RAID controller cards in order to maximize the throughput of your I/O subsystems.)
- With multiple MAIL.BOX databases, users (and server processes, such as agents and routers) can deposit messages into any available MAIL.BOX database. This reduces contention created by many users simultaneously sending messages for delivery to a single MAIL.BOX.
- The R5 database format improves the overall performance of databases, so database operations require less I/O.

Note: For in-depth information on using any of these features, see the **Domino 5 Administration Help**.

Test methodology and test data

Now, let's look at the performance analysis of these I/O improvements in Domino R5. Our test scenarios used the **NotesBench** R5 workloads for R5 mail, Webmail, IMAP, and SMTP/POP3. We compared the average response time (in seconds), CPU utilization, and disk utilization. This section describes the system configurations, and more in-depth information about the workloads.

System configurations

To run our tests, we set up three systems with the following configurations:

System 1 (Windows NT)
Our first system ran Windows NT on:

- System: IBM Netfinity 7000 M10 server
- CPU: Four Xeon PII 400MHz processors, 1MB L2 cache per CPU
- Memory: 4GB
- Drives: Varied across the tests (more details below)
- Two IBM Netfinity ServeRAID-3H Ultra SCSI (40MB/s) controllers
- Network: TCP/IP
- OS: Windows NT server EE 4.0, Service Pack 4
- Domino: Release 5 for Windows NT

System 2 (AIX)
Our next system ran AIX on:

- System: RS/6000 Enterprise Server S70
- CPU: 12 200MHz PowerPC processors
- Memory: 16GB
- Drives: 18 4.5GB drives (more details below)
- Network: TCP/IP
- OS: AIX 4.3.1
- Domino: Release 5 for AIX

System 3 (Sun Solaris)
Our final system ran Sun Solaris on:

- System: SUN Enterprise 4500
- CPU: Eight 336MHz UltraSPARC11 processors, 4MB external cache
- Memory: 8GB
- Drives: Two 8.4GB external drives and one Sun StorEdge A3500 Ultra SCSI array with 180GB of disk space (more details below)
- Network: TCP/IP
- OS: Solaris 2.6
- Domino: Release 5 for Solaris

About the workloads
Our tests used the NotesBench R5Mail, Webmail, R5IMAP, and SMTP/POP3 workloads. These workloads all have local (same server) delivery of mail. For additional details on these workloads, see the **User Profiles document** on the NotesBench Web site. (You must **register on the NotesBench site** in order to access this document.)

Note: The following workload descriptions show the time interval for sending messages that we used in our tests. You can specify a different time interval.

The **R5Mail workload** models an active user reading and sending mail, as well as scheduling an appointment, and sending meeting invitations. The script sends six messages (two memos, two appointments, and two invitations) to three recipients every 90 minutes. For each iteration of the 15-minute script, the client:

- Reads five documents
- Updates two documents
- Deletes one document
- Adds one document
- Scrolls down one view
- Opens and closes one database

The **Webmail** workload models an active user reading and sending mail using the HTTP protocol. All messages are sent to and received by other simulated Webmail users on the same Domino server. In the script, a user prepares and sends a 10K message to three recipients dynamically selected from the server's directory every 90 minutes. For each iteration of the 15-minute script, the client:

- Reads five documents
- Deletes one document
- Opens and closes one database

The **R5IMAP** workload models an active user in connected IMAP mode (IMAP online mode) with a mail file on the server and working with its contents interactively. The script sends one SMTP message every 90 minutes. For each iteration of the 15-minute script, the client:

- Checks twice for mail messages
- Reads five documents
- Deletes one document
- Adds one document

The **SMTP/POP3** workload models an active user retrieving POP3 mail and sending SMTP mail. The script sends one SMTP message to three recipients every 30 minutes. About 20 percent of the users receive 80 percent of the mail messages sent. For each iteration of the 15-minute script, the client checks and retrieves POP3 mail messages.

About the results
The following table shows the summary of our test results. For more details on the results, see the following sections of this article.

Workloads	Server	# of Users	Avg. Resp Time (in sec)	CPU Util (%)	Disk Util (%)
R5Mail	Netfinity 7000 M10	10,000	2.0	90*	100
	RS/6000 Enterprise S70	10,000	1.0	55	12
Webmail	Netfinity 7000 M10	2,000	0.5	85	45
	SUN Enterprise 4500	2,000	0.5	65	6
R5IMAP	SUN Enterprise 4500	3,000	0.3	35	25
SMTP/POP3	Netfinity 7000 M10	5,000	0.4	28	75

* We used Domino R5 debug code for the first R5Mail test, which may have resulted in the high CPU and disk utilization numbers.

Scenario 1: Windows NT

In our first evaluation scenario, we ran the R5 mail, Webmail, and SMTP/POP3 workloads on our Windows NT system. The following sections reveal the details and results of each test.

R5 mail
We ran the R5 mail workload with 10,000 users on our Windows NT system. For storage, the system had 22 9GB 10K RPM drives with the following configuration:

- Four internal 9GB drives, RAID0 Logical Unit Number (LUN), for the OS, Domino executables, and the page file
- Nine 9GB drives in an EXP15 enclosure configured as a five-member RAID0 LUN and a four-member RAID0 LUN, both for databases (on

Channel 1 of ServeRAID controller 1)

- Nine 9GB drives in EXP15 enclosure configured as a five-member enhanced RAID1 LUN and a four-member RAID0 LUN. The enhanced RAID1 LUN was used for transaction logging, and the RAID0 LUN was used for databases (on Channel 1 of ServeRAID controller 2)

In our first test, we did not use transaction logging. We were only able to run about 7,500 users, and encountered I/O bottlenecks. Then, we turned on transaction logging, which enabled us to attain our goal of 10,000 users. A hardware RAID1, a mirror set made up of two physical disk drives, should be adequate for the transaction log. We used RAID1 Enhanced for transaction logging because it was already configured on our system. Domino R5 supports a maximum transaction log size of 4GB. (For more information on transaction logging, see "**Optimizing server performance: Transaction logging**.")

For this testing, we varied the number of MAIL.BOX databases from one, to two, five, seven, and 10. We saw the largest gain in performance when we moved from one to two MAIL.BOX databases. The performance gains were small as we increased the number of MAIL.BOX databases beyond two. The number of MAIL.BOX databases should not be more than 10.

Webmail
We ran the Webmail workload with 2,000 users on our Windows NT system. For storage, the system had two Netfinity ServeRAID-3H Ultra SCSI adapters/controllers, eight 9GB 10K RPM drives, and two 2GB drives with the following configuration:

- Two internal 2GB drives, one for the OS and Domino executables, and a 1GB partition on the other drive for the page file
- Four 9GB drives in an EXP15 enclosure configured as a four-member RAID0 LUN, for databases (on Channel 1 of ServeRAID controller 1)
- Four 9GB drives in an EXP15 enclosure configured as a four-member RAID0 LUN, for databases (on Channel 1 of ServeRAID controller 2)

In our first test, this system had one large LUN, made up of two four-disk drives, RAID0 LUN. Each set of drives was on each controller. We concatenated the LUNs at the OS level into one logical unit for the databases. We also used one MAIL.BOX database. With this configuration, we encountered an I/O bottleneck at 100% disk utilization on the databases LUN.

So, we switched to two separate LUNs. Then, we took advantage of the Domino R5 support for multiple data stores for the mail files. We accomplished this by creating a second data directory, *data2*, on the second LUN. Then, we linked the two data directories together using directory links. (For information on creating directory and database links, see the **Domino 5 Administration Help**.) We distributed 1,000 mail files per data directory. With this change, the disk utilization went from 100 percent for the single large LUN, to 45 percent each for the two LUNs because of concurrent message deliveries.

In addition, we increased the number of MAIL.BOX databases from one to two. When we went from one to two MAIL.BOX databases, we saw a substantial performance improvement. Then, when we went from two to four MAIL.BOX databases, the performance improvement was marginal.

So, by using two data stores and two MAIL.BOX databases, we attained our goal for 2,000 Webmail users.

SMTP/POP3
We ran the SMTP/POP3 workload with 5,000 users on our Windows NT

system *(we used only two of the four CPUs)*. For storage, the system had 22 9GB 10K RPM drives with the following configuration:

- Four internal 9GB drives, configured as JBOD (Just a Bunch of Disks), one drive each for the OS, the page file, the Domino executables, and the transaction log
- Nine 9GB drives in an EXP15 enclosure, RAID0 LUN, for databases (on Channel 1 of ServeRAID controller 1)
- Nine 9GB drives in an EXP15 enclosure, RAID0 LUN, for databases (on Channel 1 of ServeRAID controller 2)

With this configuration, we did not run into an I/O performance bottleneck because we distributed and balanced the mail files over the two ServeRAID controllers (using directory links). Also, we used transaction logging, so we had no problem attaining our goal of 5,000 SMTP/POP3 users. We used seven MAIL.BOX databases in this configuration.

Scenario 2: AIX

In our next evaluation scenario, we ran the R5 mail workload with 10,000 users on our AIX system. For storage, the system had 18 4.5GB drives with the following configuration:

- Logical volume group of two 4.5GB internal disk drives for OS, Notes binaries, and swap
- Four 7133-020 SSA Disk subsystems for databases, with:
  -- Two SSA controllers (80 MB/s each)
  -- 16 disk drives in two loops, with eight disk drives per loop. We configured the 16 disk drives into one logical volume group using inter-policy maximum and intra-policy center.

We received this AIX machine from the IBM RS/6000 group, who preconfigured the logical volume groups. They had balanced and distributed the disk drives over the two loops. With this configuration, we did not run into an I/O performance bottleneck. The SSA subsystem was very efficient with the disk utilization at 12% for 10,000 R5 mail users. We used seven MAIL.BOX databases in this configuration.

Scenario 3: Sun Solaris

In our final evaluation scenario, we ran the Webmail and IMAP workloads on our Sun Solaris system. For each test, the system had the following storage configuration:

- Two 8.4GB external drives, with the OS on one drive and the Domino binaries on the other
- One Sun StorEdge A3500 Ultra SCSI storage array with 180GB of disk space, and the following configuration:

  -- 20 9GB 10K RPM drives

  -- Two controller modules with two Ultra SCSI controller cards per module

  -- Eight SBus Ultra Differential FW SCSI host adapters attached via eight Ultra SCSI (40 MB/s) buses

  -- Four hardware RAID0 LUNs, with five drives per LUN. We distributed and balanced each RAID0 LUN on an Ultra SCSI controller card. Three of the RAID0 LUNs were for databases and one was for swap space.

With this configuration, we successfully attained our goal of 2,000 Webmail users and 3,000 IMAP users. One advantage for this scenario is that we acquired the Sun machine after learning the I/O lessons on the Windows NT

and AIX platforms. So, we made sure that each A3500 array had multiple controller modules, and we balanced the four RAID0 LUNs for the databases and swap space on each controller card. Plus, each controller card had its own Ultra SCSI bus going to its own host adapter.

Domino R5 delivered mail concurrently, resulting in concurrent disk I/Os and increased I/O performance. The disk utilization was very low (6% for Webmail and 25% for IMAP), so the configuration includes room for the databases to grow. We used seven MAIL.BOX databases in this configuration.

Recommended I/O configurations

To optimize your servers for I/O performance, we recommend that you look at the configuration of your Domino databases, transaction logs, and storage.

Domino databases

As mentioned earlier, you can now use file links to distribute multiple mail files across multiple logical disk units. To distribute the files, you use directory links on Windows NT and symbolic links on UNIX. (The feature is new on Windows NT, but was already available on UNIX.) Domino can then deliver mail concurrently, resulting in concurrent disk I/Os and increased I/O performance.

The important thing here is that you must use *multiple RAID LUNs* for the data store. You should distribute and balance the LUNs over multiple hardware RAID controller cards in order to maximize the throughput of your I/O subsystems. If you use multiple RAID arrays, make sure to put each array on its own host adapter (preferably on its own I/O board) so that the host adapter/board does not affect performance.

In addition, you should use *hardware RAID0+1* (mirroring and striping) or *RAID1 Enhanced* (striping and mirroring) for the best performance of Domino databases. We developed this recommendation after comparing the ratio of write and read accesses to disk for the various workloads. Our results showed that the workloads are write-intensive. (To learn more about monitoring the disk statistics for your own applications, see the Lotusphere presentation, **Deploying R5 for Performance and Scalability in your Environment**.)

So, for write-intensive applications, you should not use RAID5. RAID5 (parity plus striping) writes require three additional disk I/Os, while RAID 0+1 and RAID1 Enhanced writes require only one additional disk I/O. For more information on RAID levels and our read/write access test results, see the sidebar "**Configuring RAID levels for I/O performance**."

Note that you should use *hardware RAID* for the Domino databases, not software RAID. Software RAID is less efficient than hardware RAID and puts additional overhead on the server's CPUs. Also, turn on the write-back cache on the hardware RAID controllers. We recommend using a stripe or chunk-size of 32KB or 64KB for the RAID LUNs, except in the case of the SMTP/POP3 workloads, where we recommend 64KB.

Finally, we realize that you may have concerns with using RAID0+1 or RAID1 Enhanced for the data store, because these RAID levels have an effective storage of 50 percent. (For more details on this, see the sidebar "**Configuring RAID levels for I/O performance**.") Therefore, consider the tradeoff between performance and cost when choosing your RAID level. For information on preferred configurations from the hardware vendors themselves, see the **vendor reports on the NotesBench site**. In particular, check out their I/O subsystems, number and type of host adapter and controllers, and disk configurations. Their reports have performance results using RAID0+1, RAID1 Enhanced, and RAID 5.

Transaction logs

With Domino R5 transaction logging, all changes are written *sequentially* to disk, which results in much faster I/Os. For a performance analysis of

transaction logging, see "**Optimizing server performance: Transaction logging**."

You should allocate a *separate* disk drive for the transaction log files -- *not* the same device as for the Domino databases, swap file, operating system, or the temporary directory that Domino uses for rebuilding views. We highly recommend using hardware RAID1 (two disk drives mirrored), which increases your system availability and reliability. The increase in cost is the price of one disk drive!

To further maximize performance and reduce I/O contention, place the transaction log file on its own port on a host adapter, if possible. Even better, place the log file on its own RAID controller. Also, turn on the write-back cache on the hardware RAID controller for the transaction log files.

Other storage

Finally, you should use separate disk drives for the swap and page files, and for the temporary space that Domino uses for rebuilding views. If you have an extra RAID controller, configure a RAID0 LUN of at least two disk drives for the temporary space. To do this, add the following NOTES.INI setting to point the temporary view rebuild to the extra disk drive or LUN:

VIEW_REBUILD_DIR=/extra_LUN

We hope that you can use these recommendations to remove any I/O bottlenecks in your own systems!

ABOUT THE AUTHOR

Razeyah Stephen is a Domino Performance Engineer, who has worked at Iris since October 1998. She came to Iris from Digital Equipment Corporation, now Compaq, where she worked for five years in their StorageWorks division. Razeyah's specialty is UNIX performance.

[back to "Optimizing server performance: I/O subsystems"]

Configuring RAID levels for I/O performance (sidebar)

The following sections first show our read/write access test results, and then describe the different hardware RAID levels.

Read/write access test results

To verify the appropriate RAID level to use for the various workloads, we tested the read and write access rates on our Sun Solaris system. Our results show that the write percentage is high for the workloads, so we recommend using the RAID0+1 or RAID1 Enhanced levels, *not* RAID5. (Note: The specific read/write access numbers may vary on other platforms.)

Workload	Writes (%)	Reads (%)	Writes Transfer Size (KB)	Reads Transfer Size (KB)
R5Mail	60	40	14KB	22KB
Webmail	90	10	8KB	26KB
R5IMAP	90	10	12KB	16KB
SMTP/POP3	70	30	12KB	38KB

About the hardware RAID levels

To configure your hardware RAID levels for optimum I/O performance, you need to understand the five basic RAID levels: RAID0, RAID1, RAID0+1, RAID1 Enhanced, and RAID5.

- **RAID0** stripes data across multiple disk drives -- stripe1 is on drive1, stripe2 is on drive2, and so on -- by using the stripe or chunk size. There is no redundancy of data, and there is no penalty for writes.
- **RAID1** is a mirrored set of two physical disk drives. Applications can read from either drive. However, they must write to both drives, resulting in extra I/O for writes.
- **RAID0+1** is mirroring, plus striping -- a combination of RAID0 and RAID1. Hardware RAID controllers mirror pairs of disk drives, and then stripe the pairs. For example, for six members in the LUN, drives 1 and 2 are mirrored (*m1)*; drives 3 and 4 are mirrored (*m2*); and drives 5 and 6 are mirrored (*m3*). Then, m1, m2, and m3 are striped. Usable space is 50 percent of the total space. You need an even number of disk drives for RAID0+1. Applications can read from the data or the mirrored copy. They must write both to the data and copy, resulting in an extra write.
- **RAID1 Enhanced** is striping, then mirroring the stripe. The first stripe is the data stripe; the second stripe is the mirror (copy) of the first data stripe, but shifted by one drive. Usable space is 50 percent of the total space. You can use an odd number of disk drives for RAID1 Enhanced. Applications can read from the data or the copy. They must write both to the data and copy, resulting in an extra write. (Note: A hardware vendor usually supports RAID0+1 *or* RAID1 Enhanced. The performance of RAID0+1 and RAID1 Enhanced should be the same.)
- **RAID5** is striping data and parity across all members of the RAID set. It requires the extra storage of one disk drive for parity. For each write operation, four disk I/Os are necessary to:
  1. Read the data block.
  2. Read the parity block (and calculate the new parity block, which is not a disk I/O)
  3. Write the updated data block.
  4. Write the updated parity block.

  Therefore, RAID5 requires three additional disk I/Os.