Notes.net Iris Today

Domino clusters (Part 2)

by
Harry Murray
and Gary Sullivan

Level: Advanced
Works with: Domino 5.0
Updated 11/01/99

Inside this article:

Related links:

Domino clusters (Part 1)

Optimizing server performance: Transaction logging

Workload balancing with Domino clusters

Fine points of configuring a cluster

NotesBench Web site

Get the PDF:

[Editor's Note: This is the second article in a two-part series on the performance analysis of Domino clusters. This article focuses on performance tests of R5 clusters, including load balancing, mail workload, and cluster replicators. The ***first article*** *introduces you to clusters, and then looks at our performance tests of R4.6 clusters. It also includes recommendations from performance testing on R4.6.]*

Introduction

As a Domino administrator, your top concern is ensuring 24x7 server availability to your user community. At the same time, you need Domino to scale well, and to continue offering fast response times as the needs and numbers of your users grows. You can address both of these concerns by creating a Domino cluster.

This is the second article in a two-part series that examines the performance benefits of using Domino clusters. Part 1 of this article covered the general concepts of Domino clustering, including a discussion on clustering performance illustrated with data from R4 testing. For completeness and continuity **read part 1** before reading part 2.

In this article, we turn our focus to Domino R5, highlighting some of the main new differences in clustering for R5. We then explore **Domino R5 cluster workload balancing**, exposing an undocumented NOTES.INI setting, and recommending how to use this new setting in concert with existing NOTES.INI variables for cluster workload balancing.

We also present **R5 clustered mail** workload data including a summary view of the cluster data testing. This section includes an investigation of the use of multiple **cluster replicators**. We show the impact of cluster replicators to aid in the decision of how many replicators to use.

We provide some ideas on sizing a new cluster and conclude with news about a forthcoming workload balancing and capacity planning tool that can help with database distribution.

Main Domino R5 clustering differences

The general improvements in performance will directly benefit clustering. For example, with transaction logging enabled the disk I/O will decrease by 10 to 20 percent. Memory utilization may drop by 30 percent; and response time may improve by 75 percent, to mention a few of the R5 general performance improvements. Upgrading to R5 and taking full advantage of the many new performance related features should help relieve many of the performance bottlenecks of existing clusters.

Enhancements have been made to the operation of R5 clustered servers to support the following features not previously available:

- Failover and workload balancing for Web clients (Internet Cluster Manager)
- Free-time calendar and scheduling lookup
- Synchronous new mail agents that execute regardless of host server
- Type-ahead addressing and address resolution

- Synchronized unread marks across cluster replicas

Domino R5 cluster workload balancing

Cluster workload balancing is the Domino Enterprise Server’s capability to distribute client requests to available servers, thus avoiding over-utilizing any specific server. Relatively transparent to users, this distribution allows heavily used servers to decline additional work. In a well-configured environment, this capability spreads the workload across a set of Domino servers. Provided adequate resources are available, this distribution of workload results in lower response times and, more importantly, improves the consistency of response times.

Functioning together, Domino cluster servers and Notes clients implement cluster workload balancing. Web clients can also take advantage of workload balancing through the use of the Internet Cluster Manager in Domino R5. For simplicity in testing, we focused our testing efforts on Notes clients and Domino servers.

As mentioned in **part 1 of this series**, **Server_Availability_Threshold (SAT)** is the NOTES.INI setting on the server that defines the point at which failover occurs. Part 1 also indicated that the **Server Availability Index (SAI)** is obtained from a calculation based on a server's response time. When the SAI goes below the SAT setting, the server is in the **BUSY** state.

When a clustered Domino server is BUSY, Notes clients that already have databases open can continue to access those databases. But when a client attempts to open a database on the BUSY server, the client gets an indication that the server is not going to respond. Since the client is automatically cluster-enabled, it then accesses the Cluster Manager on an available server in the cluster. (The Notes client stores a list of the cluster servers in a local cache for just such an event.)

The Cluster Manager accesses the Cluster Database Directory to determine which servers in the cluster have a replica of the requested database. The Cluster Manager then selects the least-busy server and returns the server name to the client, which can open the database on that server.
In this way, the workload is balanced among the servers based on how busy they are.

If Domino is not set up properly, workload balancing among cluster members will be less than optimal. From testing and production environments, we have found that failover (a cluster's ability to redirect requests from one server to another) doesn’t occur gracefully unless we set an undocumented NOTES.INI setting in addition to setting the SAT.

The undocumented setting is **Server_Transinfo_Normalize (STN)**. The value of this setting is used in the SAI calculation to "normalize" the response times at the server (in other words, the response times are divided by this value). Until now there has been no documentation on setting because little testing had been done on which to base recommendations.

For the SAI calculation to work properly, the STN value should be roughly the average Domino transaction time (for the server in question) in milliseconds multiplied by 100. The default value is 3000, corresponding to an average response time of 30 milliseconds per transaction. This setting may have been appropriate for "the average server" when Domino clustering was first shipped several years ago, but our testing shows that this default is too large for the current generation of servers.

To change the default STN setting, include the following line in NOTES.INI:

```
Server_TransInfo_Normalize = 600
```

If the STN default setting (3000) is used, today's faster servers may not fail over until so heavily loaded that recovery cannot be accomplished in a reasonable period of time. Testing on Windows NT platforms has shown that SAT and STN settings can be coordinated to provide even workload balancing among cluster members.

From the definition of STN, you would expect that the faster the server, the lower you would need to set the STN. Faster in this context means a server with quick response times due to such things as its superior CPU, faster disk subsystem, or high bus speed.

To determine the optimal values of SAT and STN we varied them in a test environment while a server was under load.

Optimizing SAT and STN
Testing demonstrated that a value lower than 95 for SAT may cause the server to get overwhelmed prior to failing over with slow recovery times. Higher values of SAT generally cause the server to fail over too quickly, possibly dumping too much of its share of the workload. The optimal SAT value for your system may vary, but 95 is a good point at which to begin exploration. An SAT value from 95 to 97 was held for the majority of the remaining testing while the STN value was varied to achieve the desired results.

Testing was done on two different Domino R5.0 servers:

Compaq 7000
Two Pentium Pro 200 MHz processors (1 MB L2 caches)
Windows NT Server 4.0 Service Pack 4
2 GB RAM
523MB page file
6 data disks using RAID 0 SCSI controller
OS, Domino executables, and page file on first partition of two-disk RAID 1 on separate Smart controller with transaction log on second partition
Network: 10 Mbit Ethernet

IBM Netfinity 7000 M10
Four Pentium II 400MHz Xeon processors (1 MB L2 caches)
Windows NT Server 4.0 Service Pack 4
44 GB RAM
2 GB page file
8 data disks using enhanced RAID 0 on IBM Netfinity server RAID 3H Ultra2 controller
1 disk each for OS, page file, Domino executables on separate controller
Network: 100 Mbit Ethernet

About the workloads
The NotesBench clustered mail workload was used to generate the workload on the server. This workload performs the same transactions as the R4 clustered mail workload. (In the future we will update this workload to be like the R5 Mail workload.) Below is a summary of the approximate mail transactions for each "user." We used this workload because it is a good representation of the most common functions used with Domino mail. We tested with 500 and 1000 of these "users."

Visit **www.NotesBench.org** for more detail on NotesBench workloads and the NotesBench consortium.

Per user per day (8-hour test run)	Number
Mail documents read	160

Mail messages sent	5
Mail documents updated	64
Mail documents deleted	64
Total mail related tasks (Includes other miscellaneous tasks)	293+

Recommendations
As a result of this testing, the recommended settings for our two test Windows NT systems are as follows:

Server	**SAT**	**STN**
Compaq 7000	95	600
IBM Netfinity 7000 M10	97	200

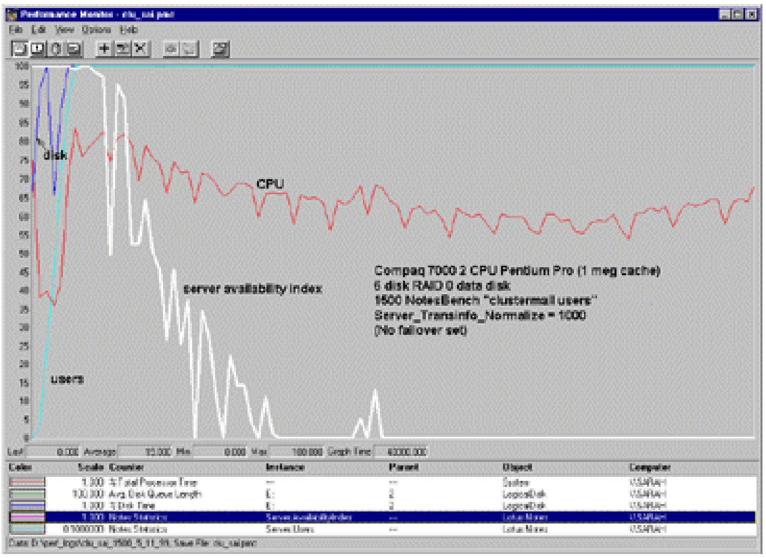
These are recommended initial settings for these specific machines based on our testing. It is highly likely that these values will need to optimized for each particular environment and system.

Determining the best SAT and STN values for your servers
To determine the best settings for these values, it is best to compare NT Performance Monitor graphs of the SAI while the server is under workload and then varying the SAT and STN values to see the behavior of the failover characteristics. You could even try this on your production systems. If you activate the Domino NT Performance Monitor statistics you can select the SAI as well as the normal NT stats. Reviewing these graphs should provide insight into your servers' performance. By using this insight and Performance Monitor (or another suitable tool) you should gain an understanding for how to optimally adjust the values of SAT and STN for your systems.

SAI Principle
As the server is nearing its capacity, the preferred behavior for the Server Availability Index (SAI) is to go quickly below the value specified for the Server Availability Threshold (SAT) and thus trigger the offloading of a portion of the server’s workload to another server in the cluster capable of accepting the workload. After rejecting additional workload, the server SAI should quickly recover and allow the server to start accepting workload again. One way to picture this desired behavior is as roughly that of a sine wave which centers on the SAT value.

The following three NT Performance Monitor graphs help illustrate the SAI principle. Note how the SAI drops off dramatically in Figure 1 with STN set to 1000. This behavior is considered undesirable. This results because the STN is set too high.

Note: Failover is disabled with the NOTES.INI setting MailClusterFailover=0.


Performance Monitor - clu_sai.pmc
File Edit View Options Help
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
disk
CPU
server availability index
users

Compaq 7000 2 CPU Pentium Pro (1 meg cache)
6 disk RAID 0 data disk
1500 NotesBench "clustermail users"
Server_Transinfo_Normalize = 1000
(No failover set)

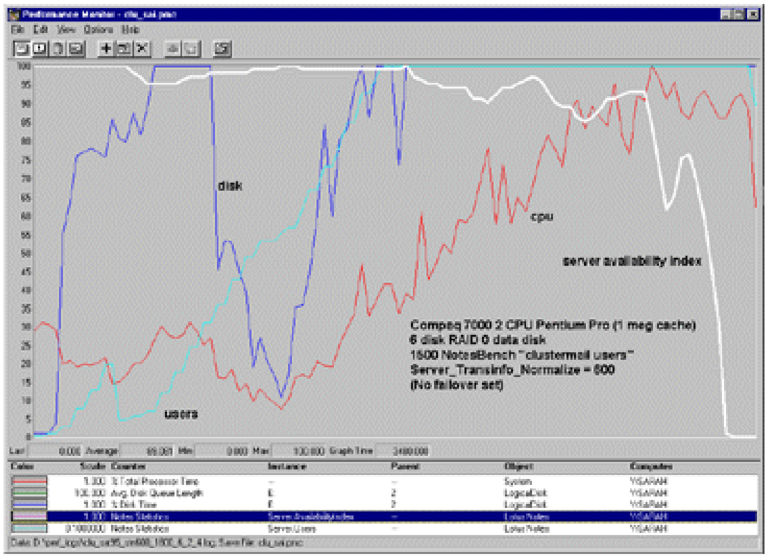
Last 0.000 Average 15.000 Min 0.000 Max 100.000 Graph Time 40000.000

Color	Scale	Counter	Instance	Parent	Object	Computer
🟥	1.000	% Total Processor Time	---	---	System	\\SARAH
🟩	100.000	Avg. Disk Queue Length	E:	2	LogicalDisk	\\SARAH
🟦	1.000	% Disk Time	E:	2	LogicalDisk	\\SARAH
🟪	1.000	Notes Statistics	Server.AvailabilityIndex	---	Lotus Notes	\\SARAH
⬜	0.1000000	Notes Statistics	Server.Users	---	Lotus Notes	\\SARAH

Data: D:\perf_logs\clu_sai_1500_5_11_99; Save File: clu_sai.pmc

Figure 1: STN set to 1000

Note how the SAI initially drops off even more gradually in Figure 2 with STN set to 600. This behavior is considered much better than in the previous case. The reason for the more gradual change in the SAI is that the value of STN is set to a value that is closer to the actual average time it takes the Domino server to complete a transaction.


Performance Monitor - clu_sai.pmc
File Edit View Options Help
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
disk
cpu
server availability index
users

Compaq 7000 2 CPU Pentium Pro (1 meg cache)
6 disk RAID 0 data disk
1500 NotesBench "clustermail users"
Server_Transinfo_Normalize = 600
(No failover set)

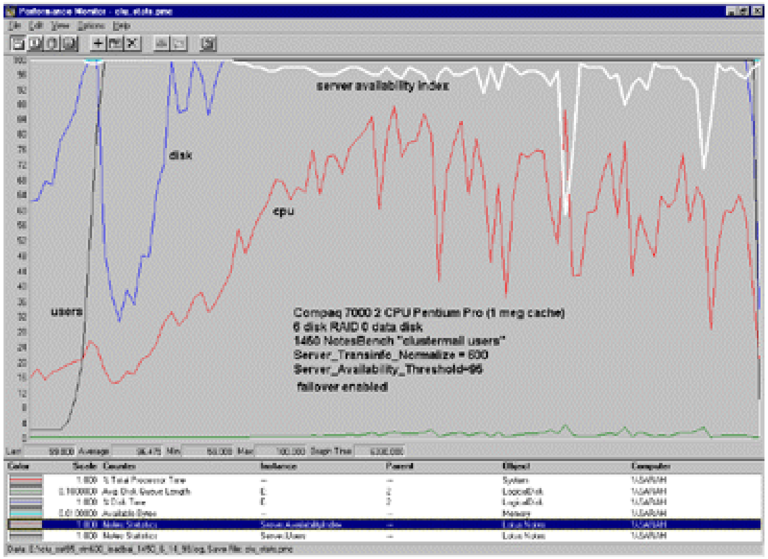
Last 0.000 Average 69.061 Min 0.000 Max 100.000 Graph Time 3480.000

Color	Scale	Counter	Instance	Parent	Object	Computer
🟥	1.000	% Total Processor Time	---	---	System	\\SARAH
🟩	100.000	Avg. Disk Queue Length	E:	2	LogicalDisk	\\SARAH
🟦	1.000	% Disk Time	E:	2	LogicalDisk	\\SARAH
🟪	1.000	Notes Statistics	Server.AvailabilityIndex	---	Lotus Notes	\\SARAH
⬜	0.1000000	Notes Statistics	Server.Users	---	Lotus Notes	\\SARAH

Data: D:\perf_logs\clu_sai95_stn600_1500_6_2_4.log; Save File: clu_sai.pmc

Figure 2: STN set to 600

Next we enable failover with an STN of 600 and with an SAT of 95. Notice the SAI generally remains high. This is the desired behavior. The server redirects new requests to other cluster members when the server availability index drops below the SAT of 95. It then quickly recovers and is ready to accept new requests. This is the sine wave mentioned in the SAI Principle above.


Performance Monitor - clu_stats.pmc
File Edit View Options Help
server availability index
disk
cpu
users

Compaq 7000 2 CPU Pentium Pro (1 meg cache)
6 disk RAID 0 data disk
1450 NotesBench "clustermail users"
Server_Transinfo_Normalize = 600
Server_Availability_Threshold=95
failover enabled

Last 99.000 Average 96.475 Min 68.000 Max 100.000 Graph Time 600.000

Color	Scale	Counter	Instance	Parent	Object	Computer
🟥	1.000	% Total Processor Time	--	--	System	\\SARAH
🟩	0.1000000	Avg. Disk Queue Length	E	2	LogicalDisk	\\SARAH
🟦	1.000	% Disk Time	E	2	LogicalDisk	\\SARAH
🟦	0.0100000	Available Bytes	--	--	Memory	\\SARAH
⬜	1.000	Notes Statistics	Server.AvailabilityIndex	--	Lotus Notes	\\SARAH
⬛	1.000	Notes Statistics	Server.Users	--	Lotus Notes	\\SARAH

Data: E:\clu_sat95_stn600_loadbal_1450_6_14_99.log; Save File: clu_stats.pmc

Figure 3: STN set to 600, SAT set to 95

Testing summary

We concluded that to improve cluster workload balancing, one should vary the value of SAT and STN in NOTES.INI so that it better reflects the Domino server transaction performance of the high end machines available today. The benefits of this type of tuning are that it allows you to better utilize the capabilities (CPU, memory, faster disk subsystem) of your Domino server.

Determining R5 clustered mail workload and optimal number of cluster replicators

In this section, we present the data and implications resulting from clustered mail testing. We found that the use of multiple cluster replicators enhances the performance of the cluster. We also determined that transaction logging enhanced disk utilization on the clustered server. Read on for more specifics about our test environment and how you can test the benefits of multiple cluster replicators in your environment.

Our cluster test environment contains two servers. “Users" are on one server referred to as the active server. The other server is referred to as the standby server. Mail messages are not routed. All databases are replicated to the standby server.

Active server

Compaq 7000 with 2 Pentium Pro 200 MHz processors (1 MB L2 caches)
2 GB RAM
512 MB page file
2 Smart-2DH-Array SCSI controllers (one array has 6 RAID 0* data disks and one array has 2 disks for the OS, page file, Domino executables, and transaction log**)
Network: 10 Mbit Ethernet
OS: Windows NT Server 4.0 Service Pack 4
Domino Release 5.0

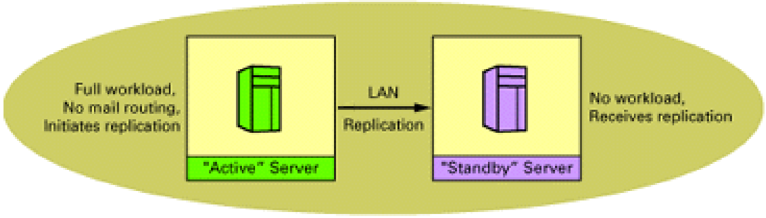
Standby server

Dell Dimension with 2 Pentium II/300MHz processors
512 MB RAM
1024 MB page file

6 disk PowerEdge enhanced RAID 1 disks (4 disks for data and 2 disks for the OS, page file, Domino executables, and transaction log**)
Network: 10 Mbit Ethernet
OS: Windows NT Server 4.0 Service Pack 4
Domino Release 5.0

*RAID 0 is not recommended for data disks since it has no fault tolerance features. Enhanced RAID 1 is generally recommended for data disks.

**The OS and Domino executables share one disk and the transaction log and page file share another. Ideally, the page file should have its own disk and the transaction log should have its own RAID 1 disk set.



The server on the left above (active server) that has the workload on it is called an “active” cluster member. The one on the right (standby server) is called a “standby” member because the only load on it is cluster replication.

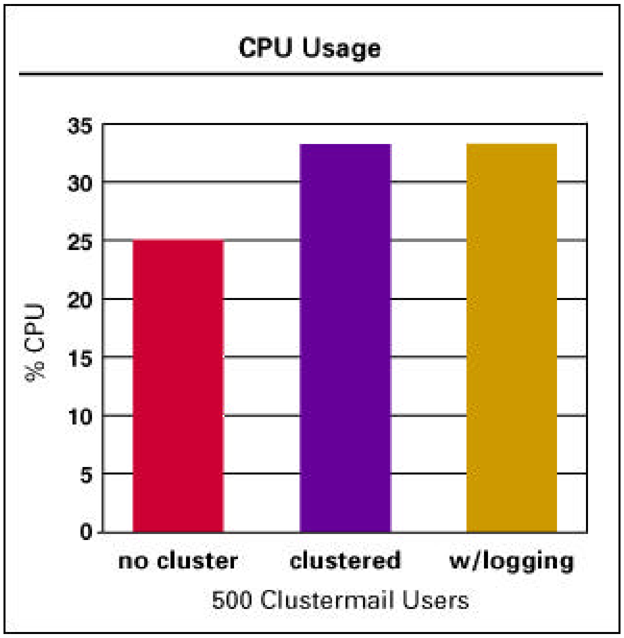
Although this configuration is not normally used in production systems, it is the preferred testing configuration because it is easier to measure the replication loads of the receiving system and system pushing the cluster replications separately. Once you know what each load is separately, then it’s easy to predict what will happen when you add a user load to the second server. Thus our use of NotesBench testing allows us to show accurate CPU utilization increases and disk utilization separate from any cluster replication impact on the server.

About the workloads
The NotesBench clustered mail workload as described in the previous section was used to generate the workload on the server.

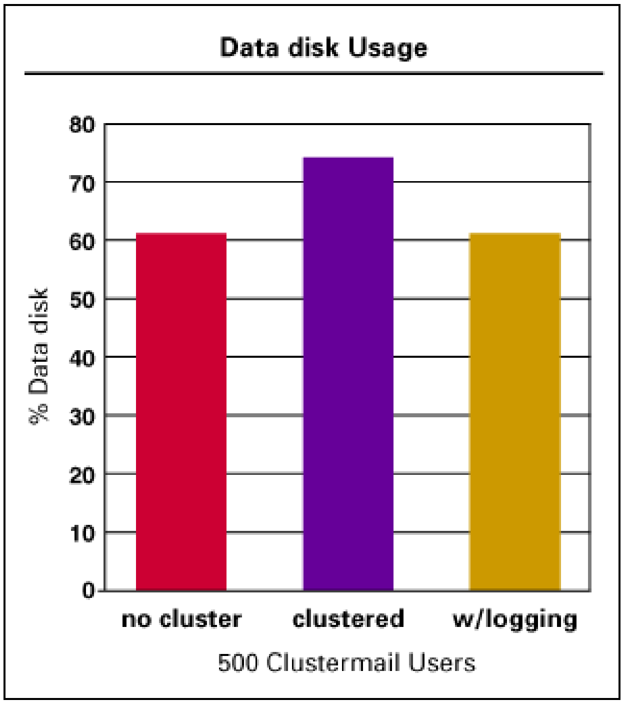
Test parameters
Child NOTES.INI settings:
NthIteration=6 (standard setting; sends mail 5 times in 8 hours)
NormalMessageSize=10000 (messages are 10K in size)
NumMailNotesPerUser=100 (100 messages in the Inbox at start of test)
NumMessageRecipients=3

Cluster testing with 500 “users” and one cluster replicator running on each cluster member indicates the following:

1. On the active server with clustering enabled, the CPU usage on the server under load increased by 32% (CPU percentage up to 33% from 25%) on a 2-CPU 200 MHz Pentium Pro system, and the disk utilization increased by 21%.



2. On the active server after transaction logging was enabled, the CPU usage remained the same and the disk utilization decreased by 21% (Data disk percentage down to 61% from 74%).



3. On the standby server, which had no direct workload running against it, the cluster replication activity for the 500 user test created a CPU load of 8% and 35% disk activity. The resource utilization went up slightly on the standby server after transaction logging was enabled on both servers. This slight increase is presumably due to the active server being able to replicate faster.

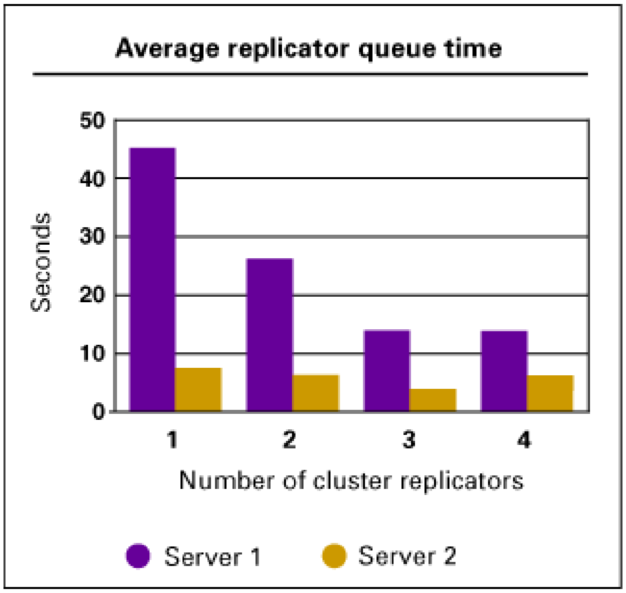
Note: If a workload were added to the standby server cluster member, it would create additional workload on the active server consisting of the cluster replication activity from the standby server to the active server.

Workload measurements when varying the number of cluster replicators
The number of cluster replicators was varied from 1 to 4 on the active server during the 500-user clustered mail test.

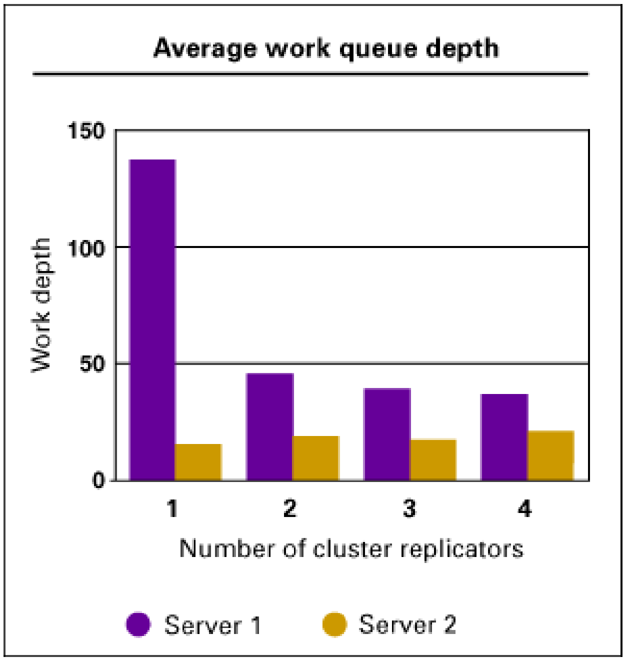
In the charts that follow, Server 1 was the active server and Server 2 was the standby server. The standby server had two cluster replicators in all tests. We can see, for the servers tested, noticeable benefits from the use of two or three cluster replicators. When multiple replicators are used, the cluster replication queue and work queue depth times are reduced. The cluster replicator CPU usage is lowest with two cluster replicators.

Note: If sufficient system resources are not available to support the additional replicators, adding additional replicator tasks will be counterproductive.

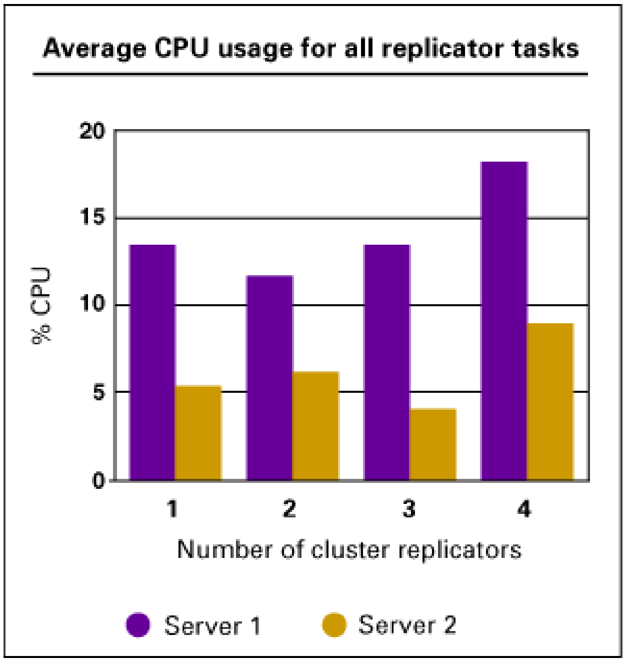
So you can see, for example, it took an average of 45 seconds to replicate a change from Server 1 to Server 2 with 1 cluster replicator; and it went down to 14 seconds when we used three cluster replicators.



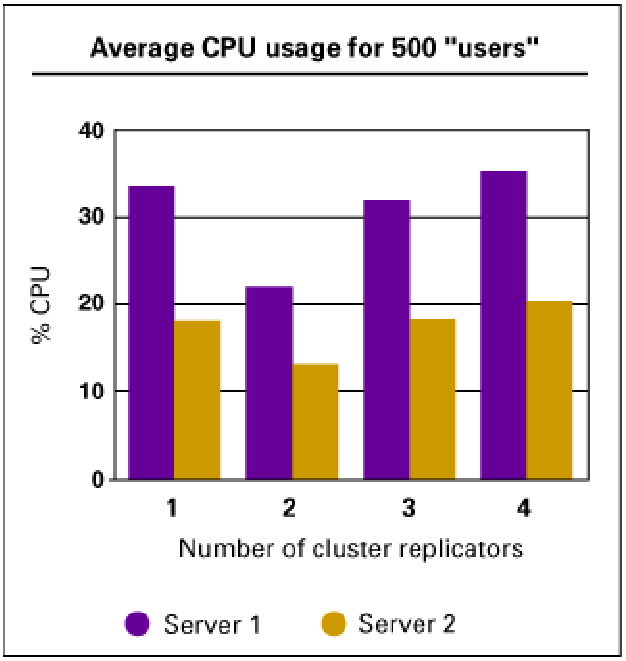
The work queue depth is the number of replications waiting to be processed by the cluster replicator. So, the smaller the queue the better.



The CPU usage is reduced when two cluster replicators are used.



So once again, this chart shows the CPU usage is reduced when two cluster replicators are used.



Recommended number of cluster replicators for a two node cluster
Our recommendation is to use two cluster replicators on each active cluster member for a two-node cluster. If you have plenty of reserve CPU, you may even run three cluster replicators to minimize the time for the replications to occur. To add a cluster replicator, simply add an additional "clrepl" to the ServerTasks setting in NOTES.INI. For example:

```
ServerTasks=
ROUTER,UPDALL,ADMINP,CLDBDIR,CLREPL,CLREPL,CLREPL
```

Testing summary

We recommend that you try running one more cluster replicator on each cluster member than there are cluster members to which the member is replicating. That is, if there are n nodes in the cluster to which a member is replicating, then try running (n+1) cluster replicators on that member. You can then vary the number from that baseline while monitoring how it affects the replicator queues and the CPU usage to find the optimal value for your installation.

Also, our testing showed that the addition of clustering can add approximately 32% more to your CPU usage on the server under load on a 2 CPU 200 MHz Pentium Pro system, and the disk utilization increased by 21% for the clustered mail workload. The addition of transaction logging helps reduce disk utilization. If both members of the cluster have the same active 500 user load, then you would need to add approximately 8% to the CPU utilization, and add approximately 35% to the disk utilization due to the replication from Server 2 to Server 1.

Sizing a cluster

Our cluster testing data (see part 1 of this series) indicates that if the Domino server has an intense workload with a large amount of write updates, then the resource impact that results from replication can be substantial. It should be mentioned that there is no exact way to accomplish sizing a cluster, however, an “educated guess” can be very beneficial.

To size a cluster, we will first assume that you have a running Domino server. We will use the resource performance information on the running server to be able to estimate the hardware for the cluster. We will assume that you want to create a two-server cluster with users and databases equally distributed. We assume this because customers who typically start out with clustering have a two-server cluster.

If there are multiple cluster servers with multiple replicas, then the sizing becomes even more difficult, but is still possible using the ideas contained in the article and combining those with your own production testing.

Additional workload is proportional
The additional workload that results from creating a cluster is proportional to how update-intensive the workload on the server is. The additional CPU requirements for a cluster member can range from a minimum of about 5% in an environment that has mostly reads, to more than three times that needed for the non-clustered base workload in a very intense workload environment. (Again, see **part 1** of this series for data.)

So the trick is to compare the activity of your typical user with the activity described in the tests in part 1 and 2 of this series, and work out a rough scaling factor. Then measure your existing CPU utilization, and estimate the resulting CPU using your scaling factor.

Remember that you will need to leave enough capacity to pick up the workload of a failed cluster member. It’s also not recommended to exceed an average of approximately 70% total CPU usage for a server under normal load.

Additional disk capacity is also proportional
The additional disk capacity requirements can also be estimated in the same way using the data examples. Measuring the current disk activity taking place will be the starting place in working out a scaling factor by comparing your typical user with the workloads in our tests.

Network bandwidth
To determine if you have adequate network bandwidth you must first measure the existing workload on the network with sniffer software or hardware. If the workload is already at or near 40% or more, you will need to upgrade the existing network or create a new intra-cluster network (as described in **part 1** of this series).

The additional network workload will be proportional to the increase in disk activity due to cluster replication. Usually network workload does not become a problem in a 100 MB network.

Memory requirements
The memory requirements will increase only slightly from what is currently needed in R5, which is approximately 200K per Notes client.

Forthcoming workload balancing and capacity planning tool -- "Activity Trends"

Distributing databases among cluster members can be one of the best ways to balance the cluster workload, however, it can be very time consuming to properly distribute the databases. Customers have been asking for a tool to help distribute databases, and a new IBM product is nearly complete that will make it easy to see, graphically, how the databases are distributed in your entire domain and which databases are "hot," or heavily used.

When run on an R5 server, the new tool can optimally redistribute your databases. Of course, this product will be very useful even if you don't have a cluster. Server consolidation tasks, for example, can also be made easier using this tool. In general, it can be used to collect and analyze server databases and connection activity within a group of heterogeneous Domino servers. The tool can perform workload balancing and capacity planning functions based on the data collected.

At the time of this article the release date for this product has not yet been set. Stay tuned for a future *Iris Today* article when this tool is released.

For more information on Activity Trends read *Lotus Notes and Domino 5 Scalable Network Design: Web Server Network Infrastructure* by John Lamb and Peter Lew, published by McGraw-Hill, ISBN 007913792X.

Conclusion

We hope this article will help Domino administrators of existing clusters improve their cluster performance, and encourage those without clusters to create one. The important thing to remember is to do the best possible job of correctly sizing the cluster members prior to deployment. After creating the cluster, closely monitor performance to anticipate bottlenecks. Those activities should provide you with a useful and efficient Domino server cluster.

ABOUT HARRY

Harry Murray joined the Iris Performance Group in 1998. He is currently involved in the testing of Domino R5 on Windows NT systems. Prior to joining Iris, he worked for Digital Equipment Corp. in their performance group doing NotesBench testing of Domino on Digital servers. Before that, Harry was involved in the system management of many Digital production systems and was manager of System Technical Support in a number of Digital facilities.

ABOUT GARY

Gary Sullivan joined IBM in 1987 and is currently a marketing support specialist in the IBM Lotus Integration Center. Prior to joining IBM, Gary was the Capacity Planning manager for FMC Corporation. Before that, he worked as a research consultant for Atlantic Richfield.

What do you think about this article?

Register Here!

About this Site | **Feedback**
Lotus Home | **IBM Home** | **Iris Home**