

# A Primer on Domino Clustering

Let Domino clusters solve your availability and server load problems.

by Danny Lawrence

**D**omino clustering is a way to ensure high availability for your Domino data or provide load balancing over a group of servers. Using a specialized version of the Domino replicator and some additional server tasks, your Domino server can be configured to route database requests to any one of several servers working together in a cluster.

In this article, I will describe Domino cluster setup, clustered server configuration, the management tasks that make clustering happen, and several different scenarios for Domino cluster deployment that provide failover or load balancing for your Domino data. This article is aimed at experienced Domino administrators who are familiar with server setup, configuration, and management. There are some differences in Domino clustering between R4 and R5, but the general concepts apply to both versions. Unless there is a specific mention of a version, you can assume that this article applies to both versions.

## What Is Domino Clustering?

Domino clustering is included in the Enterprise server license; it is not a part of the Mail or Application server. Each server in the cluster requires an Enterprise license. When you

set up a Domino cluster, the Cluster Replicator (CLREPL) server task runs on each server in the cluster. Unlike the standard replicate task that activates on a fixed schedule, CLREPL is event-driven. Any change to a database that CLREPL is monitoring forces replication of that change to other replicas of that database throughout the cluster. The cluster replicator batches changes to a given database in order to effectively use network bandwidth while keeping all the replicas as up to date as network throughput allows.

Domino clustering is application-level clustering in that it synchronizes only application (e.g., Domino) data. Because of this and because the Domino application is operating system independent, Domino clusters can include servers running different operating systems or even different versions of Domino.

Domino clustering was originally designed for Notes clients, but with the advent of R5, Domino clustering has been enhanced to provide failover to browser clients through the Internet Cluster Manager (ICM). ICM is a separate server task that intercepts HTTP requests to a Domino server and distributes them among the servers in a cluster. It has its own configuration settings in the server document, but still uses Domino clustering at its core.

What Domino clustering is *not* is hardware-level fault tolerance such as IBM's High Availability Cluster Multi Processing for AIX (AIX HA/CMP) or Microsoft's Wolfpack cluster server. However, these operating system features can be used along with Domino clustering to provide additional high availability functionality to your user base.

## Create a Cluster

As previously stated, you'll need Enterprise licenses for all Domino servers in the cluster; an administrative server should also be assigned to the Notes Address Book (NAB). To create a cluster, open the Server/Servers view in the NAB, select the servers you want to be clustered and click the Add to Cluster action button, then choose a name for the cluster. This generates an AdminP request that makes the changes to the NAB and also creates a new database, the cluster database directory, which is also replicated to all of the servers in the cluster. After that, you must replicate the NAB around to all the servers in the cluster so that all of the servers receive the change. Next, you must replicate all of the databases you want clustered onto all of the servers that will be serving that database.

Note the difference here between hardware fault-tolerant clustering and Domino's application clustering. With Domino cluster-

ing, all of the databases do not need to be on all of the servers. For example, in a three-server setup, you could have two servers each back up half of the databases on the other server. Most hardware fault-tolerant systems require that each machine in the cluster have the exact same configuration.

After the initial setup is complete, you should add the two clustering tasks—Cluster Database Directory (CLDBDIR) and Cluster Replicator (CLREPL)—to the server tasks line in Notes.ini. Make sure that CLREPL follows CLDBDIR, as the cluster replicator requires the information generated by the cluster database directory. It is a good idea (although not required) to dedicate a network card and subsection of your LAN to handle cluster traffic only, to ensure that the cluster replicator has maximum bandwidth available. To do this, add a second network interface card (NIC) to your server and use the operating system to bind an IP address to it. Then, in the Domino server configuration, create an additional Notes network port on each server that uses the TCP/IP driver and that additional IP address. Next, you need to configure your server so that it uses these ports for the cluster traffic. To do this, place the `Server_Cluster_Default_Port= PortName` and `Server_Cluster_Probe_Port= PortName` in the Notes.ini of the servers. The first command tells CLREPL to use the specified port for cluster replication; the second command tells CLREPL to use the port to exchange information about the status of the servers in the cluster. Connect the secondary NICs to a hub or an isolated portion of the network, and your cluster traffic will be unencumbered by any other network traffic.

If you are using clustering for your users' Mail databases, you must also add the line `MailClusterFailover= 1` to the Notes.ini file in all servers in your Notes domain. This tells the Notes router to deliver mail to the other servers in the cluster if the user's home server becomes unavailable.

### Go Configure

After setting up the cluster, you need to configure the servers in the cluster. The key variable that you need to deal with on a clustered server is the server availability threshold. This allows you to set how busy your server gets before it starts to route requests to another server. The server availability threshold is compared to the current server availability index. If the current server availability index is greater than the server availability threshold, then the server is designated as busy, and requests to that server are routed to other servers in the cluster. Note that even if a server is designated as busy in the cluster, it will continue to serve requests if there is no other

database replica available.

Though the server availability index is measured on a scale from 1 to 100, it's not measuring the percentage of the capacity that the server has in use. It's measuring how much longer a given action takes, as opposed to how long it takes when the server is lightly loaded. The actual formula is  $100 - (\text{current response time}/\text{optimal response time}) = \text{availability index}$ . So, if an action (for example, Database open) currently takes 5 seconds, but when the server is unloaded takes 0.5 seconds, then the availability index would be 90. The current server availability index is available from the server console by typing `show stat server.availabilityindex`; it is also logged by the STATREP task whenever it runs and is stored in the statistics database.

Now that you have created your Domino cluster, you need to set up the servers for the type of services you need. The following three scenarios describe different uses for Domino clusters and show you how to set up a cluster for either load balancing or high availability. Domino cluster can be used in other contexts, but these scenarios are good examples of basic Domino clustering.

### Scenario One: Failover Clustering, One to One

In this scenario, the secondary Domino server (server 2) is waiting in the wings in case the primary server (server 1) goes down. If server 1 fails, users are routed to server 2 for Domino services. When server 1 comes back up, all users are routed back to it. This is the basic high-availability server scenario. Because Domino clustering is at the application level, server 2 could also be used as the server from which to get good backups, because most of the time it is not in use. Server 2 could also be

## ***Domino clusters can include servers running different operating systems or even different versions of Domino.***

less powerful than server 1 (although when server 2 is in use, your users will want similar levels of performance, so don't skimp too much on server 2).

### Setting Up the Servers

Set up server 1 and server 2 in a cluster. However, since server 1 will perform all the work when it is up, set the server availability threshold on server 2 to a high number, such as 90, while leaving server 1 set to zero. This setup will force server 1 to handle most of the requests when both servers are running. Should server 1 go down, all requests will failover to server 2 (since server 2 is the only active server in the cluster, the availability threshold is ignored). When server 1 is ready

to go back online, set its availability threshold to 100 so that when it comes back up, it will not serve any requests until the cluster replicator has re-synchronized all the databases. Then set the availability threshold of server 1 to zero and move server 2 up to 100, for a few hours (or maybe overnight) before resetting the default to 90. You must do this to force those users who were switched over to replicas on server 2 while server 1 was down back over to server 1, thus maintaining normal usage levels at server 1.

This scenario has several benefits:

- Users experience no degradation in performance when server 1 goes down (assuming servers 1 and 2 are of similar capabilities).
- Server 1 failure is no longer a crisis for the administration staff, who can now spend more time identifying the causes of the failure and taking corrective action before bringing the server back up.
- Server 2 can be used as the source of your backups of all the databases—since it is not actively in use (except when server 1 is down)—or to run the directory catalog, billing, or another background, low-intensity server task.

### Scenario Two: Failover Clustering, Many to One

In this scenario, server 3 functions as the backup server to servers 1 and 2. If server 1 or 2 (or both!) fail, users are routed to server 3. This is a variation of the failover scenario, where one server acts as the backup to several other servers. Also note that while this article presents a three-server model with one server backing up the other two, Domino clusters can contain up to six servers. In theory, one server could provide failover to five other servers.

This scenario is similar to the previous one, in that there is a hot server waiting in the wings for either one of the other servers to fail. The variation to this scenario is that server 3 acts as a backup for both servers 1 and 2.

### Setup

To set up this configuration, create a cluster that contains servers 1, 2, and 3, and replicate all the databases (or, all the databases that you want to be accessible should server 1 or 2 go down) onto server 3. Server 3 should have as much disk space as servers 1 and 2 combined, because it might have to store all the contents of server 1 and server 2.

During normal operation, set server 3's availability threshold to 90 and leave server 1


and 2 set at zero. Then, when server 1 or 2 fails, all database requests will get routed to server 3. When the down server is brought back up, set its availability threshold to 100 for a brief time to allow the cluster replicator to synchronize the databases. Then, reset the down server's threshold back to zero and change the threshold of server 3 to 100, to force the users back onto server 1 or 2. Finally, reset server 3's availability threshold back to 90. This scenario gives you all the same benefits of the previous one, and, in addition, you don't need to double your hardware investment to provide high-availability services to your user base.

### **Scenario Three: Load Balancing**

In this scenario, a cluster of servers, servers A, B, C, and D, act as a single "megaserver" to handle all requests to any of the databases in the cluster. This scenario is useful for heavily trafficked databases in production environments that also need high availability. This scenario is less useful for mail files because mail files tend to remain open all day; load balancing would not function properly because the server availability threshold is only checked when a database is opened, so all requests will be routed to one server throughout the day. An example of an efficient usage for this type of clustering would be a centralized listing of parts that were accessed via lookups from other databases or a knowledge base that was intermittently

server availability statistics are kept in the STATREP database along with the other statistics in the Statistics Reports—Clusters view. You can also enter the console command *show stat server.availabilityindex* to get an instant reading of the server's availability. If one server seems to be handling more requests than the others, you should raise its availability threshold slightly so that it is less available to the cluster, thereby forcing more requests to the other servers in the cluster.

### **Circle the Wagons**

Domino clustering is a useful and flexible tool that can add high availability or load balancing to your Domino infrastructure. Domino clustering makes life easier for administrators, improves the user's experience, and expands infrastructure capability at a lower cost than other similar solutions. 

Danny Lawrence is the president of Tiassa Technologies, Inc., an Advanced Lotus Business Partner specializing in Lotus Notes and Domino infrastructure support and development. He is also a frequent contributor to Usenet's [comp.groupware.lotus-notes.admin](mailto:comp.groupware.lotus-notes.admin) forum. Danny can be reached as [tiassa@ix.netcom.com](mailto:tiassa@ix.netcom.com).

#### REFERENCES AND RELATED MATERIALS

- *Domino R5 Clustering with Netfinity Servers* (SG24-5141-01)
- *High Availability and Scalability with Domino Clustering and Partitioning on Windows NT* (SG24-5141-00)

## ***Domino clusters can contain up to six servers. In theory, one server could provide failover to five other servers.***

accessed by many people.

### **Setup**

Set up all the servers in a single cluster and place replicas of all the databases that you want served by the cluster on each of the servers in the cluster. Once the cluster is running, you will need to tune the cluster so that all of the users are served as efficiently as possible. Do this by setting the server availability threshold on each of the servers. Start by setting the threshold to  $100/n$ , where  $n$  is the number of servers in the cluster. In this example, the availability threshold on each server would be set to 25. However, unless each server in the cluster is of the exact same specifications (and even identical servers can perform differently in the field), their capabilities will differ. So, each server may need a different threshold setting in order to balance the workload. You should monitor the servers during peak time to see that each is handling its share of the server traffic. The