In this chapter

- What Is Cluster Cats for ColdFusion Application Server 4?
- Understanding Cluster Cats

What Is Cluster Cats for ColdFusion Application Server 4?

Included with the Enterprise version of ColdFusion is Cluster Cats by Bright Tiger Software. A software-based load-balancing solution, Cluster Cats monitors your ColdFusion Application Server (CFAS) and can redirect requests away from a server that is beginning to enter a "busy" state. Note that Cluster Cats does not work on the network layer. When Cluster Cats redirects requests to another server, it does so by redirecting to the URL of another machine in the cluster. This means that if your server is completely out of commission (that is, not turned on), Cluster Cats cannot communicate with it and therefore cannot redirect requests away from it.

Perhaps the most attractive thing about using Cluster Cats for your load-balancing solution is its integration with the CFAS. Because Cluster Cats responds to elements of the CFAS, you get load balancing that is specific to your ColdFusion-based application. You get this benefit in addition to general failover and machine alerts.

An important point to note about Cluster Cats is that the version shipping with CFAS 4 does not include support for replication; however, the full version of Cluster Cats, available from Bright Tiger, does have replication built in. Replication is the ability to put your ColdFusion scripts on one machine in your cluster and have those scripts update on all machines in the cluster. Many administrators overlook replication when planning their load-balancing strategy. However, as code bases grow, and server architectures get more complex, replication plays a crucial role in content synchronization. Later this chapter outlines some simple strategies to include replication from the onset, and thus keep your job as server administrator manageable as your server cluster grows.

Understanding Cluster Cats

As you have learned in the preceding two chapters, many different methods for clustering servers are available. In particular, the methodologies employed in software-based clustering differ greatly. This fact will most definitely have an impact on how your server cluster serves your application. This section will give you a broad overview of the methodologies employed in Cluster Cats.
To begin, you need to note that Cluster Cats uses HTTP redirection to balance load across a cluster. Although, in general, this is a protocol-based redirection, it lacks certain network-level controls; therefore, I place it in the application layer of the networking hierarchy. In contrast, another popular software-based clustering strategy, Convoy Cluster (acquired by Microsoft in August 1998), attempts to use the network layer by advertising multicast MAC addresses to routing equipment. Because this discussion is about Cluster Cats, I will not go into the details of multicast MAC addressing. However, suffice it to say that Convoy makes your cluster look like one big IP address (see Figure 3.1), whereas Cluster Cats makes your cluster look like a lot of HTTP servers grouped together (see Figure 3.2). Depending on your situation, one approach may be better than the other, but they both do the same job.

**FIGURE 3.1 Diagram of HTTP server redirection.**

**FIGURE 3.2 Diagram of multicast MAC addressing.**

---

**CAUTION**

By extension of these differing methodologies, you must make concessions in the way you code your application to support the type of redirection being employed. In some cases, this solution may be as simple as not using absolute paths in your URL references; in other cases, you might need to make use of ColdFusion’s advanced Client State management options (discussed in a later chapter) to mitigate these server redirections.

Now, take a look at the components of Cluster Cats. The software works with two primary components: the Server component and the Explorer component. Each plays a critical role in the configuration and support of your Cluster Cats clusters. Note that although you can have a cluster consisting of a mix of Sun Solaris and Windows NT-based servers running Cluster Cats, you must have at least one Windows NT server to run the Cluster Cats Explorer. You cannot configure your cluster without the Explorer component.

**Cluster Cats Server**

The Cluster Cats Server component runs on Windows NT 4.0 and Sun Solaris. This Server component is the heart of the cluster. It controls configuration of a machine’s role in a particular cluster, handles redirection from the server in the event that load thresholds are breached, and controls access to the server based on restriction rules. The Server component must reside on all machines you are using for your cluster operations.

**Cluster Cats Explorer**

The Cluster Cats Explorer component runs only on Windows NT 4.0. It controls configuration of Cluster Cats clusters. I found that it was best to configure the server that the Explorer sits on as the primary server for all your incoming HTTP requests. I like to call it the "Controller Server." Its purpose is to handle all the incoming HTTP requests first and then redirect to other machines in the cluster as needed.

Tasks handled by the Cluster Cats Explorer include the following:
Creating and removing clusters

Adding and removing servers from a cluster and setting server load threshold levels

Restricting or providing access to servers

Registering cluster administrators

Selecting events for alarms and specifying the recipients of alarm email distributions

How Cluster Cats Works

As mentioned previously, Cluster Cats uses HTTP redirection as its principal methodology for distributing load across a cluster. This means that the user’s request is sent to another machine in the cluster and stays there until that machine needs to redirect the request to yet another machine.

For example, if a request comes to www.yourcompany.com, and www is too busy to handle new requests, it sends that request to www2 (or another machine based on availability of all machines in the cluster). The URL in the browser location box now reads www2.yourcompany.com (refer to Figure 3.1). Therefore, subsequent requests will go to www2 until that machine cannot accept more requests. At that point, www2 will attempt to redirect the request to another machine based on the information it has about other machine availability.

CAUTION

In the preceding example, the cluster essentially "cedes" control of redirections to the HTTP protocol. This means that there is no way to control what happens after that redirect is issued. If the target server crashes or otherwise does not respond to requests at all, a redirect fails with a Server unavailable response to the user. As of the writing of this book, the CFAS 4 implementation of Cluster Cats does not provide catastrophic server failover capabilities. This capability is expected in a later release of CFAS 4 and will mitigate this dilemma. With catastrophic server failover in place, the Cluster Cats HTTP redirection would know whether a server is available and thus not redirect a request to a failed server. This result is accomplished via each server listening for a heartbeat from other servers in the cluster. If a machine does not respond to the heartbeat within a specific period of time, another machine in the cluster assumes the IP address of the down machine (a process otherwise known as IP aliasing).

Let me contrast this shortcoming of Cluster Cats in the CFAS 4 implementation: If your server is up and running, and you just want to take it out of the cluster for a while for maintenance, you can restrict that server. Restricting a server in a Cluster Cats cluster causes all requests to that server to be sent to other machines automatically.

Sample Application

Say that you have a ColdFusion application that needs to be clustered across two ColdFusion application servers. This application does not use session variables but instead employs a "cluster neutral" method of saving state: client cookies. The name of the Web site you are clustering is
www.mycompany.com, and you will be using machines www1.mycompany.com and www2.mycompany.com in the configuration.

To provide two layers of load balancing, you can employ round-robin DNS in addition to Cluster Cats. By doing so, you can distribute general requests to both servers in the cluster. In this configuration, you can keep all load from initially being directed at one machine and better use resources. Although this result may seem inconsequential in the context of what’s being described, it will have an impact on what you set in the load threshold level section of each cluster member.

Creating Your Application Cluster

Begin by installing the integrated Cluster Cats software included with ColdFusion 4 on both machines. If you are using Windows NT on both machines, the setup routine installs the Cluster Cats Explorer. In this example, you will use the Cluster Cats Explorer only on www1.mycompany.com.

After you install the software, you will see a new service (in addition to the ColdFusion 4 services) named Bright Tiger Server service in the Service Control Manager. It must be set to Automatic and should be running so that Cluster Cats can work properly. You should always check the Service Control Panel after a significant installation. Doing so helps you understand the components of your server application.

From the www1.mycompany.com console, open the Cluster Cats Explorer. Your screen should look like Figure 3.1.

Right-click the Cluster Manager icon in the left pane of the Cluster Cats Explorer. You then see a dialog box that looks like the one in Figure 3.3.

FIGURE 3.3 The Create New Cluster dialog box.

You need to enter the Cluster Name (see the following sidebar), the Server Name, which is the name of the primary server in your cluster, and the License Key included in your ColdFusion 4 Administrator Guide. Be sure to enter the Fully Qualified Host Name of the first machine in your cluster in the Server Name box. In the example, you would enter www1.mycompany.com.

---

**Naming, and Why It Is Important**

Note that the first name you enter in the Create New Cluster dialog box will become the default administrative manager for the cluster. The administrative manager is responsible for allowing other Web servers to be added and deleted from the SmartCluster. Apart from this, you also need to name your SmartCluster something that is specific to your task. For example, if you plan to host several ColdFusion applications responding to multiple IP addresses on your server, you might want to create clusters that have the same name that exists in your CFAPPLICATION tag. Doing so helps you balance load based on application and not just on the server. If you have a particular application that is CPU intensive, for example, you can give it more of a particular machine in your cluster and redirect other application requests away.

---

Creating Cluster Members

After you create your initial cluster member from the Create New Cluster dialog box, you can add
other cluster members by right-clicking on your cluster name, choosing New, and then selecting Cluster Member. In the resulting dialog box (see Figure 3.4), enter the Fully Qualified Host Name. In the example, you would enter www2.mycompany.com.

**FIGURE 3.4** The Add New Server to Cluster dialog box.

At this point, you should have two servers listed in the Cluster Cats for ColdFusion Explorer. Cluster Cats might take a few seconds to connect to the servers in your cluster, depending on the size of your cluster or network load. I found this process to be uneventful, and after the cluster is synchronized, it does not have an impact on performance at all. Figure 3.5 shows an example of a cluster in use at my company.

**FIGURE 3.5** Cluster Cats Explorer with two active servers.

**Setting Load Threshold Levels**

You can set each server in your cluster to respond to two load thresholds:

- Load Threshold
- Gradual Redirection Threshold

Load Threshold is the top level you allow your server to accept before it enters a busy state. When this situation occurs, Cluster Cats begins redirecting requests to other servers in the cluster based on their availability. Cluster Cats continues redirecting requests until the actual machine load dips beneath this line.

Gradual Redirection Threshold defines a secondary threshold at which user requests start being redirected. As the name implies, if the Gradual Redirection Threshold is reached, Cluster Cats redirects a portion of the load—but not all of the load—to available machines in the cluster. The Gradual Redirection Threshold and the Load Threshold work together to provide a smooth transition of load (see Figure 3.6).

**FIGURE 3.6** Cluster Cats Load Monitor with examples of both types of redirection set.

---

**NOTE**

Be sure to remember the relationship of the Load Threshold to the Gradual Redirection Threshold. If the Gradual Redirection Threshold is set close to the Load Threshold, a greater number of requests is redirected from the server. Conversely, if the Gradual Redirection Threshold is set far from the Load Threshold, fewer requests are redirected from the server.

---

**Replication: Keeping Your Content in Sync**

Unfortunately, the replication features built into the Bright Tiger implementation of Cluster Cats are not included in the version shipping with CFAS 4. Replication is a mechanism employed to keep content on all your servers in sync with a central source. Anyone who runs multiple Web servers, grouped in a cluster, needs to be able to replicate content. Fortunately, many shareware and freeware...
utilities are available to help you synchronize your content. I like to use a utility named Robocopy from the Windows NT Resource Kit.

**Sample Replication Strategy Using Robocopy**

Robocopy and AT (the Windows NT Scheduler service) can combine to be a very effective replication strategy. Robocopy copies only changed files from a central source to target sources. It has an extremely large number of switches that allow you to customize your synchronization rules. Because Robocopy is a command-line utility, you can script fairly elaborate interactions with file systems using UNC names and the program’s switches. To use Robocopy, you must have the Windows NT Resource Kit. The following is an example of a Robocopy script I use to replicate content to another Web server and backup RAID drive. Note that I also use the command-line utility called blat to send an email of the Robocopy log back to a central administration desk. If you are writing scripts like these, you would be wise to include some contact information at the top; this way, if your script fails, systems staff know whom they should contact.

```bash
cls
echo off
echo **********************************************************
echo *****This is a G. Triad Automated Maintenance Script *****
echo *****Report Problems to gerryl@gtriad.com *****
echo *****or 908-497-0510 *****
echo **********************************************************
del C:\temp\latest2Repl.txt
del C:\temp\latest3Repl.txt
echo Replication to Server 2 Initiated.....working....
robocopy D:\Inetpub\wwwroot\ \Server2\90195 /E /PURGE >> C:\Temp\latest2Repl.txt
echo Replication to Server2 completed.
echo Replication to Server3 Initiated.... (BackingUp)....
robocopy D:\Inetpub\wwwroot\ \Server3\90195 /E /PURGE >> C:\Temp\latest3Repl.txt
echo Replication to Server3 completed.
echo Notifying central administration........
blat C:\temp\latest2Repl.txt -s "[MAINT] Server2 Replication Completed" -t gerryl@gtriad.com
blat C:\temp\latest3Repl.txt -s "[MAINT] Server3 Replication Completed" -t gerryl@gtriad.com
echo Notification Completed.
```

**Strategies for Clustering with Cluster Cats**

Because Cluster Cats works on the HTTP level, you can easily layer load-balancing strategies on top of the system. Combining hardware-based load balancing or round-robin DNS with Cluster Cats creates a two-tiered solution that can be configured in numerous detailed ways. Cluster Cats also provides a means for dealing with Web applications that require use of ColdFusion’s session variables.

**Dealing with Client State in Cluster Cats**

CFAS 4 uses certain types of client state variables that are written into the memory of a Web server. They are known as session and application variables. These variables apply to a user’s session or globally to an application defined with the CFAPPLICATION tag.

Obviously, if a user’s session is reliant on persistent variables stored on a particular server in your cluster, you have to keep the user on the same server. If the ColdFusion application you are clustering
requires that you not redirect users after they initially access a Web server, you can enable session-aware clustering. Do so by right-clicking the cluster you want to enable it for, selecting Configure, and selecting Administration (see Figure 3.7).

**FIGURE 3.7 The Administration tab in the Cluster Cats Explorer.**

Selecting this option simply keeps users confined to the same server after they initially hit the server. No redirections are performed.

---

**CAUTION**

Using session-aware clustering has a direct impact on how you balance load between the servers in your cluster. I believe it is best to set a Load Threshold of approximately 60 percent. Adjust this setting according to the load that your ColdFusion application generates.

---

**Using Round-Robin DNS**

Round-robin DNS (RRDNS), as you learned in Chapter 1, "Monitoring Server Performance," alternates requests from Web server to Web server based on entries found in your DNS server. In the example here, if you want to set up your cluster for RRDNS, you would enter two entries for the www machine: one pointing to www1.mycompany.com and another pointing to www2.mycompany.com. Thus, resolution requests for www.mycompany.com would alternate between the two servers in the cluster.

The big problem with RRDNS has been that if you have to take a server offline, every other new request to www.mycompany.com will hit a dead machine. With Cluster Cats, you can simply put that machine into a busy state and redirect requests. This type of general failover is included in the version of Cluster Cats shipping with CFAS 4.

Layering RRDNS on top of Cluster Cats provides a wider spread of load to your cluster. Whereas RRDNS never truly provides a true balance, Cluster Cats can help make that balance a little smoother by redirecting requests based on machine load. This load can then be controlled by the Load and Gradual Redirection Thresholds.