



BY GUY C. YOST

WinFrame Strategies: Using Load Balance

Load balancing in a WinFrame environment is a good way to leverage the server farm's ability to host the maximum number of sessions while ensuring that no single server attempts to handle too much. It also provides a level of fault tolerance, admittedly not as transparent or seamless as provided by clustering, but suitable for many applications.

A previous article about Citrix' WinFrame technology (*Technical Support*, February 1998) provided a functional overview and explored installation tips. Continuing the WinFrame series, this article examines the benefits and caveats of deploying a WinFrame server farm using the load balance option. Although the load balance option was possible using WinFrame version 1.6, it was clumsy at best and required explicit registry hacks. This and subsequent articles on WinFrame imply version 1.7.

To help understand what load balancing does for you, it's helpful to know what it doesn't do.

LOAD BALANCING VS. SERVER CLUSTERS

A WinFrame server farm configured with load balancing is sometimes mistaken for a "server cluster." A fundamental difference between WinFrame load balancing and server clustering is the overall client type being serviced. A typical server cluster will host the server portion of a traditional

client/server application while the "fat" client actually runs the front-end services. WinFrame clients are "thin" and attach to a WinFrame server through a remote control, terminal-like session. From the server session, single-tiered applications stored locally on the WinFrame server can be run. Alternately, a client/server application can be established between the WinFrame Server and a back-end server (which incidentally may be a true server cluster).

A server cluster refers to two or more physical servers acting as one logical server. That is, the users (or front-end applications) need only be aware of one server, but when all servers are functional, the user community realizes the aggregate server power of all CPUs and I/O bandwidth in the cluster. Fault tolerance is inherent in the standard cluster design. If a server fails, then the remaining servers assume the workload. Logically, a server cluster is simple, as depicted in Figure 1.

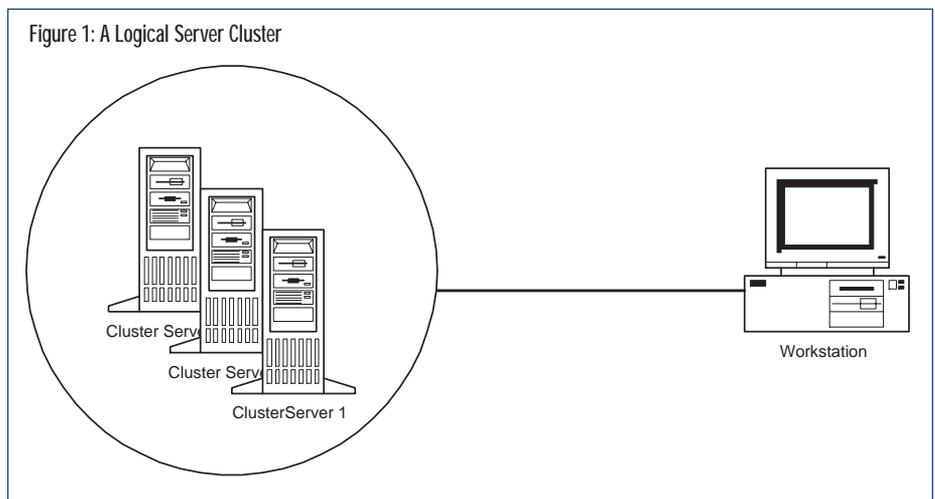
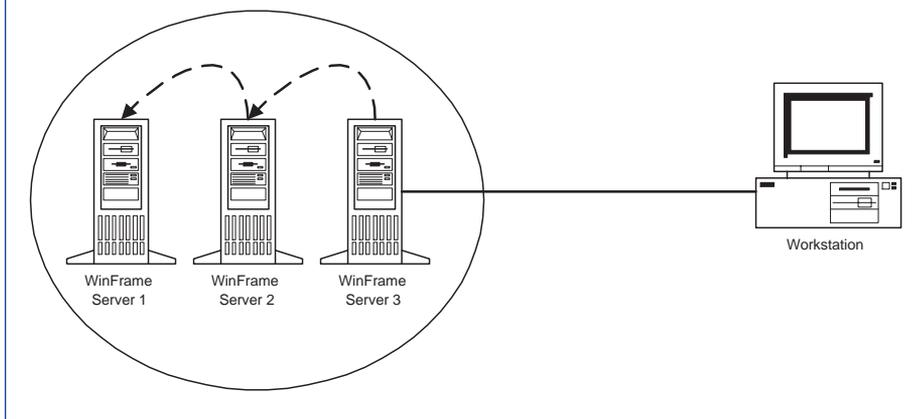


Figure 2: WinFrame Server Farm



Physically, the servers in Figure 1 would be attached to separate ports on a network hub, as would the workstation. However, in a TCP/IP network, each server would have a unique management IP address but would share a “cluster” IP address. The shared cluster address is what the user community sees, whereas network administrators can Telnet, FTP, etc., into specific hosts within the cluster using a host-specific IP address.

A load-balanced WinFrame Server farm can be networked physically identical to a cluster; however each server has its own IP address and the farm does not share a single cluster address. From a network and performance standpoint, load balancing a WinFrame server farm is similar to server clusters with the overall benefits offered, but there are some notable differences, as well. Even though WinFrame clients can be configured to request a single logical connection, when clients attach to the farm, they attach to a particular server. The performance realized by a client is determined by the abilities of the currently attached server. If the server fails, then the currently attached sessions are lost and the users will need to re-establish a WinFrame connection to another available server. For many applications, this offers an acceptable level of fault-tolerance but it is not totally transparent to the user as in a true-cluster environment. Logically, connections to the server farm behave as shown in Figure 2.

DOMAIN PREREQUISITES

All servers that participate in a load-balanced group must belong to the same logical MS domain. This implies that

every server farm will have at least one computer configured as a domain controller (PDC) and perhaps two (a BDC) for fault tolerance. The Domain Controller can be a straight NT server, but it is recommended to use a Citrix WinFrame as the Domain Controller in order to leverage advanced features such as the Citrix Browse Master as explained in the section “Adding a Master Browser.”

**A load-balanced WinFrame Server farm
can be networked physically identical to a cluster;
however each server has its own IP address
and the farm does not share a single cluster address.**

INSTALLING THE LOAD-BALANCE OPTION

Each WinFrame server in a server group must have the WinFrame Load Balancing Option Pack enabled to participate. It is possible, however, to have multiple load balance groups within one domain. For example, you could configure multiple instances of Figure 2 within a single domain, each with its own load-balance application or “group” name. Clients would then specify the group name in their connection setup, as shown in Figure 3.

To install the WinFrame Load Balancing Option Pack on a WinFrame server, you simply logon to the server as Administrator, run the WinFrame Licensing program in the Administrative Tools group, enter the License Serial number in the pull-down “Add License” menu, and activate the license by contacting Citrix via phone or web. In actuality, then, you really don’t install the

load balance option; you enable it. This tends to annoy consumers when the expense of this option is \$4,995/15 users or \$333/user even though the software is inherent to the OS product. All that needs to be done is enter a licensed serial number, and then activate the license.

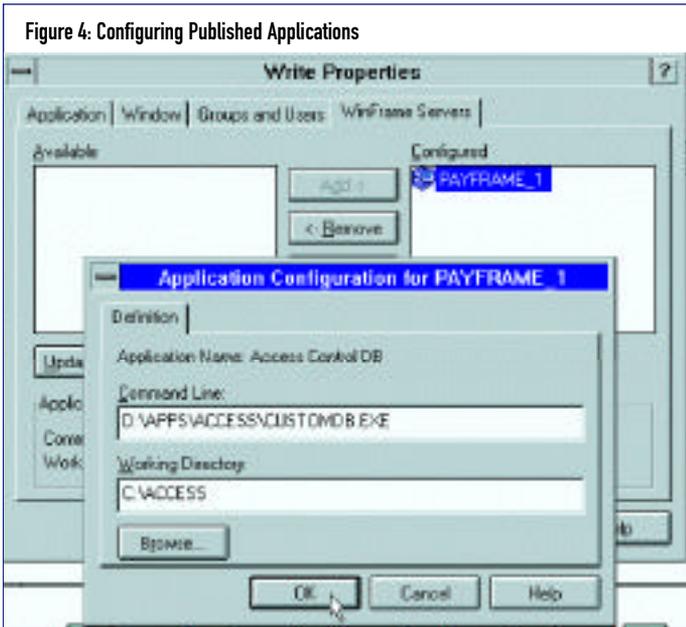
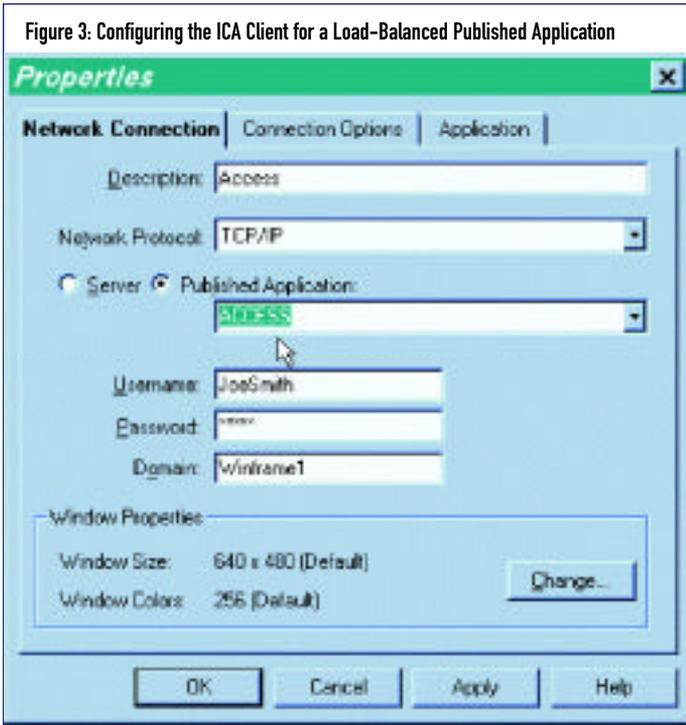
The next step is to configure the server to “advertise” its load-balance group or application name to other servers in the farm. This is done in the Application Configuration utility in the Administrative Tools program group which has a “wizard” style routine to guide you through the screens. The only non-intuitive prompt you may encounter would be the Application path menu field. Leave this blank if you want the users to connect to the desktop, but fill it in with the application path/executable file name and working directory if a specific application is to be launched upon connecting to the server farm. This is shown in Figure 4.

After configuring basic load-balance functionality, you can optimize the load-balance criteria based on your specific application needs. Still within the Application Configuration utility, simply highlight the published application or group name and choose Configure from the main menu. From there, choose Load Balancing as shown in Figure 5.

LOAD-BALANCE FACTORS

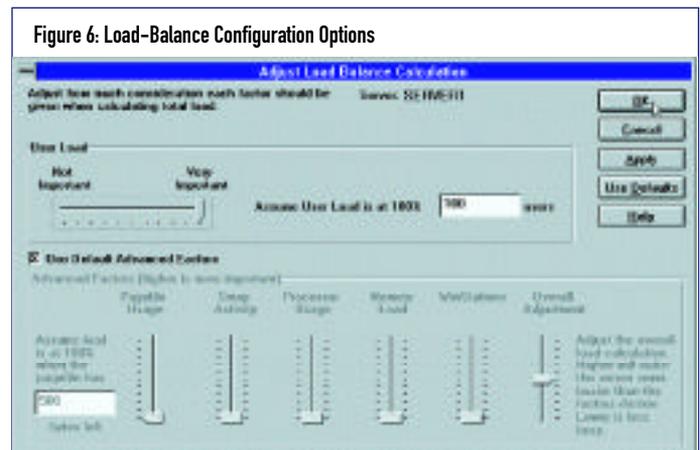
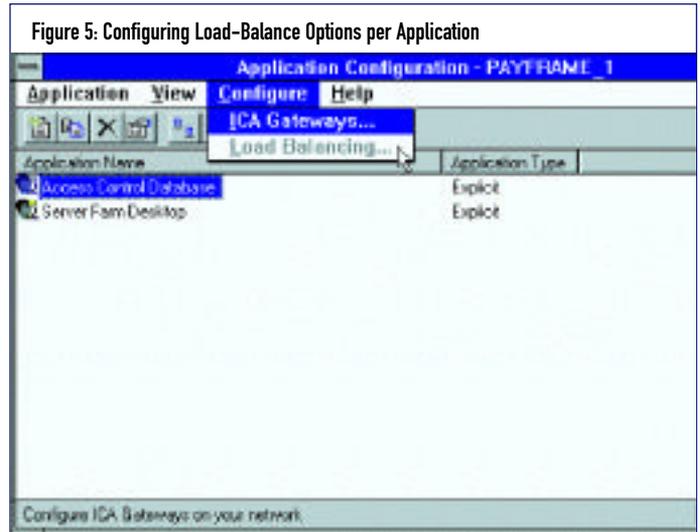
The manner in which network connections are distributed to other servers is based on overall server utilization (comprised of CPU utilization, disk activity, number of connections, and available RAM). Load balancing can query and test a variety of performance factors, depending on the type of work the servers are performing. Because the maximum number of sessions a WinFrame server can handle is mainly based on how much RAM is installed in the server, it is possible to configure load balancing so that a server’s availability is weighed primarily according to the amount of free RAM.

In addition, you can configure load balancing to pay more attention to CPU utilization or the number of client connections. This makes sense because it’s possible that one server could be hosting 30 low-impact sessions while another server might only have five sessions, each of which is running a CPU-intensive application thereby making the server less “available” to the farm.



Accordingly, if a 30-user license is installed on the “stressed” five-user server, other servers can use the remaining 25 licenses if the licenses are pooled. (Pooling is configured from the Licensing Manager in the Administrative Tools program group.)

You have the option to configure and weigh all utilization factors to best suit your specific application. The only factor that I’ve noted missing is the ability to measure network traffic on the server’s NIC. By default, all factors are taken into consideration and an overall “score” is used to pass client connections onto other servers. Specifically, if a client connects to a server that already has working connections, load-balancing software then queries other servers about their utilization and passes the connection on to a less utilized server. In any case, the ultimate client connection is to a specific server within the farm and not to the farm as a whole. This



behavior persists even at the low-end when only one or two connections are being used per server.

Figure 5 shows the load-balance configuration options. You can accept the defaults or simply adjust the file, processor, memory, or connections parameters by moving the corresponding slide bar. Is it a good idea to adjust these parameters? It depends entirely on your ability to judge an application’s real “needs.” A simple and effective way to determine what resources are most used by an application is through the Performance Monitor, located in the Administrative Tools group. Simply monitor overall CPU, memory, and disk usage to determine the application’s characteristics.

When you have an idea of how an application behaves, you can adjust the following options:

User Load — the ratio of currently attached users to the number of pooled licenses or maximum number of users the system can support, whichever is smaller. The maximum number of users is the number of WinStations defined per protocol in the WinStation Configuration utility or the number of users specified in the User Load section, as shown in Figure 5.

Pagefile Usage — the ratio of the current pagefile size to the allowed minimum space left in the pagefile. Note that you can specify when page files have consumed a measurable amount of the hard drive space.

Swap Activity— the number of times the pagefile is accessed per second. If increasing the importance of this setting increases server farm performance, than this is an indication that the servers would definitely benefit by adding more RAM.

Processor Usage — the percent of time the server's CPUs are busy processing user or OS requests. Raise this setting if the applications prove to be more processor-intensive than compared to the other parameters.

Memory Load — the ratio of total installed memory to currently available memory. Raise this value if your application requires more proportional memory resources than CPU or file.

WinStations — the ratio of maximum allowed/configured clients to currently free clients.

Overall Adjustment — determines the overall calculated load of a server. The middle (default) setting means no adjustment over other servers.

ADDING A MASTER BROWSER

We've seen how various and specific application resources can be promoted or demoted in terms of importance. This makes for practical negotiation between each server with load balancing enabled, but we can further improve how client traffic is distributed by making an "educated guess" from a central distribution server before forwarding connections to specific servers. Thus, to further the efficiency of distributing clients across a load-balanced server farm, a Master Browser can be used.

Without a Master Browser, a client request comes into a server farm and attaches to

some server that then negotiates with other servers where the connection should finalize. If we had a means to point the connection to a 'likely' server to begin with, then the amount of negotiation between servers would be decreased, as would the client connect and overall response time.

With a domain Master Browser, initial connections are passed on to servers that have the least number of attached workstations. Based on previous understandings, this is not a fool-proof way of determining the best suited server, but when used in conjunction with load balancing, the overall efficiency of client distribution is increased by more than 35 percent.

A Master Browser is an inherent part of Microsoft Domain networks, and is by default part of the PDC in the domain. It regulates the available licenses to attached server and resources. In a WinFrame environment, however, the Master Browser should be of "Citrix" flavor, instead of a generic NT server. This is to ensure that the Master Browser is load-balance aware.

If using the PDC as the default Master Browser is not desirable, there are methods to explicitly designate the Master Browser to other servers. The most explicit method requires editing the server's registry, which is rather involved but fully documented in the WinFrame online help documents.

I haven't encountered a reason not to use the PDC as the Master Browser, but that's not to suggest that there isn't a reason not to.

SUMMARY

Load balancing in a WinFrame environment is a good way to leverage the server farm's ability to host the maximum number of sessions while ensuring that no single server attempts to handle too much. It also provides a level of fault tolerance, admittedly

not as transparent or seamless as provided by clustering, but suitable for many applications.

The key to getting the most out of WinFrame's load-balance option is to understand an application's demands on a server, and adjust the load-balance criteria accordingly. The most effective way to monitor a server's specific affected components is to run Performance Monitor from the Administrative Tools program group and chart the memory, disk, and CPU usage while the program in question is being used by one or more attached users. Based on the Performance Monitor results, you will be able to determine the best way to allow the maximum number of users per server in your WinFrame farm. 



NaSPA member Guy C. Yost is the owner of Redstone Consulting, an IT management consulting firm in New York. He has authored several books on networking for Que Publishing, including *Learning NetWare 4.1*, and *NetWare 4.1 SmartScan*, and contributes to *Technical Support* magazine as an author, columnist and technical editor. Guy also develops and conducts seminars on networking with Windows NT, UNIX, NetWare and Internet/intranet technologies across the United States and Canada. He can be reached at (518) 674-5606 or gyost@logical.net.

©1998 Technical Enterprises, Inc. For reprints of this document contact sales@naspa.net.