Contents

Understanding

NetWare 5 is based on a new multiprocessing kernel (MPK) that supports as many as 32 processors per server. MPK combines the outstanding performance of a uniprocessing kernel with the superior scalability of a multiprocessing kernel.

MPK intelligently handles load balancing for the most efficient utilization of multiple processors. For single-processor servers, MPK's enhancements in the area of memory protection provide greater reliability and efficiency.

MPK includes a new application preemption function—Fair Share scheduling—that enables system administrators to prioritize applications running on the server. By assigning shares you can ensure that high-priority and mission-critical applications can run faster with fewer interruptions during heavy network traffic.

MPK (MultiProcessing Kernel)

The kernel is the core of an operating system and provides the most fundamental operating system services. It handles interrupts and the I/O system, manages threads and processes, and allocates and schedules processor resources.

To compare NetWare 5's multiprocessing kernel with intraNetWare's SMP, see "Comparing MPK and SMP" on page 2.

For a summary of multiprocessing advantages, see "Benefiting from Multiprocessors" on page 3.

For an explanation of Fair Share scheduling and MPK's built-in load balancing, see "Controlling Processor Utilization" on page 3.

Comparing MPK and SMP

When multiple processors are detected during NetWare 5 installation, you have to option to simply load Platform Support Module (PSM). The PSM is a hardware abstraction layer that shields software layers above it from the details and complexities of your hardware platform. Loading the PSM is required for MPK to use multiple processors.

NetWare 4.11, in contrast, included two kernels: one for uniprocessors and another for multiprocessors. When multiple processors were detected, NetWare 4.11 gave you the option to load the NetWare SMP.NLM (symmetric multiprocessing) to support additional processors as well as the Platform Support Module to server as a hardward abstraction layer.

NetWare SMP enabled a server to run resource-intensive services, such as large databases, document management software, and multimedia applications on a NetWare 4.11 server. Although SMP provided early access to the potential benefits of multiple processors, it was limited in using more than one processor effectively. NetWare has traditionally used non-preemptive, round-robin scheduling for speed and efficiency. In a NetWare 4.11 SMP environment, a thread—the unit of execution—is scheduled to run in the order that it reaches the ready state. The drawback in this implementation was that an application with several threads of execution could "hog" the processor while a critical application with a single thread waited.

A multithreaded application is necessary to take full advantage of multiple processors . Multithreading allows multiple paths of concurrent and parallel execution through the code path. A multi-threaded application can have threads simultaneously executing on more than one processor.SMP.NLM is not fully multi-threaded.

The NetWare Symmetric MultiProcessing (SMP) technology enables the NetWare 4.11 operating system to run on a multiprocessor server.

NetWare SMP provides:

Increased processing power and better network performance because multithreaded processes are split among the additional processors, leaving addition processing time on processor 0 for non multithreaded processes.

Support for up to 32 processors, depending on the hardware platform.

Support for Advanced Programmable Interrupt Controllers (APICs),which means that if any processor (except 0) fails, the server continues to function and that you can load and unload APIC processors without bringing down the server.

Priority-based Preemptive Scheduling--Fair Share?

Threads

Interrupts

Applications

One benefit that carrie over from SMP is the platform support technology. New PSMs (Platform Support Modules) detect the appropriate pieces of hardware and the correct PSM is loaded.

Unlike intraNetWare, there are no longer separate kernels for single processors and multiprocessors. MPK is both a uniprocessor and multiprocessor kernel in one. Intran

During installation of NetWare 5, MPK detects multiple processors and uses both preset and user settable parameters to tune itself for maximum advantage of the processors you have and the bandwidth they provide.

Benefiting from Multiprocessors

Multiprocessing advantages begin immediately with the installation of a second processor. NetWare can currently recognize up to 32 processors per server. The ideal number of processors for you can only be determined by your experience combined with the developers of the applications you use. Only applications specifically designed and written to use multiple processors can take advantage of multiple processors.

If you only have one processor it is numbered Processor 0. If you have multiple processors, the first processor is still numbered Processor 0 and all additional processors (up to 31) are numbered sequentially totalling a maximum of 32 processors. They are referred to as secondary processors.

Since MPK is both a uniprocessor as well as a multiprocessor kernel, its advances in other areas become very important. Improvements in memory protection and scheduling procedures provide the potential for greater speed and reliability to uniprocessor systems.

Controlling Processor Utilization

Two features control processor utilitzation:

- **Fair Share Scheduling**—allows you to assign a share value to each multithreaded application whether you have a single or a multiprocessor system. In the sense used here, an application is a group of threads that are grouped for the purpose of assigning resources. The share value determines the amount of processor time an application is allowed to use relative to competing applications. The share values you assign to an application are based on your own business and operational priorities.

- **Load Balancing Threshold**—provides the optimal threshold to determine when to address load imbalance by migrating an application's thread to another processor. This threshold has been preset by Novell and, in most cases, provides the most efficient use of each processor. Without this control, constantly updating the processor assignments would result in thrashing.With the threshold set at the optimum, processor thrashing is eliminated. The Load Balancing Threshold applies only to multiprocessor servers.

Adjustment to this threshold in a production environment should be done only with considerable research and consultation with Novell. Experimenting with different threshold levels can be done in non-production test environments.

To adjust these values, see Optimizing Processor Utilization.

Planning

With the introduction of MPK (Multi Processing Kernel) in NetWare 5, efficient multiprocessing has moved from a dream to reality. This doesn’t mean however, that you need to buy a multiprocessing server right away—at least not until you’ve done some research and planning. This section explains some of the issues you should explore and how to determine your needs for multi processing.

If you are already using multi processing servers, determine your actual needs before expanding your existing servers. But even as a uniprocessor kernel, MPK delivers advanced benefits not previously available. To make the best use of MPK you need to recognize its strengths as a uni processor first. Then expand those benefits with all that multi processing affords you.

And finally, learn the theory behind the controls given you with MPK—Virtual Machines with their Share Values and the Load Balancing Threshold. This is all discussed below to help you understand the direct benefit you can realize from MPK.

Determining Needs

The first part of the process is determining whether you currently run any multi processor aware applications. You should also determine if any currently used uni processor applications will be rearchitected for multi processor use. Contact the companies providing these programs to find out if and when they will convert their software to being multi processor aware.

Next, take some time to plan ahead by considering your future needs. Perhaps you only have one multi-processor capable application right now. What is the current performance level of that application? Can you gain enough increase to justify US$25,000 for a new multiprocessor server? If you have a large network installation, perhaps you would need to consider 500 or 5000 servers.

Will your application(s) allow you to implement multi processor technology in only a limited number of servers to begin with? If you run the same application on one multi processor server and 499 uniprocessor servers, will it work? Can you test various applications this way?

For all applications, you need to plan ahead to take advantage of “Share Values” on page 7. For multi processor aware applications, learn the ramifications of adjusting the “Load Balancing Threshold” on page 8.

Share Values

A Virtual Machine (VM) and the Share Value work in conjunction with NetWare's round-robin scheduler. They have been developed to allow you better control of processor scheduling. The share value parameter is available even on uniprocessor servers. Before NetWare 5, all applications received equal treatment. Earlier versions of NetWare allocated processor time based on the number of threads ready for execution within an individual application.

For example, an application with 20 threads ready for execution would receive 10 times more processor time than an application with only 2 threads ready. This was true even if the application with only two threads ready was more critical than the application with 20 threads ready.

With MPK, you can declare NLM programs to be in different applications or groups called Virtual Machines (VMs). Processor time is allocated based on the number of share values assigned to these VMs rather than just on the number of threads ready for execution. Each VM takes enough processor time during each cycle to complete all threads ready for execution during that cycle.

There is one default VM called NetWare Application. All NLM programs are loaded into this VM until you assign them otherwise. By default, every NLM receives a share value of 100 and is given absolute equality. When you determine that an NLM program needs to be given higher or lower priority, create a VM and load the NLM as a member of that VM and assign it a higher or lower number of share values. You can put multiple NLM programs into a single VM. Each NLM program is then treated equally.

After you have studied the following example, you can go to "Adjusting Share Values" on page 17 for step-by-step procedures.

For Example

If you load a database NLM into a separate VM with a share value of 500 and leave GroupWise in the default VM with a share value of 100, each database user is given 5 times the processor utilization as each GroupWise user. That's because assigned processor time is proportional to the number of users and the threads their requests generate.

Therefore the total number of threads ready for execution could increase based on the number of users or connections to the NLMs associated with a VM. This means that no matter how many GroupWise users are using the application at the same time, that each user receives the same amount of processor time for each thread generated by requests.

Go to "Formula 4 Chemical Company" on page 9 to read how others have solved their problems by adjusting the share value for certain applications. (All companies and incidents depicted in these scenarios are fictitious. Any similarities to actual people, places, organizations or incidents is purely coincidental.)

Load Balancing Threshold

As soon as you install a second and subsequent processor in any server, the Load Balancing Threshold becomes active. This threshold is the point at which requests for processor time are moved from the current processor in use to another processor.

The load balancing threshold is a value set above the mean average load (defined as 1.0) of all processors. The default threshold value is 1536 (equating to 1.5) or 50% above the mean average load. It is not recommended but you may set this value to any number above 1.0 that you want. Never set the value to 1.0 (the mean load value) or below.

The load value between the mean and threshold levels provides working room for the automatic threshold balancing algorithms. If the threshold value was based only on the mean average, nearly every thread would constantly move from one processor to another, creating processor thrashing and destroying the system's productivity.

There are additional issues you need to consider. Some NLM programs can be bound to a specific processor by their authors.In such a case, the threads generated by these NLM programs are never moved and are taken out of the mean equation. All of this is done in the background. Through extensive testing, Novell has discovered the most efficient threshold value is generally 1536.

Figure 1
Calculating the Threshold Value (missing for now)

You may find that in certain cases for your company and network, a different number is more efficient. The higher the number, the fewer times threads are reassigned. The lower the number, the more frequently threads are reassigned. But remember the CPU cache.

Every CPU has its own cache in which threads are stored after being retrieved the first time. Retrieving threads from cache can be up to 100 time faster than having to look outside the cache. This implies that you should be very careful about reducing the threshold value. If, however, you find that your secondary processors are being under utilized, you may be able to speed things up even more by lowering the threshold value.

Figure 2
Determining the Load Balancing Threshold (missing for now)

Go to "National Furniture Giant (NFG)" on page 10 to read how others have solved their problems using multiple processors (all companies and incidents depicted in these scenarios are purely fictitiously. Any similarities to actual people, places, organizations or things is purely coincidental.

Go to "Set Load Balancing Threshold" on page 20 for step-by-step procedures.

Scenarios

Formula 4 Chemical Company

The situation: As the IS manager at the Formula 4 Chemical Company, you're responsible for all IS operations. And with this particular company, you're tasked with more than just helping make money, your job may include helping to save lives. Your operation runs 24 hours a day, 7 days a week. If you're operation is down when a call comes in about a chemical spill or some other accident involving ingestion, someone may die.

To assure minimum down time, you run everything in duplicate. You use Hot plug technology, raid drive arrays, replicated trees, independent and online power backup systems, etc. You're covered in case of a failed component. You're actually sleeping well at night again. Ahhh! This is a comfortable feeling.

Then you get a call at 2:00 a.m. Your night shift system administrator, Mike, is frantic. He describes the worst night of his life. You ask what happened. It seems John had an upset stomach when he came to work and it got progressively worse. He didn't want to go home so he went to the nurses office. She had gone into the next office for just a minute, but John didn't want to wait. He saw some pink stuff in the cabinet that looked like an antacid he had gotten from the nurse before. Instead, he got a hold of some trichloroethyloxynitrate.

Just then the nurse came back in but it was too late. John had already downed a couple tablespoonsful. In an instant he was curled on the floor, rolling back and forth, screaming. The nurse heads right to her PC and enters the name into the poison control database to look up the antidote. When nothing happens she calls me. I try it, nothing happens, the system just sat there. She kept yelling at me to hurry up but I couldn't find out what was wrong. I just couldn't get any response. Finally, after about 45 seconds, she got the answer and went to work. I received the answer immediately after.

The nurse said if the system had taken any longer it would have been too late. If John hadn't received the antidote within the next 15 seconds it wouldn't have mattered. He would have been dead.

Finally you get to ask what caused the delay. Mike had no idea. Everything seemed to be working just fine except he couldn't get the normally quick response to his query. You ask Mike if everything is working now. Mike says he did two additional queries after hanging up with the nurse. The first one worked fine, then the second one took just as long as the real query for John's problem.

You tell Mike you'll be right over. All the way there you relive the nightmare again and again. You try to think of what could be causing such a problem. The company has grown a lot over the last couple of months due to an acquisition of another company. Is the system overloaded? No, it's been doing just fine. Is there a bug somewhere in your poison control database? You don't think so. You've not seen this problem before. So what could it be.

Finally you arrive and check on John. He's been taken to the hospital but should be able to return to work in a couple of weeks. You head over to the computer center to try and figure things out. After trying the same things Mike did, you get the same results. You start checking everything you can think of. Nothing seems to be wrong. Everything is running as usual. Then, all of a sudden, your queries start responding quickly again.

The problem: You finally determine that some other application must have been hogging all the resources. As you check to see what was running during the problem period, you eventually trace the problem to the backup software. But why did it cause this problem? Why hadn't you seen this problem before?

Of course, backup was tying up all the processing power for a few seconds or a few minutes at a time. That's just what it's supposed to do. After all, that's why you run backup in the middle of the night, so it can do its job. Everything else gets to keep running, the response time is just a little slower. For the kind of operations running at night, it doesn't matter. At least, not until now.

The solution: You need some way to give database queries a higher priority than backup. In fact, you decide you need to give it a higher priority over everything else that runs. But how? A quick call to Novell support provides the answer.

You get a quick explanation about share values, how they work and how to implement them. You and Mike look over the NLM programs you run on your server and make some initial decisions. The most critical is to give the database a higher priority for right now. You put the database in its own application and set its share values to five time the default. You decide to spend the next few days doing a more detailed investigation to fully implement the use of share values in the most efficient manner.

National Furniture Giant (NFG)

The situation: Hundreds of trucks roll through each of five distribution centers around the country each day. Dozens of trucks arrive from the different furniture and appliance manufactures. Each truck needs to be unloaded, the merchandise checked in and stored on racks. Tens of thousands of pieces of furniture are stored in these five locations so they can be distributed to each store as needed. Nearly 100 NFG trucks head out to different stores each morning.

There is one server with approximately two dozen workstations in each store. These systems handle in-store inventory, local delivery scheduling, customer service and repair, credit applications and general administrative applications, etc. In addition, each night these systems connect with the main application server at their regional distribution center.

Each of the regional distribution centers have one application server for their inventory and distribution scheduling database; GroupWise also runs on this server. A second server collects a complete set of reports for items sold, current inventory and restocking needs. This is completed within one hour of each stores closing. Additional data on sales volume, store overhead, number of credit applications accepted and so on are also sent.

NFC has opened 55 additional stores during the past two years. This has required an average 50% increase in the pieces of furniture handled through each distribution center.

As Jerry, the system administrator, continuously works with his online inventory and distribution systems, he has managed to squeeze the extra work load out of his existing equipment. Last week, however, he was told that another 35 stores will be opened during the coming year and he was charged with making sure his systems would handle the additional load.

The problem: Jerry already knows he can only accommodate about 5% additional growth before his systems are overloaded. He's been planning ahead, but he didn't expect another 35 stores so soon. In the back of his mind, Jerry's been thinking another distribution center would be needed before too much more growth could be handled. This option was specifically denied before he could even ask about it.

Jerry has always had to give strict and accurate accounting for his system and software expenditures, so he begins thinking about what will be approved to help him meet this challenge. A new server plus workstations are known and accepted commodities for each of the new stores. Even the software for them is pre established. But how will he handle the increased demands upon the distribution centers' central systems? He doesn't want to spread his databases over multiple servers. He needs to find a system that can handle the announced growth plus what ever comes during the next three to five years, his expected upgrade cycle. An entirely new kind of system might require all new software. Just the thoughts of the expense, time and inherent problems with such a change made Jerry uneasy. There's got to be a better way.

Three years ago, NetWare 4 with NDS lightened his administrative tasks so much that he hates to give it up. But he knows it just won't handle any more on these single servers. Because of the good relationship established with his Novell reseller, Jerry calls to get some advice on where he can go from here.

The solution: Margaret has been servicing Jerry's account for about four years and visits with Jerry that same afternoon. After listening to Jerry's explanation and asking a few pointed questions of her own, Margaret begins sharing some thoughts to be sure she understands the situation and to make sure she's on track with her thoughts. Together, they make a list of what they have and what needs to be considered and done:

- Verify the versions and upgradeability of each application on both servers and clients
- Verify the current server inventory throughout the company
- No workstations need upgrading
- Any changes made must be able to satisfy the companies future growth for the next three to five years without disrupting current operations
- More processor power will be needed for the database application servers at the distribution centers
- Other servers at the distribution centers can easily handle their daily administrative tasks without any changes
- Servers at individual stores can easily handle their loads without any changes.

Jerry had the results of his inventory within two days. Due to their equipment upgrade about two years ago, all servers and workstations are nearly identical. The findings include:

- Every system has a 166 MHz Pentium processor
- Each workstation has 32 MB of RAM
- Each server has 128 MB of RAM
- All applications are standard name brands and have been upgraded when new versions were available
- The database application is up to date and multi processor ready

This information provided some specifics to govern the discussions that followed. Within a couple of hours, Jerry was satisfied that he had found the probable solution because it was fairly simple to implement, could easily be put in place in a timely and safe manner, met all the conditions dictated above and was well within the general expense guide lines he had received from his boss.

But, as do all of us, Jerry wanted proof. Margaret was talking about an expensive solution (all be it less money than Jerry had anticipated). So, he put Margaret on the spot and asked for the proof up front. She and Jerry agreed to work together on three tasks before Jerry would take a proposal to his management.

Checklist:

- ❑ Margaret's company would provide a four processor server on loan for three months without cost to NFG.

❑ She would work with Jerry's staff to implement his database and a mock environment that mimics his current situation. This would be created within two weeks and would run for two weeks.

❑ When the mock trial proved to work, Margaret would help add the four processor server to the on line system and allow it to be used as a duplicate system along side the existing server for up to two months. It had to do everything the current server and system was already doing. This included handling the currently daily receivables and stocking, nightly connections and order processing with the retail stores, truck loading and delivery schedules. GroupWise would be added after the trial period and when Jerry's proposal was accepted by management.

If this trial functioned as needed for three months, Jerry would then write the full proposal and present it to his management. The proposal would include the purchase of one multi processor server with four processors to be put online along side the existing system for one more month. If it continued without major problems, the multi processor server would become the primary server and the uni processor server would become the backup. Two more months allowed time to fine tune things such as the load balancing threshold.

This plan puts one distribution center on line with the new multi processor server within six months. One additional distribution center would be changed over to a new multi processor server each month for the next four months. By targeting the busiest centers first for the change over, Jerry would be ahead of schedule right from the beginning and complete the process about five months before required.

Just in case the multi processor servers didn't do as expected, Jerry continued to work with Margaret and others on possible alternate solutions. Even though every other likely solution he could find would have been considerably more expensive to implement, Jerry had to at least keep his options open. He looked at other PC network operating systems and proprietary hardware for mid-range systems. All options had to meet the same conditions Margaret had agreed to for the NetWare 5 server upgrade.

One other decision was made no matter which alternate system Jerry decided on. The current servers would be kept as back up systems at each distribution center. They could be temporarily used as emergency servers or workstations in case of a major breakdown. Having this insurance added to Jerry's comfort level for the future.

Well, of course this scenario has a happy ending. That's why we used it. Everything worked just as Margaret promised and Jerry needed. In fact, Jerry was able to move up everything in the schedule by more than a month. The specific parameters he implemented after testing included leaving the default load balancing threshold, putting the database in its own virtual machine (VM) with a share value of 300 and GroupWise in another VM with a share value of only 80. Every distribution center reported complete satisfaction with the changes.

Jerry had one additional benefit in mind as he made this decision. If NFG is fortunate to be able to continue to expand, the multiprocessor servers can accommodate additional processors as needed. Even if some unforeseen applications need to be added to these main servers, Jerry has a way to handle them without great expense for either hardware or inconvenience.

Setting Up

No setup is necessary. Multiprocessing is a function of the kernel itself.The NetWare 5 installation program detects the existence of multiple processors and loads the Platform Support Module.

After installation, you may choose to add additional processors or even to remove existing secondary processors. Begin with instructions for Stopping and Starting Processors and then see Adding, Registering, and Activating Processors.

If you need to optimize uni processor and multi processor usage for special circumstances, see Optimizing Processor Utilization for information on how to change default settings.

Note:

Adjustment to the Load Balancing Threshold in a production environment should be done only with considerable research and consultation with Novell. Experimenting with different threshold levels can be done in non-production test environments.

Optimizing

To improve the speed and performance of your system, you can use MONITOR to determine which threads are utilizing the processors and then take steps to avoid bottlenecks by reassigning shares or by adjusting the load balancing thresholds.

To view MPK components from MONITOR's Available Options > Kernel:

- Applications > NetWare Application (or other application in the list) > Threads
- Processors > Processor Information
- Interrupts > Registered System Interrupts > Service Routines for Interrupt *n*

Optimizing Processor Utilization

With a multiprocessor machine, you can add additional processors and you can set the processor utilization level when threads are migrated.

See the following sections.

- “Determining Which Threads Are Using a Processor” on page 16
- “Adjusting Share Values” on page 17
- “Adding, Registering, and Activating Processors” on page 18
- “Set Load Balancing Threshold” on page 20

Determining Which Threads Are Using a Processor

One possible cause of network problems is a single thread that monopolizes a processor. Use this procedure to determine which threads are monopolizing a CPU.

Note:

For information about optimizing processor utilization in a multiprocessing server, see “Set Load Balancing Threshold” on page 20.

Procedure:

1. **From MONITOR’s Available Options, select Kernel > Processors.**
2. **Press Tab to expand or activate the upper window that displays Processor Information.**

   Note the statistics for each processor.
3. **Press Tab to return to the Kernel Options menu.**
4. **Select Applications to display a list of applications running on the server.**

   In this case, an application is defined by one or more loaded modules and consists of a group of threads that are treated as a unit when processing resources are assigned. Modules that do not create their own application are assigned to the NetWare Application.
5. **Select an application to display a list of its threads.**
6. **Highlight a thread to display its information in the upper window.**

   Note the setting for Scheduling Priority.
7. **Press Escape to return to the Kernel Options menu.**
8. **Select Interrupts to display a list of registered system interrupts.**
9. **Highlight an interrupt to display its information in the upper window.**

   The information includes per-processor statistics.
10. **Press Tab to expand the information window and press Tab again to return to the list of system interrupts.**
11. **Select an interrupt to display a list of its associated Interrupt Handlers (service routines).**

    The Interrupt Information window provides per-processor statistics.
12. **Compare statistics for threads to determine which threads are taking the most processor time. In a multiprocessor server, compare processor loads.**
13. **(Optional) To reduce processor utilization temporarily, unload non-critical NLM programs that monopolize processors.**

Adjusting Share Values

Novell’s multi-processor kernel (MPK) allows you to assign priority to applications. One or more loaded modules can define an application. All NLM programs within a single application are treated equally. Remember that in this context an application is a group of threads.

An application is given a default share value of 100. You can adjust the share value of each application with the following procedure.

Procedure:

1. **From MONITOR’s Available Options, select Kernel > Applications.**

   A list of defined applications is displayed

2. **Highlight the application for which you want to adjust the share value.**

   The current share value is displayed in the upper window along with other information. The default is 100.

3. **To adjust the value, press F3, type in the new value and press Enter.**

   A value lower than 100 gives all NLM programs in this application a lower priority than the NLM programs assigned to the NetWare Application.

   A value higher than 100 gives all NLM programs in this application a higher priority than the NLM programs assigned to the NetWare Application.

4. **Press Escape to return to the Kernel menu.**

For an explanation of Applications and Share Values, see “Share Values” on page 7.

[Remove this, but come up with a link? “For a scenario describing the use of shares, see “]

Adding, Registering, and Activating Processors

Adding or removing physical processors can be done only when the server is powered off. When you boot the server, all processors are automatically registered and activated.

Although Processor 0 cannot be stopped, each secondary processor can be started or stopped at any time while the server is running. You should first look at a display of all processors and note which NLM programs are running on each processor.

Hint:

Because Processor 0 cannot be taken offline while the server is running, server console commands for adding, registering, and activating processors affect only secondary processors.

Displaying Processor Status

To determine which processors are currently online, enter at the console prompt:

```
display processors
```

Displaying Generic Processor Information

To determine processor speed and other generic information, enter at the console prompt:

```
cpucheck
```

Stopping Processors

To take a specific secondary processor offline, enter at the console prompt:

```
stop processor #
```

To take multiple processors offline, but not all secondary processors, enter at the console prompt:

```
stop processor # # #
```

To take all secondary processors offline, enter at the console prompt:

```
stop processors
```

Starting Processors

To bring a specific secondary processor online, enter at the console prompt:

```
start processor #
```

To bring multiple processors online, but not all secondary processors, enter at the console prompt:

```
start processor # # #
```

To bring all secondary processors online, enter at the console prompt:

```
start processors
```

Set Load Balancing Threshold

The threshold value for the processing load specifies when it is worth the cost in processor time to address processor load imbalance by migrating threads. Use this procedure to set the load balancing threshhold.

Procedure:

1. **In MONITOR’s Available Options, select Server Parameters > Miscellaneous.**

   The miscellaneous parameters appear in the upper window. The scroll thumb on the right of the window indicates that the window is scrollable.

2. **Scroll the window until you highlight the System Threshold field.**

   The pop-up window displays the current setting and the range of valid values.

3. **To change the threshold setting, press Enter and type a new value in the field.**

4. **To exit MONITOR, press Alt-F10.**