

Windows
& .NET MAGAZINE

eBooks

Windows
& .NET MAGAZINE

TECHNICAL REFERENCE

Winternals

Windows Performance Tuning

Curt Aubley
Jordan Ayala
Cris Banson
Pierre Bijaoui
Sean Daily
Kalen Delaney
Troy Landry
Darren Mar-Elia
Tony Redmond
Joe Rudich
Tao Zhou

Windows & .NET Magazine Technical Reference

Windows Performance Tuning

*By Curt Aubley, Jordan Ayala, Cris Banson, Pierre Bijaoui,
Sean Daily, Kalen Delaney, Troy Landry, Darren Mar-Elia,
Tony Redmond, Joe Rudich, and Tao Zhou*

Windows
& .NET MAGAZINE

A Division of Penton Media

Windows

& .NET MAGAZINE

Copyright 2003

Windows & .NET Magazine

All rights reserved. No part of this book may be reproduced in any form by an electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

It is the reader's responsibility to ensure procedures and techniques used from this book are accurate and appropriate for the user's installation. No warranty is implied or expressed.

ISBN 1-58304-506-6

About the Authors

Curt Aubley (caubley@oao.com) is chief technology officer for OAO and an MCSE. He is the author of *Tuning and Sizing Windows 2000 for Maximum Performance* (Prentice Hall).

Jordan Ayala (jordanayala@hotmail.com) is a contributing author for *Windows & .NET Magazine*. He is an MCP and an independent consultant and writer in Seattle.

Cris Banson (cbanson@itref.com) is a freelance writer who specializes in enterprise networking, storage, and global connectivity. He is an MCSE and a CNE.

Pierre Bijaoui (pierre.bijaoui@compaq.com) is a senior solution architect for Compaq's Applied Microsoft Technologies Group in Sophia-Antipolis, France.

Sean Daily (sdaily@winnetmag.com) is a senior contributing editor for *Windows & .NET Magazine* and the CEO of Realtimepublishers.com. His most recent books are *The Definitive Guide* series (Realtimepublishers.com).

Kalen Delaney (<http://www.insidesqlserver.com>) is an independent SQL Server trainer and consultant, an MCT, and an MCSE. She is the author or coauthor of several books about SQL Server, including *Inside Microsoft SQL Server 2000* (Microsoft Press).

Troy Landry (tlandry@oao.com) is a senior system architect for OAO and an MCSE. He works extensively in system and network management solutions.

Darren Mar-Elia (dmarelia@winnetmag.com) is a contributing editor for *Windows & .NET Magazine* and senior product architect for Windows at Quest Software. His most recent book is *The Tips and Tricks Guide to Windows 2000 Group Policy* (Realtimepublishers.com).

Tony Redmond (exchguru@winnetmag.com) is a contributing editor for *Windows & .NET Magazine*, senior technical editor for *Exchange & Outlook Administrator*, vice president and chief technology officer for Compaq Global Services, and author of *Microsoft Exchange Server 2000: Planning, Design, and Implementation* (Digital Press).

Joe Rudich (joe@rudich.com) is a network administrator with the St. Paul Companies in St. Paul, Minnesota.

Tao Zhou (tao@winnetmag.com) is a contributing editor for *Windows & .NET Magazine* and an Internet services engineer for a New Jersey-based networking company. He is an MCSE and a Master CNE and holds a master's degree in computer science.



Table of Contents

Introduction	ix
Part 1: Windows Server and Client Performance	1
Chapter 1: Windows 2000 Performance Tools	1
Performance Monitor vs. Task Manager	1
Performance Monitor	1
Performance Monitor in the Real World	2
Sidebar: Activating All Performance Counters	3
General Performance Monitoring	2
Long-Term Performance Analysis	3
Sampling Rates	4
Which Process Is the Bottleneck?	6
Task Manager	7
Use Your Imagination	8
Chapter 2: NT Performance Tuning	9
Memory	11
Processor	13
Disk	15
Network Interface	16
Understand Your Environment	17
Chapter 3: Tuning NT Server Disk Subsystems	19
Common RAID Characteristics	19
Sidebar: What to Look for When Selecting a Scalable RAID Array	21
Fault Tolerance of the RAID Array	21
RAID Manageability Tools	21
Support for Dynamic Growth of the RAID Array	22
Support for Additional Disks	22
I/O Technology in Place: Follow Your Data	22
Grouping Similar Disk Activities	20
Sidebar: Why Is RAID 5 Slow on Writes?	24
Disk Workload Characteristics	23
Load Balancing Your Disks	25
Tuning the Allocation Unit Size	26

Your Results May Vary	27
Know Your Environment	28
Chapter 4: Take It to the Limit	29
Rely on Vendors' Recommendations	29
Simulate the Server's Actual Use	29
Use Software to Simulate a Heavy Workload	29
What Is Response Probe?	29
Creating the Script Files	30
Configuring the Server	33
Using Response Probe	33
No Longer in the Dark	34
Chapter 5: Optimizing Windows Services	35
What Is a Service?	35
Service Administration	35
Taking Action	40
An Application or Service Has a Problem	40
You Need to Cause Configuration Changes to Take Effect	40
You Need to Troubleshoot	40
You Want to Change the Way the System Behaves	41
You Need to Shut Down Services to Install Applications or System Updates	41
You Need to Install New Services or Applications	41
Installing and Uninstalling Services	41
Evaluating and Tuning Services	42
What Installs Which Services?	46
What Can You Afford to Lose?	49
Security Tune-Up	51
Tune Up or Tune Out	52
Chapter 6: Measuring and Managing Windows NT Workstation 4.0	
Application Performance	53
What's Important in Performance Monitor	53
Monitoring for Memory Leaks	54
Committed Bytes and the Pagefile	55
Processor Utilization	56
Resource Kit Utilities for Performance Management	57
Sidebar: Performance Management Utilities	57
Part 2: Networking Performance	59
Chapter 7: Optimize GPO-Processing Performance	59
GPO-Processing Basics	59
Performance Boosters	60
Slow-Link Detection	61

GPO Versioning	61
Asynchronous Processing	62
Sidebar: Group Policy Logging	63
Greater Control	64
Disable Unused Settings	64
Set a Maximum Wait Time	64
Design Matters	64
Limit GPOs	65
Limit Security Groups	65
Limit Cross-Domain Links	65
GPOs: Complex but Powerful	66
Chapter 8: Web Server Load Balancers	67
What Is a Load Balancer?	67
Sidebar: Microsoft's Load-Balancing Services	70
Server Monitoring	71
Server Selection	72
Traffic Redirection	73
MAT	73
NAT	74
TCP Gateway	74
Global Site Selection and Traffic Redirection	74
Load Balancer Redundancy	76
Balance Your Environment	76
Chapter 9: Monitoring Win2K Web Site Performance and Availability	77
Customer Perspective: Site Monitoring Methodology	77
Sidebar: Commercial Tools for System and Network Management	79
Implementing and Testing the Monitoring Modules	78
Sidebar: Monitoring Modules and Windows 2000 Network Load Balancing Service	83
System Performance and Trend Analysis	85
Leveraging Win2K Troubleshooting Tools	86
Proactive System Management	87
Act Now	89
Part 3: .NET Server Performance	91
Chapter 10: Exchange 2000 Performance Planning	91
Partitioning the IS	91
Storage Performance Basics	94
Storage Configuration	97
Example Configuration	97
Expanding Boundaries	100

Chapter 11: 3 Basics of Exchange Server Performance	101
Fundamental No. 1: Hardware	101
Sidebar: Making Sense of Benchmarks	102
CPU Statistics	101
Memory and Cache	102
Maximum Disk Performance	104
Fundamental No. 2: Design	106
Sidebar: Exchange 2000 and AD	106
Fundamental No. 3: Operations	107
Stay in the Know	109
Chapter 12: The 90:10 Rule for SQL Server Performance	111
Sidebar: Knowing is 9/10 the Battle	112
Tip 1: Don't skimp on hardware	112
Tip 2: Don't overconfigure	112
Tip 3: Take time for design	112
Tip 4: Create useful indexes	112
Tip 5: Use SQL effectively	113
Tip 6: Learn T-SQL tricks	113
Tip 7: Understand locking	113
Tip 8: Minimize recompilations	113
Tip 9: Program applications intelligently	113
Tip 10: Stay in touch	114
Tip 1: Don't skimp on hardware	111
Tip 2: Don't overconfigure	114
Tip 3: Take time for design	114
Tip 4: Create useful indexes	115
Tip 5: Use SQL effectively	115
Tip 6: Learn T-SQL tricks	116
Tip 7: Understand locking	116
Tip 8: Minimize recompilations	117
Tip 9: Program applications intelligently	117
Tip 10: Stay in touch	118
Tips of Icebergs	118
Chapter 13: Performance FAQs	119
Which Performance Monitor counters do I need to observe to evaluate my Windows NT server's performance?	119
What causes my Windows NT server's available memory to drop to 0MB during file transfers?	120
How can I ensure that both processors in a 2-way system are fully utilized?	120
How do I use Windows 2000's Extensible Storage Engine (ESE) counters?	122
How do I improve performance running SQL Server 7.0 and Exchange Server 5.5 SP2?	122

Introduction

Good network administrators know that their job isn't finished after they successfully install Windows. Instead, their real work is just beginning. As is true for all computing systems, a Windows network's performance will suffer if you don't tune it. As users run more applications and clutter up the file system, a once snappy system will gradually lose its zip. Fortunately, you can take several steps to ensure that your system continues to perform optimally.

This book will guide you through the basics of Windows performance tuning and will provide tips for tuning Windows products such as Exchange Server and SQL Server. Be forewarned that no magic Windows performance tuning formula exists. Tuning your system will depend on many factors, including your system's hardware attributes and the tasks you're performing. For example, the settings you use to tune a SQL Server system are different from the settings you might apply to a specialized Windows Active Directory (AD) domain controller (DC). Likewise, optimizing an I/O-bound system is different from optimizing a CPU-constrained system.

Part 1 is a guide for tuning Windows server and client performance. In Chapter 1 you'll learn about built-in Windows Server performance tools such as System Monitor and Task Manager. Chapter 2 guides you through the Windows NT performance tuning process. Although the examples in this chapter are for NT, the same concepts apply to Windows 2000 Server. Chapter 3 dives into the details of tuning the disk subsystem. This topic is important because the disk is typically the slowest performance component; optimizing the disk can give you the biggest performance gains. Chapter 4 shows how to use the Response Probe tool to stress test your Windows server systems. Chapter 5 explains various Windows services' purposes so that you can safely disable unused services. In Chapter 6 you'll learn how to use Performance Monitor under NT to check common system performance counters. Again, although some of the screens might be different, the concepts in this chapter also apply to Win2K.

Part 2 focuses on network performance. In Chapter 7 you'll see how to optimize Win2K Group Policy Objects (GPOs) to speed up domain logon performance. Chapter 8 presents Windows network load balancing and shows you how to load balance your Web servers to achieve better Web site scalability. Chapter 9 shows you how to monitor Web sites for availability and perform trend analysis for proactive tuning.

In Part 3 you'll learn how to tune some Windows family members. Chapter 10 focuses on performance planning for Exchange 2000 Server. Chapter 11 explains the 3 fundamental components of Exchange server performance. Chapter 12 gives you 10 quick rules to apply to your SQL Server systems to ensure they're running well. The book concludes with some common Windows performance FAQs.

This book provides a guide for you to keep your Windows systems up and running with tip-top performance.

Part 1: Windows Server and Client Performance

Chapter 1

Windows 2000 Performance Tools

—by *Curt Aubley*

Whether you're sleuthing server-performance problems, determining how to tune your system, or sizing a server for new applications, the first step is to learn how to leverage your OS's native performance tools. As Windows NT 4.0 became more popular and IT professionals creatively used it for more complex and larger solutions, the OS's native performance tools quickly began to show their age. Although NT 4.0's core performance tools are still available in Windows 2000, Microsoft has enhanced them to keep up with today's IT professionals. Win2K's primary performance tools include System Performance Monitor and Windows Task Manager. If you're familiar with NT 4.0's Performance Monitor and Task Manager, you'll quickly master Win2K's enhanced versions and enjoy taking advantage of their new features.

Performance Monitor vs. Task Manager

Which tool is best for you? Most likely, you'll use both Performance Monitor and Task Manager depending on your mission. Performance Monitor is the tool of choice for obtaining detailed information, logging data for extended analysis, and collecting performance information based on performance events that occur within your system. Task Manager provides a quick look into what is occurring on your system but doesn't provide a mechanism for logging. However, Task Manager lets you manage applications (i.e., processes) that might be adversely affecting your system.

Performance Monitor

Performance Monitor is a Microsoft Management Console (MMC) snap-in. To invoke this tool, select Start, Programs, Administrative Tools, Performance. Alternatively, you can invoke Performance Monitor by selecting Start, Run, inputting Performance Monitor in the Open text box, then pressing Enter. Win2K's Performance Monitor provides the following features to monitor and analyze your server's performance.

- Realtime performance monitoring in chart, reporting, or histogram mode lets you highlight a counter on the Performance Monitor screen and press Ctrl+H, which highlights the current counter selection on your screen. After you perform this action, as you scroll through the counters, Performance Monitor highlights in the associated graph each counter as you select it. When you're displaying multiple counters on the GUI, this feature helps denote which counter is doing what. (The Backspace key doesn't provide this functionality as it did in NT 4.0.)

2 Windows Performance Tuning

- Trace logs provide an advanced mechanism to analyze your system. Third-party tools usually leverage this feature.
- Counter logs let you log performance data at a designated interval for local or remote Win2K systems.

In addition to these monitoring tools, Performance Monitor provides enhanced functionality: Alerts let you generate an action (i.e., run a command or script) based on the counter value thresholds you set in Performance Monitor. In addition, all your settings move with you from one reporting mode to another reporting mode. When you start Performance Monitor, the tool recalls your last settings. Thus, you don't have to save your default settings to a .pwm file and recall them to begin analyzing your system. These settings are system based, so the next person who logs in will see the view that you left. The new tool offers more flexibility in how you store the data that Performance Monitor generates (e.g., you can store data as HTML, binary, .csv, .tsv, and binary circular) than previous versions offered. You can start and stop performance logging based on a date/time group. You can automatically start another copy of the tools based on Performance Monitor events that you configure on your system. Finally, the new tool has a friendlier mechanism to simultaneously collect performance data from multiple servers.

Although NT 4.0 provides some of this functionality (if you install tools from the *Microsoft Windows NT Server 4.0 Resource Kit*), Win2K provides these features in an integrated, friendlier tool that saves you the extra step of loading additional resource kit tools. In addition, Win2K's Performance Monitor can't read performance logs that you generate with NT's Performance Monitor.

Performance Monitor in the Real World

To find details about the basic mechanics of using Win2K's Performance Monitor, click About Performance Monitor in the tool's Help menu. This file provides helpful information and useful directions.

The following scenarios show you how to leverage Performance Monitor's capabilities. To take full advantage of Performance Monitor's functionality, you must activate all your system's performance counters. For information about how to activate these counters in Win2K, see the sidebar "Activating All Performance Counters."

General Performance Monitoring

When you start Performance Monitor, the tool presents you with the default Performance window, which Figure 1 shows. To add any combination of counters to the right display pane, click the addition sign (+) button in the toolbar at the top of the right display pane. Table 1 outlines the minimum counters you should monitor for general performance monitoring. When you're examining specific resources, include the appropriate counters for analyzing that area.

In the Performance window, you can quickly change between chart, report, or histogram views by selecting the appropriate icon below the Performance Monitor's menu bar. Figure 1 shows an example of the report view. You can view the performance of a remote server by clicking the + button, selecting the *Select counters from computer* option, and entering the remote computer's name using the Universal Naming Convention (UNC) format. (Performance Monitor enters the name of the local computer by default.) You must have administrative rights on the remote system you want to monitor.

Activating All Performance Counters

—by *Curt Aubley*

By default, Windows 2000 doesn't activate two of the core performance counters—network and logical disk monitoring. If you don't activate these counters, half of your performance tuning and sizing puzzle will be missing, which makes analyzing your system's performance extra challenging. If you're concerned about a performance-related problem, you need all the help you can get!

To activate Win2K Performance Monitor's network counters, install SNMP and Network Monitor services by clicking Add Networking Components in the Control Panel Network and Dial-Up Services applet. Next, select Management and Monitoring Tools.

By default, Win2K starts the physical hard disk counters. You can use the Diskperf command at a command prompt to control which disk counters are on or off. For more information about the Diskperf command options, type

```
diskperf -?
```

at the command prompt. If you want to activate both the logical and physical hard disk counters, run

```
diskperf -y
```

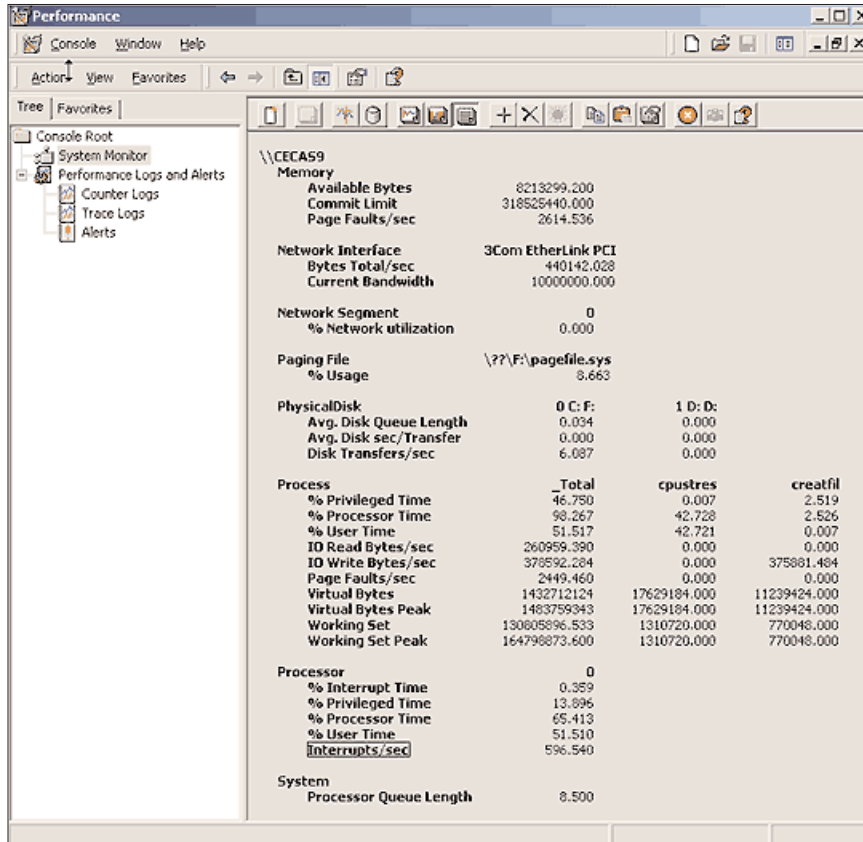
from the command line. You must reboot your system to activate these counters. In addition, you can use the Diskperf commands to start disk counters on remote systems, if you have the proper administrative privileges.

Long-Term Performance Analysis

What if you want to collect performance information over time to develop a baseline? With NT 4.0, your biggest hurdle is the physical size that the performance logs might grow to. To work around this limitation, Win2K's Performance Monitor lets you schedule log collection by time or date. This enhancement lets you isolate the collection of data to times of interest, thus lowering the amount of data Performance Monitor collects. To set a schedule, expand the Performance Logs and Alerts object in the left pane of the Performance window, right-click a log, and select the Schedule tab. On the Schedule tab, you can configure start and stop times. Collecting performance data during typical operations (i.e., from 8:00 A.M. to 6:00 P.M.) is common.

Depending on your environment, you might want to collect data for several weeks at a time for trend analysis. To avoid having to perform maintenance on these files as they grow, select Counter Logs, right-click the file you want to manage, select Properties, click the Log Files tab, select Binary Circular File from the *Log file type* drop-down list, and input a limit in the *Limit of* text box, as Figure 2 shows. Leveraging this performance-collection strategy lets you limit the amount of disk space a performance file uses. If you match the sampling rate to the amount of disk you want to use for performance collection, you can monitor and access several weeks worth of performance data without worrying about performance log size maintenance.

Figure 1
Performance Monitor's default Performance window



Sampling Rates

How often do you need to sample your system for performance data? The answer depends on your goals. If you sample more often than every 5 seconds, you place a slightly higher load on your system (i.e., 1 to 3 percent) and your performance log files require more disk space than if you sample at a usual rate (i.e., less often than every 5 seconds). If you don't sample often enough, you risk not monitoring the system when it experiences a problem.

Win2K provides a much broader range of objects and counters than previously available. If you collect all possible performance data on a system with one disk and one network connection, each sample you collect requires more than 200Kb per sample. Most administrators don't need to monitor every possible performance object and its associated counters. If you collect performance data from the counters that Table 1 lists, each sample consumes approximately 2Kb. Using this information as a baseline, Table 2 provides general guidelines about performance collection rates.

Table 1
Key Performance Metrics to Monitor

Object	Counter	Reason to Monitor
PhysicalDisk and LogicalDisk	Disk Transfers/sec (all instances)	On average, a modern SCSI hard disk can support about 80 to 100 transfers/sec before its response time erodes past an acceptable limit.
	Avg. Disk sec/Transfer (all instances)	These counters measure the time required to complete a read or write transaction. Developing your performance baseline and comparing it with this value shows whether your disk subsystem is running faster or slower than usual.
	Avg. Disk Queue Length (all instances)	If these counters' values are greater than 2 on one drive, you might have a disk-bottleneck problem. For RAID arrays, if the LogicalDisk Avg. Disk Queue Length is greater than twice the number of disks in the array, you have a disk bottleneck.
Memory	Pages/sec	If this value is high (i.e., a high value for Pages/sec is relative to your system) for consistent periods of time (i.e., longer than 5 minutes) and the physical disk where your pagefile resides is experiencing a high workload, you have a memory bottleneck. On a lightly loaded server, a Pages/sec value of 20 is high. On a workstation, a value of 4 might be high. Baseline your environment, and watch closely for dramatic increase in this counter; increased disk activity on the disk that contains your pagefile, the pagefile usage size, and low available memory bytes.
	Available Bytes	This counter shows the amount of RAM still available. You want your system to use all its RAM, but if this counter consistently drops below 4MB, you need more RAM.
Paging File	% Usage (all instances)	These values are helpful in assessing whether you have a memory problem. If Pages/sec increases and the pagefile grows, your system is running low on memory.
Processor	% Processor Time	This counter tracks CPU usage. If this value is consistently greater than 90 percent and the system work queue is greater than 2 over time, you have a CPU bottleneck.
System	Processor Queue Length	One queue exists for processor time even on systems with multiple CPUs. This counter measures the number of threads in the queue that are ready for execution. If this value is greater than 2 for a single-CPU system (or twice the number of CPUs in a multi-CPU system) and the processor usage is greater than 90 percent, you probably have a CPU bottleneck.
Network Interface	Bytes Total/sec	This counter lets you isolate performance-related network problems. If this value is greater than 50 percent of its network medium, a network bottleneck is forming.

Figure 2
Collecting data for several weeks at a time

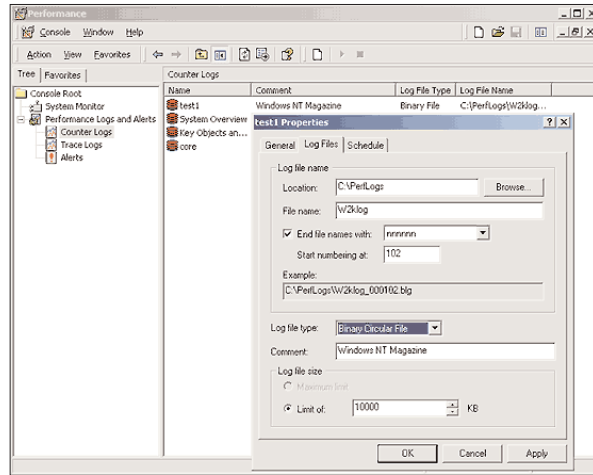


Table 2
Sample Performance Rate Guidelines

Goal	Sampling Rate	Counters	Disk Space Required Per Hour (Kb)
Detailed Troubleshooting	Sample once per second	All Possible Key Counters	720,000 7200
Short-term Analysis	Sample once per 5 seconds	All Possible Key Counters	144,000 1440
Long-term Analysis	Sample once every 10 minutes	All Possible Key Counters	1200 12

Which Process Is the Bottleneck?

Has a customer complained about poor system performance, but when you investigated everything looked fine? Performance Monitor's alert feature comes to the rescue in this type of situation. First, monitor using the counters that Table 1 lists and set performance thresholds on each counter. This setup will provide you with your system's general performance baseline, but you'll need more data to determine which application or process is swamping your system. To obtain this information, use Performance Monitor's alert feature to start any action based on an event you define (e.g., when your counters reach their maximum performance thresholds).

For this example, set an alert to start a copy of the Performance Monitor counter logs when CPU usage exceeds 98 percent. (Occasional peaks in CPU usage might trigger this alert even when a problem doesn't exist. You can use third-party tools to start additional performance collection based on more advanced logical sequences—e.g., when CPU usage exceeds 90 percent for 5 minutes, start additional performance data collection.) To configure this alert, start Performance Monitor, expand Performance Logs and Alerts, and select Alerts. Right-click in the right pane, and select

New, Create New Alert Settings and a name. Add the counters you want to monitor and their threshold for triggering an action; select the Action tab, the *Start performance log* option, a counter log to start, and the Schedule tab; and fill in the times you want to run the monitor. Use a counter log that collects data from at least the counters that Table 1 lists and all the counters and instances under the Process object.

With this setup, Performance Monitor will alert you when your system has a performance problem, and the software will provide you with quantifiable and empirical data that illustrates which process is causing the problem. (Performance Monitor will provide this information in the detailed counter logs that the tool started only after your system reached a certain threshold.)

Performance Monitor's alert feature is flexible. You can tell the alert function to start any script or application. You can have the system send you an email message or start a batch file that pings (i.e., ping.exe), then trace routes (i.e., tracert.exe) the network path to a distant system with which you want to interact. In this manner, you can measure the network response time to determine whether your network has problems.

Task Manager

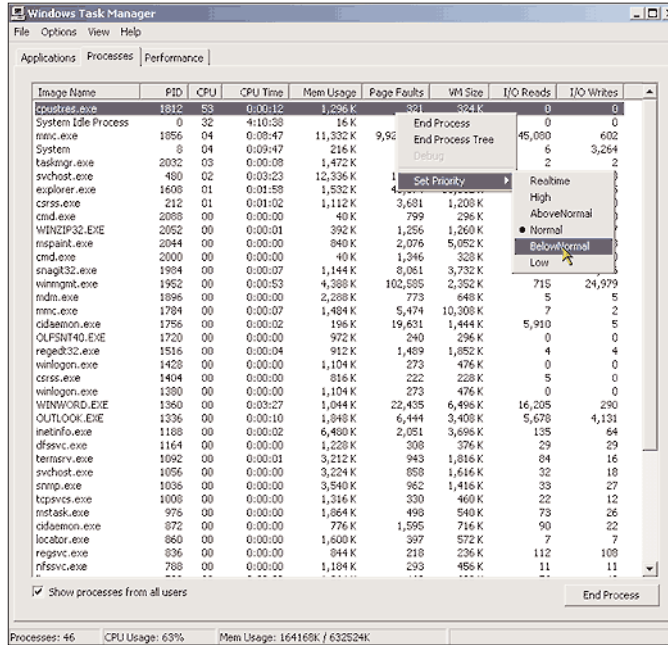
Performance Monitor helps you track problems over time, but what can you do about problem processes in realtime? Task Manager provides mechanisms to monitor in realtime and resolve performance problems. For example, say you have a hunch that cpustres.exe is your system's CPU hog. To activate Task Manager, press Ctrl+Alt+Del and click Task Manager. Alternatively, you can run taskmgr.exe from the command line. After you start this tool, you can view numerous columns of performance data on the Processes tab. The amount of data available on Win2K's Task Manager Processes tab is much greater than on NT 4.0's Task Manager Processes tab—particularly finer grain I/O information is available on a per-process basis (e.g., I/O reads, I/O writes). Within the Processes view, you can quickly determine what amount of CPU, memory, and disk resources each process is consuming. The Applications tab lets you see which processes or applications are not responding.

To find out whether cpustres.exe is your system's CPU hog, select the Processes image name column to place the process list in alphabetical order. This action simplifies finding cpustres.exe. After you find the filename, highlight it by clicking it, then right-click it. Task Manager presents you with several system control options, which Table 3 defines. You can lower cpustres.exe's priority by selecting Set Priority, BelowNormal, as Figure 3 illustrates.

Table 3
Task Manager System Control Options

Option	Purpose
End Process	If the process isn't crucial, you can select this option and stop the process. This action might let another critical process get CPU time and operate properly.
End Process Tree	Some processes have multiple threads that might act independently. You can select this option to bring your system under control.
Set Priority	You can select this option if you don't want to kill the process, but you want it to have less CPU time. Lowering a process' priority level lowers its access to CPU cycles.
Set Affinity	This option is only available on multi-CPU systems. You can select this option to bind an application to a specific CPU or set of CPUs to manage CPU usage. For example, if you have four CPUs and one application is hogging all four CPUs, you can restrict the application to only one CPU so that other applications can use the freed resources.

Figure 3
Lowering a process's priority



In the unlikely event that an application has gone astray and you must terminate it, some applications won't terminate when you select this Task Manager option—even if you have administrator privileges. In this situation, you can use the *Microsoft Windows 2000 Resource Kit* kill.exe /process ID command to terminate the application. You can add the process ID column that corresponds to the application you want to kill to Task Manager. However, this command is powerful and can crash your system.

Use Your Imagination

You can use the primary Win2K performance monitoring and tuning tools to manage the performance of systems in your enterprise. The new functionality these enhanced tools provide lets you be more proactive in tuning your enterprise's systems. You now know how to take advantage of new features such as Performance Monitor alerts and Task Manager information resources. However, don't limit your monitoring and tuning to these features—be creative. With a little experimenting, you'll be surprised at how helpful these enhanced tools can be.

Chapter 2

NT Performance Tuning

—by *Cris Banson*

When you think of a computer system's performance, imagine a chain: The slowest component (or weakest link) affects the performance of the overall system. This weak link in the performance chain is also called a bottleneck. The best indicator that a bottleneck exists is the end user's perception of a lag in a system's or application's response time. To tune a system's performance, you need to determine where—CPU, memory, disk, network, applications, clients, or Windows NT resources—a bottleneck exists. If you add resources to an area that isn't choking your system's performance, your efforts are in vain.

You can use NT Server's native tools (or those of third-party vendors) to optimize the performance of your system and identify potential bottlenecks. NT Server's primary performance tools are Task Manager, which Figure 1 shows, and Performance Monitor, which Figure 2 shows. Task Manager can give you a quick look at what's happening in your system. Although it doesn't provide a logging mechanism, Task Manager displays specific information about your system's programs and processes. Task Manager also lets you manage the processes that might be adversely affecting your system. You can use Performance Monitor to obtain more detailed performance information (in the form of charts, alerts, and reports that specify both current activity and ongoing logging) based on system events. The *Microsoft Windows NT Server 4.0 Resource Kit* also contains tools that you can use for troubleshooting. (For a sampling of NT performance-monitoring tools, see Table 1.)

Figure 1
Task Manager

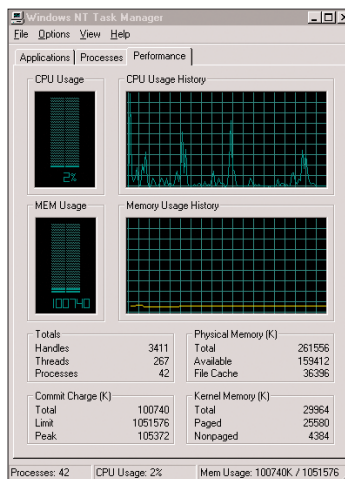


Figure 2
Performance Monitor

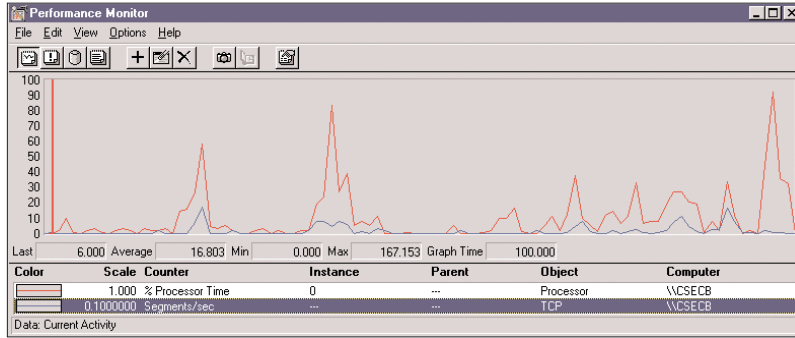


Table 1
A Sampling of NT Performance-Monitoring Tools

Tool	Source	Function
Data Logging Service	Microsoft Windows NT Server 4.0 Resource Kit Supplement 4	Performs the same function as the Performance Monitor Alert and Logging facility and is useful for remotely administering log files for many computers
Network Monitor	NT	Lets you view network traffic on a specific server
Page Fault Monitor	Supplement 4	Lets you monitor page faults that occur as you run an application
Pentium Counters	Supplement 4	Let you monitor the inner workings of Pentium and Pentium Pro chips
Performance Data Log Service	Supplement 4	Logs data from performance counters to a tab-separated-value or Comma Separated Values (CSV) file
Performance Monitor	NT	Lets you perform short-term and long-term data collection and analysis
Process Explode	Supplement 4	Provides accurate and detailed information about the system's processes, threads, and memory
Process Monitor	Supplement 4	Displays process statistics in text format in a command window
Process Viewer	Supplement 4	Displays information about processes on local and remote computers and is particularly useful for investigating process memory use
Quick Slice	Supplement 4	Provides a graphical view of CPU utilization by process
Response Probe	Supplement 4	Lets you design a simulated workload
SC Utility	Supplement 4	Lets you view a command-line interface for the service controller, which displays a specific computer's configuration
Task Manager	NT	Lets you monitor, start, and stop your computer's active applications and processes

Before you start performance tuning, you must understand your system. You should know what server hardware you have, how NT operates, what applications you're running, who uses the system, what kind of workload the system handles, and how your system fits into the network infrastructure. You also need to establish a performance baseline that tells you how your system uses its resources during periods of typical activity. (You can use Performance Monitor to establish your baseline.) Until you know how your system performs over time, you won't be able to recognize slowdowns or improvements in your NT server's performance. Include as many objects in your baseline measurements as possible (e.g., memory, processor, system, paging file, logical disk, physical disk, server, cache, network interface). At a minimum, include all four major resources (i.e., memory, processor, disk, and network interface) when taking a server's baseline measurements—regardless of server function (e.g., file server, print server, application server, domain server).

Because all four of a server's major resources are interrelated, locating a bottleneck can be difficult. Resolving one problem can cause another. When possible, make one change at a time, then compare your results with your baseline to determine whether the change was helpful. If you make several changes before performing a comparison, you won't know precisely what works and what doesn't work. Always test your new configuration, then retest it to be sure changes haven't adversely affected your server. Additionally, always document your processes and the effects of your modifications.

Memory

Insufficient memory is a common cause of bottlenecks in NT Server. A memory deficiency can disguise itself as other problems, such as an overloaded CPU or slow disk I/O. The best first indicator of a memory bottleneck is a sustained high rate of hard page faults (e.g., more than five per second). Hard page faults occur when a program can't find the data it needs in physical memory and therefore must retrieve the data from disk. You can use Performance Monitor to determine whether your system is suffering from a RAM shortage. The following counters are valuable for viewing the status of a system's memory:

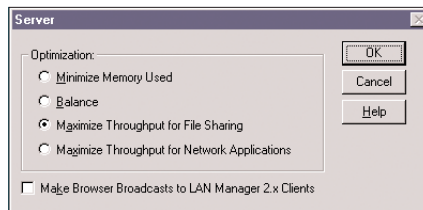
- **Memory: Pages/sec**—Shows the number of requested pages that aren't immediately available in RAM and that must be read from the disk or that had to be written to the disk to make room in RAM for other pages. If this number is high while your system is under a usual load, consider increasing your RAM. If Memory: Pages/sec is increasing but the Memory: Available Bytes counter is decreasing toward the minimum NT Server limit of 4MB, and the disks that contain the pagefile.sys files are busy (marked by an increase in %Disk Time, Disk Bytes/sec, and Average Disk Queue Length), you've identified a memory bottleneck. If Memory: Available Bytes isn't decreasing, you might not have a memory bottleneck. In this case, check for an application that's performing a large number of disk reads or writes (and make sure that the data isn't in cache). To do so, use Performance Monitor to monitor the Physical Disk and Cache objects. The Cache object counters can tell you whether a small cache is affecting system performance.
- **Memory: Available Bytes**—Shows the amount of physical memory available to programs. This figure is typically low because NT's Disk Cache Manager uses extra memory for caching, then returns the extra memory when requests for memory occur. However, if this value is consistently below 4MB on a server, excessive paging is occurring.

12 Windows Performance Tuning

- **Memory: Committed Bytes**—Indicates the amount of virtual memory that the system has committed to either physical RAM for storage or to pagefile space. If the number of committed bytes is larger than the amount of physical memory, more RAM is probably necessary.
- **Memory: Pool Nonpaged Bytes**—Indicates the amount of RAM in the nonpaged pool system memory area, in which OS components acquire space to accomplish tasks. If the Memory: Pool Nonpaged Bytes value shows a steady increase but you don't see a corresponding increase in server activity, a running process might have a memory leak. A memory leak occurs when a bug prevents a program from freeing up memory that it no longer needs. Over time, memory leaks can cause a system crash because all available memory (i.e., physical memory and pagefile space) has been allocated.
 - **Paging File: %Usage**—Shows the percentage of the maximum pagefile size that the system has used. If this value hits 80 percent or higher, consider increasing the pagefile size.

You can instruct NT Server to tune the memory that you have in your system. In the Control Panel Network applet, go to the Services tab and select Server. When you click Properties, a dialog box presents four optimization choices, as Figure 3 shows: Minimize Memory Used, Balance, Maximize Throughput for File Sharing, and Maximize Throughput for Network Applications. Another parameter that you can tune—on the Performance tab of the System Properties dialog box—is the virtual memory subsystem (aka the pagefile).

Figure 3
Choosing a server optimization option



If you have a multiuser server environment, you'll be particularly interested in two of these memory-optimization strategies: Maximize Throughput for File Sharing and Maximize Throughput for Network Applications. When you select Maximize Throughput for File Sharing, NT Server allocates the maximum amount of memory for the file-system cache. (This process is called dynamic disk buffer allocation.) This option is especially useful if you're using an NT Server machine as a file server. Allocating all memory for file-system buffers generally enhances disk and network I/O performance. By providing more RAM for disk buffers, you increase the likelihood that NT Server will complete I/O requests in the faster RAM cache instead of in the slower file system on the physical disk.

When you select Maximize Throughput for Network Applications, NT Server allocates less memory for the file-system cache so that applications have access to more RAM. This option optimizes server memory for distributed applications that perform memory caching. You can tune

applications (e.g., Microsoft SQL Server, Exchange Server) so that they use specific amounts of RAM for buffers for disk I/O and database cache.

However, if you allocate too much memory to each application in a multiapplication environment, excessive paging can turn into *thrashing*. Thrashing occurs when all active processes and file-system cache requests become so large that they overwhelm the system's memory resources. When thrashing occurs, requests for RAM create hard page faults at an alarming rate, and the OS devotes most of its time to moving data in and out of virtual memory (i.e., swapping pages) rather than executing programs. Thrashing quickly consumes system resources and typically increases response times. If an application you're working with stops responding but the disk drive LED keeps blinking, your computer is probably thrashing.

To ease a memory bottleneck, you can increase the size of the pagefile or spread the pagefile across multiple disks or controllers. An NT server can contain as many as 16 pagefiles at one time and can read and write to multiple pagefiles simultaneously. If disk space on your boot volume is limited, you can move the pagefile to another volume to achieve better performance. However, for the sake of recoverability, you might want to place a small pagefile on the boot volume and maintain a larger file on a different volume that offers more capacity. Alternatively, you might want to place the pagefile on a hard disk (or on multiple hard disks) that doesn't contain the NT system files or on a dedicated non-RAID FAT partition.

I also recommend that you schedule memory-intensive applications across several machines. Through registry editing, you can enable an NT server to use more than 256KB of Level 2 cache. Start regedit.exe, go to the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management subkey, and double-click SecondLevelDataCache. Click the decimal base, and enter the amount of Level 2 cache that you have (e.g., 512 if you have 512KB). Then, click OK, close the registry editor, and reboot. I also recommend disabling or uninstalling unnecessary services, device drivers, and network protocols.

Processor

To determine whether an NT Server machine has a CPU bottleneck, remember to first ensure that the system doesn't have a memory bottleneck. CPU bottlenecks occur only when the processor is so busy that it can't respond to requests. Symptoms of this situation include high rates of processor activity, sustained long queues, and poor application response. CPU-bound applications and drivers and extreme interrupts (which badly designed disk or network-subsystem components create) are common causes of CPU bottlenecks.

You can use the following counters to view the status of your system's CPU utilization:

- Processor: % Processor Time—Measures the amount of time a processor spends executing nonidle threads. (If your system has multiple processors, you need to monitor the System: % Total Processor Time counter.) If a processor consistently runs at more than 80 percent capacity, the processor might be experiencing a bottleneck. To determine the cause of a processor's activity, you can monitor individual processes through Performance Monitor. However, a high Processor: % Processor Time doesn't always mean the system has a CPU bottleneck. If the CPU is servicing all the NT Server scheduler requests without building up the Server Work Queues or the Processor Queue Length, the CPU is servicing the processes as fast as it can handle them. A processor bottleneck occurs when the System: Processor Queue Length value is growing; Processor: % Processor Time is high; and the system's memory, network interface,

14 Windows Performance Tuning

and disks don't exhibit any bottlenecks. When a CPU bottleneck occurs, the processor can't handle the workload that NT requires. The CPU is running as fast as it can, but requests are queued and waiting for CPU resources.

- Processor: % Privileged Time—Measures the amount of time the CPU spends performing OS services.
- Processor: % User Time—Measures the amount of time the processor spends running application and subsystem code (e.g., word processor, spreadsheet). A healthy percentage for this value is 75 percent or less.
- Processor: Interrupts/sec—Measures the number of application and hardware device interrupts that the processor is servicing. The interrupt rate depends on the rate of disk I/O, the number of operations per second, and the number of network packets per second. Faster processors can tolerate higher interrupt rates. For most current CPUs, 1500 interrupts per second is typical.
- Process: % Processor Time—Measures the amount of a processor's time that a process is occupying. Helps you determine which process is using up most of a CPU's time.
- System: Processor Queue Length—Shows the number of tasks waiting for processor time. If you run numerous tasks, you'll occasionally see this counter go above 0. If this counter regularly shows a value of 2 or higher, your processor is definitely experiencing a bottleneck. Too many processes are waiting for the CPU. To determine what's causing the congestion, you need to use Performance Monitor to monitor the process object and further analyze the individual processes making requests on the processor.

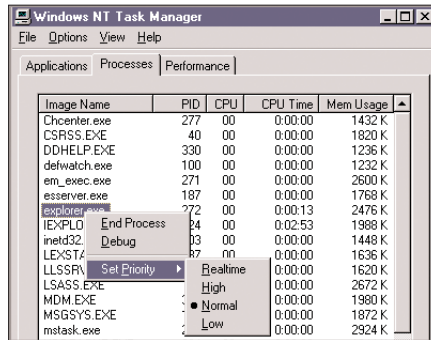
One way to resolve processor bottlenecks is to upgrade to a faster CPU (if your system board supports it). If you have a multiuser system that's running multithreaded applications, you can obtain more processor power by adding CPUs. (If a process is multithreaded, adding a processor improves performance. If a process is single-threaded, a faster processor improves performance.) However, if you're running the single-processor NT kernel, you might need to update the kernel to the multiprocessor version. To do so, reinstall the OS or use the resource kit's `uptomp.exe` utility.

Another way to tune CPU performance is to use Task Manager to identify processes that are consuming the most CPU time, then adjust the priority of those processes. A process starts with a base priority level, and its threads can deviate two levels higher or lower than the base. If you have a busy CPU, you can boost a process's priority level to improve CPU performance for that process. To do so, press `Ctrl+Alt+Del` to access Task Manager, then go to the Processes tab. Right-click the process, choose Set Priority, and select a value of High, Normal, or Low, as Figure 4 shows. The priority change takes effect immediately, but this fix is temporary: After you reboot the system or stop and start the application, you'll lose the priority properties that you set. To ensure that an application always starts at a specified priority level, you can use the Start command from the command line or within a batch script. To review the Start command's options, enter

```
start /?
```

at the command prompt.

Figure 4
Setting a process's priority level



Disk

To determine whether your system is experiencing disk bottlenecks, first ensure that the problem isn't occurring because of insufficient memory. A disk bottleneck is easy to confuse with pagefile activity resulting from a memory shortage. To help you distinguish between disk activity related to Virtual Memory Manager's paging to disk and disk activity related to applications, keep pagefiles on separate, dedicated disks.

Before you use Performance Monitor to examine your disks, you must understand the difference between its two disk counters. LogicalDisk counters measure the performance of high-level items (e.g., stripe sets, volume sets). These counters are useful for determining which partition is causing the disk activity, possibly identifying the application or service that's generating the requests. PhysicalDisk counters show information about individual disks, regardless of how you're using the disks. LogicalDisk counters measure activity on a disk's logical partitions, whereas PhysicalDisk counters measure activity on the entire physical disk.

NT doesn't enable Performance Monitor disk counters by default; you must enable them manually. Enabling these counters will result in a 2 to 5 percent performance hit on your disk subsystem. To activate Performance Monitor disk counters on the local computer, type

```
diskperf -y
```

at a command prompt. (If you're monitoring RAID, use the `-ye` switch.) Restart the computer.

To analyze disk-subsystem performance and capacity, monitor the Performance Monitor's disk-subsystem counters. The following counters are available under both LogicalDisk and PhysicalDisk:

- **% Disk Time**—Measures the amount of time the disk spends servicing read and write requests. If this value is consistently close to 100 percent, the system is using the disk heavily. If the disk is consistently busy and a large queue has developed, the disk might be experiencing a bottleneck. Under typical conditions, the value should be 50 percent or less.
- **Avg. Disk Queue Length**—Shows the average number of pending disk I/O requests. If this value is consistently higher than 2, the disk is experiencing congestion.

16 Windows Performance Tuning

- Avg. Disk Bytes/Transfer—Measures throughput (i.e., the average number of bytes transferred to or from the disk during write or read operations). The larger the transfer size, the more efficiently the system is running.
- Disk Bytes/sec—Measures the rate at which the system transfers bytes to or from the disk during write or read operations. The higher the average, the more efficiently the system is running.
- Current Disk Queue Length—Shows how many disk requests are waiting for processing. During heavy disk access, queued requests are common; however, if you see requests consistently backed up, your disk isn't keeping up.

If you determine that your disk subsystem is experiencing a bottleneck, you can implement several solutions. You can add a faster disk controller, add more disk drives in a RAID environment (spreading the data across multiple physical disks improves performance, especially during reads), or add more memory (to increase file cache size). You also might try defragmenting the disk, changing to a different I/O bus architecture, placing multiple partitions on separate I/O buses (particularly if a disk has an I/O-intensive workload), or choosing a new disk with a low seek time (i.e., the time necessary to move the disk drive's heads from one data track to another). If your file system is FAT, remember that NTFS is best for volumes larger than 400MB.

You can also provide more disk spindles to the application. How you organize your data depends on your data-integrity requirements. Use striped volumes to process I/O requests concurrently across multiple disks, to facilitate fast reading and writing, and to improve storage capacity. When you use striped volumes, disk utilization per disk decreases and overall throughput increases because the system distributes work across the volumes.

Consider matching the file system's allocation unit size to the application block size to improve the efficiency of disk transfers. However, increasing the cluster size doesn't always improve disk performance. If the partition contains many small files, a smaller cluster size might be more efficient. You can change the cluster size in two ways. At the command line, enter

```
format <disk>:/FS:NTFS /A:<cluster size>
```

or use Disk Administrator. Select Tools, Format, and change the allocation unit size. NTFS supports a cluster size of 512 bytes, 1024 bytes, 2048 bytes, 4096 bytes, 8192 bytes, 16KB, 32KB, or 64KB. FAT supports a cluster size of 8192 bytes, 16KB, 32KB, 64KB, 128KB, or 256KB.

Network Interface

After you consider a system's memory, CPU, and disk metrics, your next step is to examine the network subsystem. Client machines and other systems must be able to connect quickly to the NT Server system's network I/O subsystem so that they provide acceptable response times to end users. To determine where network bottlenecks reside and how to fix them, you must understand what type of workload your client systems generate, which key network architecture components are in use, and what type of network protocol (e.g., Ethernet, NetBEUI) and physical network you're on. Performance Monitor collects data for each physical network adapter. To determine how busy your adapters are, use the following counters:

- Network Interface: Output Queue Length—Measures the length of an adapter's output packet queue. A value of 1 or 2 is acceptable. However, if this measurement is frequently at or higher

than 3 or 4, your network I/O adapter can't keep up with the server's requests to move data onto the network.

- Network Interface: Bytes Total/sec—Measures all network traffic (number of bytes sent and received) that moves through a network adapter, including all overhead that the network protocol (e.g., TCP/IP, NetBEUI) and physical protocol (e.g., Ethernet) incur. If, for example, in a 10Base-T network, the value is close to 1Mbps and the output queue length continues to increase, you might have a network bottleneck.
- Network Interface: Bytes Sent/sec—Shows the number of bytes sent through a specific network adapter card.
- Server: Bytes Total/sec—Shows the number of bytes that the server has sent and received over the network through all of its network adapter cards.
- Server: Logon/sec—Shows the number of logon attempts per second for local authentication, over-the-network authentication, and service-account authentication. This counter is useful on domain controllers (DCs) to determine how much logon validation is occurring.
- Server: Logon Total—Shows the number of logon attempts for local authentication, over-the-network authentication, and service-account authentication since the computer was last started.

If you determine that the network subsystem is experiencing a bottleneck, you can implement numerous measures to alleviate the problem. You can bind your network adapter to only those protocols that are currently in use, upgrade your network adapters to the latest drivers, upgrade to better adapters, or add adapters to segment the network (so that you can isolate traffic to appropriate segments). Check overall network throughput, and improve physical-layer components (e.g., switches, hubs) to confirm that the constraint is in the network plumbing. You might also try distributing the processing workload to additional servers.

In a TCP/IP network, you can adjust the TCP window size for a potential improvement in performance. The TCP/IP receive window size shows the amount of receive data (in bytes) that the system can buffer at one time on a connection. In NT, the window size is fixed and defaults to 8760 bytes for Ethernet, but you can adjust the window size in the registry. You can either modify the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize subkey to globally change the setting on the computer or use the `setsockopt()` Windows Sockets call to change the setting on a per-socket basis. The optimal size depends on your network architecture. In TCP, the maximum achievable throughput equals window size divided by round-trip delay or network latency.

Finally, don't use Autosense mode in your network adapters. Set your NICs to the precise speed that you want. To change the setting, use the configuration program that came with your network adapter.

Understand Your Environment

Performance analysis requires logical thinking, testing, and patience. NT's primary monitoring and tuning tools can help you manage the performance of your company's systems.

The key to achieving your objective is understanding what you have, how your applications work, and how your users use your network. To resolve any performance problems and plan for

18 Windows Performance Tuning

future requirements, combine the knowledge you gain from these tools' output with an understanding of your applications and your environment.

Chapter 3

Tuning NT Server Disk Subsystems

—by Curt Aubley

Performance! Everyone wants Windows NT servers and workstations to run as fast as possible. Keeping your disk subsystem running at its best is an important step in improving the overall performance of your NT solution. In this chapter, I'll show you how to maximize the performance of your extra disk capacity when implementing NT disk subsystems and RAID arrays, regardless of any vendor-specific hardware tweaks. You can use NT's built-in tools and a freeware tool to quickly optimize your disk subsystem (hardware- or software-based RAID). To accomplish this task, you must understand the performance characteristics of the standalone disk or RAID technology you are working with and the workload characteristics of your existing disk subsystem. Using this information, you can load balance your disk subsystem and tune the disk allocation unit size. Finally, because your performance increase can vary according to your computing environment, I'll show you how I tuned and tested my disk subsystem.

Common RAID Characteristics

Most systems administrators add disk subsystem capacity in the form of RAID arrays. When you use NT's internal software RAID technology or hardware-based vendor RAID technology, RAID 0, 1, and 5 are the most common RAID levels for NT Server environments. (For information about RAID level technology, see Table 1.)

Table 1
*Relative RAID Performance Guide**

Disk I/O Characteristics (1st is fastest, 2nd is second fastest, 3rd is slowest)

Fault Tolerance Level Provided	RAID Level	Random Read	Random Write	Sequential Read	Sequential Write
None	Stripe (0)	1st	1st	1st	1st
Capable of surviving a single disk failure without loss of data	Mirror (1)	2nd 2nd	2nd (up to two times as slow as one hard disk or a RAID 0 stripe)	3rd	2nd (up to two times as slow as one hard disk or a RAID 0 stripe)
Capable of surviving a single disk failure without loss of data	Stripe w/parity (5)	1st	3rd (up to four times as slow as other RAID options)	1st 1st	3rd (up to four times as slow as other RAID options)

* This table illustrates which RAID level provides the best performance under four common disk workload environments. The lower the ranking, the better the performance that particular RAID level provides, compared to the other RAID levels. Interpret this table by matching up your disk I/O characteristics and the performance level you are seeking.

When you select RAID configurations, you need to consider many important factors, such as cost and availability level required, in addition to just performance; however, this chapter focuses on performance. For information about other factors to consider, see the sidebar “What to Look for When Selecting a Scalable RAID Array.”

Regardless of the RAID level you select, use a hardware-based RAID solution if you implement any NT solution that supports more than three disks and requires any level of high availability and performance. Hardware-based RAID adapters and controllers provide much better performance than NT’s built-in software-based solution provides and are particularly valuable when you implement RAID 5. With RAID 5, hardware-based RAID adapters offload parity processing and some of the associated interrupt traffic, unlike software-based RAID. This offloading results in improved overall NT performance.

When you’re selecting the appropriate RAID level, consider the relative performance characteristics that each RAID level provides. Table 1 illustrates the relative performance level of the various RAID levels in several I/O workload environments and can help you match the appropriate performance to your system’s disk I/O characteristics.

Grouping Similar Disk Activities

Configuring one large RAID 5 array to handle all your disk I/O needs might appear to be the easiest solution, but this approach is not always a wise decision. You can dramatically improve performance by matching the performance characteristics of each RAID level with your disk workload patterns. For example, a Microsoft Exchange Server environment contains both sequentially write-intensive log files and random information-store data files. Instead of using one RAID 5 array for both activities, you’ll achieve the greatest performance by placing the write-intensive log files on their own RAID 1 array and leaving the random information-store data files on the RAID 5 array. This approach provides better performance because you’re moving the write-intensive workload away from the RAID 5 array, which exhibits slower write performance than a RAID 1 array exhibits. (For more information about RAID 5, see the sidebar “Why Is RAID 5 Slow on Writes?”) Configuring the RAID levels to match your workload patterns improves the response times of the disk subsystem and ultimately the NT system.

If you use a stress-testing tool to measure your server’s performance, you can quantify the overall server performance benefit of using multiple RAID levels or adding extra standalone disks. If you don’t have a stress-testing tool available, you can use two counters (Avg. Disk sec/Write and Avg. Disk sec/Read) under the LogicalDisk object in NT’s Performance Monitor to help you determine whether using multiple RAID levels or standalone disks will increase performance. The Avg. Disk sec/Write counter measures the average time in seconds to write data to the disk, and the Avg. Disk sec/Read counter measures the average time in seconds to read data from the disk. Look at these two counters before and after you change your disk subsystem configuration. If the workload on your server is roughly the same before and after you make changes to your disk subsystem, you will see significant improvements in these two metrics after you implement multiple RAID levels or add standalone disks. Remember, these values will always be zero if you haven’t run Diskperf -ye from the NT command prompt and rebooted your NT system to activate NT’s collection of disk metrics. Numerous other techniques and Performance Monitor counters can help you measure increased application performance after you have made your changes to your disk subsystem.

What to Look for When Selecting a Scalable RAID Array

—by Curt Aubley

Suppose you're looking for a robust and scalable RAID array solution to store your important enterprise data. You need to consider the fault tolerance of the RAID array, the RAID array manageability tools, support for dynamic growth of the RAID array, support for additional disk drives, and the I/O technology in place.

Fault Tolerance of the RAID Array

Regardless of how well you tune and size your RAID array, it can still fail to a point at which you can't access your data. The importance of your data directly affects the level of fault tolerance you apply to your RAID array solution. Improved fault tolerance is synonymous with higher cost. If you use a RAID level other than RAID 0, you've already decided that your data has some level of importance because you're providing disk-level redundancy in the form of RAID. Beyond this precaution, consider these desirable fault-tolerant features to further bolster your overall RAID array solution: redundant hot-swappable power supplies, redundant hot-swappable fans, and hot-swappable hard disks. These items are becoming more common, and you should consider them mandatory.

Some external RAID arrays, such as those from Data General and Symbios, provide embedded RAID array controllers. These units internally support multiple SCSI channels to each hard disk, so that if one disk array adapter channel fails, you can still access the disks in the array. For the ultimate in RAID array reliability, you can add multiple RAID array adapters in your server, where each provides a separate channel to your RAID array. Again, if one RAID array adapter fails, the server and subsequently your customers can still access the information on the array. You can also leverage these extra channels for improved performance of larger RAID arrays. Whenever possible, try to remove single points of failure.

RAID Manageability Tools

Manageability tools are the most commonly overlooked component of a scalable and reliable RAID array system. How do you know whether a disk in your RAID 5 array fails or a power supply fails? Unfortunately, such failures can kill performance because a RAID array that is operating in fault mode (e.g., with a failed disk in an array) runs slowly. You need to ensure that your RAID array includes built-in manageability tools, and you need to learn how to use them. Typically, these tools let you configure and monitor your RAID array. These tools often include an array agent that runs as a service under Windows NT. The array agent closely monitors the performance of your RAID array solution. If a disk or another component in the array fails, the array agent records

Continued on page 22

What to Look for When Selecting a Scalable RAID Array *continued*

the failure in NT's event log. Also, you can configure array agents to send email, provide a visual alert, or even page you as needed (depending on the vendor's implementation). Diagnostic lights on the outside of the RAID array are cute, but who wants to sit in a computer room and watch some lights blink on and off or listen for audible alerts? Instead, you can easily automate the diagnostic monitoring and ensure that your RAID array is running flawlessly.

Support for Dynamic Growth of the RAID Array

To configure a RAID array under NT a few years ago, you had to boot your NT system into DOS mode, run a utility to create the RAID array, and then format the array under NT with a file system. Later, if you needed to add an extra disk to the array for capacity or performance, you had to back up the data on the array, go into DOS mode, reconfigure the array, reformat the array, and restore the data. Fortunately, you don't have to endure this process today. When selecting your RAID array, ensure that it includes tools to let you dynamically add hot-swappable disks. This functionality allows for a truly scalable RAID array solution.

Support for Additional Disks

Now that you can dynamically add disks to your array, ensure that your RAID array supports additional disks. Wide SCSI-2 lets you connect up to 16 devices to one SCSI channel. Depending on your workload, you might consider configuring up to 10 disks in one large array and still not saturate the SCSI channel. You need to ensure that your RAID array enclosure will support the disks that you need in the future. Also, higher-density SCSI adapters now support one to four channels per adapter. This support lets you conserve those precious PCI expansion slots in your NT system. For even more flexibility when adding and configuring your RAID arrays, you might consider a RAID array system that leverages fibre channel technology. Both fibre channel- and SCSI-based RAID arrays must still wait on the physical limitations of the disks when obtaining data. However, fibre channel allows for more arrays per channel and greater distances between the server and its arrays.

I/O Technology in Place: Follow Your Data

Regardless of whether you leverage fibre channel- or SCSI-based RAID arrays, you must closely monitor the data path between the RAID arrays and your server. Each disk can provide a certain amount of throughput. As you group more and more arrays on a particular channel, you need to ensure that the data path between your arrays and your server doesn't become saturated. To avoid saturation, select the RAID array technology that meets your needs. Fibre channel can run at speeds up to 100Mbps, and the latest

Continued on page 23

What to Look for When Selecting a Scalable RAID Array *continued*

SCSI standard can support speeds up to 80Mbps. Follow the data path from the disk to your NT system. You can take this precaution to ensure that you haven't configured too many RAID arrays and avoid overwhelming the data channel, the disk array controller in the RAID array, the RAID adapter in the NT system, or even the PCI bus that the RAID adapter attaches to.

NT doesn't provide any one counter in Performance Monitor to accomplish this task, but you can review Performance Monitor's LogicalDisk object's Disk Bytes/sec counter for each array connected on a specific channel, add these values together, and ensure they aren't exceeding 70 percent of the theoretical throughput your data path can support. If the total does exceed this threshold, consider adding a second data (SCSI or fibre channel) connection to your RAID array configuration. The converse of this concept is helpful also. If you have plenty of bandwidth available from your current data channel, you can use this data channel even further by adding more RAID arrays with confidence.

Disk Workload Characteristics

How can you determine what type of disk workload your server is experiencing so that you can appropriately distribute the disk activities across multiple RAID levels or standalone disks? Performance Monitor provides two counters (% Disk Read Time and % Disk Write Time) under the LogicalDisk object that let you identify disk subsystem workload characteristics. The % Disk Read Time counter measures the percentage of elapsed time that the selected disk is busy servicing read requests, and the % Disk Write Time counter measures the percentage of elapsed time that the selected disk is busy servicing write requests.

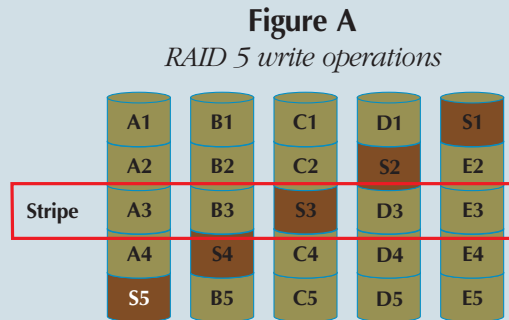
Using these counters, you can determine how much time your disk spends writing and reading data. These counters provide a high-level view of the type of disk activity you must plan for. Use this information with the information in Table 1 to select the RAID levels that provide the best performance for your environment. For example, if the value for % Disk Read Time is 10 percent and the value for % Disk Write Time is 90 percent, consider either RAID 1, RAID 0, or a standalone disk (for the latter option, remember that you forfeit any disk fault tolerance for improved performance). As Table 1 shows, RAID 5 write performance is lower than other RAID levels and standalone disks because every RAID 5 write incurs four *disk* operations: read data, read parity data (compare the two using the CPU), write the data, write the parity data. Conversely, if the value for % Disk Read Time is 80 percent and the value for % Disk Write Time is 20 percent, RAID 5 is a good choice.

Many enterprise networks have a mixed read and write environment with some RAID devices experiencing much higher workloads than others. In these instances, you need to load balance your disk devices for optimal performance.

Why Is RAID 5 Slow on Writes?

—by Tony Redmond and Pierre Bijaoui

Books about storage often refer to RAID 5 as striping with distributed parity. RAID 5 comprises a logical volume, based on three or more disk drives, that generates data redundancy to avoid the loss of an entire volume in the event of disk failure. The RAID controller creates a special parity block for each stripe of information, as Figure A shows. (Features at the OS level, such as Windows 2000's disk striping with parity, can also perform this function.) The parity is typically a binary exclusive OR (XOR) operation—indicated here with the \oplus symbol—on all the data blocks of the stripe. In Figure A, the RAID controller calculates parity as $S3 = A3 \oplus B3 \oplus D3 \oplus E3$.



When a write operation occurs on a RAID 5 volume (e.g., on block B3), the controller must update parity block S3. Because the controller must read all the blocks in the stripe to recreate the parity block, most RAID 5 controllers will go through the following steps, in which single quotation marks and double quotation marks represent modifications:

1. Read the block that needs modification (B3).
2. Read the parity block (S3).
3. Remove the knowledge of block B3 from parity S3 ($S3' = S3 \oplus B3$).
4. Calculate the new parity ($S3'' = S3' \oplus B3'$).
5. Update the data block (B3').
6. Update the parity block (S3'').

In other words, one application I/O requires four disk I/Os, and these four I/Os occur on two spindles, potentially disturbing other transfer operations on those volumes.

Some high-end controllers optimize the disk activities so that if the controller needs to write an entire stripe (e.g., during a restore operation, in which the I/Os are typically large), the controller calculates the new parity on the fly and overwrites the old parity. However, this feature requires that the controller update all the other data blocks in the stripe.

Load Balancing Your Disks

Throwing more hardware at a bottleneck in the disk subsystem isn't always effective. By load balancing your existing disk subsystem, you can capitalize on your investment and better understand when and where to add disk hardware, if needed. The key to load balancing your NT Server disk subsystem is to identify which disk device is the bottleneck, understand the characteristics of the application using the disk device, and determine which files on the disk device the application is using. Performance Monitor is an excellent tool to help you identify which disk device is the bottleneck. Unfortunately, Performance Monitor alone isn't enough. To understand how and where to distribute your disk workload, you must know which applications (processes) access which files on the disk device.

After you identify which disk device is the bottleneck, you can use two techniques to help you isolate the files on the disk device your applications use. The first technique involves using NT's internal auditing feature (available under Administrative Tools/UserManager/Policies/Audit) to alert you when applications access specific files. This technique is helpful, but the auditing feature generates and sends output to the Event Monitor and can be challenging to decrypt. Auditing also adds a lot of overhead to both the CPU and the disk subsystem selected for auditing.

The second technique, which I find easier to implement, is using Sysinternals' freeware tool, Filemon.exe, which is available at <http://www.sysinternals.com>. After you download this utility, you just unzip the file and run the utility during high usage periods to get a feel for the specific files your applications are accessing on the disk subsystem, as Figure 1 shows.

Figure 1

Using Filemon.exe to help load balance your NT disk subsystem

#	Time	Process	Request	Path	Result	Other
143	1:23:59 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 6164480 Len...
144	1:23:59 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 6197248 Len...
145	1:23:59 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 7020544 Len...
146	1:23:59 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 7032832 Len...
147	1:23:59 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 12603392 Len...
148	1:24:01 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 3379200 Len...
149	1:24:01 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 8192 Length:...
150	1:24:01 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 0 Length: 409
151	1:24:01 PM	NTFILMON...	IRP_MJ_READ	C:\WINNT\system32\COMCTL32.DLL	SUCCESS	Offset: 279040 Leng...
152	1:24:02 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 0 Length: 409
153	1:24:06 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 3387392 Len...
154	1:24:06 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 12288 Length...
155	1:24:06 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 4096 Length:...
156	1:24:08 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 0 Length: 409
157	1:24:08 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 45056 Length...
158	1:24:11 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 3391488 Len...
159	1:24:11 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 8192 Length:...
160	1:24:11 PM	System	IRP_MJ_WRITE	C: DASD	SUCCESS	Offset: 0 Length: 409
161	1:24:14 PM	Control.exe	IRP_MJ_CREATE	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Attributes:0080 Opti...
162	1:24:14 PM	Control.exe	FASTIO_QUERY...	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Size: 7705
163	1:24:14 PM	Control.exe	IRP_MJ_READ	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Offset: 0 Length: 770
164	1:24:14 PM	System	IRP_MJ_CLOSE	C:\Program Files\ExecSoft\Diskeep\Defr...	SUCCESS	
165	1:24:14 PM	Control.exe	IRP_MJ_CLEANUP	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	
166	1:24:14 PM	Control.exe	IRP_MJ_CREATE	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Attributes:0080 Opti...
167	1:24:14 PM	Control.exe	FASTIO_QUERY...	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Size: 7705
168	1:24:14 PM	Control.exe	IRP_MJ_READ	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	Offset: 0 Length: 770
169	1:24:14 PM	Control.exe	IRP_MJ_CLEANUP	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	
170	1:24:14 PM	Control.exe	IRP_MJ_CLOSE	C:\program files\ExecSoft\Diskeep\Defr...	SUCCESS	

Filemon gives you the data you need to load balance your disk subsystem. As Figure 1 shows, you can determine which process is accessing which part of the disk subsystem. By using Filemon with Performance Monitor, you can decide which disk devices to move files from (disks with a high % Disk Time—a combination of read time and write time), which files to move, and where to move them (disk devices whose % Disk Time is low).

Not only does Filemon tell you which applications access which files on which disk devices, but it can also show you whether the disk requests are a read or write activity. As a result, you have more granular data to use when deciding which RAID level to use or when adding standalone disks.

Be careful when you move files around to different volumes under NT. Make sure that the shares and permissions are properly set after you move the files. If the files are associated with a specific application, you might need to move the files using the application or update the registry. You can determine how you move the files in the disk subsystem according to the applications running in your environment. Regardless of your environment, you will want to have a complete system backup available to restore from if necessary. I also suggest that you make sure no active users are on your system when you make changes to your disk subsystem; otherwise, frustrated end users might complain. Always remember to test your application before and after you move any files to ensure you don't accidentally break anything.

Tuning the Allocation Unit Size

NTFS uses clusters as the fundamental unit of disk allocation. A cluster consists of a fixed number of disk sectors. When you use the Format command or NT's Disk Administrator, clusters are known as the allocation units. In NTFS, the default allocation unit size depends on the volume size. Using the Format command from the command line to format your NTFS volume, you can specify a variety of allocation unit sizes for a specific NT disk volume. For example, to format a volume using an 8KB allocation unit size, go to the command prompt and type

```
Format k: /f:NTFS /A:8192
```

For information about the Format command, go to the command prompt and type

```
Format /? | more
```

The default allocation unit size that NT provides is a great place to start if you're unaware of the disk workload characteristics. Before you set up a RAID array or new standalone disks, you need to determine the size of the average disk transfer on your disk subsystem and set the allocation unit size to match it as closely as possible. By matching the allocation unit size with the amount of data that you typically transfer to and from the disk, you'll incur lower disk subsystem overhead and gain better overall performance. To determine the size of your average disk transfer, use Performance Monitor to review two counters (Avg. Disk Bytes/Read and Avg. Disk Bytes/Write) under the LogicalDisk object. The Avg. Disk Bytes/Read counter measures the average number of bytes transferred from the disk during read operations, and the Avg. Disk Bytes/Write counter measures the average number of bytes transferred to the disk during write operations.

After you measure the number of bytes written to and read from your disk subsystem, you can set the allocation unit size so that you can achieve maximum performance. Of course, if you

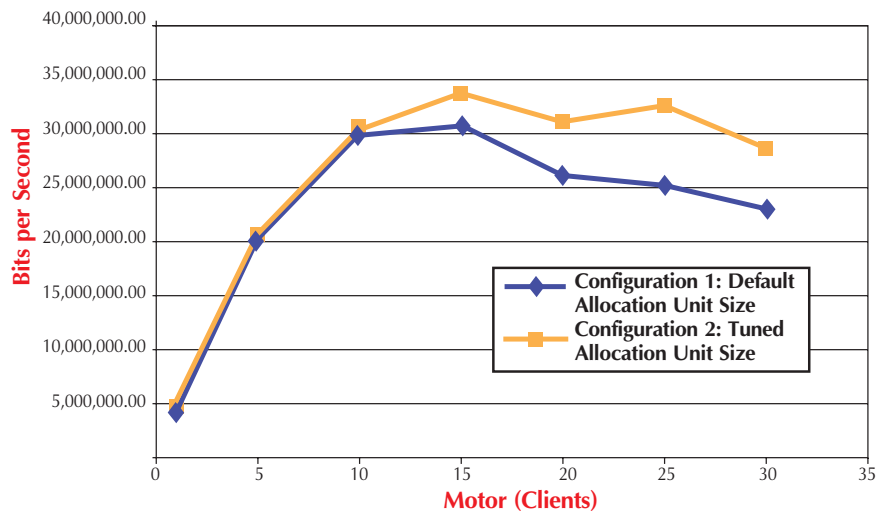
want to change the allocation unit size for a file system on the disk device, you have to back up the data from the disk device, use the new allocation unit size to format the disk device, and restore your data. Just as you need to use multiple RAID levels or standalone disks to accommodate different disk workloads, you will want to use this same approach when formatting multiple RAID arrays and standalone disks. Customize each disk device with the allocation unit size appropriate for the projected workload characteristics. If you can't determine the disk workload characteristics, use a smaller allocation unit size on devices that you expect to be random or read-intensive and use a larger allocation unit size for disk devices that will experience sequential and write-intensive workloads.

Your Results May Vary

Just how much your disk subsystem performance will improve using the tuning techniques I've described in this chapter depends on your environment. To test the differences in matching the allocation unit size, I ran a stress test. My test system consisted of an NT Server 4.0 file server configured with a 100Base-T network card, a 200MHz Pentium Pro processor, 64MB of RAM, a SCSI-based hardware RAID array controller with 16MB of onboard cache, a hard disk for NT, a hard disk for the paging file, and a three-disk RAID 5 array. I tested this system by using the Dynameasure file-server stress-testing tool from Bluecurve operating from four 400MHz Pentium II processor clients with 128MB of RAM and 100Base-TX networking cards. While running the baseline test, I used Performance Monitor to obtain the disk characteristics of the RAID 5 array. I used this information to reformat the RAID 5 array from the default allocation unit size of 4KB to an allocation unit size of 64KB according to my Performance Monitor results. When I retested the system with the same workload, the results were dramatic, as Figure 2 shows. In this chart, higher values are better. As you can see, once the load increased on the file server, the performance of the tuned RAID array improved by more than 5Mbps in some instances.

Figure 2

Default Allocation Unit Size vs. Tuned Allocation Unit Size



Know Your Environment

Although this chapter presents general performance recommendations, you need to recognize that you'll achieve the best RAID or standalone disk performance improvements when you understand what is occurring on your particular NT system. Then you can make informed decisions when you customize your NT system to match your specific needs.

Chapter 4

Take It to the Limit

—by Joe Rudich

When civil engineers design a bridge, they try to determine the point at which demand and capacity intersect. Demand is composed of factors outside the physical structure of the bridge, such as the maximum number of vehicles that can be on it at one time and the number of vehicles that will cross it in a year. Capacity has to do with the known strength of bridge construction materials.

Server engineers face the same demands as bridge builders, although in a somewhat more virtual fashion. If a server supporting a mission-critical Windows NT application becomes processor bound and unresponsive during the busiest time of day, cars won't fall into rivers, but the engineer who attested that the server would be able to handle the demand might feel like jumping.

Administrators face questions about capacity nearly every time they need to purchase or configure a server or an application, but they're at a disadvantage compared with their counterparts in civil engineering. Determining a server's capacity is an inexact science. You can use one of three methods to attempt it.

Rely on Vendors' Recommendations

Hardware manufacturers and application software makers usually provide recommendations for the type of server hardware you should use and how you should configure the hardware and software. However, those estimates tend to be general and conservative.

Simulate the Server's Actual Use

If time and a suitable test environment are available, an application server's engineering and construction phase can include a period during which that server is tested under "live fire" (i.e., actual use by users). Companies often use this method, however, for "proof of concept" rather than for a true capacity test (i.e., they don't test the application server to the point of failure).

Use Software to Simulate a Heavy Workload

Simulation solves the other two options' major shortcomings. On NT servers, you can perform workload simulation with a utility called Response Probe, which Microsoft provides in the *Microsoft Windows NT Server 4.0 Resource Kit*. (Response Probe is incompatible with Windows 2000.)

What Is Response Probe?

Server workloads can be difficult—even impossible—to duplicate because factors such as background network traffic and number of users logged on are hard to control. Response Probe's strength is its controllability. Response Probe lets you design a reproducible, application-independent workload and use it to test the performance of a server's hardware and software configuration

without anyone having to use the server. The tool doesn't manipulate the application running on the system; rather, it evaluates performance by generating unique threads that create a specific load on the server. In other words, Response Probe replaces the application in question. You run Response Probe instead of the system's primary application but along with the system's secondary applications.

Response Probe works by simulating real computer use. By *real*, I mean that the tool simulates a typical workload cycle—a variable think time followed by file access followed by computation. Response Probe assumes that many characteristics of real workloads follow a standard bell curve, or *normal distribution*, and designs the test workload accordingly. For example, the time that users spend thinking about what action to take varies from user to user and task to task, and the time spent finding a record in a file varies according to the location of the record on the hard disk relative to the read arm's current position. Response Probe assumes that in a typical workload, these and other workload variables are distributed normally and so can be described by specifying a mean and a standard deviation. (For details about how to use a mean and standard deviation to specify a normal distribution, see Chapter 11 of the *Microsoft Windows NT Workstation 4.0 Resource Guide* at <http://www.microsoft.com/technet/prodtechnol/ntwrkstn/reskit/03tools.asp?frame=true>.) Response Probe lets you specify means and standard deviations for several such workload characteristics, as you'll see later.

To test your server's performance, Response Probe relies on three closely linked text files that describe the simulated workload. These three files are

- a process script file (.scr), which creates each process in the test workload
- a thread definition file (.scp), which creates the threads that will run in the process
- a thread description file (.sct), which describes a thread process (You need a thread description file for each thread you define.)

Response Probe requires at least one of each of these files for every test it performs.

Using these files, Response Probe generates a precise activity level that's identical each time you run the same script configuration. By varying the script parameters from test to test, you can first establish a baseline for performance, then use Response Probe to find the upper limits of a server's capacity.

Creating the Script Files

You use the process script file to create each process in the test workload. Each line in the file creates a process and uses the syntax

```
REPEAT <n>
PROCESS <ThreadDefFile.scp>
  <DataPages>
  <ProcessName>
  <PriorityClass>
```

REPEAT *n* is an optional parameter that runs the process *n* times. *ThreadDefFile.scp* specifies the name of the file that defines the process's threads. *DataPages* specifies the number of pages designated to simulate data pages. (*Data pages* refers to NT's internal buffer storage of data.) *Pro-*

cessName is an optional parameter that specifies a name for the process in which the test will run, and *PriorityClass* is an optional parameter that sets a relative priority for the process. Possible values for *PriorityClass* are I (Idle), N (Normal), H (High), and R (Realtime). For example, the line

```
PROCESS servbase.scp 10
```

creates one process, associates it with thread definition file *servbase.scp*, and specifies a 10-page datapage file.

In the thread definition file, you create the threads that you name in the process script file by using the syntax

```
REPEAT <n>
THREAD <ThreadDesFile.sct>
<ThreadPriorityAdjustment>
```

REPEAT *n* is an optional parameter that creates *n* instances of the thread. *ThreadDesFile.sct* specifies the name of the thread description file, and *ThreadPriorityAdjustment* lets you set the thread to a priority other than the one you specified for the process. Possible values for *ThreadPriorityAdjustment* are T (TimeCritical), H (Highest), A (AboveNormal), N (Normal, the default), B (BelowNormal), L (Lowest), and I (Idle). For example, the line

```
THREAD servbase.sct
```

creates the thread that you've described in the thread description file named *servbase.sct*. Because this thread definition line doesn't adjust the priority, the thread runs at the same priority as the process.

Finally, you describe the thread in each process in the thread description file. You set most of Response Probe's configuration values in this file, including the means and standard deviations for several parameters that Response Probe uses to generate each thread. Table 1 shows the parameters that you can configure.

You build the thread description file in table format with one line per parameter. The parameters can be in any order. For example, the lines

```
THINKTIME      0      0
CPUTIME 0
CYCLEREADS     100    30
FILESEEK       0      0
DATAPAGE       0      0
FUNCTION       500    0
FILEACCESS     workfile.dat
FILEATTRIBUTE  SEQUENTIAL
FILEACCESSMODE UNBUFFER
RECORDSIZE     2048
FILEACTION     R
```

set the parameters for a thread that performs reads from a file. The parameters indicate that this thread

Table 1
Thread Description File Parameters

Parameter	Description	Valid Values for Mean
CPUTIME	Mean and standard deviation describing the amount of time spent in the compute state	0 to trial time
CYCLEREADS	Mean and standard deviation describing the number of times Response Probe executes the FILEACTION and CPUTIME parameters between think times	1 (minimum)
DATAPAGE	Mean and standard deviation describing the position of the page to be written to in Response Probe's simulated data page file	0 to the data size you specified in the process script file
FILEACCESS	The filename of the test workload file	N/A
FILEACCESSMODE	An optional parameter that specifies how Response Probe accesses files; valid values are BUFFER (uses the system cache), UNBUFFER (no cache), and MAPPED (array in memory)	N/A
FILEACTION	An optional parameter that specifies the read and write pattern; valid values are R and W (the default is R—one read—but you can specify numerous Rs and Ws in any combination)	N/A
FILEATTRIBUTE	An optional parameter that specifies the type of file access; valid values are RANDOM (the default) and SEQUENTIAL	N/A
FILESEEK	Mean and standard deviation describing the position of the record Response Probe will access in a file	1 to the number of records in the file
FUNCTION	Mean and standard deviation describing the position of the function that Response Probe will read from the internal codepage file	1 to 1000
RECORDSIZE	Mean and standard deviation describing the size (in bytes) of each read from or write to the FILEACCESS workload file (the default size is 4096 bytes)	N/A
THINKTIME	The mean and standard deviation (in milliseconds) describing the amount of idle time between processing cycles	0 to trial time

- requires no think time or additional CPU time
- reads data an average of 100 times between think times, with a standard deviation of 30 times
- uses sequential access and thus ignores file-peek time
- triggers no datapage activity
- repeatedly reads the function in the center of the codepage to simulate minimal codepage access

- reads records from the `workfile.dat` workload file
- reads records sequentially
- doesn't use system cache
- reads data in 2KB chunks
- reads only (doesn't write)

Unlike earlier versions of Response Probe, Response Probe 2.3 automatically generates and reads from an internal codepage file. The codepage file that Response Probe 2.3 uses contains a set of 1000 built-in function calls or actions that the tool can perform. This file simulates application code from which instructions are read during a process. You can tweak specific parameters after determining which parameter values best represent your application load.

You don't need to create Response Probe's script files wholly from scratch; you can modify the sample files in `\ntreskit\perftool\probe\examples`. Among these sample files are several sets of excellent baseline scripts that might suit many system designers' testing needs:

- `Diskmax.scr` determines a disk drive's maximum throughput by performing sequential, unbuffered reads of 64KB records from a 20MB file.
- `Minread.scr` measures how disk performance varies when reading one sector of the disk at a time; this sample file performs sequential, unbuffered 512-byte reads from a 20MB file.
- `Sizeread.bat` tests a disk configuration's performance in reading records of increasing size. This file is a batch file rather than a process script file. `Sizeread.bat` runs a series of tests of unbuffered, sequential reads from a 20MB file in which the size of the record read increases from 2KB to 8KB, 64KB, 256KB, 1MB, and 4MB.

Configuring the Server

Before you launch Response Probe, configure the server hardware as you'd configure it for production work, then configure any secondary applications and utilities (e.g., antivirus utilities, system-monitoring software) that will run concurrently with the primary application. Run only those programs and services whose activity level won't vary significantly over time. For example, a monitoring service such as NT's SNMP Agent adds to the demand on a server, but that demand is constant—it doesn't vary with the number of user connections or any other factor—so you can incorporate this service into the configuration. Conversely, you shouldn't include Microsoft SQL Server in a test because that application's impact varies.

Using Response Probe

Response Probe is installed on a server when you install the resource kit. Its installation has no parameters. The tool consists of the files in the `\ntreskit\perftool\probe` folder (including the `examples` subdirectory).

Running Response Probe is easy after you've defined the three script files. You run Response Probe from a command prompt. The syntax is

```
probe <ProcessFileName.scr>
<TrialTime>
<OutputFileName>
```

where *ProcessFileName.scr* is the name of the process script file that describes the threads you're testing, *TrialTime* is the length of time (in seconds) that the test will run, and *OutputFileName* is an optional parameter that creates an output (.out) file to which Response Probe will save test results (the default output file name is *ProcessFileName.out*). You can't direct the utility at another computer on the network; you must launch it from the system you want to test.

No Longer in the Dark

One of Response Probe's most important uses is determining just how much more powerful a new server will be than the one it's replacing. Manufacturers market server hardware with plenty of "performance figures," but how do increases in processor speed, disk-seek time, RAM capacity, processor bus size, and network-interface bandwidth really translate into overall performance? Many of these factors change with each new server model, so comparing servers based on the rated performance of their components becomes nearly impossible. Yet the most common question customers ask about a new server is, "How much faster is it going to run?"

I find Response Probe particularly valuable in answering this question because you can use it to create a true performance-based baseline. Run Response Probe on a 2-year-old application server that you're using now, and you'll have a relevant baseline for comparing candidates to replace it. Then, run Response Probe against the same software configuration on the newer servers, and you'll have a result that correlates directly with the results you saw on the old server, clearly showing the performance difference between the two systems. Other tools, such as Microsoft Performance Monitor, provide information only about a server's separate components. Response Probe gives a complete picture of the server's overall performance.

Chapter 5

Optimizing Windows Services

—by *Jordan Ayala*

Windows' services are central to the function of the OS and your loaded applications. Your system hosts dozens of services that control just about everything you can do with Windows. They enable actions from logging on to file-and-print functionality to supporting legacy networks. First, I give you a definition of services and what they do as well as tools and tips about how to manage them. I discuss the services available in Windows 2000 Server, Standard Edition without service packs applied (the other Win2K server products and Win2K Professional have different service options). This information sets the foundation for configuring and performance-tuning your Win2K services.

What Is a Service?

The Windows NT architecture has always used the services model (i.e., operational modules that encapsulate specific system functions). For example, Win2K's Indexing Service contains all the compiled code and interfaces necessary to index content on your server. Other applications and services can rely on and use the Indexing Service to provide even more functionality. A case in point is Microsoft SQL Server 2000's cross-store search function, which uses the Indexing Service to let you search for content across files and directories, on the Web, in SQL Server databases, and even in the Microsoft Exchange Server Web Store.

Isolating functions neatly into individual services makes the functions more manageable and easier to use. In addition, the isolation gives you fine-grained control over execution and availability of the OS or application features, helps you troubleshoot problems, and lets you easily access system information, such as network properties and performance counters.

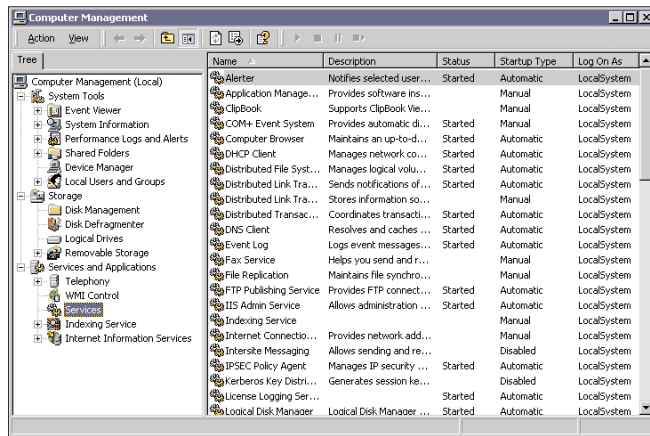
A service can be a core part of the OS's Service Controller, which is the services.exe file in the `\winnt\system32` directory. You can also find running services on the Processes tab in the Task Manager dialog box. A service can also be an application or the compartmentalized functionality of a server application such as SQL Server or Exchange. For example, installing Exchange loads a new selection of services that makes up the application's messaging and collaboration functionality. You can even set an individual end-user or desktop application to run as a system service.

Service Administration

Managing services on your Win2K Server system is fairly straightforward. Log on to your server as Administrator or to an account in the Administrators group that lets you change services' settings. The Microsoft Management Console (MMC) Computer Management snap-in gives you complete control over services' attributes and execution and lets you view services' status. To access the Computer Management snap-in, click Start, Services, Programs, Administrative Tools, or right-click the My Computer icon on your desktop and select Manage. In the Computer Management

window, which Figure 1 shows, expand the Services and Applications tree and click Services. The right pane will list all the installed services on your system, regardless of whether they're running.

Figure 1
Viewing system services



Right-clicking a service brings up a menu of tasks, which Table 1 defines. You can also use the icon that accompanies each listed service to start, stop, pause, or restart the service. Just select the service and click the appropriate icon.

Table 1
Services' Task Menu Selections

Task	Action
All Tasks	A submenu of the Pause, Restart, Resume, Start, and Stop tasks.
Help	Provides helpful information about the selected service.
Pause	Pauses a running service.
Properties	Brings up the Properties dialog box for the selected service.
Refresh	Refreshes the Services display pane.
Restart	Stops and restarts a service and any dependent services in one action.
Resume	Returns a paused service to its typical operational state.
Start	Starts a service that you stopped or that failed to start automatically at boot.
Stop	Stops a running service. (Stopping a service also stops a running application that depends on the stopped service, which can cause a crucial application fault. You should be very careful about stopping a service on a production server.)

By default, the Computer Management console displays five columns of information for each currently installed service: Name, Description, Status, Startup Type, and Log On As. However, you

can configure the Computer Management console by selecting Choose Columns from the View menu. At the resulting dialog box, you can select the data that you want the console to display.

To view a service's properties, right-click or double-click the service name and select Properties from the pop-up menu. Figure 2 shows the Properties dialog box for the Fax Service. This dialog box offers far more control than NT 4.0 provides. On the General tab, you can't change the service name, but you can change both the display name and the description, which might help you organize your services in the management interface for easier recognition and use. The General tab also displays the path to the service's executable file. The Fax Service is a standalone executable that the system runs as a service, whereas other system services, such as the Alerter service, are part of the core Windows functionality and the OS's Service Controller. From the *Startup type* drop-down list on the General tab, you can select whether the system service or application is loaded automatically or manually or is disabled. Table 2 outlines these options. To govern how the Fax Service operates, you can change the *Service status* parameter to Start, Stop, Pause, or Resume. In the *Start parameters* text box, you can specify start parameters that you want the system to pass to the service when it loads, such as file locations and control attributes.

Figure 2

A service's general properties window

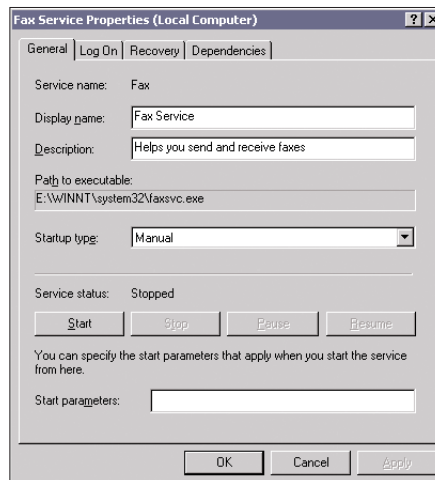


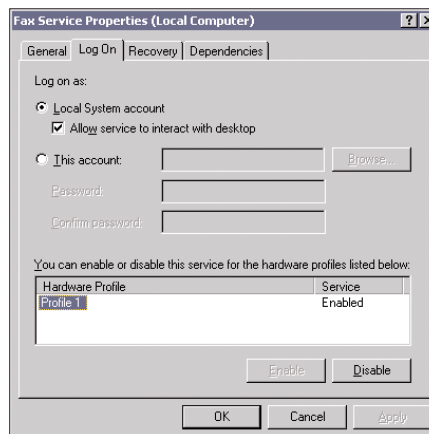
Table 2

Services' Startup Parameters

Parameter	Description
Automatic	The service is loaded and runs automatically when the server boots.
Disabled	The service isn't loaded when the server boots; dependent services that need to start and use the disabled service won't be able to do so.
Manual	The service isn't loaded when the server boots; you must manually launch the service from the Services container.

The Log On tab of the Properties dialog box, which Figure 3 shows, gives you control over a service's security attributes. On this tab, you can select which security attribute the service runs under, determining the scope of its operation in the system as well as that of any application that depends on the service. Thus, use caution when you configure this tab's settings. By default, services log on and run with the security attributes of the LocalSystem account (a unique system user account that's local to the server and doesn't provide logon or network access). LocalSystem is similar to the Administrator account in that LocalSystem runs under the authority of the OS. Therefore, the service has the access necessary to run properly. You can also set a service to run under a user account that you create for a specific purpose, such as centralizing the management of certain applications. (For example, you might create an Exchange Server user account that only Exchange Server services use and that's associated with specific user rights and policies.) On the Log On tab, you can also associate service-execution parameters with hardware profiles to control which applications and services run under various hardware configurations. For example, you might have a special maintenance-mode profile in which the server boots to a minimum configuration that has all peripheral devices, storage arrays, and the related system services disabled so that you can install or upgrade applications without interference.

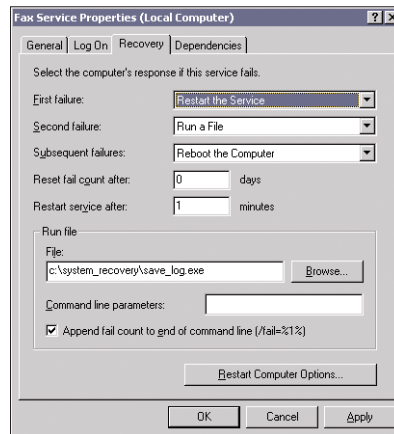
Figure 3
A service's logon settings



The Recovery tab provides new functionality in Win2K. If a service fails under NT 4.0, all you can do is look in the event logs for a clue about the cause. You need additional management tools to solve the problem from there. As Figure 4 shows, the Recovery tab provides a drop-down list of actions the system can take after a service's first failure, second failure, and subsequent failures. You can choose from four options: Take no action, Restart the Service, Run a File, or Reboot the Computer. Judiciously selecting actions makes your server self-maintaining to a certain extent. For example, if the Fax Service configured in Figure 4 stops for some reason, the server will first try to restart the service. On the second failure, the server will run a file containing an application

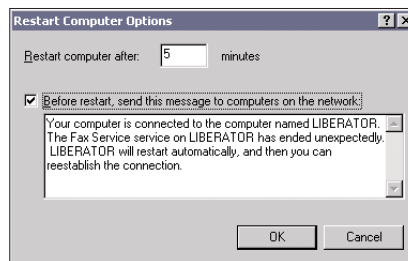
that sends a page alert to the systems administrator. If the service fails again, the system will reboot.

Figure 4
A service's recovery properties



The Recovery tab also gives you access to an attribute that resets the failure count so that the count doesn't get caught in a terminal loop. In addition, you can set the length of time the server will wait before it tries to restart the service. Clicking Restart Computer Options takes you to a dialog box, which Figure 5 shows, in which you can specify what will happen if an automatic system reboot becomes necessary.

Figure 5
Configuring the Restart Computer Options settings

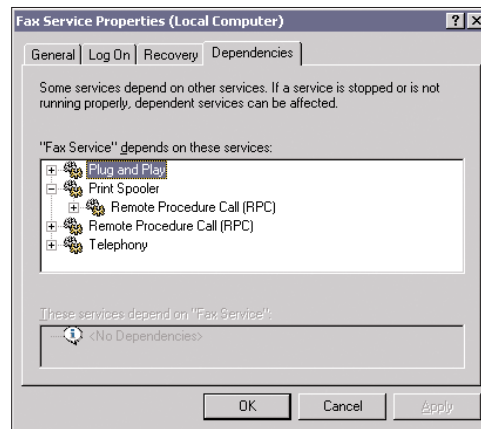


The Dependencies tab shows which services the selected service relies on for its operation and which services depend on the selected service. Figure 6 shows that the Plug and Play and Print Spooler services, among others, are necessary for the Fax Service to function, but that no other applications or services currently depend on the Fax Service.

Some systems management applications let you remotely control services (e.g., restart a failed application service over a dial-up connection). You can find this useful functionality in many pop-

ular systems management tools and out-of-band management applications as well as in the *Microsoft Windows 2000 Resource Kit*.

Figure 6
A service's dependencies



You can also manage services from a command prompt. Use the Net Start and Net Stop commands to pass parameters to a service and control its execution. Use the command prompt to create system startup and shutdown scripts that let you simultaneously control many services, or to create batch commands that restart multiple services after an error.

Taking Action

Why would you want to look up or modify a service's settings? Many situations require you to investigate the status of the services on your system and take action, as the following examples show.

An Application or Service Has a Problem

If you receive an event-log warning or a system error or if an application isn't functioning properly, you'll need to navigate to the services management tool that I discussed in the previous section.

You Need to Cause Configuration Changes to Take Effect

Changing a service's properties or server network settings sometimes requires you to reboot the system or at least restart a service. You can use the services management tool's tasks to manually stop and restart the required services, causing the changes to take effect without rebooting the system.

You Need to Troubleshoot

If your server is having difficulties, a useful method of tracking the cause is to shut down services one by one until you discover which service is causing the problem. Or, you can shut down all application services until you can diagnose the one you know to be problematic.

You Want to Change the Way the System Behaves

If you want to control access to the server by shutting down network authentication services, change service security settings, or turn applications on and off, you can do so from the services management tool.

You Need to Shut Down Services to Install Applications or System Updates

Sometimes you need to use the services management tool to shut down services and install various applications or system updates. For example, you must terminate all services that use ODBC before you can install the Microsoft Data Access Components (MDAC) update.

You Need to Install New Services or Applications

Before installation, some applications, services, and system updates require you to stop a server application. For example, before you can apply an MDAC update, you must shut down all ODBC-dependent services.

Installing and Uninstalling Services

If you installed and then removed applications from pre-Win2K Windows installations, the uninstallation process invariably left little program gremlins on your system. In Win2K, Microsoft has significantly improved this situation by designing the OS to maintain much tighter control over application components and their code. The result is that you can use Win2K's native packaging and development tools to minimize the instances of nonfunctional services remaining in your configuration.

In many cases, applications manage their components by themselves, configuring or creating new services as needed. When you use an application's original installer tool to remove the application, the tool can usually clean up and remove any remaining pieces. However, if you've changed the application's configuration since you installed the tool, you might have to manually change the configuration back to the original for the installer tool to successfully remove the application.

Win2K uses the Control Panel Add/Remove Programs applet to add and remove application, network, and data services. However, I strongly recommend that you don't try to manually remove a service by tracking down its code in the `\winnt` directory and deleting it. Doing so can have unpredictable but negative effects on your server's operation. Alternatively, the resource kit provides an array of tools that you can use for service-management tasks. Table 3 provides a list of resource kit tools for managing system and application services.

If you prefer to manage from a command prompt, the resource kit offers command-line versions of service-management tools that let you install and uninstall services' executable files, but only on the local machine. In addition, these tools require you to use the Service Controller (`sc.exe`) tool to start and set services' attributes. The Service Installation Wizard is a GUI tool that lets you install, start, and set services' attributes from one interface, even on remote servers. However, you should use this tool only with applications and services that don't have an install/uninstall program.

The resource kit's `Srvany` tool lets you run an end-user or other Windows application as a system service. (Some applications install as system services, but you have to force others to run this way.) Running Windows applications as system services lets you set an application to load automatically when the server boots, without requiring a user to log on. (However, you should test this configuration on a nonproduction system first—some applications don't function properly

when forced to run as services.) The application runs in the background regardless of the security attributes of the current user. The resource kit includes instructions for how to install and use Srvany. For more information about Win2K's resource kit tools, invest in the resource kit and pay close attention to the Win2K core Help files and the resource kit's documentation.

Table 3
Resource Kit Tools for Managing System and Application Services

Tool	Full Name	Description	Function
Delsrv.exe*	Delete Service	Unregisters a service through the Windows Services Control Manager.	Manually removes parts of a service that the service's uninstallation program as left running on the system.
Instrsv.exe*	Service Installer	Installs and uninstalls services.	Gives a specific name to a new program service and installs it with the OS (e.g., creates an instance of Srvany to run an application in the background).
Netsvc.exe*	Command-line Service Controller	Lists, queries the status of, and controls the execution of services.	Locally and remotely controls services and retrieves status information.
Sc.exe*	Service Controller Tool	Interacts directly with the Windows Service Controller, giving you full access to services information through a command prompt.	Starts and stops services, tests problematic services locally or remotely, debugs newly installed services, creates and deletes services, alters properties, and queries services' status.
Sclist.exe*	Service List	Lists running, stopped, or all services on a local or a remote system.	Gives a quick status update about all system services, rather than individual services.
Srvany.exe	Applications as Services Utility	Runs any Windows application as a background service.	Loads applications automatically at boot and runs them unattended.
Srvinstw.exe	Service Installation Wizard	Installs and deletes services and device drivers.	Adds a new service or device driver either locally or remotely.
Svcaccls.exe*	Service ACL Editor	Delegates services control through ACLs.	Grants, sets, revokes, and denies access to specific services.
Svcmon.exe	Service Monitoring Tool (SMT)	Monitors services locally or remotely and notifies you of any changes.	Polls for any changes to running services on your server, and uses SMTP or Exchange to notify you about stop and start events.
Smconfig.exe	SMT Configuration Wizard	Configures Svcmon to notify you of changes in services' status.	Configures which services Svcmon monitors and whom the tool notifies. (Svcmon.exe must be in your %systemroot%\system32 directory.)

* A command-line utility

Evaluating and Tuning Services

By default, Win2K Server, Standard Edition (without service packs applied) installs 65 services. (The other Win2K Server products and Win2K Pro install different services. For descriptions of the 65 default services that Win2K Server, Standard Edition installs, see Table 4. In the previous sections, I provide a definition of those services and what they do as well as tools and tips for how

to manage them. With that foundation, you can begin to evaluate the services running on your system and tune them to your ideal configuration.

Table 4
Default Windows 2000 Server Services

Service	Description	Startup Type	Logon Account
Alerter	Notifies selected users and computers of administrative alerts.	Automatic	LocalSystem
Application Management	Provides software installation services such as Assign, Publish, and Remove.	Manual	LocalSystem
ClipBook	Supports ClipBook Viewer, which lets remote ClipBooks see pages.	Manual	LocalSystem
COM+ Event System	Provides automatic distribution of events to subscribing COM components.	Manual	LocalSystem
Computer Browser	Maintains an up-to-date list of computers on your network and supplies the list to programs that request it.	Automatic	LocalSystem
DHCP Client	Manages network configuration by registering and updating IP addresses and DNS names.	Automatic	LocalSystem
Distributed File System	Manages logical volumes distributed across a LAN or WAN.	Automatic	LocalSystem
Distributed Link Tracking Client	Sends notifications of files moving between NTFS volumes in a network domain.	Automatic	LocalSystem
Distributed Link Tracking Server	Stores information so that files move between volumes for each volume in the domain.	Manual	LocalSystem
Distributed Transaction Coordinator	Coordinates transactions that are distributed across two or more databases, message queues, file systems, or other transaction-protected resource managers.	Automatic	LocalSystem
DNS Client	Resolves and caches DNS names.	Automatic	LocalSystem
Event Log	Logs event messages issued that programs and Windows issue. Event Log reports contain useful diagnostic information that you can view in Event Viewer.	Automatic	LocalSystem
Fax Service	Helps you send and receive faxes.	Manual	LocalSystem
File Replication	Maintains file synchronization of file directory contents among multiple servers.	Manual	LocalSystem
FTP Publishing Service	Provides FTP connectivity and administration through the Microsoft Management Console (MMC) Internet Information Services snap-in.	Automatic	LocalSystem
IIS Admin Service	Allows administration of Web and FTP services through the Internet Information Services snap-in.	Automatic	LocalSystem
Indexing Service	Indexes contents and properties of files on local and remote computers; provides rapid access to files through flexible querying language.	Manual	LocalSystem
Internet Connection Sharing (ICS)	Provides Network Address Translation (NAT) addressing and name-resolution services for all computers on your home network through a dial-up connection.	Manual	LocalSystem

Continued on page 44

Table 4 continued
Default Windows 2000 Server Services

Service	Description	Startup Type	Logon Account
Intersite Messaging	Lets users send and receive messages between Windows Advanced Server sites.	Disabled	LocalSystem
IP Security (IPSec) Policy Agent	Manages IP security policy and starts the Internet Security Association and Key Management Protocol (ISAKMP)/Oakley Internet Key Exchange (IKE) and the IP security driver.	Automatic	LocalSystem
Kerberos Key Distribution Center	Generates session keys and grants service tickets for mutual client/server authentication.	Disabled	LocalSystem
License Logging Service	Tracks concurrent user connections against those licensed.	Automatic	LocalSystem
Logical Disk Manager	Logical Disk Manager Watchdog Service.	Automatic	LocalSystem
Logical Disk Manager Administrative Service	Provides administrative service for disk-management requests.	Manual	LocalSystem
Messenger	Sends and receives messages transmitted by administrators or by the Alerter service.	Automatic	LocalSystem
Net Logon	Supports pass-through authentication of account logon events for computers in a domain.	Manual	LocalSystem
NetMeeting Remote Desktop Sharing	Lets authorized people use NetMeeting to remotely access your Windows desktop.	Manual	LocalSystem
Network Connections	Manages objects in the Network and Dial-Up Connections folder, in which you can view both LAN and WAN connections.	Manual	LocalSystem
Network DDE	Provides network transport and security for Dynamic Data Exchange (DDE).	Manual	LocalSystem
Network DDE DSDM	Manages shared DDE and is used by network DDE.	Manual	LocalSystem
NTLM Security Support Provider	Provides security to remote procedure call (RPC) programs that use transports other than named pipes.	Manual	LocalSystem
Performance Logs and Alerts	Configures performance logs and alerts.	Manual	LocalSystem
Plug and Play	Manages device installation and configuration and notifies programs of device changes.	Automatic	LocalSystem

Continued on page 45

Table 4 continued
Default Windows 2000 Server Services

Service	Description	Startup Type	Logon Account
Print Spooler	Loads files to memory for later printing.	Automatic	LocalSystem
Protected Storage	Provides protected storage for sensitive data, such as private keys, to prevent access by unauthorized services, processes, or users.	Automatic	LocalSystem
Quality of Service (QoS) RSVP	Provides network signaling and local traffic control setup functionality for QoS-aware programs and control applets.	Manual	LocalSystem
Remote Access Auto Connection Manager	Creates a connection to a remote network whenever a program references a remote DNS or NetBIOS name or address.	Manual	LocalSystem
Remote Access Connection Manager	Creates a network connection.	Manual	LocalSystem
RPC	Provides the endpoint mapper and other miscellaneous RPC services.	Automatic	LocalSystem
RPC Locator	Manages the RPC name service database.	Manual	LocalSystem
Remote Registry Service	Lets you remotely manipulate the registry.	Automatic	LocalSystem
Removable Storage	Manages removable media, drives, and libraries.	Automatic	LocalSystem
RRAS	Offers routing services to businesses in LAN and WAN environments.	Disabled	LocalSystem
RunAs Service	Lets you start processes under alternative credentials.	Automatic	LocalSystem
SAM	Stores security information for local user accounts.	Automatic	LocalSystem
Server	Provides RPC support and file, print, and named-pipe sharing.	Automatic	LocalSystem
SMTP	Transports email across the network.	Automatic	LocalSystem
Smart Card	Manages and controls access to a smart card inserted into a smart card reader attached to the computer.	Manual	LocalSystem
Smart Card Helper	Provides support for legacy smart card readers attached to the computer.	Manual	LocalSystem
SNMP Service	Includes agents that monitor the activity in network devices and report to the network console workstation.	Automatic	LocalSystem
SNMP Trap Service	Receives trap messages that local or remote SNMP agents generate and forwards the messages to SNMP management programs running on this computer.	Manual	LocalSystem
System Event Notification	Tracks system events such as Windows logon, network, and power events. Notifies COM+ Event System subscribers of these events.	Automatic	LocalSystem
Task Scheduler	Lets a program run at a designated time. Automatic LocalSystem	Automatic	LocalSystem
TCP/IP NetBIOS Helper Service	Enables support for NetBIOS over TCP/IP (NetBT) service and NetBIOS name resolution.		

Continued on page 46

Table 4 continued
Default Windows 2000 Server Services

Service	Description	Startup Type	Logon Account
Telephony	Provides Telephony API (TAPI) support for programs that control telephony devices, IP-based voice connections on the local computer, and, through the LAN, on servers that are also running the Telephony service.	Manual	LocalSystem
Telnet	Lets a remote user log on to the system and run console programs by using the command line.	Manual	LocalSystem
Terminal Services	Provides a multisession environment that lets client devices access a virtual Windows 2000 Professional desktop session and Windows-based programs running on the server.	Disabled	LocalSystem
UPS	Manages a UPS connected to the computer.	Manual	LocalSystem
Utility Manager	Starts and configures accessibility tools from one window.	Manual	LocalSystem
Windows Installer	Installs, repairs, and removes software according to instructions contained in .msi files.	Manual	LocalSystem
Windows Management Instrumentation (WMI)	Provides system management information.	Manual	LocalSystem
WMI Driver Extensions	Provides systems management information to and from drivers.	Manual	LocalSystem
Windows Time	Sets the computer clock.	Manual	LocalSystem
Workstation	Provides network connections and communications.	Automatic	LocalSystem
WWW Publishing Service	Provides Web connectivity and administration through the Internet Information Services snap-in.	Automatic	LocalSystem

What Installs Which Services?

To see which services Win2K Server installs by default, I started with a clean Win2K Server installation and accepted all the default settings (except that I opted to install the management and monitoring tools, which Win2K Server doesn't install by default). Next, I ran the Active Directory Installation Wizard (dcpromo.exe) and accepted all the default settings. Using the wizard, I made the server the first domain controller (DC) in the new domain homedomain.com, and I installed DNS locally. The Active Directory (AD) installation process installed only one new service, the DNS Server service, which answers DNS name queries and update requests.

Although the AD installation added only one new service, the installation changed the status of some of the Win2K Server default services from manual or disabled to automatic. Table 5 shows the services that AD requires but that don't run in a default standalone server configuration unless you manually turn them on.

Table 5
Services that Change Status After AD Installation

Service	Startup Type	New Startup Type
Distributed Link Tracking Server	Manual	Automatic
File Replication	Manual	Automatic
Intersite Messaging	Disabled	Automatic
Kerberos Key Distribution Center	Disabled	Automatic
Net Logon	Manual	Automatic
NTLM Security Support Provider	Manual	Manual
RPC Locator	Manual	Automatic
Telephony	Manual	Manual
Windows Installer	Manual	Manual
Windows Management Instrumentation (WMI)	Manual	Automatic
Windows Time	Manual	Automatic

Finally, using the Control Panel Add/Remove Programs applet, I installed every possible native Windows service and accepted all the default configuration parameters. (Under most circumstances, I would never take this step on a production server. I did so in this case simply to research the services and their options.) This installation added 24 services to my system and changed the Startup Type parameter of the already installed Win2K Server Terminal Services from Disabled to Automatic. Table 6 lists and describes the 24 services that this step added.

Table 6
Optional Win2K Services

Service	Description	Status	Startup Type	Logon Account
Boot Information Negotiation Layer	Lets you install Win2K Pro on Preboot Execution Environment (PXE) remote boot-enabled client computers.	Not started	Manual	Local System
Certificate	Issues and revokes X.509 certificates for public key-based cryptography technologies.	Started	Automatic	Local System
DHCP Server	Provides dynamic IP address assignment and network configuration for DHCP clients.	Started	Automatic	Local System
File Server for Macintosh	Lets Macintosh users store and access files on the local server.	Started	Automatic	Local System

Continued on page 48

Table 6 continued
Optional Win2K Services

Service	Description	Status	Startup Type	Logon Account
Internet Authentication Service (IAS)	Enables authentication, authorization, and accounting of dial-up and VPN users. IAS supports the Remote Authentication Dial-In User Service (RADIUS) protocol.	Started	Automatic	Local System
Message Queuing	Provides a communications infrastructure for distributed asynchronous messaging applications.	Started	Automatic	Local System
Network News Transfer Protocol (NNTP)	Transports network news across the network.	Started	Automatic	Local System
Online Presentation Broadcast	No description available.	Not started	Manual	Local System
Print Server for Macintosh	Lets Macintosh users send print jobs to a spooler on a Win2K server.	Started	Automatic	Local System
Remote Storage Engine	Coordinates the services and administrative tools used for storing infrequently used data.	Started	Automatic	Local System
Remote Storage File	Manages operations on remotely stored files.	Started	Automatic	Local System
Remote Storage Media	Controls the media that stores remote data.	Started	Automatic	Local System
Remote Storage Notification	Notifies the client about recalled data.	Not started	Manual	Local System
Simple TCP/IP Services	Supports the following TCP/IP services: Character Generator, Daytime, Discard, Echo, and Quote of the Day.	Started	Automatic	Local System
Single Instance Storage Groveler	Scans Single Instance Storage volumes for duplicate files, and points duplicate files to one data storage point, conserving disk space.	Not started	Manual	Local System
Site Server Internet Locator Service (ILS)	Enables IP multicast for network conferencing.	Started	Automatic	Local System

Continued on page 49

Table 6 continued
Optional Win2K Services

Service	Description	Status	Startup Type	Logon Account
TCP/IP Print Server	Provides a TCP/IP-based printing service that uses the Line Printer protocol.	Started	Automatic	Local System
Terminal Services Licensing	Installs a license server and provides registered client licenses when connecting to a terminal server.	Started	Automatic	Local System
Trivial FTP Daemon	Implements the Trivial FTP Internet standard, which doesn't require a username or password. Part of Remote Installation Services (RIS).	Not started	Manual	Local System
Windows Media Monitor	Monitors client and server connections to the Windows Media services.	Started	Automatic	.\NetShowServices
Windows Media Program	Groups Windows Media streams into a sequential program for the Windows Media Station service.	Started	Automatic	.\NetShowServices
Windows Media Station	Provides multicasting and distribution services for streaming Windows Media content.	Started	Automatic	.\NetShowServices
Windows Media Unicast	Provides Windows Media streaming content on demand to networked clients.	Started	Automatic	.\NetShowServices
WINS	Provides a NetBIOS name service for TCP/IP clients that must register and resolve NetBIOS-type names.	Started	Automatic	Local System

What Can You Afford to Lose?

With 90 services running on your Win2K Server system, won't all that code bring your server to its knees? The answer depends on the server's horsepower. Most of these services don't drain system resources unless they're active. For example, if you don't maintain an active Web site on your server, having Microsoft IIS installed and running won't significantly slow your system's performance.

By default, many services are disabled or set to manual start, but the more services your server loads automatically, the more memory and CPU resources it uses during typical operation. Therefore, if fewer services are running, more resources are available to the system, and the system will run faster. Thus, to improve performance, you should enable applications to load automatically

only when necessary and disable or remove (or set to manual start) the other services on your server.

However, be very careful about which services you disable or remove. A good rule of thumb is that if you don't know what it does, don't disable or remove it. Turning off a necessary or dependent service can crash an application, corrupt files, or cause your system to fail. Whether you can safely disable or remove a service depends on your server's configuration, but Table 7 shows services you might be able to turn off to boost performance (provided you've verified that the system or other applications aren't using the services). To properly remove a service, use the Add/Remove Programs applet. Click Add/Remove Windows Components to launch the Windows Components Wizard, which presents a list of available Win2K services. Currently installed services appear with selected check boxes. To remove a service, clear the service's check box; to modify a service, select its check box, then click Next to step through configuration for the services you selected (some services include multiple components). Be sure to clear a check box only if you want to remove that service.

Table 7
Services You Might Disable or Remove

Service	Considerations
Alerter	Disable only if you don't need the ability to send messages, such as <i>Shut down now</i> , to users.
DHCP Client	Disable only if you're statically assigning IP addresses.
Distributed File System	Disable only if you aren't using DFS volumes.
DNS Client	Disable only in a development or test environment.
IISAdmin	Disable only if you aren't running a Web server. However, be aware that many Win2K components are Web based, and disabling this service might affect those components.
Messenger	Disabling this service might affect applications that need to send messages between systems or other applications.
Print Spooler	Disable only if the system isn't a print server.
Remote Registry	Disabling this service might protect your server from attack.
RunAs	Disable only if you don't need the ability to use the Run As command to start an application under a different user security context.
SMTP	Disable only if you don't need SMTP.
SNMP	Disable only if you aren't running any SNMP-based management applications. However, most management applications use SNMP.

Should you turn on any services that don't run by default? The answer depends on your situation. For example, you might want to enable the Indexing service, but this service slows server performance every time it indexes the server's content. If you need fax capability or RRAS functionality, you should turn on those services. Table 8 lists useful system services that you might want to enable.

Table 8
Useful System Services to Enable

Service	Reason to Enable
Net Logon	Enable only if this server will support user logons.
NetMeeting Remote Desktop Sharing	Useful for supporting remote Help desk activities.
RRAS	Lets you support dial-in and Internet logons directly.
SNMP Trap	Necessary when running management applications that use SNMP.
Telnet	Useful for server access in a mixed Windows and UNIX environment.
Windows Time	Lets other computers on the network sync their clocks to this server.

When tuning your system's services, perform a full backup before you significantly alter your server's configuration and to log configuration changes. Backups and logs are your primary vehicles for troubleshooting problems if a configuration change results in a broken application or performance degradation.

Security Tune-Up

Disabling security-related services on any server—but especially on a DC—sacrifices the system's protection and endangers your network environment. However, you can tune service settings to ease systems management.

I discussed how to create service accounts for applications and services. These accounts control the security context under which the applications and services run, help you control the access rights and interactivity of multiple related services, and secure the system's core management and application functions.

Using Win2K's native security object model, you can control access to individual server properties and actions. So, for example, you can control which services your Help desk technicians can access, what actions they can take, and even what management information they can view. By setting ACLs on individual services, you can delegate control and access rights to those services. Alternatively, you can use Microsoft BackOffice Server 2000 to determine, through logon credentials and locked-down MMC files, what a technician has permission to do. For example, you can customize a context menu to display only Start Service (and not Stop). The *Microsoft Windows 2000 Resource Kit* Service ACL Editor tool also lets you administer services at a granular level.

You can set logon credentials for services, enter passwords, and set interaction with the desktop through the Log On tab of a service's Properties window. Through the logon account, you can determine which rights a service or application will have on your server. Thus, for services that are potential security risks, you can limit access to server resources. You can create a unique user account and manually assign the account to the groups that contain the permissions necessary to work with that service. When you do so, create the user account in the Local User and Groups container. (If your system is a DC, create a unique domain account rather than a local or system account.) Make sure that you limit the account's functional scope as much as possible (e.g., provide limited logon rights and no general server access unless the service requires it). Setting up service-management accounts that have different names and strong passwords will make cracking your network more difficult.

However, creating a multitude of service accounts can result in a hassle when you must change accounts' passwords (according to your company's password policies). One option is to set these accounts' passwords to never expire. This setting protects you from finding yourself with a dead server if a password times out and prevents the associated service from logging on and running. But this setting is also a security risk. Rather than create many accounts with passwords that don't expire, you can create a few, nonprivileged service accounts and develop a process for changing their passwords as needed.

Desktop interaction for a service means that the service can bring itself up in the Windows desktop environment for input by anyone logged on to the system. Selecting the *Allow service to interact with desktop* check box in the service's Properties window exposes the service's UI so that users can change the service's settings. Leaving this check box clear prevents logged-on users from interfering with the service. This configuration option is available only when a service is running under the Local System account. Usually, you wouldn't change the interaction settings of common Windows components and services because doing so could have detrimental effects on your server's operation. However, in a development environment or if you're running an application as a service, permitting desktop interaction might be necessary to control a service or to provide user-input settings.

What if you mess up? You mistakenly set the Server service to log on under a user account with an expired password. Now, you find that you can't log on to your system. Don't panic. Reboot the server into Safe Mode, which is a minimal service and driver configuration. Through one of the various Safe Mode startup options, you can get back into Windows and fix your error.

Tune Up or Tune Out

You've learned your way around services' administration tools and interfaces, and now you know how to apply that knowledge through enabling and disabling services and tweaking services' security-related settings. You can use this chapter as a Win2K services primer to ease service management, and you can consult Windows Help and the resource kit documentation for more information about tuning your system's services.

Chapter 6

Measuring and Managing Windows NT Workstation 4.0 Application Performance

—by Darren Mar-Elia

When you want to maximize Windows NT Workstation 4.0 performance, my immediate thought is that you simply strip off the explorer shell, use a command-prompt interface, and don't run those pesky GUI applications. NT Workstation will rip. I call this approach the “nix mentality,” which is common among UNIX followers, who believe, in some cases correctly, that graphical applications slow you down. However, no matter how fast the NT kernel is, NT Workstation operates in a graphical environment and runs graphical applications. In most cases, you can't disable the Windows Explorer shell without crippling the system's functionality. Given that reality, it's time to take a fresh look at how you can measure and manage your NT Workstation applications' performance to get the most bang for your buck. You can use Performance Monitor counters to identify problem applications, which is a good starting point and one that I find consistently useful. You can also use some *Microsoft Windows NT Workstation 4.0 Resource Kit* utilities that help you measure and watch for performance problems.

What's Important in Performance Monitor

Performance Monitor is a great tool for measuring NT Workstation 4.0 or Server performance, and you can find a lot of information about using the program to gather metrics on NT computers. However, I want to focus on the Performance Monitor features that measure NT Workstation application performance. I spend most of my time working with two performance objects—process and memory. The process performance object returns metrics that are related to all running processes, whether system processes, user applications, or NT services. The memory object returns metrics that are related to NT's memory management subsystem elements, including file cache, physical memory, and several paged and nonpaged pools that NT uses for system processes.

When you're considering NT Workstation performance, you might want to consider the system's disk subsystems performance. However, I'm not going to focus too much attention in this area. (For information about disk subsystems, see Chapter 3, “Tuning NT Server Disk Subsystems.”) I'm interested in getting to the problem's source, and I want to see how my application is using the system's memory and processor resources and how that usage affects my overall system performance.

To a degree, disk problems, such as thrashing, are symptoms of other problems within an application. A pagefile that grows excessively might present problems on a slow disk subsystem or highly fragmented volume, but you need to know why the pagefile is growing in the first place.

Table 1 lists objects and counters, which are good starting points for monitoring application performance, and briefly describes the value each feature provides. If you use these counters to create a Performance Monitor workspace (.pmw) file, you can quickly load the files whenever you need to monitor application performance. However, when you're using workspace files, they embed the name of the workstation or server on which you've configured them into the .pmw file. You'll need to edit the name after you load the new workspace file on a different computer.

Table 1
Objects and Counters

Object: Counter	What the Object: Counter Measures	What the Object: Counter Helps You Do
Process: Working Set	The amount of physical RAM a process is consuming.	Monitor an application's memory usage over time and detect memory leaks.
Process: Pagefile Bytes	The amount of bytes this process uses in the pagefile.	Monitor an application's total memory usage over time.
Memory: Committed Bytes	The total amount of committed virtual memory all user processes on the system are using at a given time.	Determine when the pagefile will grow (when you compare this value against Commit Limit).
Memory: Commit Limit	A calculated value that determines how much virtual memory the system can commit before NT Workstation needs to grow the pagefile size.	Know how the current pagefile size matches your system's memory needs. You use this value to calculate % Committed Bytes In Use.
Memory: % Committed Bytes In Use	The fraction of Committed Bytes to the Commit Limit.	Monitor when a workstation will start growing the pagefile.
Process: % Processor Time	Current processor utilization for the selected process.	Pinpoint applications with high CPU utilization.

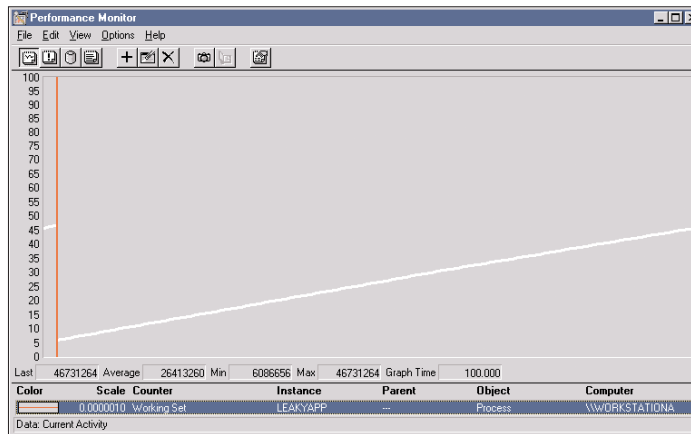
Monitoring for Memory Leaks

The first two counters that Table 1 lists, *Process: Working Set* and *Process: Pagefile Bytes*, let me monitor my application's memory consumption footprint. The working set is an important metric for application performance because it tells you how much physical memory (i.e., actual pages in RAM) an application is consuming. You can monitor the working set over time to detect memory leaks in applications. If you see a steady increase in the working set, as Figure 1 shows, the application might not be properly releasing previously allocated memory. However, you need to know the application to understand how it's supposed to behave. For example, if I leave Microsoft Word running but inactive on my desktop and Word's working set steadily increases over time, I can be pretty sure that Word has some kind of memory leak. However, if I have a data acquisition software program that might be collecting data into larger and larger arrays as it runs, then that software's working set might increase, which is typical behavior (although perhaps not desirable).

Process: Pagefile Bytes tracks an application's working set pretty closely as the application's memory consumption increases. For example, if you use the working set to monitor an application

that leaks over time, its Pagefile Bytes counter will follow the working set in a nearly linear fashion.

Figure 1
Monitoring a working set over time



Committed Bytes and the Pagefile

The Committed Bytes, Commit Limit, and % Committed Bytes In Use counters are handy tools you can use to determine the systemwide memory pressures on NT Workstation. Before I discuss these counters, you need the background information about the pagefile.

Microsoft recommends that you use a minimum pagefile size of physical RAM plus approximately 12MB. However, you can optimize this number as needed for your system's real memory requirements. If you're curious about the maximum pagefile size, forget it! In every NT version that I've measured the maximum pagefile size on, including NT 4.0 Service Pack 6 (SP6), NT ignores whatever value you enter. NT will increase the pagefile to meet increasing memory pressures until the OS runs out of disk space. To test how NT responds to increasing memory needs, enter a maximum pagefile size value in the Control Panel System applet's Performance tab. In the resource kit, look for the leakyapp.exe tool in the \perftool\meastool directory. Microsoft designed My Leaky App to test your system's behavior as it continues to allocate memory. My Leaky App grows over time, consuming more and more system memory. You start the application and select the Start Leaking button to begin the process. My Leaky App shows current pagefile usage and lets you stop and start the leaking process, as Figure 2 shows. If you let My Leaky App run long enough, it will start to increase the pagefile size and will continue to increase the size well past the limit you've specified in the NT Performance tab. After the pagefile size increases beyond the initial minimum value you've specified, you need to reboot to shrink the pagefile back to its original size.

Figure 2
Using My Leaky App to display pagefile usage



When NT starts increasing the pagefile to accommodate memory pressures, performance deteriorates rapidly, especially if the pagefile grows on a slow or badly fragmented disk partition. You can use Committed Bytes and Commit Limit metrics to determine when memory pressure on the system is causing the abrupt pagefile growth. NT calculates the Commit Limit as roughly equal to the sum of the system's installed physical memory plus the minimum pagefile size that the user specified in the NT Performance tab. Committed Bytes is the running processes' total commit charge. As Committed Bytes grows, it approaches the Commit Limit; you can reach the limit when one or more applications increasingly allocate more memory. When you monitor % Committed Bytes In Use, you'll see that as this metric approaches 100 percent, the pagefile will begin to grow to meet increasing memory demands. To try to keep up with memory demands, NT will increase the pagefile until no more disk space is available. You'll also see the message *Out of Virtual Memory*, which Figure 3 shows. If you receive this message, run Performance Monitor. Select the Process object, working set, and Pagefile Bytes counter, then select all running applications. You'll see fairly quickly whether one application is responsible for the precipitous growth in memory demands. You can also use the % Committed Bytes In Use metric to tune your pagefile's size. If you monitor this metric over time, you can adjust your minimum pagefile size to meet the needs of your particular set of applications.

Figure 3
Viewing the Out of Virtual Memory message



Processor Utilization

Process: % Processor Time measures how much processor time an application is using, which is important for determining system bottlenecks. However, you need to be careful when you use Performance Monitor to look at this metric. For example, certain applications might introduce loops in their processing, which can happen when they're waiting on a particular event. These loops can show up as 100 percent processor utilization, which doesn't necessarily mean that the workstation can't process anything else. In most cases, these loops are low priority and will con-

cede processor cycles to other applications that start up and request processing. Earlier Netscape Browser versions introduced loops that showed 100 percent utilization, and you couldn't tell whether Netscape was a CPU hog or was simply waiting on a certain event. Of course, if excessive disk activity, memory utilization, and overall system slowdown accompany 100 percent processor utilization on an application, then you might have just found a bug in that application. The resource kit's CPU Stress tool lets you artificially load a processor to get an idea of how the system will behave under heavy processor load. You can use this tool to adjust thread priorities and the activity level for four threads, control how much load applications place on the CPU, and determine how crucial a thread is (i.e., you can see which low-priority threads cede control to higher-priority ones).

Resource Kit Utilities for Performance Management

The resource kit includes several handy utilities in addition to My Leaky App and CPU Stress for managing your NT computers' performance. You'll find most of these tools in the resource kit's Perftool folder. For a list of some interesting tools you can use to manage and monitor NT performance, see the sidebar "Performance Management Utilities."

The key to managing NT Workstation performance is to be familiar with your applications and how they use NT's resources. The techniques I've described are a first step toward meeting that goal. After you thoroughly understand the Performance Monitor metrics, I encourage you to take a look at the resource kit's Response Probe utility. This tool lets you take a proactive approach to designing high-performance applications. You can create artificial workloads that let you simulate a user's likely stresses on a system. After all, optimizing performance for an application that is running alone is easy. The real fun begins when you must contend with 20 applications, various services, and desktop utilities that might be running at the same time.

Performance Management Utilities

—by *Darren Mar-Elia*

The *Microsoft Windows NT Workstation 4.0 Resource Kit* includes several tools for managing NT Workstation performance. You'll find the following tools in the resource kit's Perftool folder.

- **\cntrtool\counters.hlp.** This Help file lists and explains the NT default Performance Monitor counters.
- **\logtools\typeperf.exe.** This command-line version of Performance Monitor dumps metrics to Comma Separated Values (CSVs), as Figure A shows. Typeperf expects parameters that refer to Performance Monitor objects and counters and returns current values. In Screen A, I've specified to monitor Memory: Committed Bytes and the Microsoft Internet Explorer (IE) process working set at 1-second

Continued on page 58

Performance Management Utilities *continued*

intervals. I can also use Typeperf to monitor performance objects on remote machines if I specify a Universal Naming Convention (UNC) path (e.g., “\\machine\memory\committed bytes”) to precede the object.

- **\\meastool\empty.exe.** This tool empties a running application’s working set. You use this tool to release currently allocated physical memory for an application.
- **\\meastool\ntimer.exe.** This tool shows how long a particular application takes to run, including what percentage of the time it was in privileged mode vs. user mode.
- **\\meastool\pview.exe.** This Process Explode tool shows detailed information per process about how a system is allocating memory. You can also use this tool to see the security token that the system has associated with a given process and threads.
- **\\meastool\top.exe.** This command-line tool constantly displays the top processes by CPU usage.
- **\\meastool\wperf.exe.** This graphical utility displays a variety of metrics about system memory and processor usage. This tool is similar to the UNIX Perfometer utility.

Figure A

Viewing Performance Monitor’s \logtools\typeperf.exe utility

```

C:\WINNT\System32\cmd.exe - C:\NTRESKIT\PerfTool\LogTools\typeperf.exe 1 "memory\committ...
G:\>C:\NTRESKIT\PerfTool\LogTools\typeperf.exe 1 "memory\committed bytes" "process(IEXPLORE)\working set"

"Sample Time" "memory\committed bytes" "process(IEXPLORE)\working set"
"12/17/1999 13:09:38.280" "56045568" "6815744"
"12/17/1999 13:09:39.282" "56045568" "6815744"
"12/17/1999 13:09:40.283" "56045568" "6815744"
"12/17/1999 13:09:41.285" "56045568" "6815744"
"12/17/1999 13:09:42.286" "56045568" "6815744"
"12/17/1999 13:09:43.288" "56045568" "6815744"
"12/17/1999 13:09:44.289" "56045568" "6815744"
"12/17/1999 13:09:45.290" "56045568" "6815744"
"12/17/1999 13:09:46.292" "56045568" "6815744"
"12/17/1999 13:09:47.293" "56045568" "6815744"
"12/17/1999 13:09:48.295" "56045568" "6815744"

```

Part 2: Networking Performance

Chapter 7

Optimize GPO-Processing Performance

—by *Darren Mar-Elia*

If you've deployed Active Directory (AD), you know the benefits that it brings to your Windows environment. Among these benefits is the use of Group Policy Objects (GPOs)—powerful tools for managing your Windows 2000 servers and your Windows XP and Win2K workstations. As with any technology, however, too much of a good thing can hurt your systems' performance. You can link GPOs to multiple levels of your AD hierarchy, so a particular computer or user in your infrastructure might be subject to tens of GPOs at system startup or at logon. The result: long startup and logon times while your systems complete GPO processing.

To manage GPO processing and optimize your GPO infrastructure so that the impact on your systems and users is minimal, you need to understand how Win2K stores and applies GPO settings, how you can adjust those settings, and how to design an effective yet efficient Group Policy infrastructure.

GPO-Processing Basics

You link GPOs to container objects (i.e., sites, domains, or organizational units—OUs) within AD, and all user and computer objects under that container process those GPOs. This process can be complicated because user and computer objects must process any GPOs that you link to the domain, parent and child OU, and site in which the object resides. You can link one GPO to multiple container objects, or you can link multiple GPOs to one container object. The former situation has little effect on GPO-processing performance, but the latter situation makes all the difference in the world. The more GPOs that a given computer or user must process, the more time the computer needs to boot or the user needs to log on.

Win2K stores a GPO's settings in two places: the GPO's Group Policy Container (GPC) in AD, and the GPO's Group Policy Template (GPT) within the Sysvol share on your domain controllers (DCs). The process of creating a new GPO through the Microsoft Management Console (MMC) Active Directory Users and Computers snap-in or the MMC Active Directory Sites and Services snap-in creates the GPC and GPT and links the GPO to the selected container object. When you use the MMC Group Policy snap-in to change a GPO, your actions modify both the GPC and the GPT.

Processing the settings in the GPC and GPT is the job of a set of DLLs called client-side extensions. Your XP and Win2K workstations' local registries reference these client-side extensions in separate subkeys under the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions subkey. The values in each globally unique identifier (GUID)-named subkey list the name of the DLL, the Group Policy processing category that the

extension provides (e.g., Folder Redirection, Software Installation), and the settings that control the extension's behavior. These settings determine, for example, whether the extension will process a GPO when the computer connects to the DC over a slow network link, whether the extension will refresh policy settings periodically, and whether the extension will process GPOs that haven't changed since the last processing time.

Client-side extensions are the primary worker bees of GPO processing. But certain network interactions must occur before a client-side extension can do its work. Network communications usually represent a significant portion of your servers' and workstations' total GPO-processing time. When a Win2K workstation boots in an AD domain that contains GPOs, the following processes take place:

1. The workstation queries a DNS server to locate a DC in the workstation's site. To be precise, the workstation queries DNS for the `_ldap._tcp.sitename._sites.dc._msdcs.domain-name` SRV record. This record returns the name of the DC (in the site *sitename*) that handles Lightweight Directory Access Protocol (LDAP) requests for the domain.
2. The workstation establishes a secure-channel connection with the DC.
3. The workstation pings the DC to determine whether the workstation's network connection to the DC (e.g., dial-up, T1) constitutes a slow network link. (By default, Win2K considers a transfer rate of less than 500Kbps to be slow. See the Microsoft article "How a Slow Link Is Detected for Processing User Profiles and Group Policy" at <http://support.microsoft.com/?kbid=227260> for information about how Win2K calculates slow links.)
4. The workstation binds to AD over LDAP.
5. The workstation uses LDAP to query AD and get a list of all the GPOs linked to the workstation's OU or parent OU.
6. The workstation uses LDAP to query AD and get a list of all the GPOs linked to the workstation's domain.
7. The workstation uses LDAP to query AD and get a list of all the GPOs linked to the workstation's site.
8. The workstation uses LDAP to query the GPC (in AD) and determine the path to each GPO's GPT (in Sysvol).
9. The workstation reads the `gpt.ini` file that resides in each GPO's GPT. This file lists the GPO's current version number.
10. The workstation's client-side extensions process the retrieved GPOs.

These steps represent the processing of only computer-specific GPOs, which occurs at computer boot. After a user logs on to the system, Win2K must process any user-specific GPOs. During that procedure, the OS repeats Steps 4 through 10 (from a network perspective, Steps 1 through 3 have occurred already).

Performance Boosters

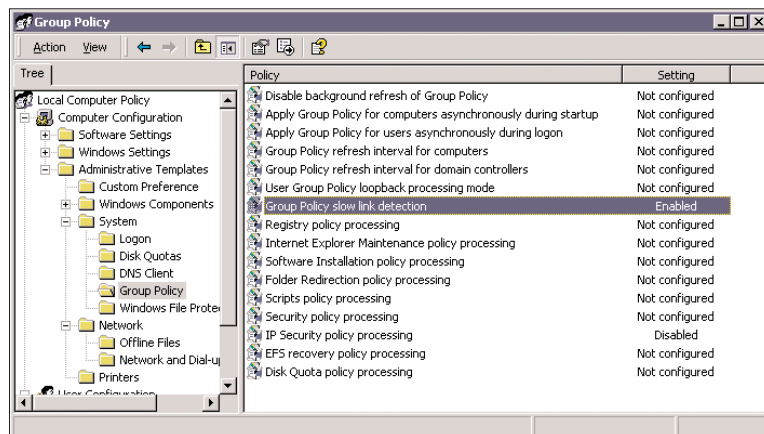
Besides the sheer number of GPOs that a computer or user object must deal with, numerous steps within the GPO-processing operation can affect the amount of time that a computer needs to boot or that a user needs to log on and gain control of the desktop. The ability to promptly resolve the required DNS names and locate a DC in the workstation's site also is important to good GPO-processing performance. The more time these basic setup tasks take, the more time GPO processing

consumes. And if your XP or Win2K devices can't resolve the correct SRV records, GPO processing might fail outright.

Even basic GPO processing can be time-consuming. However, several Group Policy settings and features can affect GPO-processing performance. As Figure 1 shows, you can access client-side extension and Group Policy options through the Group Policy snap-in. Open the Group Policy console, then drill down to Computer Configuration, Administrative Templates, System, Group Policy. Select a policy in the right-hand pane and open the policy's Properties dialog box to view or modify the policy's settings. In particular, the policies that control slow-link detection, processing despite GPO version, and synchronous or asynchronous processing can affect performance significantly.

Figure 1

Policy options for controlling GPO-processing behaviors



Slow-Link Detection

By default, the client-side extensions that control Folder Redirection, Software Installation, Scripts, and Disk Quota won't process a GPO when the workstation detects a slow link. Enabling slow-link detection means that fewer client-side extensions will work to process GPOs, so GPO-processing time will lessen under slow-link conditions. You can modify the default slow-link value of 500Kbps through the *Group Policy slow link detection* policy. (However, increasing the threshold to force slow-link detection isn't the best strategy for improving GPO-processing performance.)

GPO Versioning

Each GPO's GPC and GPT contain the GPO's version number. Win2K increments this number each time you change the GPO. XP and Win2K workstations keep a history of each round of GPO processing in their local registries, under the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\History and HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\History subkeys. By default, client-side extensions won't

process a GPO if its version number hasn't changed. When a GPO's version number is 0 (meaning that no settings have been made within the GPO), the client-side extensions won't even attempt to process the GPO.

Forcing the client-side extensions to process all GPOs regardless of version number will increase processing time. From the Group Policy folder, select a policy from the right-hand pane, open the policy's Properties dialog box, select the option to enable the policy, and be sure the *Process even if the Group Policy objects have not changed* check box is cleared.

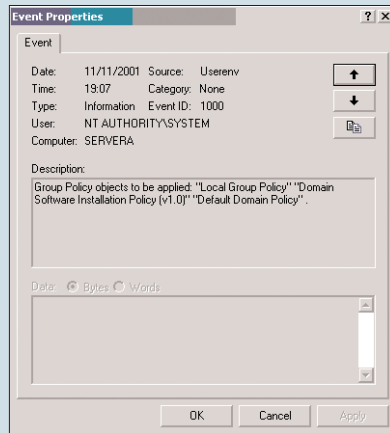
Asynchronous Processing

By default, Win2K's GPO-processing operations are synchronous: All client-side extensions must finish processing any machine-based GPOs (at system boot) before the computer will present the logon dialog. Similarly, when a user logs on to a Win2K device, the client-side extensions that process user-level GPOs must complete their work before the user can get control of the desktop and start working. If the processing of many GPOs significantly delays system startup or user logon, you can configure Win2K to process GPOs asynchronously (through the *Apply Group Policy for computers asynchronously during startup* and the *Apply Group Policy for users asynchronously during logon* policies). However, a GPO that doesn't complete processing by the time a user logs on might not go into effect until the next time the user logs on—a lapse that could present a problem for Group Policy categories such as Software Installation and Folder Redirection. (XP includes a *Fast logon optimization* feature, so XP's GPO processing is asynchronous by default. Thus, the client-side extensions on an XP device might not finish processing all GPOs before a system presents the logon dialog box or lets a user access the desktop, and Software Installation and Folder Redirection typically require two logons before they take effect.)

Win2K also uses asynchronous processing for background refresh of Group Policy. Win2K periodically refreshes certain client-side extensions, such as those responsible for security settings and administrative templates, after the initial processing at boot or logon. For example, the client-side extension responsible for security settings on a Win2K server or workstation refreshes all applicable GPO settings every 90 minutes by default. On DCs, the default refresh interval is 5 minutes. This type of periodic processing limits the damage from users who muck with security settings between logons or reboots.

Not all client-side extensions support background refresh. For example, the Software Installation policy doesn't refresh (uninstalling Microsoft Word while someone is using it would be a bad idea). Also, client-side extensions won't refresh a GPO that hasn't changed. To prevent a GPO from refreshing, open a policy's Properties dialog box and select the *Do not apply during periodic background processing* check box. To change a device's overall background processing settings, enable and modify the *Disable background refresh of Group Policy*, *Group Policy refresh interval for computers*, or *Group Policy refresh interval for domain controllers* policy.

Although background processing doesn't have a big effect on your system's performance, you should be aware that it's happening. You can enable event logging for GPO processing so that you can monitor background processing and troubleshoot processing problems (see the sidebar "Group Policy Logging" for details).



Greater Control

Performance-enhancing behaviors such as slow-link detection, GPO versioning, and asynchronous-processing options are available in XP and Win2K. You can also explicitly tune a couple other settings to further reduce the overhead of GPO processing.

Disable Unused Settings

Within each GPO, you can define settings that apply to computers or to users. However, you don't need to define both within a given GPO. Therefore, the first and easiest step to enhance performance is to disable a GPO's unused computer-level or user-level settings. Suppose that a workstation determines during boot that it needs to process four GPOs, only two of which have a defined computer-level policy. You can flag the other two GPOs as not having any computer-level policy. As a result, the workstation's client-side extensions won't bother to look for the nonexistent computer-level settings, and you'll save some time in the processing cycle.

To disable a GPO's computer- or user-level settings, open the Active Directory Users and Computers snap-in or the Active Directory Sites and Services snap-in, right-click the container to which the GPO is linked, then choose Properties from the context menu. Go to the Properties dialog box's Group Policy tab. Select the GPO and click Properties to open the GPO's Policy Properties dialog box. Use the check boxes in the Disable section to disable unused computer or user configuration settings. (You can select both check boxes, but doing so effectively disables the GPO.)

Set a Maximum Wait Time

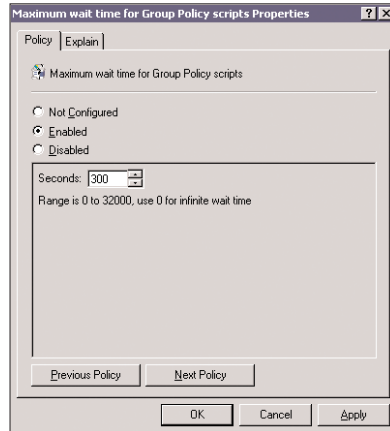
Another way to keep GPO-processing times in check is to establish a maximum interval for running scripts. GPOs support computer startup and shutdown scripts as well as user logon and logoff scripts. Such scripts can be any form of executable, batch file, or Windows Script Host (WSH) script. Because you can apply multiple GPOs to a given user or computer, you might have multiple scripts running one after the other. But ill-functioning or poorly programmed scripts could hang or run forever. For example, when you use synchronous GPO processing, your XP and Win2K systems might hang for as many as 10 minutes, and you have no easy way to determine the problem.

To mitigate this type of problem, you can set a maximum time for all scripts to run. In a worst-case scenario, a script that is hung or caught in some kind of loop will run for only the specified time. Be aware, however, that the wait time applies to the total runtime of all scripts. For example, if you've defined logon scripts in each of 10 GPOs in your AD domain and you set the wait time to 60 seconds, all those scripts must be completely executed within 60 seconds. To specify a maximum script-processing interval, open the Group Policy snap-in, drill down to Computer Configuration, Administrative Templates, System, Logon (or Administrative Templates, System, Scripts in XP), and open the *Maximum wait time for Group Policy scripts* policy's Properties dialog box, which Figure 2 shows. You can enable the policy and configure the wait time on the Policy tab.

Design Matters

Aside from tweaking Group Policy behaviors, you can mitigate or prevent performance problems through a well-planned Group Policy infrastructure. Limiting the number of GPOs you create, the security groups you use, and the cross-domain GPO links you establish can speed up processing time.

Figure 2
Setting the maximum wait time for Group Policy scripts



Limit GPOs

The most basic step is to limit the number of GPOs that a computer or user must process at startup or logon. In general, I suggest limiting this number to 10 as a starting point, but you need to test this number for yourself because it depends heavily on what each GPO does. Also keep in mind that wait times are longer the first time a computer or user processes a GPO because the client-side extensions must initially apply all the settings. After the initial processing cycle, subsequent system restarts or user logons will process only GPOs that have changed (unless you force them to do otherwise).

Limit Security Groups

The use of security groups (i.e., AD local, global, or universal groups containing computers or users) can affect GPO processing. You can use security groups to filter GPOs' effects—for example, when you want to apply a domain-level GPO to only a handful of users or computers. However, security-group filtering comes with a performance cost. The more access control entries (ACEs) you associate with a GPO, the more work the GPO's client-side extension must do to figure out whether a computer or user belongs to one of the groups to which you've applied filtering. Thus, keeping your GPOs' ACLs short and concise further improves (or at least maintains) performance. Don't use ACLs indiscriminately to filter GPOs for every computer or user. Instead, rethink the level at which you're linking your GPOs. You might get the desired effect by relinking the GPO lower in your AD hierarchy (e.g., at the OU level rather than the domain level).

Limit Cross-Domain Links

Another design aspect that can play a role in performance is the use of GPOs that are linked across domain boundaries. Every GPO belongs to one AD domain, and the GPO's GPC and GPT reside on that domain's DCs. Suppose you have a multidomain AD forest. You could link a GPO

in one domain (Domain A) to another domain in the forest (Domain B). But when a computer or user in Domain B processes the GPO that resides in Domain A, the client-side extensions on the Domain B computer must traverse trust relationships within the forest to access the GPO's GPC and GPT. Such an operation is more expensive from a performance perspective than communicating with GPOs within the same domain. Furthermore, if the Domain B computer can't find a Domain A DC within the same AD site, the computer might need to traverse WAN links to reach a DC and process the GPO.

The best solution is to avoid linking GPOs across domain boundaries. Instead, copy a defined GPO from one domain to another. (XP and Win2K don't provide an easy way to copy GPOs from one domain to another, but third-party tools can provide such functionality.)

GPOs: Complex but Powerful

GPOs can be powerful tools in your Windows systems-management arsenal, but GPO configuration and behaviors are complex and can slow down system startups and user logons. Armed with the knowledge of how to modify GPO behavior and infrastructure to improve GPO-processing time, however, you can minimize GPO performance penalties—and get the most out of your AD infrastructure.

Chapter 8

Web Server Load Balancers

—by *Tao Zhou*

As the e-commerce industry continues to grow, more businesses rely on their Web sites to communicate with customers. A high-performance Web site that quickly and reliably delivers content gains and retains customers and is crucial to a successful and competitive e-business. Few potential customers will return to a frustratingly slow Web site if the customer experiences significant delays or failure. Thus, as part of your organization's Web infrastructure planning and implementation, you need to seriously consider how to improve your Web site's performance.

You can use several methods to improve Web performance: expand Internet bandwidth, use fast network equipment, design efficient Web applications, optimize and upgrade Web server software and hardware, and use Web-caching technology. In addition to these options, you can improve your Web site's performance by adding Web servers and sites and mirroring content across all servers and sites. This method lets you share the overall load among servers and sites and reduce the information turnaround time involved in a server's internal processing of client requests. In addition, you can preserve your existing servers rather than retire them to make way for new servers.

Load sharing, or balancing, on multiple servers ensures that Web traffic doesn't overload one server while other servers sit idle. To load balance Web servers, traditionally you use the DNS round-robin feature to evenly distribute Web server IP addresses to clients; thus, your Web servers are equally accessible. However, this mechanism doesn't provide load balancing in an environment in which the Web servers have different hardware and software capacities. For example, a Windows 2000 Server system with two 450MHz Pentium III processors and 1GB of memory should handle more load in a load-balanced environment than a Windows NT Server system with one 300MHz Pentium II processor and 256MB of memory. However, DNS's round-robin feature treats these two systems equally; it doesn't know a Web server's availability because the round-robin feature doesn't detect whether the server is up or down.

Recently, vendors developed *load balancers*, which are products that balance load evenly across multiple servers. In addition, a load balancer ensures Web servers' fault tolerance by redirecting traffic and clients to another server or site in case of failure. Therefore, clients experience fewer delays and no failures. You can use load balancers in single Web site and multiple Web site scenarios. Knowing what a load balancer is and how it works will help you identify important features to consider and evaluate when choosing a load balancer.

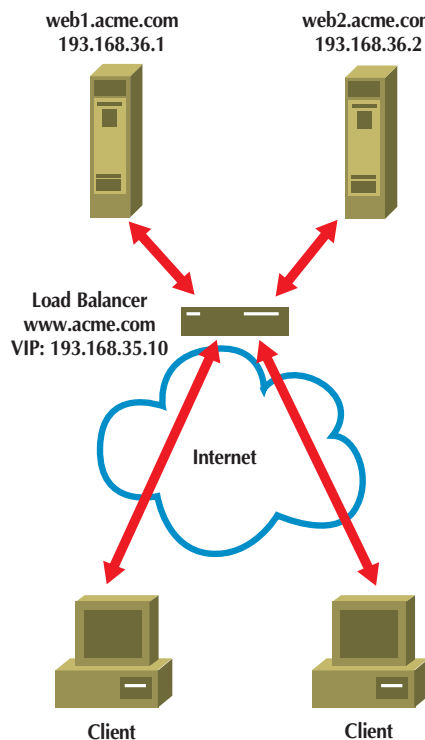
What Is a Load Balancer?

A Web server load balancer is a tool that directs a client to the least busy or most appropriate Web server among several servers that contain mirrored contents. The client transparently accesses this set of servers as one virtual server. For example, suppose you have two Web servers in a

single Web site scenario: web1.acme.com with the IP address 193.168.36.1 and web2.acme.com with the IP address 193.168.36.2, as Figure 1 shows. The load balancer uses a virtual host name (e.g., www.acme.com) and virtual IP (VIP) address (e.g., 193.168.35.10) to represent the Web site. You associate the virtual host name and the corresponding VIP address with the two Web servers by publishing the virtual host name and its VIP address in your DNS server. The load balancer constantly monitors the load and availability of each Web server. When a client accesses www.acme.com, the request goes to the load balancer instead of a Web server. Based on the load of each monitored server and conditions and policies you've defined, the load balancer decides which server should receive the request. The load balancer then redirects the request from the client to the server and usually redirects the reply from the server to the client depending on the implementation.

Figure 1

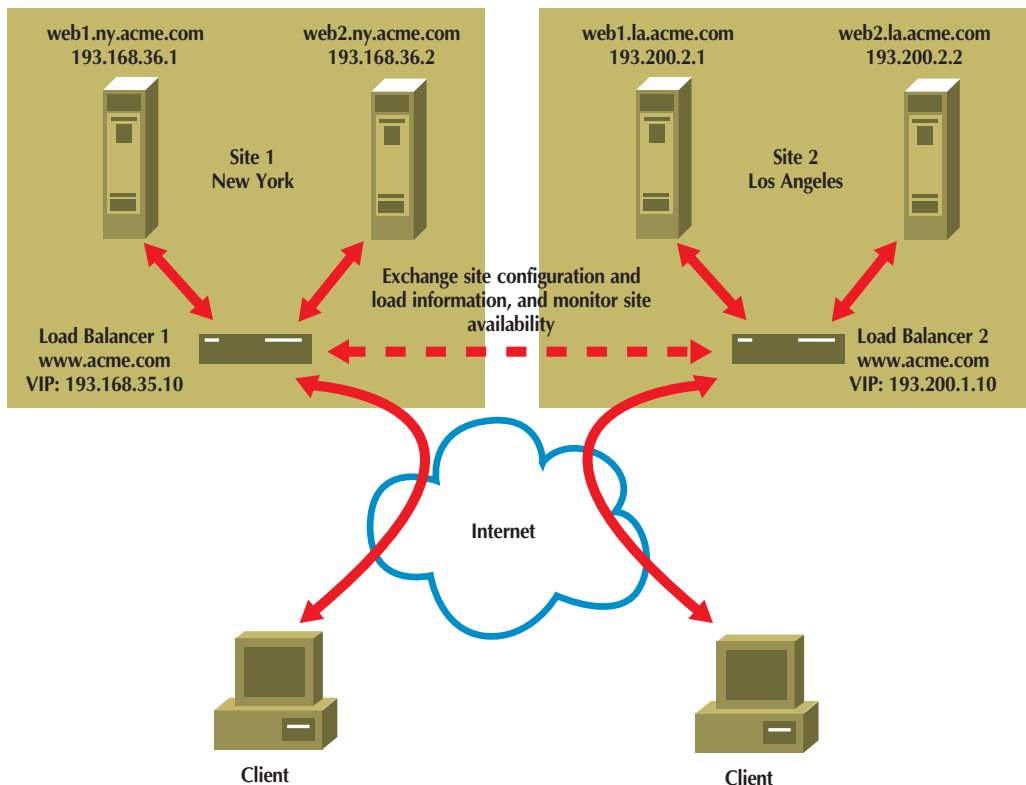
Load balancer in a single-site scenario



Load balancers can also support load balancing across multiple Web sites. Implementing multiple sites places mirrored Web servers close to customers and reduces delay between your Web site and customers. In addition, multiple Web sites provide load balancing, high availability, and

fault tolerance in case of a complete site failure (e.g., a power or Internet connection outage at a data center). In a multisite scenario, which Figure 2 shows, every load balancer at every site has the same virtual host name but a different VIP address. For example, load balancer 1 at site 1 in New York has the virtual host name `www.acme.com` and VIP address `193.168.35.10`. Load balancer 2 at site 2 in Los Angeles has the same virtual host name but a different VIP address—`193.200.1.10`. You associate each load balancer with its local Web servers using the same method you use in a single-site scenario. In addition to monitoring the load of local servers, the load balancers exchange site configuration and load information and check site availability with load balancers at other sites. Thus, each load balancer has the global load and availability information locally. Load balancers in multisite scenarios also often work as the DNS server for the virtual host name. When a load balancer receives a DNS lookup from a client for the virtual host name, the load balancer returns to the client the VIP address of the best site according to the current site load, client proximity, and other conditions. The client then transparently accesses that site.

Figure 2
Load balancers in a multisite scenario



Three types of load balancers exist: hardware appliances, network switches, and software. A hardware appliance-based load balancer is a closed box that is usually an Intel-based machine running a vendor's load-balancing software and a UNIX or proprietary OS. Hardware appliance-based load balancers provide a Plug and Play (PnP) solution for Web administrators. A network switch-based load balancer uses a Layer2 or Layer3 switch to integrate the load-balancing service. This device doesn't require an add-on box between the switch and the Web servers, but the appliance-based load balancer requires an add-on box. A software-based load balancer doesn't require you to modify network connectivity or equipment when you introduce the load-balancing service to your Web server farm. You can install the software on existing Web servers or dedicated load-balancing servers. (The sidebar "Microsoft's Load-Balancing Services" explores Microsoft's load-balancing solutions.)

Regardless of which product category a load balancer belongs to, it fulfills the following three functions: monitoring server load and health, selecting the right server for a client, and redirecting traffic between the client and server. Let's look at each of these functions and learn how load balancers implement them.

Microsoft's Load-Balancing Services

—by *Tao Zhou*

In 1998, Microsoft acquired Windows NT Load Balancing Service (WLBS) from Valence Research. This product, which Valence marketed as Convoy Cluster, is a free add-on service to Windows NT Server, Enterprise Edition (NTS/E). Microsoft implemented this service in Windows 2000 Advanced Server and Windows 2000 Datacenter Server as Network Load Balancing (NLB) service. Both services support 2 to 32 servers in the same cluster. Administrators most commonly use WLBS and NLB to distribute Web client requests among Web servers; however, both services support additional Internet applications such as FTP server load balancing.

You install WLBS or NLB on all servers in the same Web site or cluster, and a virtual IP (VIP) or cluster address represents the Web site or cluster. The software requires all servers to be on the same subnet, and both services use a media access control (MAC) multicast method to redirect client traffic. When the router that the server subnet connects to receives a client request, the router uses a MAC-layer multicast to multicast the request to the cluster. The load-balancing server uses its algorithm to choose the best available server for the client. That server responds and handles the client request. You can configure the service to evenly distribute requests to the servers or specify the percentage of the total cluster load that a server takes based on the server's capacity. Both WLBS and NLB can select a server and redirect traffic according to the client's IP address and port numbers, and the software can support a persistent connection based on the

Continued on page 71

Microsoft's Load-Balancing Services *continued*

client's IP address or Class C network address. However, the software doesn't support delayed binding. Each server provides failover for every other server, thus the software provides load-balancing redundancy in an active-and-active implementation. Although you can install Microsoft's load-balancing services in multiple Web sites or cluster scenarios, the service doesn't support global load balancing. To distribute traffic to multiple sites, you must use the DNS round-robin feature, which doesn't provide failover or good load-balancing across sites.

WLBS and NLB are useful for traffic distribution among front-end Web servers. To support high availability of back-end applications such as Microsoft SQL Server, you can use Microsoft Cluster Server (MSCS) through a 2-node cluster in NTS/E and a 4-node cluster in Datacenter. In addition, Microsoft developed a COM+ load-balancing service called Component Load Balancing (CLB). This service provides load balancing on the middle or business-logic tier of multitiered Windows applications. Microsoft originally planned to include CLB in Windows 2000, but decided to exclude this service from the final Win2K release. Microsoft included CLB in Application Center Server, which is a high-performance management solution for Win2K Web applications.

Server Monitoring

A load balancer constantly monitors the load and health of managed Web servers so that it can use the load and health information to select the best available server to respond to a client request. Load balancers use two methods to monitor servers: external monitoring and internal monitoring.

To externally monitor a server, the load balancer calculates a server's response time by inputting a request to the server and waiting for a response. Using an Internet Control Message Protocol (ICMP) ping is the simplest way for a load balancer to externally monitor a server. An ICMP ping tests a server's availability and the round-trip time between the server and load balancer. If the load balancer doesn't receive a response from the server after several consecutive pings, the load balancer assumes that the server isn't available. Administrators usually connect Web servers directly to the load balancer, so if the round-trip response time is long, the load balancer knows that the server is very busy.

However, the ICMP ping tests only a server's IP stack but can't monitor the health of the TCP stack that HTTP uses. To verify that a server's TCP stack works, the load balancer attempts to establish a TCP connection, which requires a three-way handshake, with the server. In a three-way handshake, the load balancer sends the server a TCP packet with the SYN bit set to 1. If the load balancer receives back from the server a TCP packet with the SYN bit set to 1 and the ACK bit set to 1, the load balancer sends another TCP packet with the SYN bit set to 0 and the ACK bit set to 1. A completed handshake means that the server's TCP stack is healthy. After completing the handshake, the load balancer immediately drops the connection to save server resources. The load balancer can estimate a server's TCP connection performance based on the time a completed three-way handshake takes to complete.

In addition to testing the protocol stacks, a sophisticated load balancer can monitor the response time and availability of a Web server and its applications by making an HTTP request for content or a URL. For example, suppose `web1.acme.com`'s home page is `index.htm`. The load balancer in Figure 1 can initiate an HTTP Get command asking for the content of `index.htm` on `web1.acme.com`. If the load balancer receives from the Web server a return code of 200, the home page on `web1.acme.com` is available. The load balancer measures the response time by measuring the time between sending the content request and receiving the return code.

Although external monitoring lets you ascertain useful information, it provides limited or no information about several important aspects of a server's status, including CPU, memory, system bus, I/O bus, NIC, and other system and application resources. Only internal monitoring can provide such detailed server load information. To internally monitor a server, the load balancer uses an internal-monitoring agent, which physically resides in each server, monitors a server's status, and reports the status to the load balancer. Some vendors provide scripting tools that let you write internal monitoring utilities for your Web applications. Internal monitoring is common for software-based load balancers, but few appliance- and switch-based load balancers use internal monitoring.

Server Selection

A load balancer can use the information from externally and internally monitoring a server to select which server is best for handling a client request. If all servers have the same hardware and software capacity, you can configure the load balancer to use a round-robin system to select a server based on the servers' status. However, if a load balancer manages a server with a Pentium III processor and a server with a Pentium Pro processor, you can configure the load balancer to redirect more traffic to the more powerful server. This setup is a weighted round-robin configuration.

A sophisticated load balancer lets you specify a custom policy of server selection. For example, you can configure the policy to include CPU utilization, memory utilization, number of open TCP connections, and number of packets transferred on a server's NIC. Your load balancer's load formula might look like $(10 \times \text{CPU utilization}) + (3 \times \text{memory utilization}) + (6 \times \text{the number of open TCP connections}) + (3 \times \text{the number of transferred packets}) = \text{a server's load}$. When it receives a client request, the load balancer calculates the load for each server according to the formula and redirects the request to the server with the lightest load.

In some cases, after the load balancer assigns a server to a client and the server and client make an initial connection, an application requires the load balancer to persistently send the client's traffic to that server. This connection is a persistent or sticky connection. For example, a user is shopping in an online bookstore and puts three books in the shopping cart. If the server that processes that client's request caches the shopping cart information locally, the load balancer can't switch the client's new traffic to another server even if the load becomes unbalanced across a site's servers. Otherwise, the three books in the client's shopping cart will be lost because the new server doesn't have the client's shopping cart information. Therefore, the load balancer must remember which client is accessing which server for a certain amount of time that you define based on your customers' behavior and applications. If you enable a load balancer's persistent feature, this feature will always override other load-balancing policies.

The key to maintaining a persistent connection is to find out a client's identity and bind this identity to a destination server. The load balancer usually uses the client's source IP address as the client's identity. However, the client's source address might not be the client's real IP address.

Many companies and ISPs use proxy servers to control Web traffic and hide their users' IP addresses. Thus, if 500 clients access your Web site from AOL and 10 clients access your Web site from another company, the server load will be unbalanced because the load balancer will bind all 500 AOL clients that have the same source address to one server and the other 10 clients to another server. To overcome this disadvantage, a load balancer that supports source IP address and TCP port number binding can distinguish clients even if the clients are using the same proxy server. The load balancer can make this distinction because each TCP connection has a unique source IP address and TCP port number. Another way to identify a client if the client is using a secure HTTP session is to monitor a Secure Sockets Layer (SSL) session ID. The SSL protocol assigns an ID to an established SSL session, and online shopping applications often use SSL. The most recent innovation to support a persistent connection is the Web cookie, which contains a client's identity and other information, such as which server the client last accessed. By examining Web cookies' content, a load balancer can better identify clients and select the appropriate server for them. Cookie-aware load balancer vendors include Alteon WebSystems (now owned by Nortel Networks), ArrowPoint Communications (now owned by Cisco Systems), F5 Networks, and Resonate.

In another server-selection method, *immediate binding*, load balancers can choose a server for a client and send the client to the server as soon as the load balancer receives the client's TCP SYN packet. A load balancer bases the server selection on server load-balancing policies and the IP address and TCP port numbers in the client's TCP SYN packet. Although this method is fast, a load balancer doesn't have time to ascertain other information, such as the SSL session ID, cookie, URL, or application data. To learn more about the client and make a better decision, the load balancer needs time to peek into application-layer information. In the delayed-binding method of server selection, the load balancer waits to make a server selection until the TCP three-way handshake is complete and the load balancer and client establish a connection. The load balancer becomes content-aware by examining the application-layer information before selecting a server.

Traffic Redirection

A load balancer can use several methods to redirect client traffic to the chosen server: media access control (MAC) address translation (MAT), Network Address Translation (NAT), or, for delayed binding, a TCP gateway mechanism. Let's explore how load balancers use each method to redirect traffic.

MAT

A load balancer that uses this method requires each Web server to use the load balancer's VIP address as a loopback interface address, in addition to the Web server's physical IP address. When the load balancer receives a client packet and makes a server selection, the load balancer replaces the destination MAC address in the client packet with the chosen server's MAC address and sends the packet to the server. The packet contains the client's IP address, so to directly reply to the client, the server uses the original client IP address as the destination IP address. However, the server uses the load balancer's VIP address as the source IP address, as if the traffic to the client is from the load balancer. In this way, the client's next packet goes to the load balancer rather than to the server that replied to the client.

NAT

Using the NAT method, a load balancer substitutes a received client packet's destination address (i.e., the load balancer's VIP address) for the chosen server's IP address and the source IP address for the load balancer's VIP address before the load balancer redirects the packet to the chosen server. When the load balancer redirects a server packet to the client, the load balancer replaces the destination IP address with the client's IP address and the source IP address with the load balancer's VIP address. This method hides the Web server's IP addresses from clients, so the Web servers can use any IP addresses, including private addresses. The Web servers don't need to directly connect to the load balancer (i.e., use the same LAN segment) as long as the servers and the load balancer can reach one another through a static-routing or network-routing protocol.

TCP Gateway

For immediate binding, load balancers can use the MAT or NAT method to redirect traffic at Layer2 or Layer3. However, for delayed binding, load balancers have to redirect traffic at the TCP layer and above. For delayed binding, the load balancer and client establish a TCP connection so that the load balancer can receive application data before it makes a server selection. Next, the load balancer sets up a TCP connection with the chosen server and passes the client request to the server through this connection. The load balancer then passes the server's response to the client through the load balancer and client TCP connection. This function is referred to as a TCP gateway. Resonate implements this function in its load balancer product through an agent on the server that permits a direct TCP connection between the client and the server that is acting as the load balancer. The vendor calls this implementation *TCP connection hop*.

Global Site Selection and Traffic Redirection

In a multiple-mirrored site scenario, the load balancer (aka the global load balancer) uses the same server-selection mechanisms as in a single-site scenario to choose the best site for a client. In addition, a global load balancer can use client proximity (i.e., network hops and network latency) between the site and the client as an element in site selection. To make this selection, the load balancer often uses an intelligent DNS function to redirect the client traffic to the appropriate site.

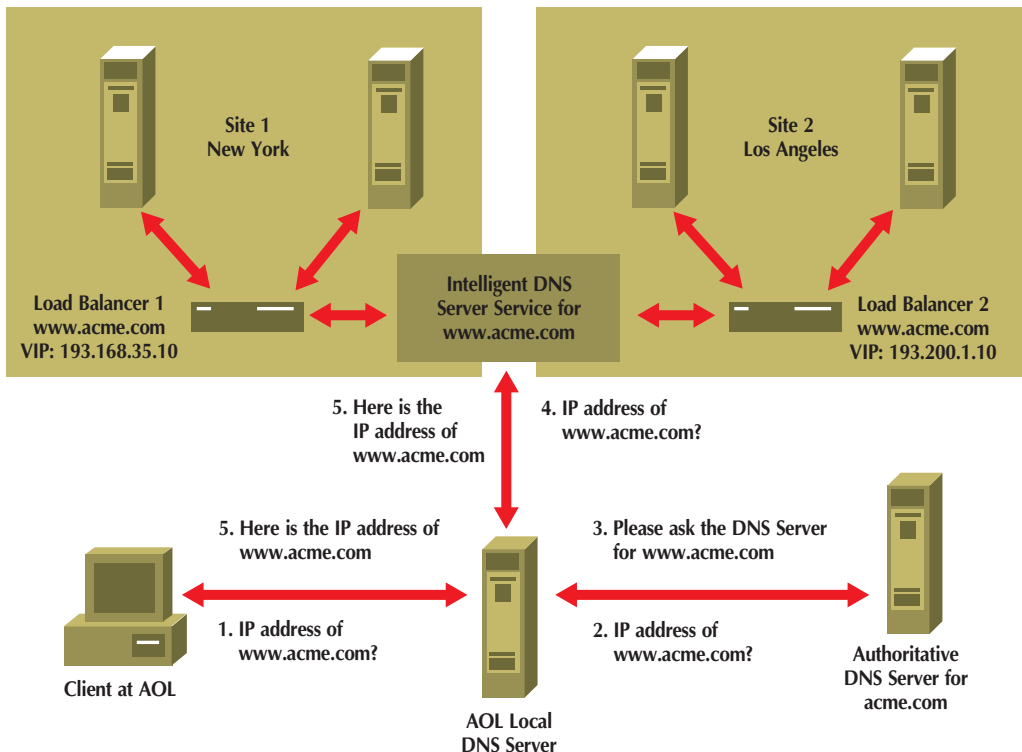
For example, www.acme.com has two sites, one load balancer in New York and one in Los Angeles, that work as DNS servers for www.acme.com. The authoritative DNS server for the Internet domain acme.com provides name resolution for FTP, mail, and other Internet servers and hosts. You can delegate the subdomain www.acme.com of the acme.com Internet domain to each load balancer; these load balancers become name servers for www.acme.com. To set up this configuration, define a DNS entry of www.acme.com in each load balancer and map the entry to the load balancer's local VIP address. The two global load balancers exchange configuration and load information, so both load balancers are aware that two VIP addresses (i.e., two sites) exist for www.acme.com. Thus, they know the load and availability of each site.

As Figure 3 shows, when a client at AOL tries to access www.acme.com, the client requests that AOL's local DNS server look up the IP address of the host name www.acme.com. If AOL's local DNS server doesn't have cached information about the requested host IP address, the server sends the request to acme.com's authoritative DNS server. [Acme.com](http://acme.com)'s DNS server delegated www.acme.com to two load balancers, so acme.com returns to AOL's local DNS server the two load balancer's IP addresses as www.acme.com's name server. (In Figure 3, I used a separate box

to highlight the intelligent DNS server service. Some vendors implement this technology in a separate server.) AOL's local DNS server then sends the DNS lookup request to one of the two load balancers. The two load balancers are name servers, so AOL's local DNS server will resend the request to the other server if the first one doesn't respond. The load balancer returns to AOL's local DNS server a VIP address based on the site load-balancing criteria. After the client receives a VIP address for `www.acme.com` from AOL's local DNS server, the client sends the HTTP traffic to the load balancer of the chosen site (e.g., New York). The load balancer in New York then selects the local server for the client. Because the local DNS server caches a resolved DNS record according to the record's Time to Live (TTL) value, most vendors suggest that you keep the TTL value of a VIP low so that clients can quickly receive a new VIP address and switch to another available site.

Figure 3

DNS redirection in a multisite scenario



Alternatively, load balancers can use HTTP redirection for global site selection and traffic redirection. This method doesn't use the load balancer's DNS function. Instead, following the `www.acme.com` example, you define in your authoritative `acme.com` DNS server the `www.acme.com` DNS record and its VIP addresses. When a client resolves `www.acme.com` and sends the HTTP request to a load balancer, the load balancer chooses the best site for the client. If the chosen site

isn't remote, the load balancer sends an HTTP redirection command to the client's browser, which accesses that site. This method lets the load balancer learn more about the client (e.g., the client's IP address) before the load balancer makes a site selection. However, the client might try to use a returned VIP address from the DNS server to access a failed site.

In addition to dynamically assigning a site to a client, load balancers can use a static mapping method to bind a specific client to a specific site. For example, suppose you have a mirrored Web site in Europe. You want European clients to access only the European site unless the site is down and the load balancer fails over the European traffic to your US site. In the load balancer, you can statically define that a request from a European IP address goes to the European site first. (To configure this setup, you must manually enter the European IP address blocks in the load balancer.) When the load balancer sees a European address, it redirects the traffic to the European site before it applies other rules.

Load Balancer Redundancy

A load balancer has the potential to become a single point of failure in a Web site because it serves as a front end for the back-end Web servers. When you design and implement a load-balancing solution, consider the load balancer's fault tolerance and choose a fast load balancer for good performance. You can choose between the two types of load-balancer redundancy: active-and-standby and active-and-active. Both methods use two load balancers at one site.

In the active-and-standby method, a backup load balancer constantly monitors the primary load balancer. When the primary load balancer is unavailable, the backup load balancer takes over the function of the primary load balancer (i.e., the backup load balancer handles traffic). When the primary load balancer comes back online, the backup load balancer transfers traffic to the primary load balancer and returns to standby mode.

In the active-and-active setup, both load balancers serve traffic and back each other up. For example, suppose you have four Web servers at a site. The first load balancer serves two Web servers, and the second load balancer serves the other two servers. When one load balancer is down, the other load balancer serves all four Web servers. This method fully utilizes load balancer resources and improves performance.

Balance Your Environment

Web hosting and e-services companies are not the only organizations that are using load balancers to direct traffic and maintain order. Many companies have adopted load balancers for their Web sites to improve Web performance and availability. Through their ability to monitor server load and health, select the best available server for clients, and redirect traffic in local sites and global environments, load balancers have become an important avenue to meet the demands of the competitive e-business market.

Chapter 9

Monitoring Win2K Web Site Performance and Availability

—by Curt Aubley

[Editors' Note: The performance-monitoring methods and diagnostics in this chapter rely on HTTP requests and on ping.exe and tracert.exe commands. Firewalls and numerous other IT-architecture security measures might not permit these protocols. You'll need to work with your security team to plan the location of your Web-monitoring modules.]

Is your Web site running well? Is your site always available, and does it provide content at an acceptable performance level? The best way to answer these questions is to monitor your Web services' performance and availability from an end user's perspective. You can leverage Windows 2000, the *Microsoft Windows 2000 Resource Kit*, a scripting language (e.g., Perl), and Internet standards (e.g., http, Internet Control Message Protocol—ICMP) to gain valuable insight into your Web servers' response times and availability. After you begin tracking your Web site's performance, you can use a combination of automated and manual processes to isolate performance bottlenecks and troubleshoot your Web site.

Customer Perspective: Site Monitoring Methodology

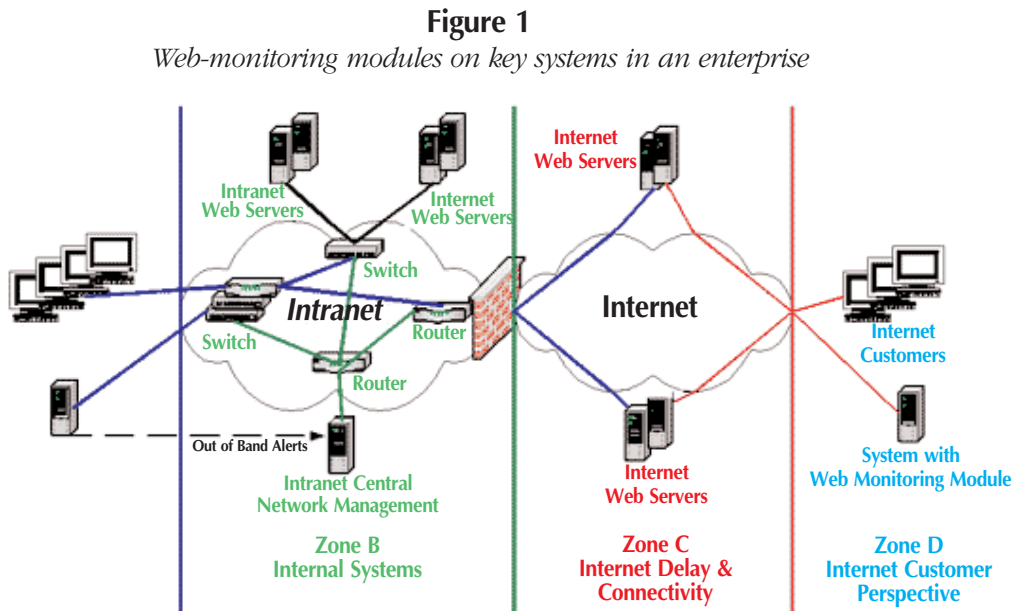
The strategy for determining your Web site's performance (i.e., how long selected Web pages take to download to your customers) and availability centers around the customers to whom you provide Web content or services. What do your customers experience when they download your home page? To determine the answer, you need to run the same application and follow the same network paths as your customers, then measure performance. You can deploy special agents on every desktop that you administer; this approach lets you directly measure customers' application performance. However, cost and manageability constraints make this solution unrealistic for most environments.

Another way to statistically sample your crucial Web servers' performance and availability is to place a few well-located Web-monitoring modules on key systems throughout your enterprise. These systems use the same network paths that your customers use to place HTTP requests to your Web sites. You must track the data path from the client, through the network, to the Web servers; tracking the entire path gives you insight into your customers' experience and helps you troubleshoot all facets of your infrastructure. Figure 1 illustrates this system-management approach.

Although this monitoring strategy isn't radical, it differs from most commercial system- and network-management packages that use ICMP echoes (e.g., basic ping responses) to determine whether a server or network device is available. (For more information about commercial management packages, see the sidebar "Commercial Tools for System and Network Management.") Basic

ping tracking is helpful when you want to monitor availability but provides information only when the server's OS is running, and the information pertains to a generic data packet's response time over the network only from a central location to the server, as Zone B in Figure 1 shows. This type of tracking can't help you determine whether a server is performing its intended application services or how the network is affecting specific application service delivery to your customers, which Zone A and Zone D in Figure 1 represent.

Conversely, you can use the system-specific data that you can collect with the Web-monitoring method to find out how much time your customers must wait when they download specific URLs. You can use Win2K (or Windows NT) event logs to track the URLs' availability over time and automatically alert you (e.g., send an email, page a systems administrator, send SNMP traps) when performance falls below an acceptable level. To automate diagnostic programs and alerts, you can integrate these monitoring tools into commercial system- and network-management tools that you might already have in place.



Implementing and Testing the Monitoring Modules

How can you put this Web-monitoring methodology into motion and make it work for you? To start, you need to determine which URLs you want to monitor, which customer locations you want to track (i.e., where you need to place your monitoring modules), which tracking method (e.g., event logs, database) you want to use, and who you want the system to alert if a problem arises.

The best way to describe the steps that you need to follow is to give you an example. For our example, we'll monitor `insideweb1.mycompany.com`, `insideweb2.mycompany.com`, `www.mycompany.com`, and `www.partnerwebsite.com`. (This combination of sites represents internal, external, and business-partner sites.) We'll monitor these URLs from two internal network

Commercial Tools for System and Network Management

—by *Curt Aubley and Troy Landry*

Why consider a commercial tool set when you can make one? After all, you can often create small management tools to meet specialized needs that commercial products might not address. In-house tools might also be more cost-effective than commercial packages. However, commercial tools for system and network management can provide numerous desirable features (e.g., graphical network maps, scalability to manage hundreds or thousands of servers or networks, automated long-term collection of performance information, OS health and event log monitoring, alert generation). A multitude of system management tools exist. The following are just a few commercial products that provide extensive system and network management functionality: Concord's eHealth Suite, Hewlett-Packard's (HP's) OpenView Suite, BMC Software's PATROL Suite, NetIQ's AppManager Suite, IBM's Tivoli Suite, and Computer Associates' (CA's) Unicenter TNG.

We've found that the best solution is a combination of commercial technologies, in combination with internally developed tools, across several large enterprises, and we've learned that you must consider several important factors when you select commercial management tools. First, make sure that the commercial tool meets the bulk of your requirements. (The product probably won't meet all your needs, which is why you also need in-house tools.) Second, how easily can you roll out the product, and how much training will your team need before you can capitalize on your investment? We recommend that if you can evaluate demonstration software first, do so. If you can wait to purchase management software until you've tested it in your lab, ensured that it will integrate with your existing in-house or third-party tools, and successfully rolled it out to your production environment, you'll know you have a winner.

locations: the network switch to which MyCompany's CEO connects and a switch to which the highest percentage of our customers connect. If our monitoring modules detect a problem, the modules will send an event to the Win2K event logs, write the diagnostic data to a flat file, and send an email alert to our systems administrator (aka SuperSA). Our example also includes an option to use an ODBC call to store the information in a Microsoft SQL Server 7.0 database, which we can use to track Web site performance over time and which provides a more robust mechanism for further analyses.

To set up a monitoring module on your Win2K system, you need to configure TCP/IP, the resource kit (the monitoring module uses the resource kit's Timethis utility to time requested Web transactions) and your chosen scripting language. For our example, we use ActivePerl 5.6, but you can use any scripting language that lets you make an HTTP request from the command line and that lets you call Win2K command-line tools. (You can obtain a freeware version of ActivePerl from <http://www.activestate.com>.)

Next, install and customize `webmonitoringmodulescript.pl` and its associated modules: `poll.pl`, `event.pl`, and `optmod.pl` (Listings 1 through 4, respectively, show these scripts). Download these listings and place them in one directory. You don't need a dedicated system to run these scripts. In our tests on a 450MHz dual Pentium III CPU-based system, monitoring 20 servers at 10-minute intervals generated a CPU load of less than 5 percent every 10 minutes. If you can run an action from the command line, you can start the action from within the Web-monitoring module scripts.

Each script includes explanatory comments and customization sections that you can edit to complete numerous actions (e.g., Listing 4, Customization Section A lets you send an email alert to more than one person) and to suit your environment. (We customized the scripts for our example.) In the script that Listing 1 shows, you'll need to review and update the following variables: `baseline`, which denotes the maximum acceptable time in seconds for a URL request to complete; `outputfile`, which defines the file in which the module keeps results; and `array`, which defines the URLs that you want to monitor. Set the `baseline` value to match your acceptable performance threshold. (In our script, we set the `baseline` to 5 seconds. One of the best ways to determine the threshold is to bring up some Web pages and gauge your reaction. How long is too long for you to wait? Set your `baseline` accordingly. If you want to adjust the `baseline` later, you'll need to change only one number.) If you use the Win2K Network Load Balancing (NLB) service in a cluster environment, you need to modify certain aspects of the script (see the sidebar "Monitoring Modules and Windows 2000 Network Load Balancing Service" for more information). In the script that Listing 4 shows, you'll need to edit Customization Section A to tell the module whom to alert in case of trouble.

Listing 1

Webmonitoringmodulescript.pl

```
NONEXECUTABLE

#####
#
#Perl Script: Rapid prototype example for taking baseline measurements
#  from a customer's point of view
#
#Typically, webmonitoringmodulescript.pl
# runs from a scheduling program every 5 to 10 minutes.
# This module uses HTTP Gets to call the specified URL or URLs and
# determine whether the site or sites responds.
# The script then compares the corresponding timing results to a designated
# baseline, logs the results in a log file, and sends an alert to the NT event log if
# the results exceed the specified baseline.
#
#Key Notes:
# -Place all four scripts in the same directory.
# -The timethis command requires installation of the Windows 2000 Resource Kit.
#
#####

#####
#
#CUSTOMIZATION SECTION A
#
#####
```

Continued on page 81

Listing 1 continued

Webmonitoringmodulescript.pl

```

#Initialization - Set up the variables.
# $baseline - The number of seconds that you determine to be acceptable performance
# $outputfile - The file in which Timethis places its results

$baseline=5;
$outputfile="output.txt";

#Enter the URL or URLs that you want to monitor.
# Place quotation marks around each URL and
# separate each URL with a comma and no spaces (e.g., "www.yahoo.com","abc.com").

@array=("insideweb1.mycompany.com","insideweb2.mycompany.com","www.mycompany.com","www.partner
website.com");

#####
#
#END CUSTOMIZATION SECTION A
#
#####

#Specify the counter for array.
# The program will loop through once for each URL.
$count=1;

while ($count <= @array){

#Run the program to time the URL connection.
# This command puts the timethis poll.pl (URL) >$outputfile command
# into a variable that the system command can run.
# Poll.pl takes the URL as an argument,
# executes a get command to the URL, then puts the return time
# in the $outputfile.
@proglst=("Timethis ", join(" ","poll.pl",$array[$count-1])," >$outputfile");

system ("@proglst");

#Open $outputfile temporary results file and assign IN as a file alias.
# If the program can't open $outputfile, the program stops.

open(IN, "$outputfile") ||
die "cannot open $outputfile: $!";

#The script looks in the output file for the keyword Elapsed. After the script finds
# the keyword, the chomp command removes the new line character (;) and assigns
# the elapsed line to the $time variable. The script repeats the process to
# obtain the start time.

while (<IN>)
{
if (/Elapsed/)
{
chomp($_); #Get time results
$time=$_;
}
if (/Start/) #Get date of tests
{
$temp_date=$_;
}
}

}

#Close the file.
close(IN);

#Determine transfer time values.

```

Continued on page 82

Listing 1 continued

Webmonitoringmodulescript.pl

```

# The script removes colons and normalizes the data.

$hour=substr($time, 28, 2);
$min=substr($time, 31, 2);
$sec=substr($time, 34, 10);
$date1=join ("",substr($temp_date, 32, 6), ",",substr($temp_date, 48, 4));
$time=substr($temp_date, 39, 8);

#Pull out the date.
$date=join (" ", $date1, $time);

#Pull out the total retrieval time.
$total_time= ($hour*3600) + ($min*60) + ($sec);

#Set up baseline numbers against which to check retrieval time.
$total_time1=sprintf("%d",$total_time);
$baseline1=sprintf("%d",$baseline);

#Generate an error if retrieval time exceeds the baseline.
# The script calls event.pl to write to the event log and
# calls an optional module (i.e., optmod.pl) to perform a ping, a tracer,
# and send the log file to the designated systems administrator (aka SuperSA).
if ($total_time1 > $baseline1)
{
system("event.pl");
$prog1=join(" ", "optmod.pl", $array[$count-1]);
system("$prog1");
}

#Enter results into an analysis file.
# The script sets up a log file.
$datatrend="webdata.txt";

#Open the log file.
# If the script can't open the log file,
# the program stops.
open (LOG, ">>webdata.txt")||
die "Cannot open $webdata: $!";

print LOG ($date);
print LOG (" $array[$count-1]);
print LOG (" $total_time \n");

close LOG;

#####
#
#OPTIONAL SQL SECTION
# If you wish to write to a SQL database,
# remove one # symbol from each line in the following section.
#
#####

##Set up ODBC Server to write results to a SQL database.
#use Win32::ODBC;
#
##Script sends ODBC authorization and writes an error message if the authorization fails.
##You need to create an ODBC connection through Control Panel.
#if (!(($data=new Win32::ODBC("DSN=[Your DSN NAME];UID=[User ID the DSN
#requires];PWD=[Password for the DSN ID]"))){
# print "Error connecting to $DSN\n";
# print "error: " . Win32::ODBC::Error();
#}
#
##$sqlStatement = "Insert INTO Baseline Values('$array[$count-1]','$total_time','$date')";
#

```

Continued on page 83

Listing 1 continued*Webmonitoringmodulescript.pl*

```

##Confirm that the database will take the previous statement.
#if ($Data->Sql($SqlStatement))
#{
# print ("SQL failed.\n");
# print "Error: " . $Data->Error() . "\n";
# $Data->Close();
#}

#####
#
#END OPTIONAL SQL SECTION
#
#####

#Let the script check the next URL.
$count++;
}

#####
#
#SCRIPT ENDS HERE
#
#####

```

Monitoring Modules and Windows 2000 Network Load Balancing Service

—by *Curt Aubley and Troy Landry*

The Windows 2000 Network Load Balancing (NLB) service permits multiple servers in a cluster to act as one virtual server. When a request comes to the virtual NIC's IP address, NLB determines which server is the least busy and sends the request to that server. (For more information about NLB, see the Chapter 8 sidebar "Microsoft's Load-Balancing Services.") When you monitor systems with NLB, remember that the Web request physically goes to only one server. For example, suppose that I cluster Server One and Server Two as the Web servers for the mycompany.com Web site and use NLB to load balance the servers. Server One's Microsoft IIS service fails, but the server remains up. Because NLB doesn't recognize that the IIS service is down, NLB continues to route requests to both servers, but Server One can't fulfill the requests that it receives.

To get an accurate view of sites that use NLB technology, you must monitor not only the Web site (i.e., mycompany.com), but you must also independently monitor each server in the NLB cluster (i.e., Server One and Server Two). First, you need to configure IIS so that multiple hosts can access the same Web site: Configure IIS to permit connections through the URL (mycompany.com) and each server's IP address. Second, add the IP addresses of Server One and Server Two to the array variable in webmonitoringmodulescript.pl (see Listing 1). With this enhancement, you can determine whether all your clustered Web servers are working.

The module emulates a customer browsing a Web site: The script makes an HTTP request to the target URLs and grabs the Web pages one at a time. And thanks to Internet standards, you don't need a Web browser to read the Web pages. The example script that Listing 2 shows uses a Perl for Win32 API call: Request(Get, \$url). After the module fetches a Web page, the script evaluates whether the Web server is responding within your performance baseline and whether the URL is available (see Listing 1). The module then logs the Web page's performance. This process can help you ensure that the Web server is responding as planned. If the server returns an error, doesn't respond at the proper performance level, or doesn't respond at all, the monitoring system sends an event to the event logs, writes the data to a flat file, and sends an email alert to our SuperSA's pager.

Listing 2

Poll.pl

```

NONEXECUTABLE
#####
#
#Perl Script: Rapid prototype example for fetching a URL through an HTTP request #
#Use poll.pl with webmonitoringmodulescript.pl
#   to measure Web site response time.
#   The script performs an http get command.
#   If the response is successful, the program continues.
#   If an error occurs, the program waits 10 seconds before continuing.
#
#####

#Start the required Perl modules.
use URI::URL;
use LWP::UserAgent;
use Getopt::Long;

#Set the variable for the monitored Web site.
# @ARGV acquires the Web site's name
#   from the timer program.

$target=join("","http://",@ARGV[0]);

#Identify the target URL.
$url = new URI::URL $target;
$ua = new LWP::UserAgent;
$ua->agent("httpd-time/1.0 ". $ua->agent);

#Request the target URL.
$req = new HTTP::Request(GET,$url);

#Perform the HTTP transaction.
$get = $ua->request($req);

#If the script successfully retrieves the page without errors, the script does nothing.
#   If an error (e.g., 404, 401) occurs, the program waits
#   10 seconds to ensure that the response time exceeds baseline requirements.

if ($get->is_success)
{
}
else
{
#Delay the program for 10 seconds.
sleep (10);
}

#####
#
#SCRIPT ENDS HERE
#
#####

```

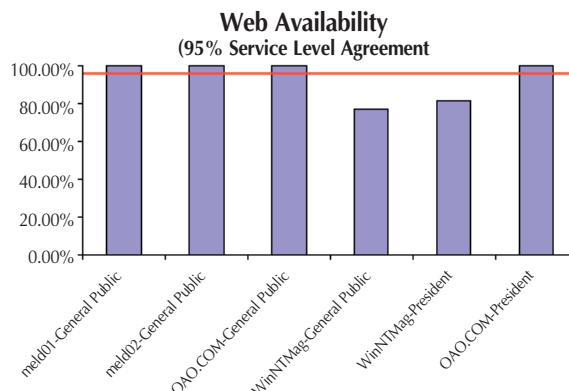
After you install the scripts on your Web-monitoring module system, you need to verify that the scripts work. We suggest that when you run your scripts the first time, you use the `perl.exe -d` command-line option, which lets you step through the script line by line. To test the scripts' availability functions, enter a bogus address into your list of URLs (i.e., the script's array variable). This address will trigger the system to notify you of at least one unavailable URL. To test the scripts' performance functions, you can lower the performance baseline to zero. This baseline will trigger the system to notify you of unacceptable performance for all pages.

After you've confirmed that the monitoring modules are working, you can schedule the monitoring system's operation from Win2K. To schedule the scripts from the Win2K Scheduler, go to Start, Settings, Control Panel, Scheduled Tasks, Add Scheduled Tasks. To decide how often to run the module, ask yourself how often your Web site operates. Typically, you can schedule the module to run every 10 minutes. For business-critical Web sites, you might want to run the module every 5 minutes. After you collect performance and availability data for at least 1 hour, you can move forward to system performance and trend analysis.

System Performance and Trend Analysis

What about situations in which you need to analyze a trend in your performance and availability data (e.g., the CEO complains that your Web site was down last week, and you want to verify that it was)? To accomplish this task, you must store the data. Listing 1 includes code to write data to a SQL Server 7.0 database. (You can use any ODBC-compliant database, such as Access, Lotus Notes, or Oracle.) In the database, we store the URL of the monitored Web site, the site's performance time, and a page-retrieval timestamp. We can then mine this data to create reports or correlate the data and generate graphs in a program such as Microsoft Excel or Seagate's Crystal Reports. From this graph, we can determine when the Web server response time began to lag. Figure 2 shows the Web services' availability—an important metric from a management point of view. You can use graphs such as these to keep your IT team, company president, or the general public up-to-date on the success of your Web implementation.

Figure 2
Web services' availability



Leveraging Win2K Troubleshooting Tools

When your Web performance is poor, you immediately need the appropriate data to track down the problem and determine a solution. Win2K and NT 4.0 include two troubleshooting tools: Ping and Tracert.

If the server doesn't respond adequately to a Web request, you can use Ping to determine general network latency. Ping provides a generic round-trip packet response time to the problematic Web server; this information can help you determine whether the server or network is completely down or whether overall network latency is slowing down Web-page delivery. If a network problem exists, you can use Tracert to determine which network device is unavailable or running slow. Tracert checks the generic packet-response time from the perspective of each network route (i.e., hop). Figure 3 shows an example Tracert output. You can use this information to determine the slowest path between your monitoring module (which represents your customers) and your Web site. You can also use this information to locate a failed network link.

In Listing 1, directly after the comment *#Generate an error if retrieval time exceeds the baseline*, we call another Perl module named `optmod.pl`, which Listing 3 illustrates. `Optmod.pl` generates `ping.exe` and `tracert.exe`, writes the results to a log file, and sends the results to the SuperSA. `Optmod.pl` doesn't include `Pathping`, which is a new tool in Win2K. `Pathping` gives you finer granular control over the number of hops that the module uses to locate your Web site, the time between attempts, the number of tries per network hop, and the available functionality of network services such as layer-2 priority and Resource Reservation Protocol (RSVP) support. To learn about this new network-troubleshooting tool, from the command line type

```
pathping.exe \?
```

If you're running the Win2K Performance Monitor on your Web server, you can review the Performance Monitor's data from the time that your monitoring system experienced Web server problems; you can thus isolate server-side problems. (For more information about Win2K Performance Monitor, see Chapter 1, "Windows 2000 Performance Tools.")

Figure 3
Tracert results

```
Tracing route to www.yahoo.com [204.71.200.68] over a maximum of 30 hops:
  1  <10 ms  <10 ms  <10 ms  205.197.243.193
  2  140 ms   78 ms   47 ms   dca1-cpe7-s6.atlas.digex.net [206.181.92.65]
  3   78 ms   63 ms   94 ms   dca1-core11-g2-0.atlas.digex.net [165.117.17.20]
  4  109 ms  141 ms  125 ms  dca1-core12-pos6-0.atlas.digex.net [165.117.59.98]
  5  109 ms   78 ms  109 ms  s2-0-0.br2.iad.gblx.net [209.143.255.49]
  6   62 ms   63 ms   94 ms  pos9-0-155m.cr1.iad3.gblx.net [208.178.254.117]
  7  219 ms  218 ms  204 ms  pos6-0-622m.cr2.snv.gblx.net [206.132.151.14]
  8   79 ms   78 ms   62 ms  pos1-0-2488m.hr8.snv.gblx.net [206.132.254.41]
  9  125 ms   78 ms   93 ms  bas1r-ge3-0-hr8.snv.yahoo.com [208.178.103.62]
 10   93 ms   94 ms   94 ms  www3.yahoo.com [204.71.200.68]
Trace complete.
```

Listing 3*Event.pl*

```

NONEXECUTABLE
#####
#
#Perl Script: Write an error to the Windows NT event log.
#
#####

#Open the Win32 Perl Module.
use Win32::EventLog;
my $EventLog;

#Lists the data to send to the application log.
my %event=( 'EventID','0xC00001003', 'EventType',EVENTLOG_ERROR_TYPE, 'Category',3,
  'Strings','Performance is unacceptable, check for bottlenecks');

#Opens the EventLog; uses Web Performance as the source.
$EventLog = new Win32::EventLog( 'Web Performance' ) || die $!;

#Writes the event to the event log.
$EventLog->Report(\%event) || die $!;

#####
#
#SCRIPT ENDS HERE
#
#####

```

Proactive System Management

What do you do if the system shows that your Web pages' performance is lacking? Don't wait for customers to complain or go to a competing Web site: Be proactive. The Web-monitoring module is flexible and lets you customize the script for various options. You can use the module script to track the problem and to alert you of other performance or availability failings (i.e., write an event to the Win2K event logs). You can then integrate the monitoring data into a commercial management program (for information about commercial programs' capabilities, see the sidebar "Commercial Tools for System and Network Management"), or you can use the Win2K Event Viewer to review the event logs. You might want to run a command-line program that can call a series of commands. For our example, from within the module we make an ActivePerl system call that runs `optmod.pl`, which in turn calls a freeware program called `Blat` (<http://www.interlog.com/~tcharron/blat.html>), which sends an email alert to our SuperSA's pager and home email account (see Listing 4).

Listing 4

Optmod.pl

```

NONEXECUTABLE
#####
#
#This listing shows examples of different notifications (e.g., ping, tracert) that you can
  configure
#   the module to call and perform. The script writes the results to a log file. The script
  also
#   sends email to a designated person; the log file is the message body.
#
#####

#Sets up the Time Perl Module to insert a time stamp into the log file.
use Time::localtime;

#Sets up the log file into the variable.
$tslogfile="tslogfile.txt";

#Opens the log file. If the log file does not open, the programs dies.
open (LOG, ">>tslogfile.txt")||
die "Cannot open $tslogfile: $!";

#Generates the ping command with the variable from webmonitoringmodulescript.pl
# and outputs the results to ouput.txt.
$target = join(" ", "ping", @ARGV[0], ">output.txt");

#Runs the ping command.
system($target);

#Opens the output file for Ping in order to move the data to the log file.
open(OUTPUT, "output.txt");

#Selects the first line in the output file.
$line=<OUTPUT>;

#Places the timestamp in the log file.
print LOG (ctime());

#Goes through each line in the output file and copies each line to the log file.
while ($line ne"")
{
  print LOG ($line);
  $line=<OUTPUT>;
}

#Closes the output file.
close (OUTPUT);

#Generates the tracert command with the variable from webmonitoringmodulescript.pl
# and outputs it the results to ouput.txt.
$target = join(" ", "tracert", @ARGV[0], ">output.txt");

#Runs the tracert command.
system ($target);

#Opens the output file for tracert in order to move the data to the log file.
open(OUTPUT, "output.txt");

#Selects the first line in the output file.
$line=<OUTPUT>;

#Goes through each line in the output file and copies each line to the log file.
while ($line ne"")

```

Continued on page 89

Listing 4 continued*Optmod.pl*

```

{
    print LOG ($line);
    $line=<OUTPUT>;
}
print LOG ("\n");

#Closes the output and log files.
close (OUTPUT);
close (LOG);

#Sets up the command line to send an email using Blat.
# Blat is freeware.
# You can download Blat from the Internet.
# You must install and configure Blat using an SMTP Server that can send email.

#####
#
#CUSTOMIZATION SECTION A
#
#####

#If you want more than one person to receive the email, separate each name with a comma
# and no spaces.
$emailto = "tlandry\@oao.com, caubley\@oao.com";
$emailsender="tlandry\@oao.com"

$emailsubject = "Warning";

#The email's message body is in a file (i.e., the log file).
$emailmessage= "tslogfile.txt"

#####
#
#END CUSTOMIZATION SECTION A
#
#####

#Generates and runs the blat command line.
$email = join(" ", "blat", $emailmessage, "-s", $emailsubject, "-t", $emailto, "-f", $emailsender, "-q");
system($email);

#####
#
#SCRIPT ENDS HERE
#

```

Act Now

Your customers value your Web services. The Web-monitoring module methodology lets you proactively monitor and track the performance and availability of your key Web sites. Monitoring your Web services is only one portion of an overall system and network management solution, but it is surprisingly easy to implement and provides valuable insight into the delivery of critical Web services. The combination of customer-perspective data, network-performance diagnostic information, and server data can give you a formidable arsenal to track, troubleshoot, and resolve Web-service performance weaknesses before they become overwhelming problems. Be creative with the example scripts—you can easily customize these tools for any enterprise—and keep ahead of your competition.

Part 3: .NET Server Performance

Chapter 10

Exchange 2000 Performance Planning

—by *Tony Redmond and Pierre Bijaoui*

Microsoft Exchange Server 5.5's performance characteristics are well known. In 1996, Exchange Server 4.0 laid down the basic principles of achieving optimum performance through efficient distribution of Exchange-generated I/Os across available disk volumes, and not much has changed since.

True, Microsoft expanded the capacity of the Information Store (IS) to a theoretical limit of 16TB, but the messaging server's essential characteristics remain. The hot spots—the files that generate the heaviest I/O load—are the IS and Directory Store databases, their transaction logs, the Windows NT swap file, and the Message Transfer Agent (MTA) work directory.

Exchange 2000 Server is a different beast. The new messaging server boasts the following improvements:

- The IS architecture has evolved from the simple partitioning of the private and public databases to a point at which, theoretically, the architecture lets you run as many as 90 databases on one server.
- Microsoft IIS handles all protocol access for SMTP, IMAP4, HTTP, Network News Transfer Protocol (NNTP), and POP3, so IIS is more important to Exchange 2000 than it was to earlier versions of Exchange Server.
- A new streaming database can hold native Internet content.
- Windows 2000's Active Directory (AD) replaces the Directory Store.
- A new SMTP-based Routing and Queuing engine replaces the older X.400-based MTA.

These improvements come in a customizable package that third-party solutions will likely extend to provide Exchange 2000 with antivirus, fax, workflow, document management, and other capabilities that aren't part of the base server. Exchange 2000 introduces important architectural changes that have a profound effect on performance. The question that system designers now face is how to best optimize these new features in terms of system and hardware configurations. To answer that question, let's start by investigating Exchange 2000's partitioning of the IS.

Partitioning the IS

Exchange Server 5.5 uses one storage group composed of the private and public stores. Exchange 2000 extends this functionality into storage groups. A storage group is an instance of the Extensible

Storage Engine (ESE) database engine, which runs in the store.exe process and manages a set of databases. Exchange Server 5.5 uses a variant called ESE 97; Exchange 2000 uses the updated ESE 98.

Each Exchange 2000 storage group has a separate set of transaction log files that as many as six message databases share. A message database consists of two files—the .edb file (i.e., the property database) and the .stm file (i.e., the streaming database). The .edb file holds message properties (e.g., author, recipients, subject, priority), which Exchange Server typically indexes for use in search operations. The .edb file also stores message and attachment content that Messaging API (MAPI) clients such as Microsoft Outlook 2000 generate. The .stm file holds native Internet content (e.g., MIME). The ESE manages the seamless join between the .edb and .stm files. The new IS architecture permits as many as 16 storage groups on a server. Exchange 2000 devotes 15 of these storage groups to regular operation and 1 to restoring or recovering databases. Each active group consumes system resources such as virtual memory. In Exchange 2003 Server and Exchange 2000, a server can support as many as 4 storage groups, each of which supports 5 databases, to a maximum of 20 databases per server. This restriction is unlikely to change until Microsoft completes its design of the next generation of Exchange.

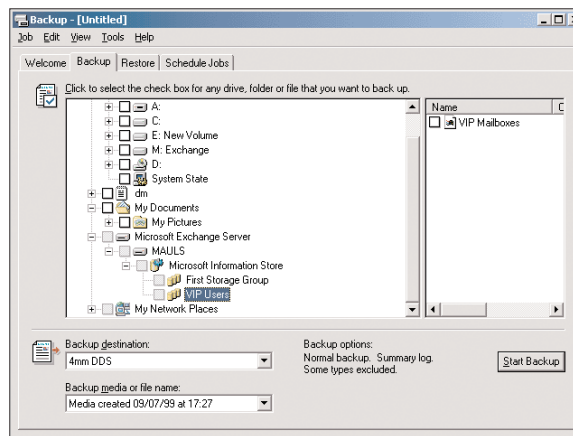
In response to criticism about the original 16GB database limit, Microsoft lifted some internal Exchange Server restrictions to let the database grow as large as available disk space permits. (The limit still exists for the standard version of Exchange Server 5.5.) A larger database lets you allocate greater mailbox quotas to users and lets a server support more mailboxes. However, when a database grows past 50GB, you need to pay special attention to backup and restore procedures, as well as the performance characteristics of the I/O subsystem. Although databases of any size require backups, the larger a database grows, the more challenging it becomes to manage. The ability to store massive amounts of data is useless if poor operational discipline compromises that data or if the data can't get to the CPU for processing because of I/O bottlenecks. In this respect, 50GB is an artificial limit.

Despite the larger store, the practical limit for user mailboxes on one Exchange server—even when you involve Microsoft Cluster Server (MSCS)—remains at about 3000. Hardware vendors have published performance data that suggests the possibility of supporting 30,000 or more simulated users on one 8-way Xeon server. Regardless of that data, if one large database experiences a problem, thousands of users will be unhappy. Large databases are potential single points of failure. Therefore, you won't find many Exchange servers that support more than 3000 mailboxes. The largest single Exchange database in production today is approximately 250GB, so functioning with very large Exchange Server databases is possible—but only when you devote great care to day-to-day operations and system performance. Running Exchange 2003 on Windows Server 2003 lets you exploit the Microsoft Volume Shadow Copy Service (VSS) to take hot snapshots of online Exchange databases, which gives administrators the confidence to run larger databases because they can more quickly recover from failures.

Partitioning the store is interesting from several perspectives. First, by removing a potential single point of failure (i.e., dividing user mailboxes across multiple databases), you can minimize the impact of database failure. Second, you can let users have larger mailbox quotas. Third, you can avoid potential I/O bottlenecks by dividing the I/O load that large user populations across multiple spindles generate. Finally, the advent in Win2K of active-active 2-way and 4-way clustering (which Exchange 2000 supports) increases overall system resilience through improved failovers.

On an operational level, Microsoft has gone to great lengths to ensure that multiple databases are easier to manage. As Figure 1 shows, Win2K's Backup utility can back up and restore individual storage groups and databases rather than process the entire IS. Third-party backup utilities (e.g., VERITAS Software's Backup Exec, Legato Systems' NetWorker, Computer Associates'—CA's—ARCserve*IT*) support Exchange 2003 and Exchange 2000; most of these utilities also include VSS support. Using the Microsoft Management Console (MMC) Exchange System Manager snap-in, you can dismount and mount an individual database for maintenance without halting all store operations, as Exchange Server 5.5 does. For example, suppose that the Applied Microsoft Technologies database in the First Storage Group is dismounted, as Figure 2 shows. Right-clicking the database brings up a context-sensitive menu in which you can choose the All Tasks, Mount Store option to bring the store online. Generally, you'll find that Exchange 2000 database operations are easier than Exchange Server 5.5 operations because the databases are smaller, and because you can process operations against multiple databases in parallel.

Figure 1
Processing storage groups with Win2K's Backup utility

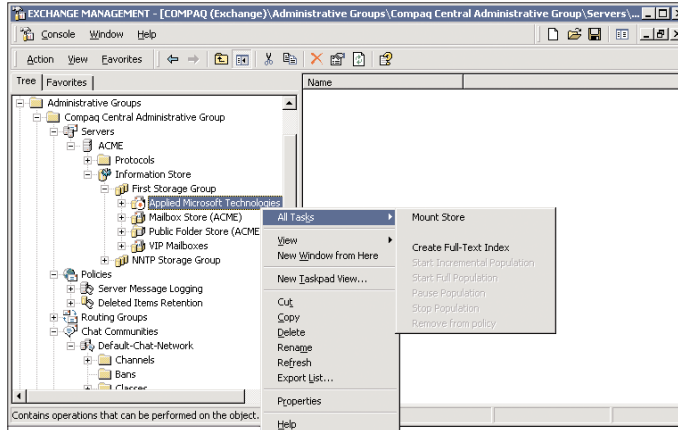


However, Exchange 2000 uses a single storage group by default. Out of the box, or following an upgrade from Exchange Server 5.5, Exchange 2000 operations proceed exactly as they do in Exchange Server 5.5. To take advantage of the new features and gain extra resilience, you need to partition the store, and you can't partition the store until you carefully consider database placement, file protection, and I/O patterns.

In terms of I/O, the Exchange Server 5.5 store is a set of hot files. All mailbox operations flow through `priv.edb`, whereas all public folder operations channel through `pub.edb`. If you partition the store and create multiple databases, you need to consider how to separate I/O across a system's available disks in such a way that you increase performance and protect your data. I don't mean to suggest that you ought to rush out and buy a set of large disks. Increased information density means that 72GB disks are now available, and the price per megabyte is constantly dropping. However, you won't attain good I/O performance by merely increasing disk capacity. The number of disks, as well as the intelligent placement of files across the disks, is much more

important. CPU power increases every 6 months, and 8-way processors are now reasonably common. However, even the most powerful system can't surpass a humble single processor if its CPUs can't get data. Therefore, storage configuration is crucial to overall system performance.

Figure 2
Mounting an individual database in a storage group



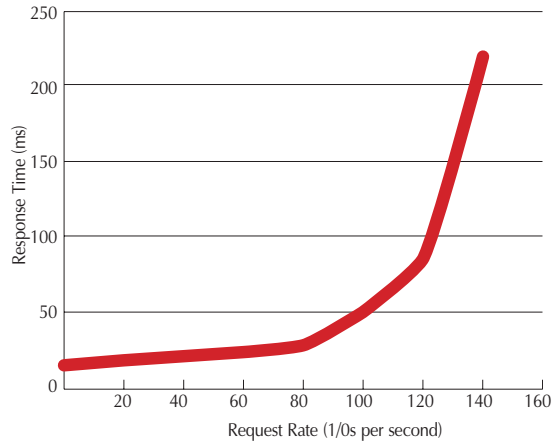
Storage Performance Basics

Figure 3 illustrates a typical disk response time. As the number of requests to the disk increase, the response time also increases along an exponential curve. Disk queuing causes this behavior, and you can't do anything about it. Any disk can service only a limited number of I/Os, and I/O queues accumulate after a disk reaches that limit. Also, the larger the disk, the slower it typically is. For example, don't expect a 50GB disk to process more than 70 I/O requests per second. Over time, disks might spin faster, get denser, and hold more data, but they can still serve I/O at only a set rate, and that rate isn't increasing.

Transactions that the ESE applies to the Exchange Server databases use a two-phase commit (2PC) process, which ensures that all database changes that are part of a transaction occur. A transaction modifies database pages as it proceeds, and the transaction log buffer stores the changes. To ensure the integrity of the database, a special memory area called the Version Store holds the original page content. When the transaction commits, the database engine writes the page changes from the transaction log buffers to the transaction log files, then removes the pages from the Version Store. If the ESE must abort the transaction, any changes related to the transaction will roll back.

Writing to the transaction log is the performance-critical part of the process. The IS orders the pages in memory and commits them in an efficient, multithreaded manner, but the writes to the transaction log are sequential. If the disk holding the logs is unresponsive, delay will occur and Exchange Server won't log transactions quickly. Therefore, you need to ensure that the I/O path to the disk holding the transaction logs is as efficient as possible. Note that the same performance characteristic is evident in AD, which uses a modified version of the ESE.

Figure 3
I/O request characteristics



For optimum performance, you need to place the transaction logs on the most responsive volume or the device with optimal write performance. Your aim is for Exchange Server to write transactions to the log files as quickly as possible. The typical—and correct—approach is to locate the log files on a disk separate from the disk that holds the database. To ensure data resilience, the logs and database must be separate. Remember that an operational database is never fully up-to-date. The transaction logs contain transactions that the IS might not have committed yet. Therefore, if the disk holding the database fails, you need to rebuild the database (by restoring the most recent full backup) and let Exchange Server replay the outstanding transactions from the logs that users created since the backup. Clearly, if the transaction logs and the database reside on the same disk and a fault occurs, you're in big trouble. To ensure resilience, mirror the transaction log disk. Don't use RAID 5 on the volume that hosts transaction logs, because it slows down the write operations to the logs and degrades overall system performance. (For more information about RAID 5, see the Chapter 3 sidebar “Why Is RAID 5 Slow on Writes?”) RAID 0+1 (i.e., striping and mirroring) delivers the best write performance for larger volumes and is highly resilient to failure. However, RAID 0+1 is typically too expensive in terms of allocating disks to transaction logs. RAID 1 (i.e., mirroring), which provides an adequate level of protection balanced with good I/O performance, is the usual choice for volumes that host transaction logs. Never use RAID 0 for a disk that holds transaction logs—if one disk fails, you run the risk of losing data.

Each storage group uses a separate set of transaction logs. You need to separate the log sets as effectively as possible on multiple mirrored volumes. However, one storage array can support only a limited number of LUs, so compromise might be necessary. On small servers, you can combine log sets from different storage groups on one volume. This approach reduces the amount of storage the server requires at the expense of placing all your eggs in one basket. A fault that occurs on the volume affects all log sets; therefore, you need to take every storage group offline.

Exchange Server databases' I/O characteristics exhibit random access across the entire database file. The IS uses parallel threads to update pages within the database, so a multispinde volume

helps service multiple concurrent read or write requests. In fact, the system's ability to process multithreaded requests increases as you add more disks to a volume.

Since Exchange Server's earliest days, most system designers have recommended RAID 5 protection for the databases. RAID 5 is a good compromise for protecting storage and delivering reasonable read/write performance without using too many disks. However, given the low cost of disks and the need to drive up I/O performance, many high-end Exchange Server 5.5 implementations now use RAID 0+1 volumes to host the databases. Expect this trend to continue in Exchange 2000. Although you can now partition I/O across multiple databases, the number of mailboxes that an individual server supports will likely increase, thereby driving up the total generated I/O. Large 4-way Exchange 2000 clusters need to be able to support as many as 10,000 mailboxes and manage 200GB to 400GB of databases across multiple storage groups. In terms of write operations, RAID 0+1 can perform at twice the speed of RAID 5, so any large Exchange 2000 server needs to deploy this configuration for database protection.

To yield the best performance for both transaction log and database writes, use the write cache on the storage controller. However, don't use the write cache unless you're sure that you've adequately protected the data in the cache against failure and loss. You need to mirror the cache and use battery backup to protect it from power failure. You also need to be able to transfer the cache between controllers in case you want to replace the controller. Read operations to access messages and attachments from the database typically retrieve information across the entire file, so controller read cache doesn't help performance. The ESE performs application-level caching.

Don't attempt to combine too many spindles in a RAID 5 volume. Each time a failure occurs, the entire volume rebuilds. The duration of the rebuild is directly proportional to the number and size of disks in the volume, so each disk you add increases the rebuild time. Most volume rebuilds occur in the background, and the volume remains online. However, if another failure occurs during the rebuild, you might experience data loss. Therefore, reducing rebuild time by reducing the number of disks in the volume set is good practice. Deciding the precise number of disks to place in a volume can be a balancing act between the size of the volume you want to create, the expected rebuild time, the data that you want to store on the volume, and the expected mean time between failures. If you want to store nonessential data on a large volume for the sake of convenience, you can combine many disks into the volume. However, an Exchange Server database tends to be sensitive to failure. I recommend erring on the side of caution and not placing more than 12 disks into the volume.

Examination of Exchange 2000 servers' I/O pattern reveals some interesting points, some of which differ significantly from Exchange Server 5.5 patterns. The streaming database delivers sparkling performance to IMAP4, POP3, and HTTP clients because they can store or retrieve data much faster from the streaming database than they can from the traditional Exchange Database (EDB). Clients access the streaming database through a kernel-mode filter driver called the Exchange Server Installable File System (ExIFS). Like the EDB, the ExIFS processes data in 4KB pages. However, the ExIFS can allocate and access the pages contiguously, whereas the EDB merely requests pages from ESE and might end up receiving pages that are scattered across the file. You won't see a performance advantage for small messages, but consider the amount of work necessary to access a large attachment from a series of 4KB pages that the IS needs to fetch from multiple locations. Because its access is contiguous, the streaming database delivers much better performance for large files. Interestingly, contiguous disk access transfers far more data (as much

as 64KB per I/O); therefore, to achieve the desired performance, the storage subsystem must be able to handle such demands. Advances in storage technology often focus on the amount of data that can reside on a physical device. As we move toward the consolidation of small servers into larger clusters, I/O performance becomes key. System designers need to focus on how to incorporate new technologies that enable I/O to get to CPUs faster. Exchange 2000 is the first general-purpose application to take full advantage of the fibre channel protocol, which delivers transfer rates as high as 100MBps. Systems that support thousands of users must manage large quantities of data. The ability to store and manage data isn't new, but advances such as fibre channel now let system configurations attain a better balance between CPU, memory, and storage.

Storage Configuration

Most Exchange Server 5.5 servers use SCSI connections. As a hardware layer, SCSI demonstrates expandability limitations, especially in the number of disks that you can connect to one SCSI bus and the distance over which you can connect the disks. As Exchange servers get larger and handle more data, SCSI becomes less acceptable.

As I noted, fibre channel delivers high I/O bandwidth and great flexibility. You can increase storage without making major changes to the underlying system, and fibre channel storage solutions that extend over several hundred meters are common. Win2K's disk-management tools simplify the addition or expansion of volumes, so you can add capacity for new storage groups or databases without affecting users. Better yet, fibre channel implementations let servers share powerful and highly protected storage enclosures called Storage Area Networks (SANs). For most individual servers, a SAN is an expensive data storage solution. However, a SAN makes sense when you need to host a large corporate user community by colocating several Exchange 2000 servers or clusters in a data center. You need to weigh the advantages of a SAN, as well as its additional cost, against the advantages and disadvantages of server-attached storage. A SAN can grow as storage requirements change. Its adaptability and ability to change without affecting server uptime might be a crucial factor in installations that need to support large user communities and deliver 99.99 percent or greater system availability.

Example Configuration

Let's put some of the theory I've discussed into the context of an example Exchange 2000 system configuration. Assume that your server must support 3000 mailboxes and you want to allocate a 100MB mailbox quota. This size might seem large, but given the ever-increasing size of messages and lower cost of storage, installations are raising mailbox quotas from the 10MB-to-20MB limits imposed in the early days of Exchange Server to 50MB-to-70MB limits. A simple calculation (i.e., mailboxes \times quota) gives you a storage requirement of 300GB. This calculation doesn't consider the single-instance ratio or the effect of the Deleted Items cache, but it serves as a general sizing figure.

A casual look at system configuration options suggests that you can solve your storage problem by combining seven 50GB disks into a RAID 5 volume. Although this volume would deliver the right capacity, the seven spindles probably couldn't handle the I/O load that 3000 users generate. Observation of production Exchange Server 5.5 servers reveals that each spindle in a RAID 5 volume can handle the I/O load of approximately 200 mailboxes. Spindles in a RAID 0+1 volume push the supported I/O load up to 300 mailboxes. If you apply these guidelines to our Exchange 2000 example, you'll need 15 spindles (i.e., physical disks) in a RAID 5 volume, or 10 spindles in a RAID 0+1 volume, to support the expected load.

Exchange Server 5.5 has one storage group, so splitting the I/O load across multiple volumes is difficult. Exchange 2000 lets you split the 3000 mailboxes across four storage groups. If you use one message database in each storage group, each database is 75GB, which is unwieldy for the purpose of maintenance. To achieve a situation in which each database supports 150 users and is about 15GB, you can split the users further across five message databases in each storage group.

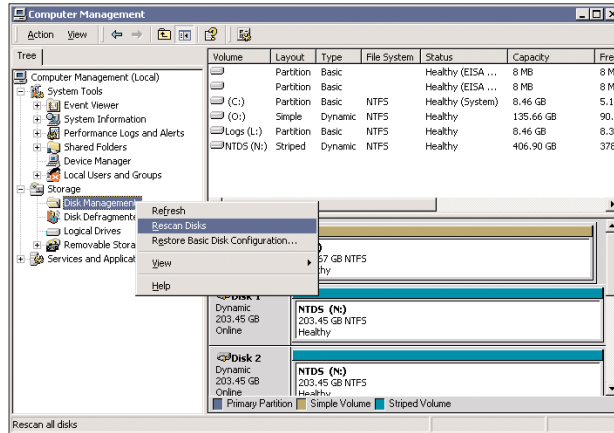
Splitting users this way affects the single-instance storage model that Exchange Server uses. Single-instance storage means that users who receive the same message share one copy of the message's content. But single-instance storage extends across only one database. After you split users into separate databases, multiple copies of messages are necessary—one for each database that holds a recipient's mailbox. However, experience shows that most Exchange servers have low sharing ratios (e.g., between 1.5 and 2.5), and dividing users across multiple databases produces manageable databases that you can back up in less than 1 hour using a DLT. Also, a disk failure that affects a database will concern only 150 users, and you can restore the database in an hour or two. Although four storage groups, each containing five databases, might seem excessive, this example realistically represents the types of configurations that system designers are now considering for early Exchange 2000 deployments.

Each storage group contains a set of transaction logs. Recalling the basics of disk configuration, you might think that you need five mirror sets for the logs and five RAID 5 or RAID 0+1 sets for each set of databases. Managing such a large amount of storage from a backplane adapter—you'd probably double the physical storage to 600GB because you don't want to fill disks and you want room to grow—is impractical because you'd probably encounter a limit to the number of disks you can connect. Also, a system this large is a candidate for clustering, so you need a solution that can deliver the I/O performance, handle the number of spindles required to deliver the capacity, and support Win2K clustering. For all Exchange 2000 clusters, consider using a SAN either to share load between servers that use the Win2K active-active clustering model or to benefit from advanced data-protection mechanisms such as online snapshots and distant mirroring. If you need to add users, you simply create a new storage group, create a new volume in the SAN, and mount the database without interrupting service. The Win2K Disk Administrator can bring new disks online without requiring a system reboot. Generally speaking, Win2K greatly improves disk administration—a welcome advance given the size of volumes in large configurations. Figure 4 shows the Disk Management MMC snap-in dealing with some very large volumes, including one whose size is 406.9GB! This volume should be large enough to keep many Exchange Server databases happy.

Each database or storage group doesn't require its own volume. You can divide the databases across available volumes as long as you keep an eye on overall resilience against failure and don't put too many databases on the same volume. Exchange 2000 clusters use storage groups as cluster resources, so you need to place all the databases for a storage group on the same volume. This placement ensures that the complete storage group and the disks holding the databases will fail over as one unit.

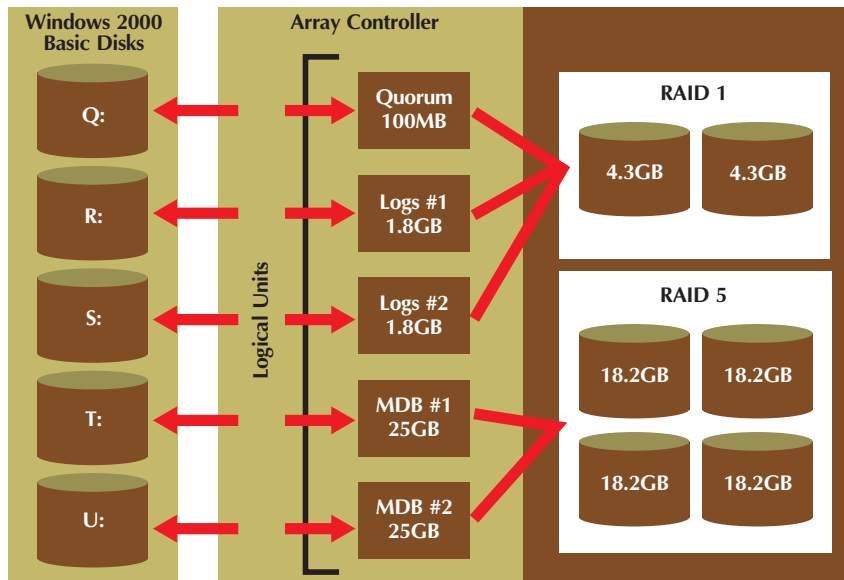
Transaction logs that handle the traffic of 600 users will be busy. In such a configuration, you could create four separate RAID 1 sets for the logs. If you use 9.2GB disks, you'll need eight disks in four volumes. A 9GB volume has more than enough space to hold the transaction logs of even the most loaded server. For best performance, don't put files that other applications use on the transaction log volumes.

Figure 4
Viewing the MMC Disk Management snap-in



Systems that run with more than a couple of storage groups can group transaction logs from different storage groups on the same volumes. You don't want to create too many volumes only for the purpose of holding transaction logs. Figure 5 illustrates how you might lay out databases and transaction logs across a set of available volumes.

Figure 5
Sample disk configuration



Disks that you use in Win2K clusters must be independently addressable, so if you want to consider a clustered system, you need to use hardware-based partitions, which let the controller present multiple LUs to the cluster or server, as well as use fewer disks. Clusters require a disk called the *quorum disk* to hold quorum data. I recommend using a hardware partition for this data; the actual data hardly ever exceeds 100MB, and dedicating an entire physical disk is a waste.

If you use RAID 5 to protect the four storage groups, you'll need five 18.2GB disks for each volume. You can gain better I/O performance by using $9 \times 9.2\text{GB}$ disks. The volumes have 72GB capacity, which is more than the predicted database size ($3 \times 15\text{GB} = 45\text{GB}$). You need the extra space for maintenance purposes (e.g., rebuilding a database with the Eseutil utility) and to ensure that the disk never becomes full. Stuffing a disk to its capacity is unwise because you'll probably reach capacity at the most inconvenient time. Exchange Server administrators typically find that databases grow past predicted sizes over time. After all, databases never shrink—they only get bigger as users store more mail.

Expanding Boundaries

The changes that Microsoft has introduced in the Exchange 2000 IS offer system designers extra flexibility in hardware configurations. Partitioning the IS means that you can exert more control over I/O patterns. I'm still investigating the opportunities that SANs, Exchange 2000 clusters, and different storage group configurations offer, but clearly the number of mailboxes that one production server can support will climb well past Exchange Server 5.5's practical limit of 3000.

Chapter 11

3 Basics of Exchange Server Performance

—by *Tony Redmond and Pierre Bijaoui*

A Microsoft Exchange Server administrator's job is all about getting the mail through, keeping the system running, and giving users the impression that Exchange Server is 100 percent dependable. To achieve these goals, you need to concentrate on several performance fundamentals: hardware, design, and operation. These three basics come into play for all Exchange Server organizations, whether you're running Exchange 2000 Server or Exchange Server 5.5. Reliable and capable hardware is the foundation of Exchange Server performance, but the design into which you place that hardware and the way that you operate the hardware are just as important when you want to achieve maximum performance over a sustained period.

Fundamental No. 1: Hardware

An adequate and balanced hardware configuration provides the platform for good Exchange Server performance. To meet your performance goals, your servers must strike a balance among these three essential components: CPU, memory, and disk.

Server configuration essentially comes down to how many CPUs the server has and how fast they are, how much memory the server has, and what type of disk subsystem the server uses. Given the speed of today's CPUs and the relatively low cost of memory and disks, you'll be hard pressed to underconfigure a server. Even the smallest off-the-shelf server—with, for example, a 700MHz CPU, 256MB of RAM, and three 18GB disks with a RAID 5 controller—can support several hundred Exchange Server mailboxes. (For an explanation of the benchmarks that vendors use in server-sizing guides, see the sidebar "Making Sense of Benchmarks.") High-end servers (i.e., servers that support more than 1000 mailboxes or servers that are designed for high availability) present more of a challenge, mostly because of the different combinations of hardware that you can apply to provide the desired performance level.

CPU Statistics

Most Exchange Server machines are equipped with only one CPU, but hardware vendor tests demonstrate that Exchange 2000 scales well using SMP. In Compaq tests, an increase from two processors to four processors led to a 50 percent improvement in capacity; an increase from four processors to eight processors also led to a 50 percent increase. Considering SMP's overhead, this performance is excellent and testifies to the Exchange Server code base's multiprocessing capabilities. Exchange 2000 exhibits good SMP support, with 4-way processors being the most popular choice for deployments. Exchange 2003 Server is even better, and 8-way processors will get more attention as Exchange 2003 deployments proceed.

CPUs get faster all the time. Level 2 cache is particularly important for good Exchange Server performance, so use systems with as much Level 2 cache as possible. This special area of the

processor caches instructions, and its size depends on the processor's model. Intel's Pentium III processors' typical Level 2 cache size is 256KB, whereas the company's Xeon processors can have Level 2 caches as big as 2MB. Currently, Xeon processors are the best platform for Exchange Server because of their cache size and the nature of the chipset, which Intel has optimized for server applications.

Keep in mind that your goal should be to saturate your machine's processing capability; to do so, you need to remove any bottlenecks. Typically, you should first address any memory deficiencies, then add storage subsystem capacity, then increase network speed (100Mbps should be sufficient for all but the largest Exchange Server systems). If you still need to increase CPU saturation after you've tuned these components, you can add more processors or increase the processors' clock speed.

Making Sense of Benchmarks

—by *Tony Redmond and Pierre Bijaoui*

Major hardware vendors publish sizing guides to give you an idea of their products' basic performance capabilities. These guides report that during a simulation, a certain configuration supported so many mailboxes for a specific workload. Messaging API (MAPI) Messaging Benchmark (MMB) and MMB2 are the two workloads that vendor benchmarks commonly use.

MMB is an older set of user activities (e.g., create and send messages, process calendar appointments) that generates unrealistic results. For example, you'll find published MMB results that reflect hardware support for tens of thousands of mailboxes. In 2000, Microsoft introduced MMB2 as a more accurate measurement of a user workload in a production environment. Typically, MMB2 supports one-seventh of the mailboxes that MMB supports. For example, an MMB result of 14,000 mailboxes would translate into a much more realistic MMB2 figure of 2000 mailboxes.

Keep in mind that these numbers are for simulated users; human beings might not produce the same results. Humans are unpredictable—think of all those users who happily swap messages containing large attachments. Benchmarks also tend to ignore network overhead, replication activity, and all the other demands for system resources that occur in production environments. Still, when you understand the benchmark game, the generated figures for different configurations can give you a useful starting point for your server designs.

Memory and Cache

To optimize the memory demands that Exchange Server makes on the OS, Exchange 2000 and Exchange Server 5.5 both implement a mechanism called Dynamic Buffer Allocation. DBA monitors the level of activity on a server and adjusts the amount of virtual memory that Exchange Server's database engine (i.e., the Extensible Storage Engine—ESE) uses. Exchange Server

implements DBA as part of the Store (Microsoft's generic term for the Information Store—IS), so you'll sometimes see the Store process grow and contract quite dramatically on a server that experiences intermittent periods of heavy demand. You'll also see the Store fluctuate on Exchange Server systems that run other applications—especially database applications such as Microsoft SQL Server—when multiple applications request memory resources.

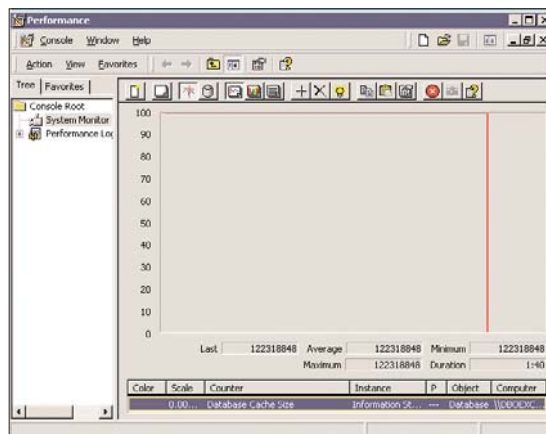
On servers that run only Exchange Server and therefore experience a consistent level of demand, the Store process tends to grow to a certain level, then remain constant. (Exchange 2000 machines aren't in this group because all Exchange 2000 servers also run Microsoft IIS to support Internet protocol access.) Don't let a large Store process worry you—it simply means that DBA has observed that no other active application wants to utilize memory and so has requested additional memory to cache database pages. The net result is a reduction in the amount of system paging; this reduction aids performance.

To see how much memory the ESE is utilizing, you can use the Database Cache Size (Information Store) counter on the Performance Monitor's Database object. Figure 1 shows a typical value from a small Exchange 2000 server under moderate load. This server has 256MB of RAM but has allocated approximately 114MB of virtual memory to the ESE. Note that the amount of virtual memory that the ESE uses will increase as you load more storage groups (SGs) and databases—one reason why experienced systems administrators stop to think before they partition the Store.

The ESE uses RAM to cache database pages in memory. When a server doesn't have enough physical memory, the ESE caches fewer pages and increases disk access to fetch information from the database. The result is an increased strain on the I/O subsystem, which must handle more operations than usual. You might conclude that you need to upgrade the I/O subsystem, probably installing additional disks and redistributing I/O. However, increasing the amount of RAM available to the server is usually more cost-effective, so always consider this step before any other action.

Figure 1

Monitoring the ESE's memory utilization



Maximum Disk Performance

Exchange Server is essentially a database application, and like all other database applications, it generates a considerable I/O load and stresses both disk and controller components. Any performance-management exercise must therefore include three steps: Properly distribute the source of I/O (i.e., the files involved in messaging activity) across disks, ensure the correct level of protection for essential files, and install the most appropriate hardware to handle disk activity.

The first step is to separate transaction-log sets and database files, even on the smallest system. This procedure ensures a maximum chance of maintaining information after a disk crashes. If you place logs on one spindle and the database on another, the loss of one won't affect the other, and you can use backups to recover the database to a known state. If you place logs and database on the same spindle, however, a disk failure inevitably results in data loss.

The second step is to properly protect the transaction-log sets. The data in the transaction logs represents changes in information between the database's current state (i.e., pages in RAM) and its state at the last backup (i.e., pages on disk). Never place transaction logs on an unprotected disk; use RAID 1 volumes with controller-based write-back cache, which provides adequate protection without reducing performance. Separate transaction-log sets on servers running Exchange 2000 Enterprise Server with multiple SGs. The ideal situation is to assign a separate volume for each log set.

The third step is to give as much protection as possible to the Store databases. Use RAID 5 or RAID 0+1 to protect the disks that hold the mailbox and public stores. RAID 5 is the most popular approach (Microsoft has recommended a RAID 5 approach since it launched Exchange Server 4.0), but RAID 0+1 is becoming more common because it delivers better I/O performance and avoids the necessity of enabling a write-back cache. Compaq has performed tests that demonstrate that one spindle in a RAID 0+1 set can support the I/O that 250 active Exchange Server users generate. Thus, a large server that supports 2500 active users would need a 10-disk RAID 0+1 set. For maximum performance, concentrate on each disk's I/O capacity rather than on how many gigabytes it can store.

After you've equipped your server with sufficient storage, you need to monitor the situation to ensure that the server delivers the desired performance. To get a complete picture, monitor each device that hosts a potential hot file (i.e., a file that generates most of the I/O traffic). For Exchange Server 5.5 machines, these devices include those that hold the Store, the Message Transfer Agent (MTA), and the Internet Mail Service (IMS) work files. For Exchange 2000, take the same approach but also monitor the device that hosts the SMTP mail drop directory, and consider that the Store might be partitioned into multiple databases, each of which you need to monitor.

To quickly assess storage performance for an Exchange Server machine, you can use several Performance Monitor counters that examine disk response times and queue lengths. (For information about Performance Monitor and similar tools, see Chapter 1, "Windows 2000 Performance Tools," or Chapter 2, "NT Performance Tuning.") To monitor response time, you can use any of the following Physical Disk object counters:

- Avg. disk sec/Read
- Avg. disk sec/Write
- Avg. disk sec/Transfer

For acceptable performance, each device should perform random I/O operations in 20 milliseconds (ms) or less. Sequential operations should take place in less than 10ms. You need to take

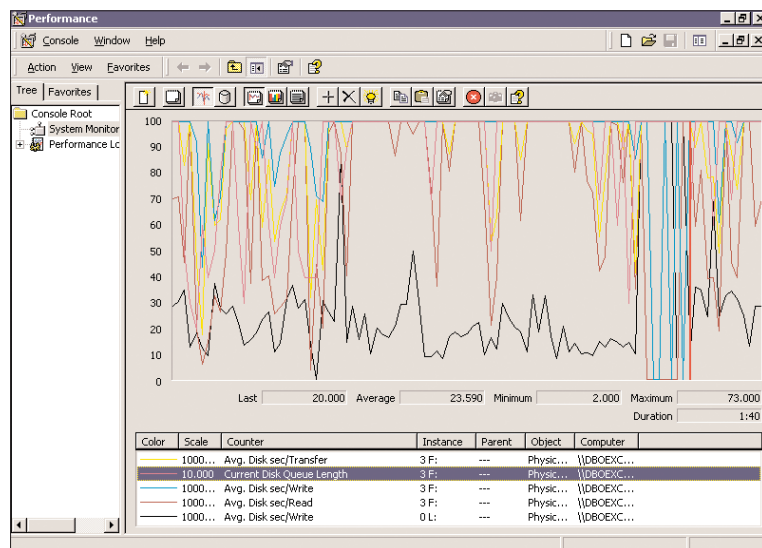
action when devices exceed these thresholds. The easiest course is to move files so that less heavily used devices take more of the load. The alternatives are more drastic: Relocate mailboxes, public folders, or connectors to another server, or install additional disks and separate the hot files.

You should also monitor the Pages Read and Pages Writes performance counters for the Memory object because they indicate the hard page faults that result in a disk access. The sum of these two values shouldn't exceed 80ms (i.e., roughly the I/O limit for one disk drive). If the sum exceeds 80ms, you should add more physical memory to your server and possibly locate the page file on a fast drive (although the latter solution is less efficient than adding memory).

The Current Disk Queue Length counter for the Physical Disk object reports the number of outstanding operations to a particular volume. Although Win2K reports separate queue lengths for read and write operations, the current aggregate value is what matters. A good rule of thumb is that the queue length should always be less than half of the number of disks in a volume. For example, if you have a 10-disk RAID volume, the queue length should be less than 5. Your aim is to ensure that the volumes have sufficient headroom to handle peak demand. Consistently high queue-length values are a signal that the volume can't keep up with the rate of I/O requests from an application and that you need to take action.

Figure 2 shows monitoring results for the F disk (on which the Store databases reside) and the L disk (on which the log files reside) on an Exchange 2000 server. The Current Disk Queue Length counter for the Physical Disk object shows that the Store disk is under considerable strain: an average queue length of 23.59 I/O operations and a peak of 73 I/O operations. This I/O load occurred when a user created some large public-folder replicas (e.g., one folder contained 32,000 items) in a public store. Exchange Server generated 230 log files—1.3GB of data—during the replication process. Users will notice such a load because Exchange Server will queue any request they make to the Store, and Exchange Server response won't be as snappy as usual.

Figure 2
Monitoring disks



Fundamental No. 2: Design

Exchange Server machines fit within a design that determines each server's workload: the number of mailboxes the server supports, whether it hosts a messaging connector or public folders, or whether it performs a specific role (e.g., key management). This design also needs to accommodate how much data the server must manage and how available you want the server to be.

You expect data requirements to increase as servers support more mailboxes, but you must also deal with the "pack rat syndrome." Users love to keep messages, so they appeal to administrators for larger mailbox quotas. Disks are cheap, so increasing the quota is the easiest response. Default quotas for Exchange Server organizations have gradually increased from 25MB in 1996 to about 100MB today. Some administrators manage to keep quotas smaller than 100MB and still keep their users happy (which is a feat in itself). Other administrators permit quotas larger than 100MB and put up with the need for more disk space and longer backup times.

Slapping a few extra drives into a cabinet and bringing them online might increase an Exchange Server machine's available storage but isn't a good way to ensure performance. Every ad hoc upgrade hurts server availability, and the chance always exists that something will go wrong during an upgrade procedure. A better approach is to plan out the maximum storage that you expect a server to manage during its lifetime, then design your storage infrastructure accordingly. If you're using Exchange 2000, your design also needs to take into consideration the interaction between Exchange Server and Active Directory (AD—for details about this relationship, see the sidebar "Exchange 2000 and AD").

Exchange 2000 and AD

—by Tony Redmond and Pierre Bijaoui

Any discussion of Microsoft Exchange 2000 Server performance necessitates mention of Active Directory (AD). Exchange 2000 depends on AD as the receptacle for organizational configuration information, the source of the Global Address List (GAL), and the basis for all routing decisions. A Global Catalog (GC) server must be in close network proximity to every Exchange 2000 server to permit Exchange 2000 to query AD as quickly as possible.

Performing a GC lookup for each and every address that Exchange 2000 needs to check wouldn't make sense, so each server maintains a cache of recently accessed directory information. Exchange 2000 always checks this cache first, then proceeds with a lookup against the GC only if it can't find the data. Although the cache can hold many thousands of items, fast access to a GC is still a prerequisite for good performance.

In small implementations, one server might run Exchange 2000 and also act as a GC. (If your Windows 2000 deployment spans only one domain, all domain controllers—DCs—are GCs.) GC placement probably won't be a concern in small environments but will be an important decision for all enterprise or distributed deployments.

New and upcoming Exchange 2003 and Exchange 2000 and hardware features offer capabilities that could increase the power of your organization's availability. Exchange's improved clustering support makes clustering a more attractive option, especially in Exchange 2003. New hardware capabilities such as Storage Area Networks (SANs) make systems more resilient with disk failures, which are Exchange Server's Achilles' heel. True online snapshot backups, now available with VSS in Windows Server 2003 and Exchange 2003, will increase server availability by making recovery from database corruption easier and faster.

Fundamental No. 3: Operations

Flawed operational procedures can render useless the best possible hardware and most comprehensive design. Your organization is only as good as its weakest link, and all too often you don't discover that link until an operational problem occurs.

Carefully observing your production systems is the key to good operations. The OS and Exchange Server write information to the event logs. You need to either scan that information manually or use a product such as NetIQ's AppManager to watch for events that point to potential problems. For example, if the Store isn't absolutely satisfied that the database engine has fully committed a transaction to a database, Exchange Server generates a -1018 error in the Application log. In versions earlier than Exchange Server 5.5, a -1018 error might be the result of a timing glitch between a disk controller and the OS, but Exchange 2000 and Exchange Server 5.5 include code to retry transactions and so overcome any intermittent problems. A -1018 error in Exchange 2000 or Exchange Server 5.5 could mean that a hardware failure has occurred and the database is corrupt. If you don't check the hardware and restore the database from backup, Exchange Server might generate more -1018 errors as the database becomes more and more corrupt and eventually fails. Of course, any backups you made during this time contain corrupted data. The Eseutil utility might be able to fix minor corruptions, but it can't fix the fundamental data loss that a hardware failure causes, so a -1018 error is a serious event.

Many other daily events provide insight into the proper workings of an Exchange Server system. For example, you can find information in the Application log about background defragmentation, an operation that the Store usually performs automatically in the middle of the night. Exchange Server logs events that report the start of a defragmentation pass (event ID 700), the end of the pass (event ID 701), and how much free space exists in the database after the pass (event ID 1221).

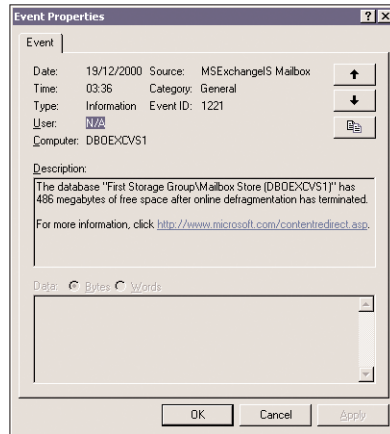
The event that Figure 3 shows reports 486MB of free space (i.e., roughly 7.2 percent of the database) after defragmentation of a 6.67GB Exchange 2000 mailbox store. Exchange 2000 will use this space to store new messages and attachments as they arrive.

Although you can use Eseutil to perform an offline rebuild and shrink the database, you should do so only when you can recover a significant amount of free space (i.e., more than 30 percent of the database) and you either need the disk space or want to reduce backup time. Because an offline rebuild prevents users from accessing email and takes a long time—at least 1 hour per 4GB of data, plus time for backups before and after the rebuild—you're better off buying more disks or considering faster backup devices than running Eseutil.

The Application log is also the place to look for signs of MTA errors, details of incoming replication messages, and situations in which someone has logged on to another user's mailbox using a Win2K or Windows NT account that isn't associated with that mailbox. (Some antivirus products

provoke the latter type of event when they log on to mailboxes to monitor incoming messages for any attached viruses.)

Figure 3
Results of a defragmentation pass



Exchange Server also logs backup-related events. Good systems administrators are paranoid about backups and always ensure that they successfully begin, process all expected data, and finish. According to Murphy's Law, backup tapes will become unreadable at the worst possible time and any readable backup tapes you fetch when you're under pressure will contain corrupt data.

Backups are the single most important and fundamental task for an Exchange Server administrator. Anyone can lose system availability because of a hardware problem, but your boss won't easily forgive extended system downtime or data loss when incorrect or lazy procedures result in a backup failure.

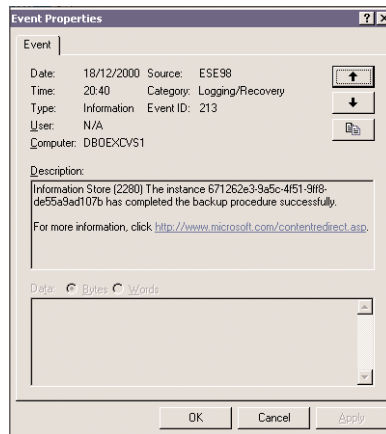
Losing system availability is the ultimate performance failure. Your goal should be to minimize the negative effects of any problem that requires you to restore data. The only way to meet this goal is to be scrupulous about observing the following requirements:

- Make daily backups, and confirm their success. Figure 4 shows event ID 213, which Exchange Server writes to the Application log at the end of a successful backup.
- Know how to restore a failed Exchange Server database. Take note: Exchange 2000 makes this task both more complex and easier than it is for Exchange Server 5.5. On the one hand, Exchange 2000 Enterprise supports multiple databases, so you might need to restore more than one database. On the other hand, the Exchange 2000 Store can keep running while you restore the databases, so service is only unavailable to users whose mailboxes are on the failed databases.
- Know the signs of imminent failure, and monitor system health to catch problems early.
- Practice a disaster-recovery plan. Make sure that everyone who might need to restore data knows where to find the backup media, how to restore both Exchange Server and the OS (in

case of a catastrophic hardware failure), and when to call for help. Calling Microsoft Product Support Services (PSS) for assistance won't help you if you've already botched up a restore. If you don't know what to do, call for help first.

Backups are boring, and performing them correctly day in and day out can be tedious. But a good backup is invaluable when a disk or controller fails, and you'll be glad (and able to keep your job) when a successful restore gets users back online quickly.

Figure 4
Event-log entry for a successful backup



Stay in the Know

Knowledge is the key to achieving and maintaining great performance within an Exchange Server infrastructure. If you don't understand the technology you deal with, you can't create a good design or properly operate your servers, and the first hint of a hardware problem could lead to data loss and extended downtime. A huge body of knowledge is available for Exchange Server 5.5 and is developing rapidly for Exchange 2000. Although newsgroups and mailing lists have a high noise-to-data ratio, you'll be surprised at how many gems you can mine from the discussions. Conferences such as Microsoft TechEd and the Microsoft Exchange Conference (MEC) offer a chance to listen to other people's experiences and learn what the future might hold.

The only thing we can be sure of is that technology will keep changing. Make sure you maintain your personal knowledge base so that you can take advantage of new hardware and software technologies. By doing so, you'll maintain—and improve—your Exchange Server organization's performance.

Chapter 12

The 90:10 Rule for SQL Server Performance

—by *Kalen Delaney*

Tuning Microsoft SQL Server 2000 and SQL Server 7.0 to boost performance can be hard work, but in most cases, you can get major benefits from expending just a little effort. It's the old 90:10 rule: You can get 90 percent improvement with only 10 percent effort, but realizing that final 10 percent performance gain will take 90 percent of your tuning efforts.

This 90:10 rule doesn't apply to all database products or even to earlier versions of SQL Server. To achieve reasonable performance, some products require you to set dozens—or even hundreds—of server-level parameters and numerous other options in your SQL code. By contrast, Microsoft made SQL Server 2000 and SQL Server 7.0 self-tuning, and these products give you reasonable performance right out of the box.

But to attain a higher level of performance than what you get on average out of the box, you only need to give SQL Server a little attention. Applying the following 10 tips will help you achieve that initial 90 percent performance gain. If you find that you need more information as you put these tips into practice, check out the resources in the sidebar “Knowing Is 9/10 of the Battle.”

Tip 1: Don't skimp on hardware

I don't usually recommend hardware improvements first on a list of tuning tips because I don't want to mislead you into thinking you can simply throw hardware at performance problems. However, in the case of SQL Server performance, consider hardware first.

If you already use a decent system, upgrading your hardware will rarely bring more than a 10 percent overall performance improvement. However, if you're running a SQL Server based application on a server to which several hundred users simultaneously connect, and the server has only one physical hard disk and the absolute minimum of 64MB of RAM, then simply increasing RAM to the recommended minimum of 128MB will bring a more drastic performance improvement.

Beyond 128MB of RAM, you ideally need another 10MB for each 10 simultaneous user connections, and you need enough additional RAM to store all user data, system data, and indexes. I recommend choosing a disk configuration with which you store user data (.mdf and .ndf) files and log (.ldf) files on separate physical disks that have separate controllers. Store the user data files on the best RAID system your budget allows. For processing power, purchase two of the fastest processors you can afford. These configurations are the very skimpiest you should consider.

Knowing is 9/10 the Battle

—by *Kalen Delaney*

Microsoft SQL Server is a rich and complex product about which you'll always be able to learn more. Each tuning tip in "The 90:10 Rule for SQL Server Performance" is also an involved topic that you might need to know more about. I recommend the following references as gateways: Each source might lead you to other helpful sources for learning more about SQL Server performance.

Tip 1: Don't skimp on hardware

For more hardware-configuration recommendations, go to the Microsoft Developer Network (MSDN) Web site, MSDN Online. The white paper "Microsoft SQL Server 7.0 Storage Engine Capacity Planning Tips" (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql7/html/storageeng.asp>) is particularly helpful.

Tip 2: Don't overconfigure

For more information about SQL Server configuration, read SQL Server *Books Online* (*BOL*) and the white paper "Microsoft SQL Server 7.0 Performance Tuning Guide" (http://msdn.microsoft.com/library/default.asp?url=/library/techart/msdn_sql7perfune.htm) by Henry Lau. Check out the Microsoft Online Seminars site (<http://www.microsoft.com/seminar/default.aspx>) for seminars about SQL Server.

Tip 3: Take time for design

Unfortunately, no introductory SQL Server book or Microsoft Official Curriculum (MOC) course covers the subject of relational database design sufficiently. Microsoft might steer clear of the topic because the subject is independent of specific software products. A good starting place for information about design is Michelle A. Poolet's Solutions by Design column in *SQL Server Magazine*. Find the articles at <http://www.sqlmag.com/articles/index.cfm?authorid=436>.

Tip 4: Create useful indexes

For more information about SQL Server indexing and the query optimizer, start by reading all the information in *BOL* about indexes. Microsoft offers two white papers about the Index Tuning Wizard: "Index Tuning Wizard for Microsoft SQL Server 7.0" (http://msdn.microsoft.com/library/default.asp?url=/library/techart/msdn_sqlindex.htm) and "Index Tuning Wizard for Microsoft SQL Server 2000" (<http://msdn.microsoft.com/>

Continued on page 113

Knowing is 9/10 the Battle *continued*

library/default.asp?url=/library/techart/itwforsql.htm). MOC Course 2013: Optimizing Microsoft SQL Server 7.0 and Course 2073: Programming a Microsoft SQL Server 2000 Database provide additional educational information about the subjects. For more information about these courses, go to <http://www.microsoft.com/traincert/default.asp>.

Tip 5: Use SQL effectively

Don't limit yourself to books about the T-SQL language. For information about programming with ANSI-SQL, I recommend Joe Celko, *Joe Celko's SQL for Smarties: Advanced SQL Programming*, 2nd edition (Morgan Kaufmann Publishers, 1999).

Tip 6: Learn T-SQL tricks

The following books supply useful examples of T-SQL programming and help you get the most bang from your SQL Server queries: Itzik Ben-Gan and Dr. Tom Moreau, *Advanced Transact-SQL for SQL Server 2000* (Apress, 2000); and Ken Henderson, *The Guru's Guide to Transact-SQL* (Addison-Wesley, 1999).

Tip 7: Understand locking

Read everything you can about SQL Server default locking mechanisms, including *BOL*, my Inside SQL Server columns for *SQL Server Magazine* (<http://www.sqlmag.com>), and these Microsoft articles: "INF: How to Monitor SQL Server 7.0 Blocking" (<http://support.microsoft.com/?kbid=251004>), "INF: Understanding and Resolving SQL Server 7.0 or 2000 Blocking Problems" (<http://support.microsoft.com/?kbid=224453>), and "INF: How to Monitor SQL Server 2000 Blocking" (<http://support.microsoft.com/?kbid=271509>).

Tip 8: Minimize recompilations

You can read more about stored procedure recompilation in *BOL*. The following Microsoft article provides information about minimizing recompilations of your application's stored procedures: "INF: Troubleshooting Stored Procedure Recompilation" (<http://support.microsoft.com/?kbid=243586>).

Tip 9: Program applications intelligently

For information about determining the source of a SQL Server performance problem, see the Microsoft article "Troubleshooting Application Performance with SQL Server" (<http://support.microsoft.com/?kbid=224587>).

Continued on page 114

Knowing is 9/10 the Battle *continued***Tip 10: Stay in touch**

In addition to searching the msnews.microsoft.com server for newsgroups you might find helpful, you can go to Microsoft's SQL Server Newsgroups Web page (<http://www.microsoft.com/sql/support/newsgroups.htm>) to search for newsgroups to which you can subscribe. If you prefer Web-based support, try the *Windows & .NET Magazine* Network's Discussion Forums (<http://www.winnetmag.com/forums>).

Tip 2: Don't overconfigure

Microsoft designed both SQL Server 2000 and SQL Server 7.0 to self-tune. For example, the SQL Server engine can determine optimal values for memory utilization, number of locks to allow, and checkpoint frequency.

Only consider changing the out-of-the-box configuration options that don't affect performance. These nonperformance configurations include user options and the two-digit year cutoff option (the user options bitmap indicates which options you want to enable for each user connection; the two-digit year cutoff option controls how SQL Server interprets a two-digit year value).

Microsoft made the performance-related options available for configuration in the rare case in which you need to make an adjustment to them. However, your best bet is to let SQL Server's configuration options continue self-tuning.

If you use SQL Server 7.0, the *max async I/O* parameter might be an exception to this recommendation. You configure max async I/O for the level of sophistication and number of controllers in your I/O system. The max async I/O value determines the maximum number of outstanding asynchronous I/O requests that the server can issue to any file. If a database spans multiple files, the value applies to each file.

The max async I/O default setting of 32—only 32 reads and 32 writes can be outstanding per file—is an optimum value for many systems. Read SQL Server *Books Online (BOL)* to determine whether you need to change the default value for your system. SQL Server 2000 doesn't include the max async I/O parameter but can determine the optimum value internally.

Tip 3: Take time for design

In this age of rapid application development (RAD), the popular goal of quick initial project implementation might tempt you to sacrifice high-quality relational database design. If you yield, performance suffers. Poor design is a difficult problem to fix because fixing design frequently requires changing much of the written and tested code.

For a well-designed database, start by creating a normalized model in at least the third normal form. This practice minimizes redundancy and reduces overall data volumes. You can then systematically denormalize design, documenting each break from the normalized form.

A denormalized design is different from an unnormalized design. A denormalized design introduces redundancy for specific performance reasons. For example, if you consistently need to look up a customer's outstanding orders by the customer's name, and one table stores the customer ID number and the customer's name whereas another table stores the customer ID number and the order status, SQL Server needs to perform a join of these tables to retrieve names of customers with outstanding orders. In some cases, joins can be expensive. A denormalized model might add the customer name to the table that contains the order status and customer ID number and eliminate the need for a join.

Tip 4: Create useful indexes

Without the help of useful indexes, SQL Server must search every row in a table to find the data you need. If you're grouping or joining data from multiple tables, SQL Server might even need to make multiple passes through the tables to get your results. Because reading from disk is one of the most expensive actions SQL Server performs, reducing the number of required disk reads is one of the most effective tuning techniques you can employ.

If you create useful indexes, you can see performance improve by orders of magnitude, rather than by a few percentage points. For example, without an index, SQL Server might need to read all 10,000 pages in a table. An index would bring a 100,000 percent improvement in the number of pages SQL Server needs to read if the index reduces the number of necessary page-reads to 10.

A thorough knowledge of index architecture and the SQL Server query optimizer will help you create the best possible indexes, but until you reach that knowledge level, you can use SQL Server's Index Tuning Wizard to create useful indexes. Open this tool from SQL Server Enterprise Manager by clicking the Wizards button in the toolbar and looking under Management Wizards.

Before the wizard makes a set of index recommendations, it needs to know how you'll be accessing data. The best way to collect this information is with SQL Server Profiler. During a few hours of peak usage, capture the SQL command batches that your client applications send to your SQL server. You can use this information to tell the wizard how client applications access a table.

If your application isn't in production yet, you can provide the Index Tuning Wizard with a set of representative SQL statements that access the tables you want to tune. You can use Query Analyzer to create this set. Simply input the names of the stored procedures you'll be running, and make your best guesses about other ad hoc SQL statements that users will run.

Tip 5: Use SQL effectively

SQL is a set processing language, not a row-at-a-time processing language. T-SQL, Microsoft's dialect of the SQL language, can use server cursors to access one row at a time; however, most solutions that use server cursors will be orders of magnitude slower than solutions that use SELECT statements and UPDATE statements to perform the equivalent task. The more SQL programming experience you have, the more comfortable you'll be using the SQL language most effectively. Taking advantage of features such as subqueries, derived tables, and CASE expressions to manipulate sets of rows will speed your solutions and help you maximize SQL Server performance.

For example, suppose a table contains a row for each product in your inventory, and another table contains a row for the quantity of each sale of that product. You want to denormalize the database and store the sum of each product's sales in the product inventory table. To generate these sums, you could use a cursor and step through the product table one row at a time. For

each row, you could then find all matching rows in the sales table, add up the quantity values, and use that sum to update the product inventory table. In this example, using server cursors to collect figures for total sales is possible but unbelievably inefficient.

You can use the following UPDATE statement and correlated subquery to perform the same task. This statement uses the titles table in the pubs database as the products table, and for each title, the statement adds the values in the sales table's qty field.

```
UPDATE titles
SET ytd_sales =
(SELECT sum(qty) FROM sales
WHERE title_id = titles
.title_id)
```

Tip 6: Learn T-SQL tricks

Microsoft's T-SQL is an enhanced version of standard ANSI-SQL. Taking advantage of these enhancements will help you improve performance.

For example, suppose you want to put all products on sale and base each product's sale price on the quantity sold in the past year. You want the sale price to be 25 percent off the current price for products that sold fewer than 3000 units; you want the reduction to be 20 percent for products that sold between 3000 and 10,000 units; and you want a 10 percent discount for products that sold more than 10,000 units. You might think you need to issue an UPDATE statement with the appropriate discount values after you use a cursor to look at the products' rows individually for quantity-sold information. However, the T-SQL CASE expression lets you use one statement to calculate appropriate discounts.

The following sample UPDATE statement uses the pubs database's titles table, which has a price field that the statement will update and a ytd_sales field that stores the year-to-date sales quantity. (If you've already run Tip 5's sample statement, this query won't work; the sample will have updated ytd_sales to a set of different values.)

```
UPDATE titles
SET price = CASE
    WHEN ytd_sales < 3000 THEN price * 0.75
    WHEN ytd_sales between 3000 and 10000 THEN price * 0.80
    WHEN ytd_sales > 10000 THEN price * 0.90
END
WHERE price IS NOT NULL
```

Other T-SQL features that improve query performance are the TOP expression, when you use it with ORDER BY; indexed views (SQL Server 2000 only); and partitioned views.

Tip 7: Understand locking

Locking and blocking problems often cause performance degradation in multiuser systems. I advise against forcing SQL Server to lock data in the ways you might think SQL Server should. Instead, I recommend that you increase your understanding of how SQL Server typically locks data, how much data it usually locks, and how long it holds the locks. After you understand how SQL Server

locking and blocking works, you can write your applications to work with SQL Server, rather than against it.

For example, after your application modifies data in a transaction, SQL Server locks that data and makes it unavailable to any other processes until you either commit or roll back your transaction. If you take a long time to issue a commit or rollback command, the data will be locked a long time. Therefore, I advise you to keep your transactions as short as possible. I also advise against allowing user interaction in the middle of your transaction.

By default, SQL Server holds exclusive locks—acquired when you insert, update, and delete data—until the end of a transaction. SQL Server holds share locks—acquired when you select data—only until you finish reading the data that you selected.

You can change your transaction isolation level to cause SQL Server to hold share locks until the end of a transaction, which means that after you retrieve and read data, no one else can modify the data. Changing transaction isolation levels might sound like a good idea for keeping data for only your use. However, these changes aren't a good idea if you have multiuser systems from which many users need to access the same data. I recommend that you keep your transaction isolation level at Committed Read (which is the default transaction isolation level) and change the level only when you can't accomplish your performance goals any other way.

Tip 8: Minimize recompilations

The first version of SQL Server could store and reuse the execution plan of a stored procedure that your application executes multiple times. However, until SQL Server 7.0, this feature didn't provide big savings. For many queries, the cost of generating the query plan through compilation (compilation also includes query optimization) was only a fraction of the cost of executing the query; you hardly noticed the millisecond or two you saved by not creating a new plan.

Microsoft rewrote the product's query optimizer for SQL Server 7.0 to include dozens of new query processing techniques. As a result of the new features, the query optimizer typically spends more time than earlier versions producing an execution plan. This extra time makes the query plan reuse feature more valuable for saving time.

SQL Server 2000 and SQL Server 7.0 also offer a mechanism for saving the execution plans of ad hoc queries. This feature can be a help when a stored procedure isn't available. This capability is automatic, but not guaranteed. The mechanism follows a strict set of rules and is less predictable than the reuse of stored procedure plans. Therefore, I recommend that you write stored procedures for all your SQL code wherever you possibly can.

Using precompiled plans can save you time, but occasionally you'll want to force a recompilation, and sometimes SQL Server will decide independently to recompile the plan for a procedure. Profiler can tell you when recompilations occur, and the System Monitor tool can tell you how often these recompilations have occurred.

Tip 9: Program applications intelligently

The more you, as a client programmer, know about how SQL Server works, the better the code you can write. For example, you'll know not to allow user interaction in the middle of a transaction, as Tip 7 also warns against.

Another poor programming practice is writing your client application to begin a transaction, send an update statement to SQL Server, then bring up a message box asking users whether they

want to continue. In this case, SQL Server would hold any acquired locks until the user, who might have stepped out to a long lunch or gone home for the day, returns and clicks OK in the message box.

Tip 5 warned against using server cursors. However, client cursors are a different topic. Programming your client application to process in a row-at-a-time fashion a result set that SQL Server used a set-based operation to generate is an acceptable practice. However, you need to read your API's documentation to maximize performance of the many variations of client cursors.

One variation of a client cursor is the Fast Forward-Only cursor, useful for fetching data sequentially for one-time-only read-only purposes. You can use this cursor to save two round-trips to the server; SQL Server fetches the first row when the cursor is opened and closes the cursor when SQL Server fetches the last row. Even if you're only fetching a few rows, if you frequently use the section with which you use the Fast Forward-Only cursor, those two round-trips you save will add up.

Tip 10: Stay in touch

If these tips don't address your particular tuning concerns, familiarize yourself with the myriad sources of free public support, in which many longtime SQL Server experts read and respond to specific questions from SQL Server users. I recommend Microsoft's public newsgroups. You can use any newsreader software (e.g., Microsoft Outlook Express) to search in the msnews.microsoft.com server for the newsgroups that have `sqlserver` in their names.

Tips of Icebergs

These tips are just that—tips—like the tips of icebergs, which show only 10 percent of the total berg. At the very least, the tips show you what's available for tuning SQL Server. Now that you know what you can tune, you can look for more information about how to tune. Before long, you might even be up to exerting that 90 percent effort to tune SQL Server to reach that last 10 percent for maximum performance.

Chapter 13

Performance FAQs

Q. Which Performance Monitor counters do I need to observe to evaluate my Windows NT server's performance?

A. Performance Monitor has several key objects for evaluating NT performance, such as LogicalDisk, Memory, Network Interface, Server Work Queues, and System. Within each of these objects, you can monitor several important counters, as Table 1 shows.

—by Curt Aubley

Table 1
Important Performance Monitor Objects and Counters

Object	Counter	Description	Indication of Performance Problem
LogicalDisk	% Disk Time	Percentage of elapsed time that the selected disk drive is busy servicing read or write requests	Value is greater than 70%
	Avg. Disk Queue Length	Average number of read and write requests that were queued for the selected disk during the sample interval	Value is greater than 2, with high % Disk Time
Memory	Available Bytes	Size of virtual memory currently on the Zeroed, Free, and Standby lists; Zeroed and Free memory is ready for use; Zeroed memory cleared to zeros	Value is 4MB or less if memory management is set to Maximize Throughput for Network Applications, or 1MB if memory management is set to Maximize Throughput for File Sharing; a low value as defined above combined with a high Pages/sec value and a busy disk drive containing the paging file
	Pages/sec	Number of pages read from the disk or written to the disk to resolve memory references that were not in memory at the time of reference	Value is greater than 100 if accompanied by a low Available Bytes value and high % Disk Time on the paging file disks
Network Interface	Output Queue Length*	Length (in packets) of the output packet queue	Value is greater than 3 for 15 minutes
Server Work Queues	Queue Length	Current length of the server work queue for a CPU	Value is greater than 2 or continuously growing with an associated % Total Processor Time greater than 90%
System	% Total Processor Time	Percentage of time that a processor is busy executing threads	A high value is acceptable, unless it is greater than 90% with an associated Queue Length greater than 2

*This value is not valid in SMP environments as of NT 4.0 Service Pack 3 (SP3).

Q. What causes my Windows NT server's available memory to drop to 0MB during file transfers?

My NT server uses system memory to buffer file transfers. My server typically has more than 100MB of available physical memory, and the paging file is about 10 percent utilized. Recently, Performance Monitor showed that available memory was below 4MB. I started a 250MB file copy from the server to my machine. After 10 seconds the available memory dropped to 0MB. What caused this drop? How can I prevent it from recurring?

A. NT's file system cache is probably the cause. You can use memory tuning to solve this problem. When you set NT's memory strategy for Maximize Throughput for File Sharing, NT favors the working set of the file system cache. When your system is under a heavy load such as a large file transfer, NT uses any available memory for the file system cache until it runs out. Then NT starts using the paging file for the additional memory space required. By itself, NT paging wastes CPU cycles and time moving data between memory and disk. After NT completes the file transfer and the file system cache is no longer in demand, NT reclaims the memory as other processes request memory. NT does not immediately free up the memory, because the last process that used the memory will probably require it again.

If your system regularly uses excessive memory, or if you want to run other applications while large file transfers are occurring, you can change NT's memory strategy, add memory to improve the performance of your paging file, or increase the amount of available memory. The easiest tactic is to set NT's memory strategy to Maximize Throughput for Network Applications. To set this memory strategy, open the Control Panel Network applet and select Server under the Services tab. When you set this memory strategy, NT doesn't use the entire file system cache for I/O transactions. In addition, one large file transfer doesn't consume all of NT's available memory. Another solution is to add memory and thus increase your paging file's performance. However, consider the applications that are running on your system before you add memory. NT can run using the server and workstation services alone. Thus, you can disable unnecessary applications and services to free up memory.

—by Curt Aubley

Q. How can I ensure that both processors in a 2-way system are fully utilized?

I have a data-processing server with dual 400MHz Intel Pentium II processors. The developer wants to use both processors at 100 percent. The processors currently balance at 50 percent each. How can we improve the server's performance without rewriting the application it runs?

A. Make sure your application isn't waiting for another Windows NT resource to complete its task. If it is, you must obtain a higher CPU utilization before you can remove the bottleneck.

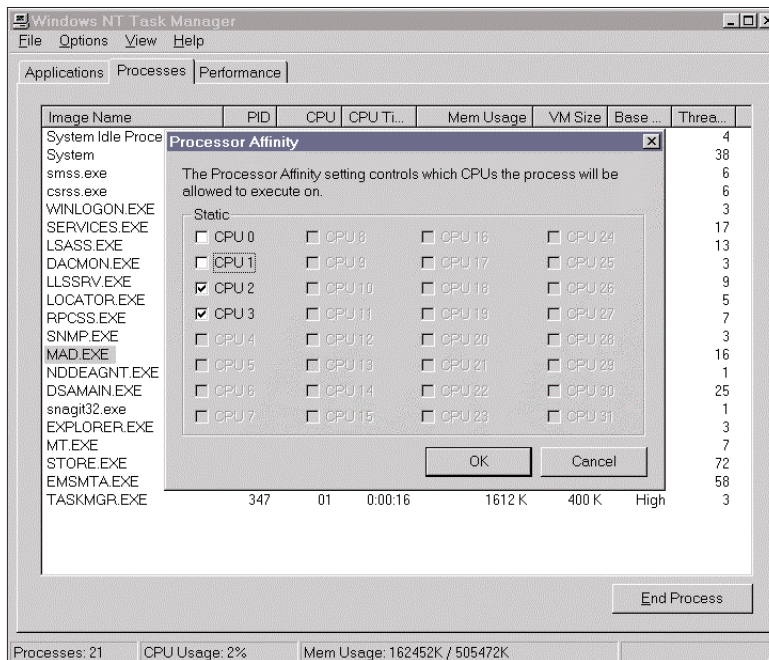
If your application operates using one thread, you can control the priority and affinity of your process to give more processor time to your application. Priority lets you assign when and for how long tasks receive time from the microprocessor. Affinity lets you limit which CPUs your application can run on or force it to use only certain CPUs. In NT's standard installation, each process has

one or more threads of execution. All these threads require CPU time simply to complete NT OS tasks. NT typically schedules these threads across multiple CPUs in an SMP server environment. If your application has only one thread, the second CPU won't provide much of a performance boost, but it will offload some of NT's housekeeping functions and give your application more processor utilization. If your application is multithreaded and designed to take advantage of multiple CPUs, the second CPU will increase performance. To tune your application's priority level, launch Task Manager and select the Processes tab. Right-click the application's name to set the priority of the application's process. Under Set Priority, select High. Monitor for any performance changes. To tune the affinity level, right-click the application's name and select Affinity. Figure 1 shows Task Manager's affinity manager.

Using Task Manager's affinity manager, clear one of the CPU boxes and limit your application to one CPU. If performance still does not improve, right-click the process again and set the priority to Realtime. This setting ensures that your application has enough CPU time and improves the chance of a cache hit. I don't recommend using the Realtime priority level on a single-CPU system because your application could deplete the CPU and make NT unstable. As with any advanced tuning techniques, you must use them correctly to avoid NT server lockups.

—by Curt Aubley

Figure 1
Controlling affinity with NT's Task Manager



Q. How do I use Windows 2000's Extensible Storage Engine (ESE) counters?

I've heard and read that you can use ESE counters in the Win2K Microsoft Management Console (MMC) Performance console to track various Active Directory (AD)-related metrics. However, I can't find the ESE object or any of its counters in the Performance console when I run the console on my domain controllers (DCs). Does this object exist, and if so, how can I enable it on my DCs?

A. You can indeed use ESE performance counters to monitor the ESE database on a DC. By default, however, Win2K doesn't install these counters on DCs. To install the counters manually, you need to use a special DLL file, named esentprf.dll. After you complete the following procedures, you can view and access the counters through the Performance console's System Monitor snap-in.

First, copy `\\%systemroot%\system32\esentprf.dll` to a different directory on your DC. (For example, you might create the directory `C:\perfcons`, then copy `esentprf.dll` into that directory.)

Next, use `regedit` or `regedt32` to create the following registry subkeys (assuming that they don't already exist): `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT` and `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT\Performance`. Under the Performance subkey, you need to add four registry values and initialize with data those values. Table 2 lists the values, their types, and the appropriate data for each.

Change the directory to the `\\%systemroot%\system32` folder (e.g., `C:\winnt\system32`). Then execute the following statement:

```
lodctr.exe esentprf.ini
```

—by Sean Daily

Table 2
ESE Performance Counter Registry Values

Value Name	Type	Data
Open	REG_SZ	OpenPerformanceData
Collect	REG_SZ	CollectPerformanceData
Close	REG_SZ	ClosePerformanceData
Library	REG_SZ	C:\perfcons\esentprf.dll

Q. How do I improve performance running SQL Server 7.0 and Exchange Server 5.5 SP2?

A. If you run both SQL Server 7.0 and Exchange Server 5.5 Service Pack 2 (SP2) on a computer running BackOffice Server 4.5 or Small Business Server (SBS) 4.5, you must explicitly configure the memory that SQL Server uses. You must increase the minimum dynamic memory setting for SQL Server 7.0 from the default value of zero to a value of at least 32MB. (You might need to set it

higher to support SQL Server's processing load because this setting determines the memory that SQL Server uses when Exchange Server is running and under load. In this environment, SQL Server won't reach the maximum dynamic memory setting.) SQL Server and Exchange Server administrators should determine the amount of memory to allocate to SQL Server that will optimize the overall performance of both applications. The SQL Server administrator must then set the SQL Server minimum memory option to this value. If the SQL Server database supports a third-party application, you might need to consult the application's documentation or vendor to find out how much memory SQL Server needs to support the application processing load.

To increase the minimum dynamic memory setting for SQL Server 7.0, perform the following steps:

1. Go to Start, Programs, Microsoft SQL Server 7.0, Service Manager. The SQL Server Service Manager dialog box will appear.
2. Make sure that MSSQLServer appears in the Services list. Click Start, and click Continue.
3. When SQL Server starts, go to Start, Programs, Microsoft SQL Server 7.0, and click Enterprise Manager.
4. In the console tree, expand the Microsoft SQL Servers node, then expand the SQL Server Group node.
5. Right-click the node of your SQL Server, and click Properties.
6. Select the Memory tab.
7. Under Dynamically Configure SQL Server Memory, drag the Minimum (MB) memory slider to the right until it says 32MB.
8. Click OK, then close the SQL Server 7.0 Enterprise Manager.

For the new settings to take effect, you must stop and then restart the MSSQLServer service.

—by *Microsoft Product Support Services*