EMC® NetWorker® Release 7.5

Performance Tuning Guide

P/N 300-008-472
REV A01

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Published December, 2008

Contents

Preface

As part of an effort to improve and enhance the performance and capabilities of its product lines, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

If a product does not function properly or does not function as described in this document, please contact your EMC representative.

Audience

This document is part of the EMC NetWorker documentation, and is intended for use by:

- System administrators responsible for installing software and maintaining the servers and clients on a network.
- Operators monitoring the daily backups who may also find this manual useful.

Readers of this document are expected to be familiar with the following software:

- EMC NetWorker software

Organization

Here is a list of where information is located in this guide.

- Chapter 1, "NetWorker Features," provides brief descriptions of software features and an overview of hardware and software performance.
- Chapter 2, "Testing and Tuning Basics," describes the hardware and software components that make up the data storage management environment, discusses their impact on storage management tasks, and provides general guidelines for locating problems and solutions.
- Chapter 3, "Testing and Tuning on Windows," explains the specific performance testing and tuning tasks that can be performed on a NetWorker server in a Windows environment.
- Chapter 4, "Testing and Tuning on UNIX and Linux," provides additional performance testing and tuning information for the NetWorker server on a UNIX or Linux platform. This chapter contains information written specifically for the UNIX and Linux environments, and provides UNIX and Linux-based examples only.
- Chapter 5, "NetWorker Configuration," explains the configuration activities that can be performed on the NetWorker server.
- Appendix A, "Troubleshooting," describes error messages and explains how to correct problems with the NetWorker Server.
- "Glossary"

Related documentation

Related documents include:

- *EMC NetWorker Release 7.5 Multiplatform Version Administration Guide*
- *EMC NetWorker Release 7.5 Release Notes*
- *EMC NetWorker Release 7.5 Installation Guides*
- *EMC Information Protection Software Compatibility Guide*

Conventions used in this document

EMC uses the following conventions for special notices.

Note: A note presents information that is important, but not hazard-related.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.

IMPORTANT

An important notice contains information essential to operation of the software. The important notice applies only to software.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, filenames, functions, utilities • URLs, pathnames, directory names, computer names, links, groups, service keys, file systems, notifications
Bold:	Used in running (nonprocedural) text for: • Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system call, man pages Used in procedures for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • What user specifically selects, clicks, presses, or types
Italic:	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis (for example a new term) • Variables
`Courier:`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text.
`Courier bold:`	Used for: • Specific user input (such as commands)
`Courier italic:`	Used in procedures for: • Variables on command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user

[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means “or”
{ }	Braces indicate content that you must specify (that is, x or y or z)
...	Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support, product, and licensing information can be obtained as follows.

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink website (registration required) at:

http://Powerlink.EMC.com

Technical support — For technical support, go to EMC Customer Service on Powerlink. To open a service request through Powerlink, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this document to:

techpub_comments@EMC.com

1 NetWorker Features

The NetWorker product is a network storage management application that is optimized for high-speed backup and recovery operations of large amounts of complex data across an entire network computer. The NetWorker software is available in four editions. Brief descriptions and distinctions among these editions are covered in this chapter, as well as an overview of hardware and software performance, in the following sections:

Introduction to NetWorker editions

The four editions of the NetWorker<sp Superscript>®<sp Superscript> software are as follows:

- Workgroup Edition

  The Workgroup Edition supports small networks that do not require clustering or contain large databases and/or file systems.

- Business Edition

  The Business Edition is similar to the Workgroup Edition as it is designed for small business needs.

- Network Edition

  The Network Edition supports large, distributed networks and a full range of add-on products.

- Power Edition

  The Power Edition addresses the storage management and data protection needs of enterprises with high-performance database servers and file servers, such as very large database (VLDB) applications, online transaction processing (OLTP) applications, data warehouses, and web servers.

Overview of hardware and software performance

Performance is the speed and efficiency by which a task is completed. Backup and recovery performance is dependent on several hardware and software variables, but in general, backups cannot go faster than the speed of their slowest component.

This guide provides tests to measure and maximize the various components of your computer. Change software settings or replace a hardware component by identifying the slowest component in the data path between the location of the data and the storage device.:

- Factors that affect the speed of the server include:
  - CPU
  - Memory
  - Input/output (I/O) bandwidth
  - SCSI bandwidth
  - Number of ports
- Factors that affect the speed of the devices include:
  - I/O transfer rate
  - Built-in compression and initialization characteristics

Note: Device speed is combined for all storage devices used concurrently.

If clients are backed up over a network, these factors can also affect performance:

- Client speed:
  - CPU
  - Memory
  - Disk speed

Note: Client speed is combined for all clients active at a single point in time.

- Network speed:
  - Network I/O bandwidth
  - Network path
  - Network load

Factors impacting software performance include:

- File-system management
- Application-specific optimization
- Backup application (for example, the NetWorker software)

2

Testing and Tuning Basics

This chapter identifies the different hardware and software components that make up the data storage management environment, discusses their impact on storage management tasks, and provides general guidelines for locating problems and solutions in the following sections:

Overview of performance testing and tuning

The benchmark tests used in the computer industry are done in controlled environments to simplify analysis and show products at their greatest advantage. When testing the performance of the NetWorker server in your environment, determine the degree to which the test is controlled:

- To run a fully-controlled test, turn off all extraneous processes to isolate the performance of the NetWorker server. The results of this test show the optimal performance of the NetWorker server with the current hardware and software configuration.
- To run a less-controlled test, do not shut down other applications and network traffic. The results of this type of test show how the NetWorker server performs while the computer is running other programs and processes.
- If running both a controlled test and one with a normal or simulated load, the difference between the results can determine:
  - How to schedule the backups.
  - Whether to run other programs on the NetWorker server.

The tests in this and the following chapters follow a basic process:

1. Measure the current performance of the NetWorker server.
2. View the results to find a blockage or limiting factor.
3. Adjust the hardware and software to correct the blockage.
4. Measure the performance of the NetWorker server.

Although there will be some blockage in a system, use this process to eliminate the greatest obstacles to fast backup performance.

IMPORTANT

Install sufficient computer and network hardware to support a storage application. Failure to meet minimum system requirements, particularly for memory, free space, and CPU speed, can have an obvious and immediate negative impact on the storage management processes.

Server hardware

The following sections address different hardware components of the NetWorker server, as well as performance tests of these components.

Physical disks

The physical disks in a computer system are in constant motion, reading and writing data for storage and recovery, and maintaining the client file indexes and media database. Problems with these disks can negatively affect every performance level within a storage management environment.

The type of data backed up and how it is laid out on the disk can cause backup performance to vary. If you have large files that are fragmented, or many small files, the efficiency of the disk suffers because the disk head moves frequently from one fragment or file to another.

If the data is compressed on the disk, the operating system or application decompresses the data before transferring it for backup. The CPU spends cycles to decompress the files, and the disk can rarely go at its maximum speed.

The backup process is most efficient for large files with minimal fragmentation.

Disk read speed testing methods

Follow these guidelines to determine the read speed of disks:

- Test the read speed of each individual disk while NetWorker software is *not* running.
- Test the read speed of all disks while the NetWorker software is *not* running.
- Test the read speed of each individual disk while NetWorker software is running.
- Test the read speed of all disks while NetWorker software is running.

Use the NetWorker **uasm** program, which saves and recovers file-system data, to test how fast the system can read from disk.

The syntax is as follows:

```
uasm -s [-benouv] [-ix] [-t time] [-f proto] [-p ppath] path
uasm -r [-nuv] [-i {nNyYrR}] [-m src = dst] [-z suffix] path
uasm -c [-nv] path
```

For example, enter the following command at the system prompt:

```
uasm -s filename > /dev/null
```

where *filename* is a large file not accessed recently. If the file was recently accessed, it may be in memory and **uasm** will not actually read the file from disk.

The **uasm** program reads from the disk at the maximum speed. In this example, the data is written to a null location so the disk read is not slowed down to allow time for writing.

The **uasm** program has three modes: save, recover, and compare. A description of how the **uasm** program performs in the three modes is provided in Table 1.

Table 1 uasm program modes

In this mode	The uasm program
save	Scans directory trees and generates a save stream on its standard output (**stdout**) representing the files and organization of the directory tree. Symbolic links are not followed by application-specific modules (ASMs), except **rawasm**, which enables NetWorker to process data in raw disk partitions.
recover	Reads a save stream from its standard input (**stdin**) and creates the corresponding directories and files.
compare	Reads a save stream from its **stdin** and compares the save stream with the files on the file system.

In save mode, **uasm** can be controlled by directive files. Directive files control the following:

- How to search descendent directories.
- Which files to ignore.
- How the save stream is generated.
- How to process subsequent directive files.

All ASMs accept the options described in Table 2 on page 18. ASMs also can have additional options.

Note: The **-s** (save), **-r** (recover), or **-c** (compare) mode must be specified when using the **uasm** program and must precede any other options. When saving, at least one path value must be specified. The path value can be either a directory or a filename.

Options valid for all modes of uasm

Table 2 lists the options that are valid with all modes of **uasm**.

Table 2 uasm options

Option	Usage
-n	Use the **-n** option to perform a dry run. When saving, scan the file system, but do not attempt to open files and produce the save stream. When recovering or comparing, consume the input save stream and perform basic checks, but do not create directories or files when recovering or do the work of comparing the file data.
-v	Use the **-v** option to turn on verbose mode. The current ASM, its specified options, and the file it is processing are displayed. When an ASM is operating in filtering mode (that is, processing the save stream of another ASM) and modifies the stream, its name, specified options, and the current file appear within square brackets.

Options for save mode

Table 3 lists the options to be used in save mode.

Table 3 Save mode options

Use this option	To
-b	Produce a byte count. This is similar to the **-n** option, but byte count estimates the amount of data to produce, instead of actually reading file data. This option is faster, but less accurate than the **-n** option. Byte count mode produces three numbers: records (files and directories), bytes of header information, and bytes of file data. Byte count mode does not produce a save stream, so its output cannot be used as input to another ASM in recover mode.
-o	Produce an "old style" save stream that can be handled by older NetWorker servers.
-e	Instruct **uasm** to not generate the final "end of save stream" boolean. Use this option only when the following applies: • ASM invokes an external ASM. • An optimization does *not* consume the generated save stream itself.
-i	Ignore all save directives from .nsr directive files found in the directory tree.
-f	Specify the location of a .nsr directive file to interpret before processing any files. Within the directive file specified by proto, path directives must resolve to files within the directory tree being processed, otherwise their subsequent directives will be ignored.
-p	Specify the path with **ppath**. When you specify **-p ppath**, this string is added to the beginning of each filename as it is output. This option is used internally when one ASM executes another external ASM. The path entered for ppath must be a properly formatted path that is either the current working directory, or a trailing component of it.
-t	Specify a date with **-t** *date* to set the date after which files are modified before they are saved.
-x	Cross file system boundaries. Normally, file-system boundaries are not crossed when walking the file-system. Symbolic links are not followed, except in the case of **rawasm**.

Options for recover mode

Table 4 lists the options to use in recover mode.

Table 4 Recover mode options

Use this option	To
-i {*nNyYrR***}**	Define the initial default overwrite response option. Only one overwrite response option is allowed. When the file being recovered conflicts with an existing file, the user is prompted for overwrite permission. The default response option, selected by pressing **Return**, is displayed within square brackets. "The -i option" on page 19 provides additional information about the **-i** option.
-m *src=dst*	Map the filenames to be created. Any files that start exactly with *src* are mapped to have the path of *dst* replacing the leading *src* component of the pathname. This option is useful when relocating the recovered files saved using absolute pathnames into an alternate directory (for example, **-m c:\win32app=.**).
-z *suffix*	Specify the suffix to append when renaming conflicting files. The default suffix is **R**.

The -i option

The **-i** option defines the initial default overwrite response option. Table 5 on page 20 lists the overwrite response options and their meanings.

Table 5 Overwrite response options

Option	Meaning
n	Do not recover the current file. This is the initial default overwrite response option. Each time a response option other than the default is specified, the new response option becomes the default.
N	Do not recover any files with conflicting names.
y	Overwrite the existing file with the recovered file.
Y	Overwrite all files with conflicting names.
r	Rename the conflicting file. A dot, ".", and a suffix (by default "R") are appended to the recovered filename. If a conflict still exists, the NetWorker software prompts you again.
R	Automatically renames conflicting files by appending a dot, ".", and a suffix. If a conflicting filename already ends in a "." suffix, the NetWorker software prompts you to avoid potential auto-rename looping conditions.

When you specify either **N**, **R**, or **Y**, you are prompted only when the NetWorker server cannot auto-rename a file ending with the rename suffix. Subsequent conflict is resolved as if the corresponding lowercase letter is selected.

Disk performance tuning methods

If there is a blockage in the disk system, make the following changes to improve performance:

- Defragment files to reduce the number of disk seeks.
- Use larger files or fewer files to reduce the number of disk seeks.
- Change to a faster SCSI adapter with synchronous I/O support and bus mastering.
- Use stripe sets on multiple disks to accommodate large files and databases, so the physical disks can work in parallel to read data.
- Spread disk-intensive programs onto different servers.
- Purchase disks with higher transfer rates.

Memory and CPU usage

Each backup session requires CPU and memory resources on the NetWorker server. If other applications are running concurrently with the backup, they impose an additional load on the system. Heavy swapping or paging activities indicate that the server is CPU or memory-bound.

Memory usage

A NetWorker server or storage node computer can never have too much memory. Data transfer within memory is much faster than over a network. Memory holds frequently used data so the disk does not have to be read as often. Open applications, processes in run mode, and disk cache all use memory. If the operating system is capable, disk storage space is used as simulated memory, or virtual memory.

For best results, install the maximum amount of memory that your computers will sustain, especially for the NetWorker server and storage nodes.

CPU usage

To increase CPU performance and bandwidth:

1. Upgrade to a faster CPU.
2. Add additional CPUs if the NetWorker server supports multiple processors.
3. Shut down other processes during backup time.

This may reduce the processor load and improve backup performance.

Set scheduled backups for low-traffic periods to avoid performance degradation or excessive network traffic. The default NetWorker backup start time is 3:33 A.M.

The operating system of a computer might contain various utilities for testing and tuning CPU speed. Additionally, you can check the CPU utilization of each NetWorker function. The processes involved in a NetWorker backup are the following:

- These NetWorker services:
  - **nsrexecd**
  - **nsrd**
  - **nsrmmd**
  - **nsrmmdbd**
  - **nsrindexd**
- The **save** program for NetWorker backups
- Application-specific services, such as for a database server

Note: To measure and record CPU speed for the NetWorker server on the Windows platform, use the Windows Performance Monitor. "Testing CPU performance" on page 40 provides more information.

Server input/output

The following performance variables relate to the input/output (I/O) throughput on the NetWorker server.

Input/output backplane

On a well-configured NetWorker server, the maximum data transfer rate of the server's I/O backplane is the limiting factor for backup throughput. The theoretical backup throughput limitation is the following formula:

Backup throughput = Maximum I/O throughput/2

At a minimum, the NetWorker server requires two I/O transactions per data block:

- A read from the disk to memory
- A write from memory to media

SCSI I/O bandwidth

SCSI bus performance depends on the technology deployed. Some specifications for different types of SCSI technology include the following:

- SCSI-2 single-ended, 10 MB/s
- FWD (fast-wide-differential), 20 MB/s
- UltraSCSI, 40 MB/s

- Fibre

Note the following:

- The rated speeds of SCSI buses are defined as "best case."
- The selection of UltraSCSI-compatible devices is limited.

NetWorker settings

The NetWorker software has several attributes that control the speed and volume of data being backed up. These attribute settings can be adjusted to modify backup levels to attain the best performance on the NetWorker server.

The following sections address the various NetWorker settings in principle. Because these attributes appear in different locations in the Windows and UNIX versions, the *EMC NetWorker Release 7.4 Multiplatform Version Administration Guide* provides specific instructions on how to access these settings based on the server platform.

Server parallelism

The server parallelism attribute controls how many savestreams the server accepts simultaneously. The more savestreams the server can accept, the faster the devices and client disks run, up to their performance limit or the limits of the connections between them.

Client parallelism

The client parallelism attribute controls how many savestreams a client can send at the same time. To avoid disk contention, do not set a value for client parallelism that is higher than the number of physical disks on the client. The client parallelism for a NetWorker server controls the number of index, bootstrap and local filesystem savestreams that start from the server at the same time.

Multiplexing

The target sessions attribute sets the target number of save streams to write to a device simultaneously. Because this value is not a limit, a device might receive more sessions than the attribute specifies. The larger the number of sessions specified for target sessions, the more save sets that can be multiplexed (or interleaved) onto the same volume.

Performance tests and evaluation can determine whether multiplexing is appropriate for the system. Follow these guidelines when evaluating the use of multiplexing:

- Find the backup rate of each disk on the client. Use the **uasm** test described in "Disk read speed testing methods" on page 17.
- Find the maximum rate of each device. Use the **bigasm** test described in "Test device performance with bigasm" on page 28.

If the sum of the backup rates from all disks involved in a backup is greater than the maximum rate of the device, do not increase server parallelism. If more save groups are multiplexed in this case, backup performance will not improve, and recovery performance could slow down.

Modifying settings for performance tuning

This section describes measures to improve NetWorker server performance related to the following:

- Parallelism and target sessions attribute settings
- Workload balance
- Multiplexing
- Backups across a network
- Setting the value for file descriptors

Parallelism and target sessions attribute settings

Follow these guidelines to adjust the parallelism and target sessions attributes for the best server performance:

- Set the server Parallelism and Target Sessions attributes so that the total performance of the disk drives equals the total performance of the tape drives. If the Parallelism attribute is set to a higher value, there is no benefit.

  To select the right values for the parallelism and target sessions attributes, use the following equation:

  Parallelism = Number of Devices * Target Sessions

  For example, if there are three tape drives available for backup, and you want each tape drive to accept two savestreams, set the value of server parallelism to six and the value of target sessions to two.

- When a NetWorker server is saving a large number of save sets, such as 500 or more, memory consumption and file descriptor consumption can reach values that are close to operating system limitations. In this event, the parallelism may need to be lowered.
- Decrease the server parallelism and target sessions attributes to unload an overworked NetWorker server. With the correct settings, the normal operation of the computer should not be interrupted by backups or other NetWorker server activities.
- Add more memory to handle a higher parallelism setting.

Workload balance

The following are recommended to avoid overloading the server:

- Increase CPU power by upgrading the CPU or adding additional CPUs to the server (if possible).
- Reduce server load by spreading backups across multiple servers or storage nodes.
- Balance the number of disks and devices, so that the backups finish at about the same time. The *NetWorker Administration Guide* provides instructions on how to change these settings.

  Here are some methods of balancing the workload on disks and devices:

  - In general, start backups for the slowest or the biggest disk volumes first, because other disks can join in later to match the maximum bandwidth of the device.
  - Use pools of media to force the bigger and faster disk volumes to back up to faster devices.

- If there is a large and fast disk volume and many tape drives, manually divide the volume into several save sets so that each one can go to each tape drive in parallel.
- Balance data load for simultaneous sessions more evenly across available devices by adjusting the target sessions attribute. This parameter specifies the minimum number of save sessions to be established before the NetWorker server attempts to assign save sessions to another device.

Multiplexing

Refer to the following guidelines for improving server performance through multiplexing:

- To fully use the bandwidth of a high-speed device when you have many slow clients or disks, multiplex the save sets on the media. You maximize the performance of the devices since they do not have to start and stop to wait for data.
- To improve recovery performance, multiplex save sets to recover together onto the same tape. For example, multiplex the disks of the same computer onto the same media because it is likely that the data from all the disks will recover simultaneously.

Backups across a network

In cases where you back up data across a network, the following suggestions also can improve performance:

- Increase the client parallelism attribute, especially for clients with a logical volume manager and several physical disks. The bandwidth of the network could limit the number of savestreams to transmit at a time.
- To reduce server load, split backups across multiple servers.
- Turn on data compression in the NetWorker client to improve effective data throughput and reduce network traffic.
- Configure clients with high-transfer rate requirements to have a preference for a particular backup server on the same subnet; avoid router hops between the NetWorker server and its clients.

Setting the value for file descriptors

You may want to increase the value for file descriptors to 2048 (or 1024 if running Solaris 2.5). The number of file descriptors can be changed using the following command line:

```
# ulimit -n 2048
```

This sets the number of file descriptors to 2048, which is considered the soft limit. This number cannot exceed the hard limit, which is set in the kernel. For Solaris 8, the default value is 1024.

Note: It is possible to set the soft limit (using **ulimit -n**) to a number greater than the hard limit, but the maximum number of file descriptors used will be the lower number. For example, if you set the soft limit to 2048, but the hard limit is 1024, the maximum number of file descriptors will be the hard limit.

To find the hard file descriptor limit and change the value:

1. Go to the */etc/system* file. If a value other than the default is used, it must be listed in */etc/system* for the value to be set after a reboot.

2. Run the following:

```
# adb -kw /dev/ksyms
physmem 5da31
Type:
rlim_fd_max/D
Result:
rlim_fd_max: rlim_fd_max: 1024

man pages adb
```

3. To override this value:
   a. Tune the system parameter by adding set **rlim_fd_max=2048** to the */etc/system* file.
   b. Shut down and run **boot -r** at the OpenBoot prompt.

Note: If you are not familiar with changing kernel parameters, verify the process with your Sun documentation.

Tuning system parameters

You can view and change the values of the following system parameters:

tcp_time_wait_interval

Get the tcp_time_wait_interval by running the following:

```
ndd -get /dev/tcp
```

The default value of the parameter is 240000 milliseconds (four minutes). To change the value to one minute, set tcp_time_wait_interval to 60000, as in the following:

```
ndd -set /dev/tcp tcp_time_wait_interval 60000
```

Note: For Solaris 2.6, the parameter is check tcp_close_wait_interval.

---

tcp_keepalive_interval

View the current value (in milliseconds) of tcp_keepalive_interval by running the following:

```
ndd -get /dev/tcp tcp_keepalive_interval
```

To set a new value, run:

```
ndd -set /dev/tcp tcp_keepalive_interval <value>
```

where *<value>* is a number representing the new value.

tcp_conn_hash

View the current value of the TCP parameter tcp_conn_hash by running the following:

```
ndd -get /dev/tcp tcp_conn_hash
```

Backup devices

NetWorker software and other storage management products use backup devices to do the following:

- Write backed-up data to the storage media volume.
- Read saved data from the storage media during recoveries.

Devices pass data to and from the NetWorker server itself, or from a NetWorker storage node through a SCSI connection. Backup devices can include optical disks, and, in most cases, tape drives. When a backup device has multiple drives, such as for autochangers, each drive is considered a separate device.

Device write performance tuning methods

If the disk read speed is noticeably faster than the ability of the device to write, that might indicate a problem with the write performance of the device. To perform a thorough evaluation of device performance, follow these guidelines:

- Measure the write speed of each individual device while the NetWorker software is *not* in run mode.
- Measure the write speed of all devices while the NetWorker software is *not* in run mode.
- Measure the write speed of each individual device while the NetWorker software is in run mode.
- Measure the write speed of all devices while the NetWorker software is in run mode.

First evaluate the device performance without the NetWorker software running, and then evaluate the devices again with the NetWorker software running.

To ensure consistency in the measurements, ensure you use the same tape drives for all tests.

Testing tape device performance without running NetWorker

The **mt** program is used to determine tape device performance when the NetWorker software is not running. Use it to evaluate the tape device without a need to consider the overhead incurred when using the NetWorker software.

The **mt** program allows you to operate a magnetic tape device, which includes rewinding, fast-forwarding, and ejecting tapes.

Note: The **mt** program cannot be used to measure the writing data rate, but can indicate performance rates related to having tapes start loaded and rewound prior to measuring write speeds to tape.

Determine tape device performance

To determine tape device performance when NetWorker software is not running, enter the following at the command line:

```
mt -f devicename command count
```

By default, the **mt** program performs the requested operation once. To perform the operation more than once, specify a value in place of `count`.

The **mt** program returns the following exit status:

- 0 if the operations were successful.
- 1 if the command was unrecognized or **mt** was unable to open the specified tape drive.
- 2 if an operation failed.

Table 6 on page 27 lists the available commands to use with the **mt** command.

Note: Older UNIX systems may have different options for the **mt** command. Before using the **mt** command, check the **mt** man page on your system for the available options.

Table 6 Commands to use with the mt command

Enter this command	To
eof, weof	Write *count* end of file (EOF) marks at the current position on the tape.
fsf	Forward space over *count* EOF marks. The tape is positioned on the first block of the file.
fsr	Forward space *count* records.
bsf	Backspace over *count* EOF marks. The tape is positioned on the beginning-of-tape side of the EOF mark.
bsr	Backspace *count* records.
nbsf	Backspace *count* files. The tape is positioned on the first block of the file. This is equivalent to count + 1 **bsf** followed by one **fsf**.
asf	Absolute space to *count* file number. This is equivalent to a rewind followed by **fsf** count.

Table 7 on page 27 lists the available commands where *count* is ignored.

Table 7 Available mt commands with count ignored

Enter this command:	To
eom	Space to the end of recorded media on the tape.This is useful for appending files onto previously written tapes.
rewind	Rewind the tape.
offline, rewoffl	Rewind the tape and, if appropriate, take the drive unit offline by unloading the tape.
status	Print status information about the tape unit.
retension	Rewind the cartridge tape completely, and then wind it forward to the end of the reel and back to the beginning of tape to smooth out tape tension.
erase	Erase the entire tape.
format	Format the tape at a low level.

Limitations

Not all tape devices support all options. Some options are hardware-dependent. For example, by entering the following command on a Windows computer:

```
D:\>mt -f \\.\Tape0 status
```

the following output is displayed:

```
\\.\Tape0:
Media Capacity  = 2.12GByte
Media Remaining = 3.06GByte
Media Block size = 0
Media Partition Count = 0
Media is not write protected
default block size = 8192
maximum block size = 1040384
```

The maximum block-size value limits the maximum number of bytes per SCSI transfer on disk reads and volume writes.

Test the native OS driver positioning

Use the **mt** command to test whether the native OS driver is able to position.

To test the driver:

1. Write a file.
2. Issue the following command:

   mt fsf 1

3. Write a second file.
4. Issue the following command:

   mt bsf 2

If the first file cannot be read, the driver is not positioning correctly.

Test device performance with bigasm

To measure SCSI throughput and the write speed of a device while the NetWorker software is running, run the **bigasm** module. The **bigasm** module is run from a directive and generates a file of the specified size, transfers it over a SCSI connection, and writes it to a tape or optical device.

The description and options of **bigasm** (and all other application specific modules) are the same as for the **uasm** module. For a full description of the **uasm** module, see "Disk read speed testing methods" on page 17.

To set up a **bigasm** test:

1. Create a file, *bigasm.file*, that contains the following command:

   bigasm -S*size* **:** *filename*

   For example:

   bigasm -S100M : bigfile

2. Save the file using the NetWorker server. For example, if the directive file is called *bigasm.file* and the NetWorker server is called *jupiter*, enter the following command:

   save -s jupiter -f /bigasm.file /tmp/*filename*

   The amount of time it takes for the backup to complete indicates the efficiency of the SCSI throughput and the write speed of the device.

   The **bigasm** module also can be used in a scheduled backup by creating a directive for it.

Device performance tuning methods

The following sections address specific device-related areas to improve performance.

Input/output transfer rate

Input/output (I/O) transfer rates can affect device performance. The I/O rate is the rate at which data can be written to the device. Depending on the device and media technology, device transfer rates can range from 500 KB per second to 20 MB per second. Default block size and buffer size of a device affect its transfer rate. If I/O limitations are hampering improved performance of the NetWorker server, try upgrading the equipment to affect a better transfer rate.

Built-in compression

Turn on device compression to increase effective throughput to the device. Some devices have a built-in hardware compression feature. Depending on how compressible the backup data is, this can improve effective data throughput, from a ratio of 1.5:1 to 3:1.

Drive streaming

To obtain peak performance from most devices, stream the drive at its maximum sustained throughput. Without drive streaming, the drive must stop to wait for its buffer to refill or to reposition the media before the drive can resume writing. This can cause a delay in the cycle time of a drive, depending on the device.

Device load balancing

Balance data load for simultaneous sessions more evenly across available devices by adjusting sessions per device. This parameter specifies the minimum number of save sessions to be established before the NetWorker server attempts to assign save sessions to another device.

File type device and staging

The NetWorker software includes file type and advanced file type devices and save set staging features, which can be used to improve backup performance. Since reading and writing data to disk is generally faster than to tape or optical media, use the file type or advanced file type device feature to direct the backups to disk media instead of tape or optical media.

The advanced file type (adv_file) is designed for large disk devices. It differs from the file type device because the volume for the advanced file type device is never marked full. When an advanced file type device is out of disk space, the current backup is suspended. The following message appears:

```
filesystem full - recover adv_file space
```

Immediately after the message displays, the NetWorker software begins deleting expired save sets until more space is made available. If enough space is cleared, the backup continues.

Use save set staging in conjunction with backups to the file type or adv_file device. Save set staging lets you move save sets backed up from one medium to another according to criteria you set, such as:

- Space remaining on the disk.
- Age of the save set.

With staging, you can automate transferring older backups from the file type device or adv_file device to a less expensive storage media. This frees up hard disk space for new backups.

Configuration of file type and adv_file devices

A possible configuration to maximize performance of file type or an adv_file device is:

1. Create one directory per user, preferably with a mnemonic name matching a client.
2. Create one file type or adv_file device per directory; set the volume size to 1 GB.
3. Create one client per system, setting the device that is always used to have the matching mnemonic device name.

This configuration limits the system to a maximum of 64 users with the NetWorker software, since one user equals one device, and that is how many devices Power Edition supports. However, if you use storage nodes, you can have 64 devices per node up to a maximum of 256 devices total for the NetWorker server and all attached storage nodes.

If you back up the data to file type devices or adv_file devices, you get high-speed backups to a hard drive in near real time, but the disk tends to fill quickly with backups.

Without save set staging

Without save set staging, disk usage will have to be closely monitored to avoid running out of space for the backups. Also, you will have to either move save sets to other storage media manually or create short browse and retention policies.

With save set staging

With save set staging, the process of moving data from disk storage to tape or optical disk is automated, and the space occupied by the save set is freed to make space for new backups. The browse and retention policies for the save sets on tape or optical disk can be as long as necessary. Also use save set staging to move files from other types of media, but the space is only reclaimed on the file-type device.

If disk contention is a concern when backing up to file type or adv_file devices, use smaller drives (for example, six drives of 4 GB each and four users per drive, instead of one 23 GB drive).

You also can use a RAID array, which provides redundancy in case of a drive failure. Microsoft Windows includes a software RAID capability that works well, but uses many CPU cycles. A hardware RAID solution is a more efficient, but more expensive solution.

Network hardware

If you back up data from remote clients, the routers, network cables, and network interface cards affect the backup and recovery operations. This section lists the performance variables in network hardware and suggests some basic tuning for networks. The following items address specific network issues:

- Network input/output bandwidth

  The maximum data transfer rate across a network rarely approaches the specification of the manufacturer because of network protocol overhead.

IMPORTANT

The following statement concerning overall system sizing needs to be taken into account when considering bandwidth.

Each attached tape drive (same for physical drive, VTL or AFTD device) uses part of the available I/O bandwidth of the system and also consumes CPU since data still has to be processed.

- Network path

  Networking components such as routers, bridges, and hubs consume some overhead bandwidth, which degrades network throughput performance.

- Network load

CAUTION

Do not attach a high number of high-performance tape drives or AFTD devices directly to a backup server.

For example, there is no system which can fully sustain 8 by LTO3 (considered normal practice in many environments).

The same rule (although in smaller amounts) applies to network interfaces; that is, processing data on each high-speed network interface card and for each IP address can use a significant amount of CPU cycles available on the server.

For example, a mid-size system with 4 by 1-Gbit, NICs will use over 50 percent of its power just to process TCP data during a backup window.

Other network traffic limits the bandwidth available to the NetWorker server and degrades backup performance. As the network load reaches a saturation threshold, data packet collisions degrade performance even more.

When using a VTL or AFTD solution from any vendor which utilizes SATA drives, do not overlap staging and backup operations. The recommendation is to have enough capacity so that staging occurs outside of the main backup window. Here, the reason is that regardless of the performance of the array, ATA technology experiences significant performance degradation on long parallel read and write streams.

Systems with a smaller number of high-performance CPUs generally outperform systems with a high number of CPUs because the majority of processing is limited to specific processes or threads inside those processes (although the overall number of NetWorker processes can be high).

How to tune network hardware performance

Employ the following measures to improve network hardware performance:

- Add additional network interface cards to client systems to expand network bandwidth available to clients.
- Upgrade to faster LAN media, such as 100 MB Ethernet or FDDI, to improve network bandwidth. Note, however, that TCP packet overhead can reduce the total transfer rate.
- Configure clients with high-transfer rate requirements to show a preference for a particular backup server (or storage node) on the same subnet; avoid router hops between the NetWorker server and clients.
- Minimize the number of network components in the data path between the client and server.

3

Testing and Tuning on Windows

This chapter addresses specific performance testing and tuning tasks to perform on a NetWorker server in a Windows environment.

The following topics are discussed in this chapter:

Windows performance monitor

Use the Windows performance monitor program to test system performance. In the performance monitor, choose:

- Which objects to watch. For example, the processor or memory.
- Which counters to test. For example, available bytes of memory.
- How to display the data.

The default view of the data is a realtime line chart. However, the data also can be logged or formatted as a report, or alerts created to warn when a counter reaches a certain limit.

Note: The following procedures have been updated for the Windows XP interface. For previous Windows OS, the interface names are in parenthesis ().

Start the Windows performance monitor

To start the Windows performance monitor on the NetWorker server:

1. From the Control Panel, select **Administrative Tools** > **Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.

   Note: The default viewing mode of performance monitor is **Graph**, but the performance monitor-related tasks in this chapter use the **Report** mode.

4. To open the **Add Counters** dialog box, click the **Data** tab and click **Add**.
5. To ensure the local computer is the one being monitored, select **Use Local Computer Counters**.
6. From the **Performance object** (Object) list, select the NetWorker server component to be tracked by the Performance monitor.

   The **Performance object** (Object) choice selected determines which **Counter** and **Instance** choices are available.
7. From the **Select counters from list** (Counter) box, select a counter. Click **Explain** to view descriptions of each counter.
8. From the **Select instances from list** (Instances) box, select an instance.
9. When all choices are selected, click **Add**.
10. Click **Close**.

    The **Performance Monitor** window displays the NetWorker server and component to be monitored.
11. Repeat steps 4 through 10 for each component on the NetWorker server whose performance you want to test.

    The **Performance object** (Object) window lists each component separately under the name of the NetWorker server:

    1. Once all the NetWorker server components to be tested are added, click **OK**.
    2. Start a backup, and view or capture the results.

To save the performance monitoring information in log form, select **Performance Logs and Alerts** in the **Performance** window.

To test the maximum performance of the NetWorker software:

1. Shut down all extraneous processes.
2. Run just the necessary software on the NetWorker server.

Testing a NetWorker server from another computer

Run the performance monitor on a different Windows computer on the network during controlled tests of the NetWorker server because the performance monitor uses system resources.

When the performance monitor is run on a different Windows system, it still uses some resources on the NetWorker system, especially adding to network traffic during remote backups. However, the performance of the processor and SCSI throughput on the NetWorker system are unaffected.

Test a NetWorker server from another Windows computer

To test the performance of one or more NetWorker servers from another Windows computer:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.

   Note: The default viewing mode of performance monitor is Graph, but the performance monitor-related tasks in this chapter use the Report mode.

4. To open the **Add Counters** dialog box, click the **Data** tab and click **Add**.
5. Select **Counters** from **Computer** and type in the name of the NetWorker server.
6. From the **Performance object** (Object) list box, select an object. The **Performance object** (Object) choice determines the available Counter and Instance choices.
7. From the **Select counters from list** (Counters) box, select a counter. Click **Explain** to view descriptions of each counter.
8. From the **Select instances from list** (Instances) box, select an instance.
9. When all choices are selected, click **Add**.
10. Click **Close**.

    The **Performance Monitor** window displays the NetWorker server and the component to be monitored.

11. Repeat steps 5 through 9 for each component on the NetWorker server to be tested.

    The **Performance Monitor** window lists each component separately under the NetWorker server.

12. Repeat steps 4 through 9 if you want to monitor the performance of any other NetWorker servers (or any other Windows computer in general).

13. When you finish adding all the NetWorker server components to be tested, click **OK**.
14. Start a backup, and view or capture the results.

    You can save, add to, and redirect the performance monitoring information that results from this procedure. The Microsoft Windows operating system documentation provides more detailed information about the performance monitor.

Physical disk

This section describes how to use performance monitor to test and evaluate physical disk performance problems related to the NetWorker server. "Physical disks" on page 17 provides a general discussion of disk-related performance problems and solutions.

Test disk performance

To test the disk performance of the NetWorker server in a Windows environment:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.
4. To open the **Add Counters** dialog box, click the **Data** tab and click **Add**.
5. From the **Performance object** (Object) list box, select **Physical Disk**.
6. From the **Select counters from list** (Counters) box, select the following counters:
   - **% Disk Time**
   - **Current Disk Queue Length**

To view details on each counter, click **Explain**.

If the NetWorker server has more than one physical disk, select these counters for each disk:

1. When all choices are selected, click **Add**.
2. Click **Close**.

   The **Performance Monitor** window displays the NetWorker server and the component to be monitored.
3. Click **OK**.
4. Start a backup, and view or capture the results.
5. If the disk performance counters do not work, open a command prompt window and enter the following command:

   ```
   diskperf -y
   ```
6. Restart the NetWorker server to activate the counters.

Disk performance evaluation methods

To determine if the NetWorker server has a disk-related blockage, check the values gathered from the performance testing:

- If the value for the Percent Disk Time counter exceeds 90 percent, tune the disk performance as suggested in "Disk tuning methods" on page 37.

  This counter measures the percentage of operating time the disk is occupied with read or write requests.

- If the value for the Disk Queue Length counter exceeds 2, tune the disk performance as suggested in "Disk tuning methods" on page 37.

  This counter measures the average number of read or write requests that were waiting to be serviced during the sampling interval.

Disk tuning methods

The methods for tuning disk performance on the NetWorker server in a Windows environment are the same for tuning computer disks in general, including:

- Defragment files to reduce the number of disk seeks.
- Use larger files or fewer files to reduce the number of disk seeks.
- Change to a faster SCSI adapter, with synchronous I/O support and bus mastering.
- Use stripe sets on multiple disks to accommodate large files and databases, so the physical disks can work in parallel to read data.
- Spread disk-intensive programs onto different servers.
- Purchase disks with higher transfer rates.

Memory

This section describes how to use the performance monitor to test and evaluate memory-related performance problems related to the NetWorker server. For a general discussion of memory-related performance problems and solutions, see "Memory usage" on page 20.

Test memory performance

To test the memory-related performance of the NetWorker server in a Windows environment:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.
4. To open the **Add Counters** dialog box, click the **Data** tab and click **Add**.
5. From the **Performance object** (Object) list box, select **Memory**.
6. From the **Select counters from list** (Counters) box, select the following counters:
   - **Available Bytes**
   - **Commit Limit**
   - **Committed Bytes**
   - **Pages/sec**

To view details on each counter, click **Explain**.

1. When all choices are selected, click **Add**.
2. Click **Close**.

   The **Performance Monitor window** displays the NetWorker server and the component to monitor.
3. Click **OK**.
4. Start a backup, and view or capture the results.

   "Memory performance evaluation methods" on page 38 provides information on how to analyze the results of these tests.

Memory performance evaluation methods

Use the values gathered from the performance testing to determine if the NetWorker server has a memory-related blockage. Table 8 on page 38 lists the counters and which values indicate that tuning is required.

Table 8 Counter values

Counter	Values
Available Bytes	This counter measures how much memory is unoccupied. The smaller the value, the slower the performance. If the value is less than 4 MB, the memory performance requires tuning.
Pages/sec	This counter measures virtual memory activity; that is, how often the memory writes to disk. When the computer is memory-bound, it pages to the virtual memory on disk more often. A value of 20 or greater indicates that tuning is required.
Committed Bytes	This counter is the total memory of all applications in use at the moment. Applications generally reserve more memory than they use during regular operation. However, an application does not write data to memory without committing the memory first. Therefore, the value for Committed Bytes shows the amount of memory the applications currently need. If more memory is committed than you have physical memory to accommodate, the computer pages memory contents to disk (virtual memory), and the computer works more slowly because physical memory is faster than virtual memory.
Commit Limit	This counter is the size of virtual memory that can be committed without having to extend the paging file. Every time the Windows computer extends the paging file, it goes to disk and searches for more space to use for virtual memory. This process has a high performance cost, and you might run out of disk space. The Committed Bytes value should not consistently exceed the Commit Limit value.

Tuning memory performance in Windows

The following section explains how to tune memory performance on the NetWorker server.

Note: To determine a good baseline size for the virtual memory file *pagefile.sys*, log the Committed Bytes counter over a period of time, add 10 to 20 percent to the maximum value, and enter that value in the Initial Size (MB) text box in the Virtual Memory dialog box.

Adjust the virtual memory

To adjust the allocated size of virtual memory on the NetWorker server in a Windows environment:

1. From the Control Panel, select **System**.
2. Click the **Advanced** tab and click the **Settings** button in the **Performance** box.

3. In the **Performance Options** window, click the **Advanced** tab.
4. Select **Change** to open the **Virtual Memory** dialog box.
5. As needed, adjust the size and location of the disk space allocated to virtual memory.
6. Click **OK** when the change is completed.

Adjust the physical memory allocation

To adjust physical memory allocation on the NetWorker server in a Windows environment:

1. From the Control Panel, click the **Network Connections** (Network and dial-up connections) icon.
2. Right-click **Local Area Connection** and select **Properties**.
3. In the **Local Area Connection Properties** dialog box, select **File and Printer Sharing for Microsoft Networks** and click **Properties**.
4. Select one of the following options:
   - **Minimize Memory Used**: Use when the server has fewer than 10 users, such as a local backup on a NetWorker server.
   - **Balance**: Use when the server has 10 to 64 users.
   - **Maximize Throughput for File Sharing**: Allocate most of the memory to the file server module, for systems with more than 64 users.
   - **Maximize Throughput for Network Application**: Use for a client-server application server, such as a NetWorker server with remote clients.
5. Click **OK**.

Memory-related server tuning methods

To speed up access to and from virtual memory, you can implement the disk recommendations in "Disk tuning methods" on page 37. You can also defragment the disks where *pagefile.sys* resides.

Do not put *pagefile.sys* on either of the following:

- A stripe set, because virtual memory is less efficient when it is fragmented.
- A mirrored drive, because every write is done twice, which slows writes to virtual memory.

Memory that the NetWorker server uses can be reduced. Suggestions to reduce memory requirements include the following:

- Reduce the number of other applications that run on the NetWorker server. Dedicate the NetWorker server to run the database server, NetWorker client, NetWorker Module client, and NetWorker server.
- Stop services you do not use. For example, if you have only SCSI devices, stop the ATDISK service, which is for IDE devices only.
- Remove communications protocols you do not use. For example, if you only use TCP/IP, but you also have SPX installed, remove SPX.

CPU

The following sections address testing and tuning issues for the NetWorker server CPU(s). "CPU usage" on page 21 provides a general discussion of how CPU issues affect NetWorker software performance.

Testing CPU performance

Total CPU utilization and CPU utilization for each individual process demonstrate the CPU performance of the NetWorker server.

Test total CPU utilization

To test total CPU utilization of the NetWorker server in a Windows environment during backup:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.
4. Click the **Data** tab and click **Add**.
5. From the **Performance object** (Object) list box, select **Processor**.
6. From the **Select counters from list** (Counters) box, select **% Processor Time**. Click **Explain** to view details on this counter.
7. Select the processors to be tested from the **Select instances from list** (Instances) box if the NetWorker server has more than one processor.
8. When all choices are selected, click **Add**.
9. Click **Close**.

   The **Performance Monitor window** displays the NetWorker server and the component to be monitored.
10. Click **OK**.
11. Start a backup, and view or capture the results.

Test CPU performance by process

You also can test CPU utilization for each process involved in the backup, to determine which processes use most of the CPU capacity.

To test total CPU utilization of the NetWorker server in a Windows environment during backup:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.
4. Click the **Data** tab and click **Add**.
5. From the **Performance object** (Object) list box, select **Processor**.
6. From the **Select counters from list** (Counters) box, select **% Processor Time**. Click **Explain** to view details on this counter.
7. Select the processors to be tested from the **Select instances from list** (Instances) box, if the NetWorker server has more than one processor.

8. When all choices are selected, click **Add**.
9. Click **Close**.

   The **Performance Monitor window** displays the NetWorker server and the component to be monitored.
10. Click **OK**.
11. Start a backup, and view or capture the results.

    There is a CPU blockage if total CPU utilization remains above 90 percent for long periods during the backup process.

Server input/output

The following sections address testing and tuning issues for the NetWorker server input/output (I/O). "Server input/output" on page 21 provides a general discussion of how I/O issues affect NetWorker software performance.

To test the I/O of the NetWorker server most effectively, you must start and configure the performance monitor, and then run specific NetWorker server I/O-related tests. "Server input/output" on page 21 provides more information about these tests.

Test input/output performance

To start and configure the performance monitor to test the NetWorker server I/O in a Windows environment:

1. From the Control Panel, select **Administrative Tools > Performance**.
2. Right-click anywhere in the **System Monitor** details pane and select **Properties**.
3. In the **General** tab, select **Report**.
4. Click the **Data** tab and click **Add**.
5. From the **Performance object** (Object) list box, select **System**.
6. From the **Select counters from list** (Counters) box, and select **File Write Bytes/sec**. Click **Explain** to view details on this counter.
7. Click **Add**.
8. Click **Close**.

   The **Performance Monitor window** displays the NetWorker server and the component to be monitored.
9. Click **OK**.
10. Run the following tests, and view or capture the results:
    - For the **uasm** program: "Disk read speed testing methods" on page 17.
    - For the **mt** program: "Testing tape device performance without running NetWorker" on page 26.
    - For the **bigasm** program: "Test device performance with bigasm" on page 28.

Tuning input/output performance

The rate at which a SCSI device can write data to media is limited by the amount of data that transfers over the SCSI bus in a single I/O operation. To tune SCSI

performance, change the number of memory pages the SCSI host bus adapter (HBA) can scatter and/or gather in a single direct memory access (DMA). The number of bytes that can be transferred in a single I/O with a SCSI device is limited by the number of memory pages that the SCSI host bus adapter can scatter and/or gather in a single DMA.

The SCSI driver of each vendor specifies the maximum number of pages that the HBA can scatter and/or gather for the DMA transfer. When you multiply the page size by the number of scatter and/or gather entries, the product is the effective limit to the number of bytes that can be transmitted in a single I/O operation.

The equation is the following:

I/O operation size = Page size * (# of scatter/gather entries -1)

User memory is not necessarily page-aligned; therefore:

If MaximumSGList = 255, then the I/O transfer size is 4 KB * 254 = 1016KB.

For the drivers of some SCSI host adapters, you can change the maximum number of scatter and/or gather pages. Many tape drives and disk drives perform better with a larger I/O limit. A change in the number of scatter and/or gather pages can make a large performance difference. For example:

- The Ampex DST tape drive writes only 3 MB per second with 64 KB SCSI transfers and 20 MB per second with 992 KB SCSI transfers.
- The SCSI disk "CYBERNET 10XP" reads 1.7 MB per second with a 32 KB read buffer size and 17 MB per second with a 200 KB read buffer size.

To change the maximum scatter and/or gather pages setting in the Windows Registry, add the following new Registry subkey to specify the maximum number of scatter and/or gather list elements for each device on a given bus:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DriverName\
Parameters\Device number\MaximumSGList
```

IMPORTANT

Modifying the Registry is dangerous. Do *not* attempt to modify the Registry unless you have an up-to-date backup of the Registry to which you can revert and you are convinced that the modification you make is correct. Inappropriate changes to the Registry can result in having to reinstall the Windows software.

Add a subkey entry to the Windows registry

To add the subkey entry to the Windows Registry:

1. Start the Windows Registry Editor program as follows:
   a. Select **Start > Run**.
   b. Enter the following value in the Open text box:

      `regedt32`

   c. Press **Enter**.

      You also can open this program at the following location:

      `%SystemRoot%\Winnt\System32\regedt32.exe`
2. Select **Window > HKEY_LOCAL_MACHINE** to make that window active.
3. Select **Edit > Add Key** and enter **Parameters** for Key Name to create a key named Parameters, if one does not exist.

4. To create a key named Device, open the **Parameters** key by selecting **Edit > Add Key** and enter **Device** for Key Name.
5. To create a DWORD named MaximumSGList, open the **Device** key by selecting **Edit > DWORD** and entering **MaximumSGList** for **DWORD Name**.
6. Open the **MaximumSGList Value Name** and replace the existing Value Data in Decimal with a number at least as large as the block size you need for the device.

   To calculate the value, use the following equation:

   ```
   (Block_Size / 4) + 1 = Value
   ```

Table 9 on page 43 provides block size and corresponding decimal and hex values.

Table 9 Block sizes

Block size	Decimal value	Hex value
64k	17 decimal	11 hex
96k	25 decimal	19 hex
128k	33 decimal	21 hex
256k	65 decimal	41 hex
512k	129 decimal	81 hex

To locate the drives and check the maximum block size:

1. Reboot the machine.
2. To locate the drives, run the **inquire** program:

   ```
   C:\>inquire
   ```

   Results similar to the following appear:

   ```
   scsidev@0.5.0:NEC CD-ROM DRIVE:4661.06|CD-ROM
   scsidev@0.6.0:ARCHIVE Python 06408-XXX8071|Tape, \\.\Tape0
   scsidev@1.0.0:HP C5173-7000 3.02|Autochanger (Jukebox)
   scsidev@1.2.0:QUANTUM DLT7000 1732|Tape, \\.\Tape1
   scsidev@1.3.0:QUANTUM DLT7000 1732|Tape, \\.\Tape2
   scsidev@2.0.0:SEAGATE ST39204LC 0002|Disk, \\.\PHYSICALDRIVE0
   scsidev@2.1.0:QUANTUM ATLAS V  9 SCA  0201|Disk, \\.\PHYSICALDRIVE1
   scsidev@2.6.0:DELL 1x4 U2W SCSI BP 5.35|Processor
   ```

3. To check if the maximum blocksize changed, run the **mt** program:

   ```
   C:\>mt -f \\.\Tape1 status
   ```

   Results similar to the following appear:

   ```
   \\.\Tape1:
   Media Capacity  = 15.20GByte
   Media Remaining = 13.60GByte
   Media Blocksize = 0
   Media Partition Count = 0
   Media is not write protected
   default blocksize = 65536
   maximum blocksize = 131072
   minimum blocksize = 1
   MaximumPartitionCount = 0
   Partition = 0
   Logical block position = 44815
   EOTWarningZoneSize = 0
   ```

```
CompressionEnabled
Features: ...
```

In the previous example, the maximum block size is 128 KB (131072 divided by 1024).

Other suggestions for tuning I/O performance are:

- Move the hardware configuration around; for example, move some disks off an overloaded SCSI bus.
- If possible, use PCI adapters in the NetWorker server. If you must use other adapters, EISA adapters are better than ISA adapters because ISA adapters have bus contention problems that degrade system performance.
- Use a dual-PCI backplane system, rather than a bridged-PCI backplane.
- Use multiple SCSI buses with devices of the same type (for example, UltraSCSI) on the same bus.
- Use short, high-quality cables with active terminators.
- Ensure that SCSI communications are occurring synchronously, not asynchronously.

Devices

The following sections address testing and tuning issues for the NetWorker backup devices within the Windows environment. "Backup devices" on page 26 provides a general discussion of how device issues affect NetWorker software performance.

Test device performance

To test device input/output, see the **bigasm** program test described in "Test device performance with bigasm" on page 28.

Tuning device performance

Most tape drives come with hardware data compression enabled by default. When compression is enabled, device performance and media capacity can double.

Hardware data compression is controlled by the **NSR_NO_HW_COMPRESS** environment variable. When you set this environment variable to Yes, data is *not* compressed by the hardware before it is written to media.

Note: For data compressed before it reaches the device, additional compression might increase the size of the data. If the data is compressed on the device, you can disable compression on the storage device.

When you change the name of an existing variable, Windows NT treats it as a new variable and does not overwrite or delete original variable.

Tune device performance

To tune the hardware compression system variable:

1. From the Control Panel, select **System**.
2. Click the **Advanced** tab and click **Environment Variables**.

3. Review the variables and values listed in the **System Variables** list box. Depending on whether the **NSR_NO_HW_COMPRESS** variable already exists or not, follow the appropriate step:

   - If the **NSR_NO_HW_COMPRESS** variable is already listed, and you simply want to change its value, select it. The variable name and its value appear in the Variable and Value text boxes.

     a. Change its value to **Yes**.

     b. Click **OK** to confirm the change.

     The updated value of the **NSR_NO_HW_COMPRESS** variable is reflected in the System Variables list box.

   - If the **NSR_NO_HW_COMPRESS** variable is not listed, you need to create it.

     a. Click **New**.

     b. Enter the variable name **NSR_NO_HW_COMPRESS**

     c. Enter the value **Yes**.

     d. Click **OK**.

4. When you have completed adding or modifying the **NSR_NO_HW_COMPRESS** variable and/or its value, click **OK**.

5. Restart the NetWorker server.

   To see the device compression setting, enter the following on the command line:

   ```
   mt -f \\.\Tape0 stat
   ```

   In the output of a device with compression enabled, the following appears:

   ```
   Compress Enabled
   ```

Windows software

The following variables affect the overall Windows operating system performance and the speed of backup:

- File-system overhead

  File-system input/output (I/O) can degrade I/O performance. If you integrate logical volume managers with the file system, you can increase disk performance and add many other features to improve manageability of large file systems. Also, depending on the level of RAID (Redundant Array of Independent Disks) you implement, RAID can improve or adversely affect backup performance.

- Server memory usage setting

  You can set server memory use to be optimized for the following:

  - Small number of clients
  - Large number of clients
  - File server
  - Domain controller

  "Memory performance evaluation methods" on page 38 provides more information.

- Page file size

You can adjust the size and location of the virtual memory file (*pagefile.sys*) on the local disks. "Memory performance evaluation methods" on page 38 provides more information.

Recoveries

Recovery performance can fluctuate based on a variety of issues, including network traffic, bottlenecks, and file sizes. One method of recovering data efficiently is to start multiple **recover** commands simultaneously through the save set recovery method.

For example, you can start three different save set recover programs instead of one save set recover program with three save sets. Separate recover programs provide the maximum parallelism possible given the number of daemons and the actual volume and save set layout.

Note: If multiple, simultaneous **recover** programs are run from the same tape, do not let the tape mount and start until all recoveries are ready. If the tape is used before all recoveries are ready, the tape will be read through multiple times and recovery performance will be slower, rather than faster.

Perform multiple, simultaneous recoveries

To perform multiple recoveries:

1. Open one NetWorker User program for each save set that needs to be recovered.
2. In each window, select the save set to be recovered, but do not start the recovery process.
3. Once all three save sets are selected in their respective window, execute each recovery consecutively.
4. Measure the time it takes to recover all three save sets.
5. Once the recoveries are finished, perform another recovery with all three save sets in the same recover request.
6. Compare the time needed for the multiple recover requests to the time needed for the single recover request.

The multiple recoveries should have finished *before* the single recovery of multiple save sets. If the single recovery completed faster, the tape may have mounted before all recoveries could start or the computer is CPU-bound rather than I/O-bound. To determine the cause of the bottlenecks, use the performance and activity monitoring tools available with the operating system.

4

Testing and Tuning on UNIX and Linux

This chapter provides additional performance testing and tuning information for the NetWorker server on a UNIX or Linux platform.

These topics are discussed:

The multitude of platform- and network-specific variables that can impact UNIX or Linux system performance cannot be covered comprehensively in a single chapter. The scope of this chapter is therefore limited to the detection and analysis of processes that have an impact on how efficiently the NetWorker server can perform backups and recoveries.

While there are no simple solutions to performance issues, this chapter provides a general testing strategy for locating performance blockages. The testing strategy uses benchmarks and baselines derived from common utilities.

This chapter assumes that you are familiar with:

- UNIX or Linux system infrastructure, including networking and devices.
- Writing shell scripts and using Linux or UNIX utilities, such as **tar**.

The scripts in this chapter use standard utilities. The examples are specific to Solaris, but the scripts can be applied to most UNIX and Linux platforms with minimal changes.

The testing strategy and scripts use this scenario:

- All backup processes are local (that is, not over a network).
- All tape devices operate at the same speed.
- Data is stored on a file system (no databases are used).

Physical disks

These sections provide information and suggestions on testing and evaluating the read speed performance on the physical disks of the NetWorker server. The UNIX and Linux operating systems provide several utilities for determining disk read speed. These sections detail their usage.

Apply this testing strategy to collect baselines to measure tape drive performance:

- Test the read speed of each individual disk while the NetWorker software is *not* in run mode.
- Test the read speed of all disks while the NetWorker software is *not* in run mode.
- Test the read speed of each individual disk while the NetWorker software is in run mode.
- Test the read speed of all disks while the NetWorker software is in run mode.

Benchmark evaluation

The utility **/bin/time** reports the amount of time required to execute a program, breaking down the total time into three components:

- Real time — The time that the program takes to run as it would be measured by a user sitting at the terminal using a stopwatch.
- User time — The actual time that the computer spent executing code in the user state.
- System time — The time the computer spent executing UNIX or Linux system code on behalf of the user.

Note: When running the **tar** or **uasm** command, do not specify a file or mountpoint accessed recently. The file may still be in memory and not read from the disk.

The result of running the **/bin/time tar cvf - /space2 > /dev/null** command on a Solaris computer provides this result for the mount point */space2*:

```
RESULTS:
real     6:53.4
user       18.2
sys      1:48.1
```

Test the read speed of a single disk

To test the read speed of an individual disk:

1. Enter the **df** command to display a list of mounted file systems and their respective sizes.
2. Run the **/bin/time** utility to determine how long it takes to read each file system. For example, on Solaris, enter:

   /bin/time tar cf - *mount_point* **> /dev/null**

To achieve a consistent result, no other disk or system activity should be running when you run this command.

Test the read speed of several disks

After you execute the **/bin/time** utility for each file system, measure the read time for all file systems on each SCSI bus. If you are running the disks in parallel, develop a baseline for the number of parallel disks. For example, if there are 300 disks, and 10 are in parallel, measure the baseline for the 10 disks.

To streamline this task, create a shell script using the editor of your choice. Following is an example shell script from a Solaris computer that was created for multiple disks (*/space*, */space2*, and so forth):

```
/bin/time tar cf - /space > /dev/null &
/bin/time tar cf - /space2 > /dev/null &
/bin/time tar cf - /space3 > /dev/null &
...
```

Test the read speed of a single disk while running the software

To test the read speed of an individual disk using the NetWorker software:

1. Enter the **df** command to display a list of mounted file systems and their respective sizes.
2. Run the **uasm** program for each disk. For example, on Solaris, enter:

   /bin/time uasm -s *mount_point* **> /dev/null**

   Depending on the configuration of the computer, you might need to specify the path for **uasm** in the command.

   "Disk read speed testing methods" on page 17 provides more information on the **uasm** command.

Test the read speed of several disks while running the software

Test the read speed for all the disks on a SCSI channel. If you are running the disks in parallel, develop a baseline for the number of parallel disks. For example, if there are 300 disks, and 10 are in parallel, measure the baseline for the 10 disks.

To streamline this task, create a shell script using the editor of your choice. The following is an example shell script for a Solaris computer that was created for multiple disks (*/space*, */space2*, and so forth):

```
/bin/time uasm -s /space > /dev/null &
/bin/time uasm -s /space2 > /dev/null &
/bin/time uasm -s /space3 > /dev/null &
...
```

To calculate the read speed of the disk:

1. Enter the **df -k** command to obtain the file size (in kilobytes) of */space2*:

   ```
   % df -k
   Filesystemkbytesusedcapacity Mounted on /dev/dsk/c0t1d0s0
   195257379793546%/space2
   ```

2. Divide the real time by the size (in kilobytes) of */space2*.

   For example, based on the result and size of */space2* above, the read speed of */space2* is 1932 KB per second.

3. Compare the results of the benchmarks generated from:
   - First, the individual disks *without* the NetWorker software running, as compared to the results of individual disks running the NetWorker software.
   - Then, multiple disks *without* the NetWorker software running, as compared to the results of multiple disks running the NetWorker software.

Devices

Apply this testing strategy to collect baselines to measure tape drive performance:

- Test the write speed of each individual tape drive while the NetWorker software is *not* in run mode.
- Test the write speed of all tape drives while the NetWorker software is *not* in run mode.
- Test the write speed of each individual tape drive while the NetWorker software is in run mode.
- Test the write speed of all tape drives while the NetWorker software is in run mode.

To ensure consistency in the measurements, test the same tape drives both when the NetWorker software is running and when it is not.

Benchmark evaluation

Compare the results of the benchmarks generated from:

- First, individual drives writing data without the NetWorker software running, as compared to the results of individual drives writing data when the NetWorker software is running.
- Then, multiple drives writing data *without* the NetWorker software running, as compared to the results of multiple drives writing data while running the NetWorker software.

Test the write speed of a single tape drive

A single tape drive can be tested by using the commands in this section.

Note: Using the **mt rewind** command, followed by the **dd** command, can erase a tape. Exercise caution when using these commands.

To test the write speed of a single tape drive, enter these commands for each tape drive:

```
mt -f /dev/rmt/drive_name rewind
dd if=/dev/zero of=/dev/rmt/drive_name \
bs=block_size count=1;\
/bin/time dd if=/dev/zero of=/dev/rmt/drive_name \
bs=block_size count=1000
```

Although the block size can vary depending on the environment, a block-size value of 32 can be used for purposes of testing and examples.

Test the write speed of several tape drives

Test the write speed of all the tape drives on the SCSI channel. To streamline this task, create a shell script using the command demonstrated in "Test the write speed of a single tape drive" on page 51.

This is an example shell script for a Solaris computer that was created for multiple drives (*0mbn, 1mbn, 2mbn,* and so forth):

```
mt -f /dev/rmt/0mbn rewind; dd if=/dev/zero \ of=/dev/rmt/0mbn
bs=32k count=1; \
/bin/time dd if=/dev/zero of=/dev/rmt/0mbn \
bs=32k count=1000 &
mt -f /dev/rmt/1mbn rewind; dd if=/dev/zero \ of=/dev/rmt/1mbn
bs=32k count=1; \
/bin/time dd if=/dev/zero of=/dev/rmt/1mbn \
bs=32k count=1000 &
mt -f /dev/rmt/2mbn rewind; dd if=/dev/zero \ of=/dev/rmt/2mbn
bs=32k count=1; \
/bin/time dd if=/dev/zero of=/dev/rmt/2mbn \
bs=32k count=1000 &
...
```

Test the write speed of a single drive while running the software

To test the write speed of a single drive while running the NetWorker software:

1. Set target sessions to 1.
2. Label the tape.
3. Set up **bigasm**:
   a. Create an empty directory:

   mkdir /empty

   b. Go to the directory *empty:*

   cd /empty

   c. Create an empty file:

   touch e

   d. Using a text editor, create a *.nsr* file.

   e. Edit the *.nsr* file by adding this directive:

   bigasm -S100M : e

   f. Save the file and edit the text editor.
4. Run this script to determine the baseline:

   save -s *server_name* **/etc/motd; /bin/time save -s** *server_name* **/empty/e**

When NetWorker encounters the .nsr file, the directive results in a 100 MB file written to tape.

The NetWorker server automatically assigns each command to a tape, since target sessions are set to 1.

Test the write speed of several drives while running the software

To test the write speed of several drives, repeat steps 1 through 4 in "Test the write speed of a single drive while running the software" on page 52 for every drive on the SCSI channel simultaneously.

Using the system activity reporter

The System Activity Reporter (**sar**) is a standard UNIX utility for most systems that monitors and reports on system performance, including buffer and block device activity, kernel memory allocation, and CPU utilization. Use this utility to evaluate and tune a UNIX system. The **sar** man page provides more information on this utility.

Monitor buffer activity

If the read and write buffer cache hit rates (%rcache and %wcache) are consistently low, increase the size of the buffer cache to improve disk I/O performance.

To monitor buffer read activity, enter this command:

```
sar -b %rcache
```

To monitor buffer write activity, enter this command:

```
sar -b %wrcache
```

The values of %rcache and %wrcache depend on the combination of applications that the system is running, the speed of its disk subsystems, and the amount of memory available. The ideal value for both is 100, but that value is unlikely on an average system. If either value is consistently below 50 percent, increase the size of the buffer cache.

Monitor block device activity

To monitor block device activity, enter this command:

```
sar -d
```

Results similar to this appear:

```
23:59:44 device %busy avquer+w/s blks/s avwait avserv
23:59:49 Sdsk-0 99.42 4.18 39.39 166.28 80.26 25.24
23:59:54 Sdsk-0 100.00 4.18 38.73 163.64 82.35 25.87
23:59:59 Sdsk-0 100.00 3.98 38.07 171.95 78.32 26.32
```

where:

- *device* is the device on which the activity is measured.
- *%busy* is the percentage of time that the system was transferring data to and from the device.
- A consistently high value for %busy indicates a heavily utilized disk. Increasing CPU power will create a worse bottleneck because the processor produces more I/Os for a disk that is already overloaded.
- *avque* is the average number of requests pending on the device including any on the device itself.
- *r+w/s* is the number of read and write transfers to and from the disk.
- *blks/s* is the number of 512-byte blocks transferred per second.
- The values for *r+w/s* and *blks/s* can be used to calculate the average size of data transfers by using this formula:

  Average size of data transfer = blks/s / r+w/s
- *avwait* is the average time in milliseconds that the request waits in the driver before being sent to the device.

- *avserv* is the average time in milliseconds that it takes a request to complete. The length of time is calculated from the time that the request is sent to the device to the moment the device signals it completed the request. The value of avserv varies depending on the type of disk and any caching on the disk controller.

Use the values generated by **sar -d** to determine which disks are overused and underused. To avoid bottlenecks, balance the disk load by moving some users from the busier disks to the less active disks.

Test CPU utilization

The **sar -u** command identifies disks to which the NetWorker software writes frequently, which could be I/O bottlenecks. To enhance performance, optimize these disks.

To test I/O-related CPU activity, enter this command:

```
sar -u
```

This information is reported:

- *%usr* is the percentage of time the system is in user mode (that is, running user or application code).
- *%sys* is the percentage of time system programs or system calls are run from user programs.
- *%wio* is the percentage of time the system is waiting for I/O.
- The *%wio* value should generally not exceed 25 percent. A higher percentage may indicate speeding up the disk throughput or add I/O bus capacity.
- *%idle* is the percentage of time the system is completely idle.
- If *%idle* is frequently equal to zero, the system is CPU bound. If the system throughput is too low, upgrade to a faster processor or reduce computational load on the system.

This example shows how to test CPU utilization by taking 5 samples at 60-second intervals.

To test CPU utilization, enter this command:

```
sar -u 60 5
```

These results appear:

```
18:22:46 %usr %sys %wio %idle
18:23:4613 9 6 73
18:24:46 25 16 13 45
18:25:46 21 9 1 68
18:26:46 22 10 1 66
18:27:46 22 11 1 66
```

Recoveries

Recovery performance can fluctuate based on a variety of issues, including network traffic, bottlenecks, and file sizes. One method of recovering data efficiently is to start multiple **recover** commands simultaneously through the save set recovery method.

For example, you can start three different save set **recover** programs instead of one **recover** program with three save sets. Separate **recover** programs provide the maximum parallelism possible given the number of daemons and the actual volume and save set layout.

If multiple, simultaneous **recover** programs are run from the same tape, do not let the tape mount and start until all recoveries are ready. If the tape is used before all recoveries are ready, the tape will be read through multiple times and recovery performance will be slower, rather than faster.

Perform multiple, simultaneous recoveries

To perform multiple recoveries:

1. Open one command window for each save set to recover.
2. In each window, enter the **recover** command with the save set to recover, but do not press **Enter**.
3. Once all three **recover** commands are ready, execute each command consecutively.
4. Measure the time to recover all three save sets.
5. Once the recoveries finish, perform another recovery with all three save sets in the same recover request.
6. Compare the time needed for the multiple recovery requests to the time needed for the single recovery request.

The multiple recoveries should finish before the single recovery of multiple save sets. If the single recovery completed faster:

- The tape mounted before all recoveries could start.
- The computer is CPU-bound rather than I/O-bound.

To determine the cause of the bottlenecks, use the performance and activity monitoring tools available with the operating system.

OS tuning recommendations

Starting with NetWorker Release 7.3, architectural changes were introduced into the NetWorker application to provide key features for the following:

- reporting
- day-to-day maintenance through a Java-based interface
- a more reliable certificate-based authentication scheme.

In order to implement these changes in existing NetWorker environments, some Operating System (OS) considerations need to be taken into account when tuning the NetWorker datazone for optimization.

As a result of features that have been implemented, versions of NetWorker 7.3.x and higher rely more heavily on internal and external communications. Large, busy NetWorker 7.2.x environments will benefit from a review of the current configuration to take into consideration a reduction in concurrent NetWorker operations which may overwhelm the underlying operating systems.

NetWorker operations which may impact operating system performance due to increased communication processes and ultimately NetWorker performance:

-Launching of a large number of NetWorker groups with a large number of clients at the same time.

-Having a large number of groups with a large number of clients which have overlapping backup windows.

- Having a large number of mount requests occurring at the same time

- Having overlapping devices requirements causing contention

If you were previously running NetWorker 7.2 and had groups starting at the same time, it may cause minor delays and no other real impact. However, with NetWorker's later releases, it may be more significant if the increase in internal and external communications cause a load that will be higher than the OS can handle.

If there are several local devices associated with the backup server, the OS resources can be exhausted due to excessive hardware resource usage. Consideration needs to be taken to ensure to avoid configuring a system with more devices than it can handle.

Before upgrade run the following:

- nsrim -x
- nsrck -m
- nsrck -L6

to ensure consistency in the NetWorker databases before upgrade.

OS settings

Starting with NetWorker release 7.3 SP4 and NetWorker release 7.4 SP1, the NetWorker savegroup will utilize two file descriptors per each active session.

Since the maximum number of sessions on a backup server is limited to 512 sessions, setting the number of available file descriptors to 1048 is sufficient for all environments. Note that some operating systems have this value by default on 256 which is insufficient for NetWorker operations.

For older versions of NetWorker release 7.3 and 7.4, set the number of file descriptors to a high value (4096). Refer to the OS documentation regarding changes to file descriptors with **ulimit**.

Issues can be encountered from TCP CRC checksuming when offloading is enabled on a network interface card. If that's expected, disable offloading.

IMPORTANT

This will increase CPU usage during high network usage.

```
# ethtool -K eth0 tso off
```

Increasing the network interface card queue length is recommended. Most new network interface cards already have this as a default.

```
# ifconfig eth0 txqueuelen 1000
```

The following settings are indicated for the specific Unix and Linux platforms that NetWorker can operate on.

AIX

Refer to powerlink.emc.com for further details

To increase TCP buffers, modify the following parameters:

```
tcp_sendspace
tcp_recvspace
```

to the value of 524 288

Solaris

Refer to powerlink.emc.com for further details

CAUTION

It is not recommended to run NetWorker 7.3 and later on a Solaris 8 box.

On Solaris 9 and Solaris 10,

```
tcp_conn_req_max_q 1024
```

(Queue depth for a number of non-processed TCP requests)

```
tcp_conn_req_max_q0 4096
```

(normally, 4 x max_q param)

```
tcp_time_wait_interval 60000
```

(this allows Solaris to free connections faster once they are closed)

HPUX

Refer to powerlink.emc.com for further details

It is recommended to increase the **max_thread_proc** kernel parameter from 64 to 256 for versions prior to 7.3.4 and 7.4.2.

TRU64

Refer to powerlink.emc.com for further details

Consider the following settings:

```
/etc/sysconfigtab
```

socket:

```
        somaxconn = 65535
```

```
sominconn = 65535
```

proc:

```
open_max_hard = 32768
open_max_soft = 32768
```

ipc:

```
sem_mni = 1024
sem_msl = 512
sem_opm = 512
sem_ume = 512
shm_allocate_striped = 1
shm_max = 2139095040
shm_min = 1
shm_mni = 256
shm_seg = 256
```

inet:

```
tcbhashnum = 16
tcbhashsize = 8192
```

Linux

Linux kernel tunings:

To increase the connection backlog (hash tables) to a maximum allowed value, modify the parameters as follows:

```
net.ipv4.tcp_max_syn_backlog = 8192
net.core.netdev_max_backlog = 8192
```

Note: A common problem occurs when the above values are too low.

To increase the memory size available for TCP buffers, modify the parameters as follows:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
```

Note: Increasing the above parameters helps to increase the transfer rate and reduce CPU usage.

The following keep alive values are suggested:

```
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_probes = 8
net.ipv4.tcp_keepalive_time = 7200
```

Note: The application still needs to ask for keep alive values - they are not forced by the kernel.

The following is a timeout after improper close:

```
net.ivp4.tcp_fin_timeout = 60
```

Note: The above value should be low, not high.

The following is an RPC value:

```
sunrpc.tcp_alot_table_entries = 64
```

Additional NetWorker related tuning parameters

- Decrease **nsrmmd** polling interval (default: 2 minutes, recommended at least 15 minutes -- even more for large environments) since polling of unavailable **nsrmmd** (if S/Ns are down) is a blocking call.
- Decrease **nsrmmd** restart interval (default: 2 minutes, recommended 10 minutes) (if SN is down, blocking calls every 2 minutes is ineffectual) If these two **nsrmmd** values are not re-evaluated, then the server can freeze because it will get too busy polling (RPC errors, backups will hang, etc.)
- Change scheduling so no savegrps start at the identical time (splitting start time for different savegroups by only 5 minutes improves stability immensely and prevents file descriptor resource exhaustion and resource exhaustion on OS level). Prior to 7.3.x a single thread was invoked during the backup, now with multi-threading in 7.3.x, there can be up to six internal threads invoked at the start of a backup.

5 NetWorker Configuration

This chapter briefly describes the features and configurations available with the NetWorker software, including details on how the NetWorker product performs backups and recoveries, in the following sections:

NetWorker software services and programs

This section provides an overview of how the NetWorker products perform backup and recovery operations.

All NetWorker servers consist of services and programs that do the following:

- Coordinate the tasks associated with a backup and recovery.
- Record information about the backup.
- Track the media containing the backed-up data.
- Manage storage management client configurations, client file indexes, and the media database.

The NetWorker client software includes a client service and user interface programs. The services and programs in NetWorker products communicate by using the Remote Procedure Call (RPC) protocol.

Services and programs on the NetWorker server

This section describes the NetWorker master service, as well as other services and programs residing on the NetWorker server. These services and programs contact the client for a backup and maintain the client file indexes and media database on the server.

The nsrd service

The NetWorker master service, **nsrd**, provides an RPC-based **save** and **recover** program function to NetWorker clients. The service also provides these other functions.

- Starts other services.
- Allocates media services on server computers.
- Authorizes backup and recovery services for the client.
- Contacts clients for scheduled backups.
- Maintains NetWorker configuration information.
- Monitors backup and recovery sessions.
- Maintains server statistics and message logs.

Other services and programs

Table 10 lists the other services or programs that reside on the server.

Table 10 Services or programs on the server (1 of 2)

Service or program	Function
ansrd	This service monitors an active **recover** session. It is called by **nsrd** in response to a **recover** session.
gstd	This master service called Generic Services Toolkit (GST) controls other services provided by the Console server.
hagentd	This service manages the NetWorker host agent modules and maintains the NetWorker host agent resource database.

Table 10 Services or programs on the server (2 of 2)

Service or program	Function
jobsd	This service manages monitoring and other communication between the NetWorker server and the NetWorker console.
nsrd	This service monitors an active **save** or **recover** program session. This is an agent process spawned by **nsrd** in response to a **save** or **recover** session. It (**nsrd**) is also the master service that controls other services on the NetWorker server, clients, and storage nodes.
nsrck	This service checks the consistency of the client file index. It is invoked by **nsrd** whenever the consistency of the client file index needs to be confirmed.
nsrim	This service automatically manages the server media database. It is invoked by **nsrmmdbd** when it starts up at the end of the **savegrp** program. It removes both aborted and expired save sets once every 24 hours after a savegroup is completed (if Volume Recycle is set to Auto). It (**nsrim**) is also called by **nsrd** when a user removes the oldest backup cycle.
nsrindexd	This service provides a method for inserting entries into the client file index based on information passed by the **save** program.
nsrlogd	This service allows logging of security audit messages from any NetWorker machine. Supports a configurable port or a variable port range, depending on the transport selected.
nsrmmd	This service provides device support, generates mount requests, and multiplexes save set data during a multiclient backup. It writes the data sent by **save** to storage media. An additional **nsrmmd** service is started while mount requests are pending.The **nsrd** service starts one **nsrmmd** service for each enabled device. Forwards storage information to **nsrmmdbd** for recording in the NetWorker server media database.
nsrmmdbd	This service provides media database management services to the local **nsrd** and **nsrmmd** services and records entries in the media database. This is the media management database service.
nsrmmgd	This service manages media library operations. It provides an RPC-based service that manages all jukebox operations on behalf of the **nsrd** service. The **nsrd** service starts one and only one instance of **nsrmmgd** on the NetWorker server as needed.
nsrlcpd	This service provides **nsrmmgd** with a uniform interface to supported libraries. One **nsrlcpd** service is started by the **nsrmmgd** service for each virtual library instance. The **nsrlcpd** services run on the storage node controlling the library that the service manages.
nsrpm	This service works with the energy-saving features of Windows to keep NetWorker components running properly during power-saving modes. If it is not required, it can be disabled. **Note:** This service is for the Windows OS only.
savegrp	This program runs a group of NetWorker clients through the **save** process.

Services and programs on the NetWorker client

The **nsrd** service calls on the NetWorker client service, **nsrexecd**, and several programs on the client when a scheduled or on-demand backup request is received. A temporary server agent, the **ansrd** service, starts on the NetWorker server to monitor the progress of the backup session.

Table 11 describes the services or programs on the NetWorker client.

Table 11 Services or programs on the client

Service or program	Function
nsrexecd	This service authenticates and processes the NetWorker server remote execution requests and executes the **save** and **savefs** programs on the client.
recover	This program browses the NetWorker server client file index and restores the specified file to primary disk storage.
save	This program sends specified files in a multiplexed data stream to the NetWorker server for backup to media by **nsrmmd,** and entry into the client file indexes by **nsrindexd** and the media database by **nsrmmdbd**. A level-based system is used to save only the files modified since some previous save (a partial save).
savefs	This program gathers all the necessary information, such as file system and level, and sends it to **savegrp**. **savegrp** then requests that the **save** program send the data to the NetWorker server.

How the NetWorker software backs up data

When you configure a backup group on the NetWorker server, you schedule a start time for the backup group. The **nsrd** service starts the **savegrp** program on the server for the backup group at the scheduled time.

The savegrp program

The **savegrp** program queries the client resources configured on the NetWorker server to determine the following:

- Which clients configured on the server are members of the scheduled group.
- What level of backup (**save**) to perform.
- How many save sets to run concurrently, determined by the parallelism value set on the NetWorker server.
- When the most recent backup of the group occurred.

If any information is unavailable on the NetWorker server, the **savegrp** program sends a request (sometimes called a probe) to the **nsrexecd** client service to run **savefs** on each client assigned to the backup group to gather the necessary details.

The **savefs** program tells **savegrp** which objects to back up for the client. After **savegrp** receives information about the objects to back up, **savegrp** assembles a work list for the server. The work list specifies the order in which clients are contacted for backup. The order of the work list is determined by the client priority attribute in the client resource. The client with the lowest value in the client priority attribute is contacted first.

If problems were encountered with the client file index during the previous backup session, **nsrd** invokes the **nsrck** service to check the consistency and state of the NetWorker server client file indexes. Then, **nsrd** starts the **nsrindexd** client file index insertion service.

The **savegrp** program contacts the first client on the server work list. The **nsrexecd** of the client is invoked and starts a **save** session of the first save set listed on the server work list. The **save** program passes all save criteria to **nsrd**, such as group, client, save sets, and level of the save data. With this information, **nsrd** determines the pool of

volumes that will store the data and forwards the information to the appropriate media service on the NetWorker server.

When there is a lull in save set activity from the client, the NetWorker server attempts to find another save set in the group to keep the process moving. The **savegrp** program attempts to back up concurrently as many save sets as possible, up to the limit set by the Parallelism attribute in the NetWorker server configuration. This way, backup devices are utilized to their maximum potential.

The **savegrp** program repeats the process for each item on the server work list until all clients in the group are backed up. Before the **savegrp** program finishes, if the NetWorker server is either part of the group being backed up or not part of any enabled group, the NetWorker server bootstrap file is backed up.

If you have set up the bootstrap notification and installed and configured TCP/IP print services, a bootstrap printout is sent to the default printer configured for the NetWorker server after the bootstrap backup finishes.

Note: Keep the bootstrap printout in a safe place to restore the NetWorker server.

The final results of the **savegrp** execution are appended to the */nsr/logs/savegrp.log* file.

The nsrmmd service

The media service, **nsrmmd**, performs the following:

- Sends a message to the NetWorker server console, requesting a mount of the media assigned to the volume pool indicated by **nsrd**.
- Writes the data sent by **save** to storage media.
- Forwards storage information to **nsrmmdbd** for recording in the NetWorker server media database.

Figure 1 on page 65 shows how all of the NetWorker client and server services and programs interact during a scheduled save.

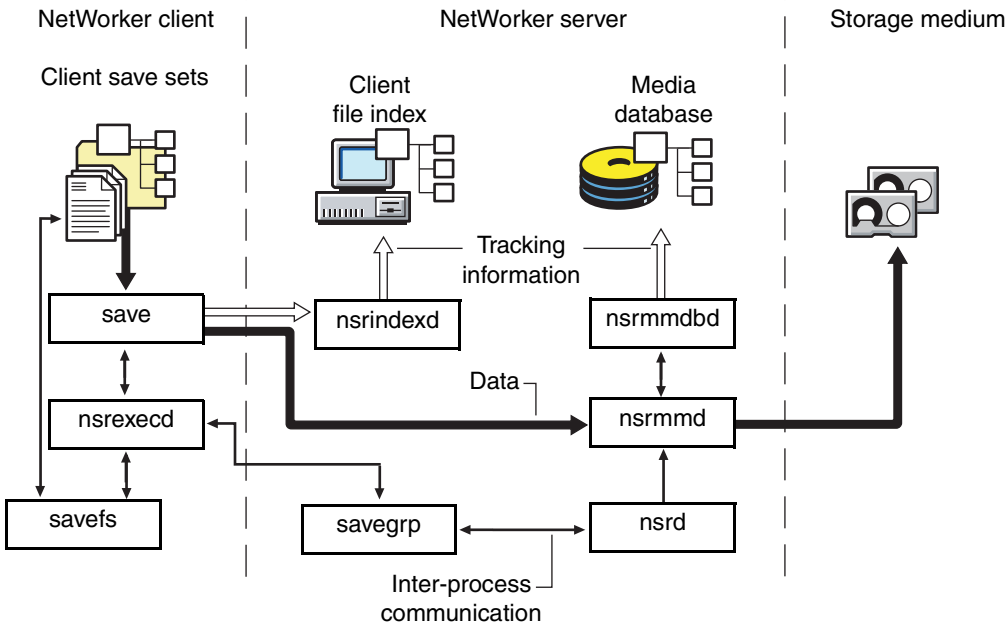

Figure 1 The NetWorker services and programs during a scheduled backup

How the NetWorker server recovers data

When the NetWorker server receives a **recover** request from a client, the **nsrd** master service of the server contacts the **nsrmmd** media service on the server. The **nsrmmd** service contacts the **nsrmmdbd** media database service of the server to determine which media contain the save set requested by the **recover** program. After **nsrmmd** finds the save set on the media, **nsrmmd** issues a mount request, the media is positioned to the beginning of the save set, and the save set stored on the mounted media is passed to **nsrmmd**. The media service forwards the save set to the **recover** program of the client, which restores the data to the client's file system.

Figure 2 on page 66 shows how the NetWorker server and client services and programs interact while recovering data to a NetWorker client.

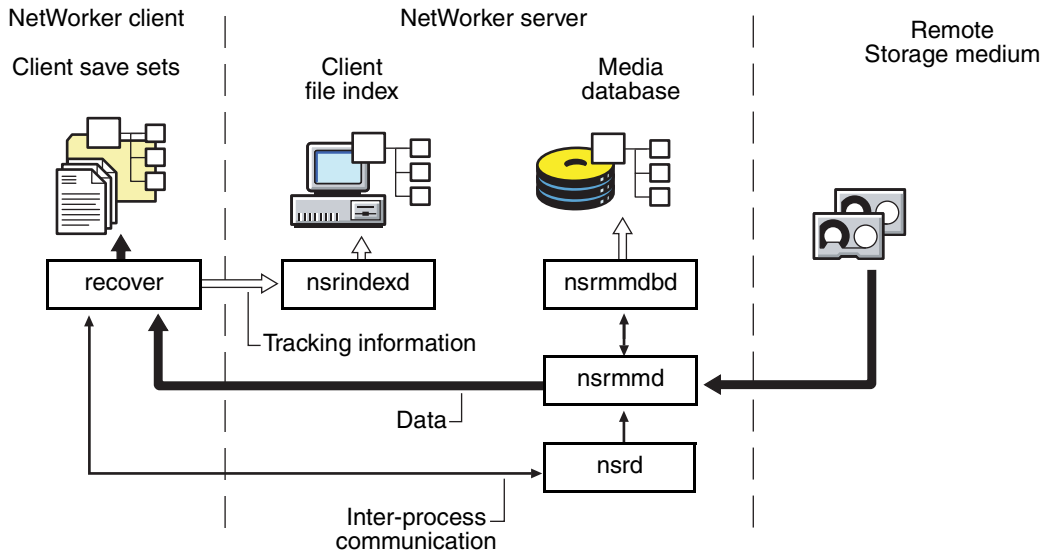

Figure 2 Processes and programs during a recover session

NetWorker configurations

This section describes various NetWorker configurations. Immediate save and recovery is a special feature available in all NetWorker editions of the NetWorker software (release 6.0 and higher). The text and figures provided here illustrate various setups, including:

Immediate save and recovery

During a backup session on a NetWorker server or storage node, the NetWorker software distinguishes whether the data resides on the same or remote computer.

- When the data resides on the same computer, immediate saves and recoveries are invoked automatically. Instead of transferring data across the network, local data is transferred within memory on the NetWorker server or storage node.
- When the data resides on a remote client, the data is transferred over the network by using TCP/IP.

Performance for remote backup is improved through RPC enhancements. The examples in the following sections illustrate when immediate save can, and cannot, be invoked.

Local backup and recovery of a client

When one computer includes either both the client and server software, or the storage node software, and they have storage devices attached, the data is backed up through immediate save. Figure 3 on page 67 shows an example of a local backup of a NetWorker server. The NetWorker software can back up data of any type supported by a NetWorker client, including a range of databases.

Figure 3 Local backup and recovery

Processes and data that travel through a protocol stack in a standard setup are passed in memory on the same computer. With immediate save, the network transmission blockage is eliminated. The same is true for recovery operations.

If a database is offline, that database can take advantage of immediate save for local backup with a NetWorker client. Some NetWorker module clients also can take advantage of immediate save for online database backups. Consult the documentation that accompanied the NetWorker module software to determine whether immediate save for online backup is supported.

Backup and recovery of remote client

Figure 4 on page 68 shows an example of the NetWorker server installed on a computer other than the one where the data resides.

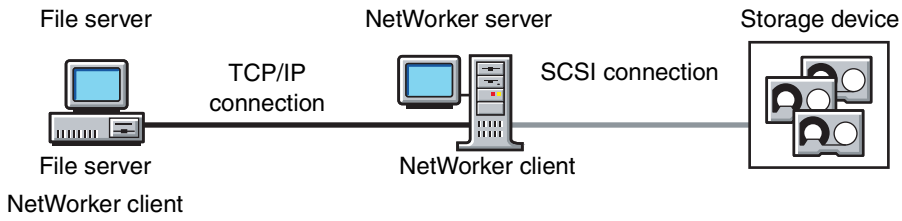

Figure 4 Remote backup and recovery

Because the NetWorker client and the fileserver data reside on a remote computer, the data is transferred across the network during backup.

For a recovery, the process is reversed. When the NetWorker client on the fileserver computer makes a recovery request, the data is transferred again across the network.

Backup and recovery performance over the network is enhanced because of Remote Procedure Call (RPC) protocol enhancements. They prevent slower clients from slowing down the backup of faster clients. Data that resides on the NetWorker server is automatically backed up through immediate save (or automatically recovered by using immediate recover).

If the data is backed up to a high-speed device, performance is improved. The device can transfer data in less time and write it in larger blocks.

Backup and recovery to a storage node

Figure 5 on page 69 shows an example of a NetWorker server that uses a storage node for backup and recovery operations. This feature is only available with the Power Edition or Network Edition of the NetWorker software.

Data hosted on each NetWorker storage node is backed up and recovered using immediate save and recover technology. The metadata associated with backups is transferred over the network to the controlling NetWorker server, and the data is transferred directly to devices attached to the storage node.

NetWorker storage nodes enable you to design scalable configurations:

- Multiple devices can be attached to multiple storage nodes.
- Data transfer operations can run concurrently.

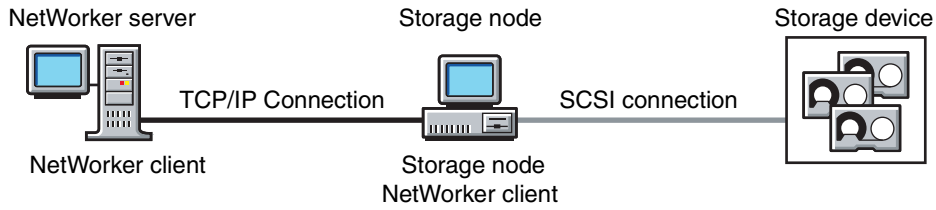

Figure 5 Backup and recovery with a NetWorker storage node

Local backup to a silo

When you add a silo to the local backup scenario, performance may improve because of the speed and number of devices in the silo. The silo requires another computer to be the silo server, which controls the robotics in the silo. The silo server cannot be the same computer as the NetWorker server.

In Figure 6 on page 69, a silo and its silo server are used in the NetWorker local backup and recovery configuration.

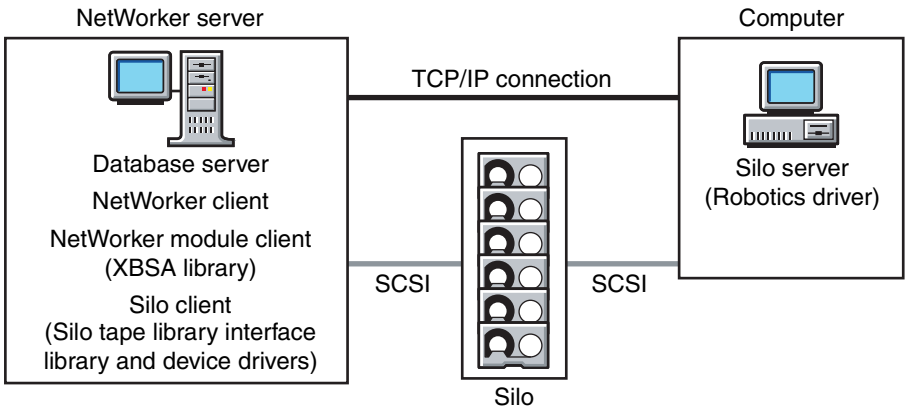

Figure 6 Local backup to a silo

During backup

During backup, save is invoked to transfer data from the NetWorker client processes to the NetWorker server processes in the memory of the NetWorker server computer. If the data is written to a device in a silo, the **nsrmmd** service initiates a Silo Tape Library Interface (STLI) call to the silo server, for example, to mount a tape.

The STLI call is transferred across the network connection to the silo server, then over a SCSI connection to the device in the silo. The data is transferred over a SCSI connection from the NetWorker server to the device.

During recovery

During a recovery, the process is similar to a backup. Media handling information is transferred across the network connection from the NetWorker server to the silo. The silo mounts the tapes, and then the data is transferred over the SCSI connection to the NetWorker server computer, where the data is transferred in memory to the NetWorker client process. The *EMC NetWorker Release 7.4 Multiplatform Version Administration Guide* provides more information about silos and how they interact with the NetWorker software.

A

Troubleshooting

This appendix contains troubleshooting information that can help to assist in fine tuning your NetWorker operation. The following information is covered:

RPC error messages

When trying to connect to the NetWorker server, you may encounter these error messages:

```
Error: 'save: RPC error: Remote system error'
Error: 'save: Cannot open save session'
```

Problem

When under stress, the current RPC implementation can handle only a small burst of connection requests. The server services only one connection request each time it goes through the select() loop. When the server is busy serving other connections, it takes more time to do one select() loop, and the server services fewer connection requests per second.

For example, there are 10 clients calling the service, and each client takes 10 ms to service. It takes 200 ms to go around the select loop, and it can only service 10 connection requests per second.

In some customer environments, over 400 clients could attempt to connect to the same server during a short time. If the server is under stress and can only handle 10 connection requests per second, it cannot handle the burst of 400 connection requests. The server can only queue up twenty requests. The connection requests that are not queued up may attempt to retry, or they may fail with multiple RPC errors.

Solution

To avoid this problem, increase the SOMAXCONN parameter, a kernel parameter that governs the maximum number of unaccepted socket connections that can be waiting in queue. It is also referred to as the listen() queue limit. When NetWorker (NSRD) starts, it requests a listen() queue of 200. Ensure that the value for SOMAXCONN is set to 200 or higher. This parameter must be set before NSRD starts.

These are examples of setting the SOMAXCONN parameter on various platforms:

- For Solaris, enter this command:

  `/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q 1024`

  where the **q** queue holds sockets awaiting an accept() call.

  `/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q0 2048`

  where the **q0** queue contains half-open sockets.

- For Tru64 UNIX:

  a. Enter this command:

  `/sbin/sysconfig -r socket sominconn=65535`

  The value of **sominconn** determines how many simultaneous incoming SYN packets the system can handle.

  b. Enter this command:

  `/sbin/sysconfig -r socket somaxconn=65535`

  The value of **somaxconn** sets the maximum number of pending TCP connections.

- For HP-UX, enter these commands:

```
/usr/sbin/ndd -set /dev/tcp tcp_syn_rcvd_max 1024
/usr/sbin/ndd -set /dev/tcp tcp_conn_request_max 200
```

- For Linux, enter this command to increase the size of the socket queue (effectively, q0):

```
/sbin/sysctl -w net.ipv4.tcp_max_syn_backlog=1280
```

Troubleshooting target sessions

The target sessions attribute sets the number of backup sessions accepted by an active device. The target sessions attribute is configured in the device resource, and a different value can be specified for each device.

If the target session value is set too high, the performance of the backup does not always improve. For example, suppose the parallelism attribute is set to 16, which means the NetWorker server can handle up to 16 streams of data from the NetWorker clients. If the target session attribute for each device is set to 16, then each device also can handle 16 streams. In this case, the 16 data streams from the clients may be sent to the same device, since each device can handle 16 streams. No benefit is gained by using the target sessions attribute.

To optimize device usage, set the target session attribute to less than the parallelism attribute so more than one device is selected for multiple backup streams.

A

Stability settings

This appendix contains information that can help to assist in stabilizing NetWorker The following information is covered:

Tuning to increase NetWorker stability

The following information concerning changes in default values can be used to add stability to NetWorker release 7.3 and later.

- Increase **client parallelism** of the server client (default: 4, recommended minimum: 12). The recommended value is due to a change in the behavior of NetWorker 7.3 and later compared to older releases.
- Increase **server parallelism** to the maximum allowed for the NetWorker Edition.
- Decrease the **nsrmmd** polling interval (default: 2 minutes, recommended: 15 minutes (at a minimum, even more for large environments)) since polling of unavailable **nsrmmd** (if S/Ns are down) is a blocking call.
- Decrease the **nsrmmd** restart interval (default: 2 minutes, recommended 10 minutes) (if SN is down, what's the point of wasting time in blocking calls every 2 minutes). If these two **nsrmmd** values are not fine tuned, then the server can freeze because it will get too busy polling (RPC errors, backups will hang, etc.).
- Implement **savegrp parallelism** (default: 0, which is unlimited) so no more than cca. Fifty clients (depends on the OS and HW) start at the same time as file descriptor resources can quickly become exhausted.
- Change scheduling so no savegroups start at the identical time (splitting start time for different savegroups by only 5 minutes increases stability immensely to prevent file descriptor resource exhaustion and resource exhaustion on OS level). Prior to release 7.3 a single thread was invoked during the backup. Now with multithreading in NetWorker 7.3 and later, there can be up to six internal threads invoked at the start of a backup.

Note: Solaris 8 is not able to handle multiple threads effectively.

- Increase maximum size of **jobsdb** (default: 20mb, recommended: 60mb) (Ck. with Debbie--need to provide instructions on how to do this.)
- Decrease retention of **jobsdb** (default 7 days, decrease to 4 days. Those two changes combined make sure that purging by time is used instead of purging by size which is much slower and can cause other problems (since **nsrjobd** slows down a lot during a purge period). Purging by size avoids forcing NetWorker to check the size of the database continuously.

Note: Size can be set even higher to ensure that it never becomes a factor.

Glossary

This glossary contains terms and definitions related to this manual. Most of these terms are specific to NetWorker products.

1-9 Intermediate backup levels. Each number represents a backup level. Lower levels back up more files.

A

Administrators group Windows NT user group whose members have all the rights and abilities of users in other groups, plus the ability to create and manage all the users and groups in the domain. Only members of the Administrators group can modify Windows NT OS files, maintain the built-in groups, and grant additional rights to groups.

annotation A comment that you associate with an archive save set, to identify that data later. Annotations are stored in the media index for ease of searching and are limited to 1,024 characters.

application-specific module (ASM) A program that, when used in a directive, specifies the way a set of files or directories is backed up and recovered.

archive The process by which NetWorker backs up directories or files to an archive volume and then optionally deletes them to free up disk space.

archive clone pool A volume pool composed exclusively of archive clone save sets.

archive pool A volume pool composed exclusively of archive save sets.

archive volume A tape or other storage medium used for NetWorker archives, as opposed to a backup volume.

ASM *See* application-specific module (ASM).

autochanger A mechanism that uses a robotic arm to move media among various components located in a device including slots, media drives, media access ports, and transports. Autochangers automate media loading and mounting functions during backup and recovery.

B

Backup Operators group A Windows NT group whose members can log in to a domain from a workstation or a server, back it up, and restore the data. Backup Operators also can shut down servers or workstations.

backup volume Backup media, such as magnetic tape or optical disk.

bootstrap Information that includes the server index, media index, and configuration files needed for recovering NetWorker after a disk crash.

browse policy The policy that determines how long entries for the files remain in the online file index.

C

carousel A tray or tape cartridge that holds multiple backup volumes.

client A computer that accesses the NetWorker server to back up or recover files. Clients may be workstations, PCs, or file servers.

clone The process by which NetWorker makes an exact copy of saved data (save sets). NetWorker can clone individual save sets or the entire contents of a backup volume.

clone volume A duplicated volume. NetWorker can track four types of volumes: backup, archive, backup clone, and archive clone. Save sets of different types may not intermix on one volume.

command line The shell prompt, where you enter commands.

compressasm A NetWorker directive used for compressing and decompressing files.

D

device The backup device (tape drive, optical drive, or autochanger) connected to the NetWorker server; used for backing up and recovering client files.

directive An instruction directing NetWorker to take special actions on a given set of files.

E

enabler codes Special codes provided by LEGATO that allow you to run the NetWorker software product.

F

file index A database of information maintained by NetWorker that tracks every file or file system backed up.

file server A computer with disks that provides services to other computers on the network.

file system 1. A file tree on a specific disk partition or other mount point. 2. The entire set of all files. 3. A method of storing files.

full (f) A backup level in which all files are backed up, regardless of when they last changed.

G

grooming The process of removing files after a successful archive.

group A client or group of clients that starts backing up its files at a designated time.

H

heterogeneous networks Networks with systems of different platforms that interact meaningfully across the network.

himt A choice in the devices window; represents half-inch magnetic tape.

I

incremental (i) A backup level in which only files that changed since the last backup are backed up.

interoperability The ability of software and hardware on multiple computers from multiple vendors to communicate meaningfully.

L

level (1-9) A backup level that backs up files changed since the last backup of any lower level.

M

media Magnetic tape or optical disks used to back up files.

media index A database of information maintained by NetWorker that tracks every backup volume.

media manager The NetWorker component that tracks save sets to backup volumes.

N

NetWorker An LEGATO network-based software product to back up and recover file systems.

NetWorker client A computer that can access the backup and recover services from a NetWorker server.

NetWorker server The computer on a network running the NetWorker software, containing the online indexes and providing backup and recover services to the clients on the same network.

notice A response to a NetWorker event.

nsrhost The logical *hostname* of the computer that is the NetWorker server.

O

online indexes The databases located on the server that contain all the information pertaining to the client backups and backup volumes.

operator The person who monitors the server status, loads backup volumes into the server devices, and otherwise executes day-to-day tasks using NetWorker.

override A backup level that takes place instead of the scheduled one.

P

pathname Instructions for accessing a file. An *absolute pathname* tells you how to find a file beginning at the root directory and working down the directory tree. A *relative pathname* tells you how to find the file starting where you are now.

preconfigured Existing selections or configurations for different NetWorker features.

Q

qic A choice in the devices window; represents quarter-inch cartridge tape.

R

recover The NetWorker command used to browse the server index and to recover files from a backup volume to a clients disk.

recycle A volume whose data has passed both its browse and retention policies and is available for relabeling.

Registry A database of configuration information central to Windows NT operations. The overall effect centralizes all Windows NT settings and provides security and control over system, security, and user account settings.

retention policy A NetWorker policy that determines how long entries will be retained in the media database and thus be recoverable.

retrieval The process of locating and copying back files and directories that NetWorker archived.

S

save The NetWorker command that backs up client files to backup volumes and makes data entries in the online index.

save set A set of files or a file system backed up onto backup media using NetWorker.

save set ID An internal identification number assigned to a save set by NetWorker.

scanner The NetWorker command used to read a backup volume when the online indexes are unavailable.

server The computer on a network running the NetWorker software, containing the online indexes and providing backup and recover services to the clients on a network.

shell prompt A cue for input in a shell window where a command is entered.

skip (s) A backup level in which files are skipped and not backed up.

stand-alone device A backup device that contains a single drive for backing up data. Stand-alone devices cannot store or automatically load backup volumes.

system administrator The person normally responsible for installing, configuring, and maintaining NetWorker.

U

user A person who can use NetWorker from a workstation to back up and recover files.

V

volume Backup media, such as magnetic tape or optical disk.

volume ID The internal identification NetWorker assigns to a backup volume.

volume name The name assigned to a backup volume when it is labeled.

volume pool A feature that allows you to sort backup data to selected volumes. A volume pool contains a collection of backup volumes to which specific data is backed up.

Index

S

T

U

V

W