

VERITAS NETBACKUP™ PERFORMANCE TUNING

George Winter
October 15, 2003

TABLE OF CONTENTS

Introduction	3
Formal Benchmarks of VERITAS NetBackup Performance.....	3
Understanding and Testing the Backup Infrastructure.....	4
Tape Backup Recommendations	4
Network Recommendations.....	5
A Network Bottleneck Case Study.....	6
Storage Recommendations	7
Tuning NetBackup.....	8
Capacity Planning.....	8
Sizing the NetBackup Media Server.....	8
Memory usage	8
Configuring Multiplexing and Multi-Streaming.....	9
Compression.....	10
Buffer Management.....	10
NET_BUFFER_SZ	10
SIZE_DATA_BUFFERS.....	11
NUMBER_DATA_BUFFERS.....	11
NUMBER_DATA_BUFFERS_RESTORE	11
Buffers and System Memory.....	11
Digital Linear Tape and the SIZE_DATA_BUFFERS.....	11
Restoring Data From UNIX to Windows.....	12
BPTM Log Contents	12
Improving Restore Performance.....	13
Disk-Based Storage	13
Indexing the NetBackup Image Catalog.....	14
Thousands of Small Backups.....	15
Determining Read and Write Performance of Backups and Restores.....	15
Log Files.....	16
Java User Interface Performance	17
Summary.....	17
Appendix.....	17
Operating System Settings	17
Minimum Solaris Kernel Parameters	17
TCP_CLOSE_WAIT_INTERVAL	19
Recommended Network Interface Settings.....	20
Additional Web Resources.....	21

Introduction

If you are an IT administrator, there is a good chance that improving backup performance ranks high on your list of priorities. Backup performance, after all, is one of the most common topics in calls to VERITAS technical support. But getting to the bottom of performance issues can be tricky. There are several interrelated hardware, software, and networking components in the backup environment, and each component can create bottlenecks that obstruct the smooth flow of data. Understanding the path that data takes – from the storage subsystem to the backup medium – is essential before determining what you can achieve by performance tuning.

This article presents guidelines for tuning VERITAS NetBackup™ performance, recommendations for testing the backup infrastructure, and information on formal benchmarks of NetBackup performance.

Formal Benchmarks of VERITAS NetBackup Performance

Classifying the performance of a backup-system as poor is often the result of unrealistic expectations. It is frequently difficult to assess performance realistically, so formal benchmarks can help shed light on this dilemma.

Although the real world has a multitude of variables that cannot be reflected in a formal benchmark study, these tests are valuable for assessing the comparative performance of different products and demonstrating a high-water mark for backup performance. The values achieved in a benchmark help you determine how well VERITAS NetBackup is performing in your environment. This is not to say that your environment will have the ability to achieve the same backup throughput; but if your performance assessment is significantly different from the benchmark results, it could indicate a performance bottleneck in the system.

The Storage Networking Industry Association (SNIA) benchmark for backup processing uses NDMP and a standard file system – without databases – to measure a backup product's performance. VERITAS NetBackup delivered backup performance of 1 TB/hour in the SNIA benchmark.

In the Oracle database backup benchmark study – previously reported in eWeek magazine – VERITAS NetBackup performance was tested while backing up an Oracle 9i database. Using NetBackup 4.5 and performing a hot backup, VERITAS achieved a rate of 2.33 TB/hour.

Understanding and Testing the Backup Infrastructure

Every backup environment has a bottleneck. Data may move around the system very fast indeed, but somewhere in the system there is a gating element that prevents the system from running even faster. In order to characterize what a configuration is capable of, you must break down the infrastructure into its component parts. This enables you to determine the limits of each element and assess the theoretical limits of the configuration.

The backup environment (see Figure 1) has three main components – the storage subsystem, the network, and the backup medium (which is usually a magnetic tape).

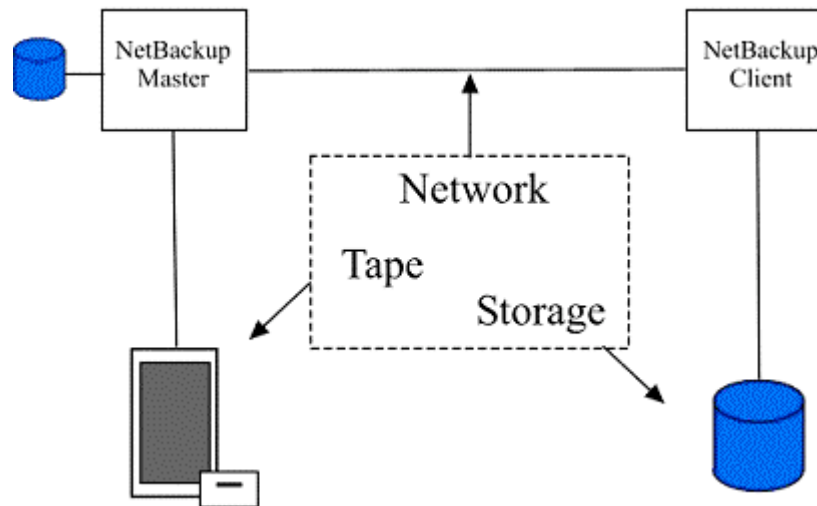


Figure 1. The Backup Infrastructure

The following sections include information on testing the backup infrastructure before tuning the performance of VERITAS NetBackup. This can help you gain an accurate assessment of the environment independent of NetBackup.

Tape Backup Recommendations

Determining the rated throughput of a tape drive is relatively easy. Many tape drive manufacturers have utilities that measure throughput. If you have no access to a utility, a simple test using “dd” or “tar” with null input achieves the same result. On UNIX systems, “iostat” and SAR also provide tape performance statistics. On Windows systems, the NetBackup performance counters can be analyzed.

As a rule of thumb, it is not advisable not to saturate a tape drive connection. There should be no more than two drives per SCSI/FC bus. Also, bear in mind that the host bus adapter (HBA) is an integral part of the hardware connection. If an HBA device is rated at 100MB/sec, then assume an optimal loading of 70MB/sec using the 70 percent rule.

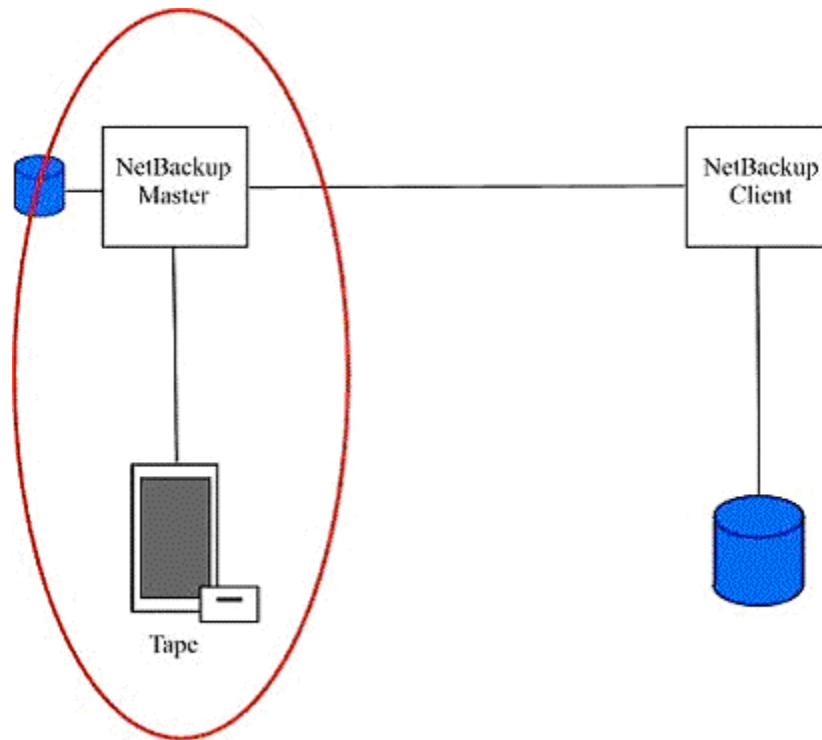


Figure 2. The Tape Configuration

Network Recommendations

The network infrastructure offers many different components that can affect backup throughput. These components include slow DNS servers, incorrect or outdated network interface (NIC) drivers, unplanned network traffic, and over-taxed CPU resources on either side of the network connection.

Iperf – which is a free utility for Windows, UNIX, and Linux that tests TCP and UDP bandwidth – is useful for testing a network connection. The Iperf utility isolates the network, removing tape and disk from the equation, and tests network throughput in either direction. It is a good idea to run tests at different times of day to gain a complete picture of network usage and throughput. If Iperf is not available, FTP can also be used to study performance.

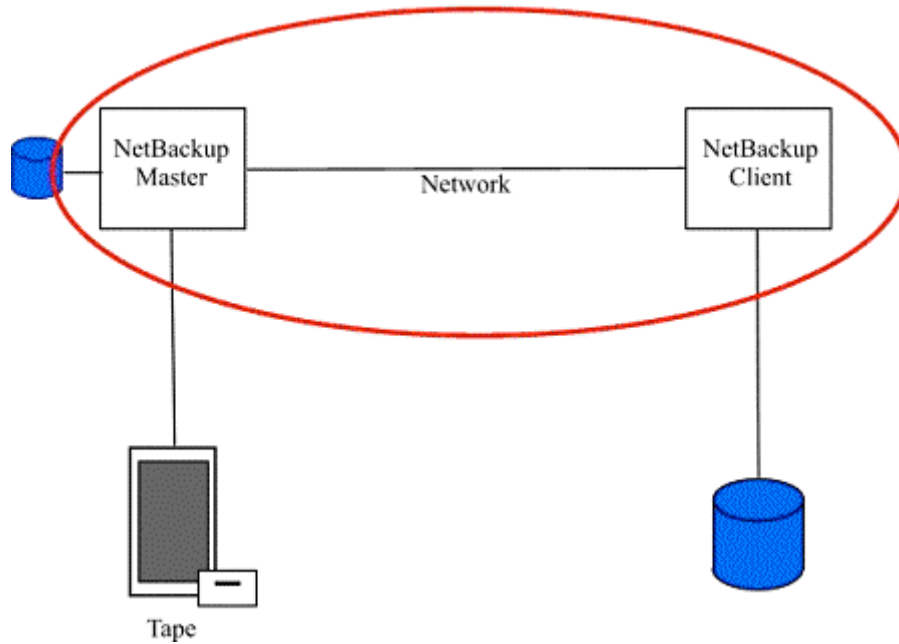


Figure 3. The Network Configuration

A Network Bottleneck Case Study

A VERITAS customer experiencing slow and inconsistent network throughput approached the technical support desk with a problem. While running backup processing, it appeared that Windows servers performed better than Solaris, and some end users experienced no problems at all.

Using Iperf, to test a 100 MB segment of the network, the customer discovered a switch with an out-of-date NIC driver. The switch was also exhibiting problems negotiating half/full duplex.

After upgrading the NIC driver and resolving the duplex negotiation problem, all Iperf tests were rerun. The result showed a dramatic improvement in network throughput.

Iperf output on 100Mb network segment

{ID}	Interval	Transfer	Bandwidth
{ 4 }	0.0-11.1 sec	0.15 MBytes	0.11 Mbits/sec

Iperf output on same segment after upgrading NIC driver and resolving Half/Full duplex problem

{ID}	Interval	Transfer	Bandwidth
{ 4 }	0.0-10.1 sec	99.5 MBytes	83.5 Mbits/sec

Figure 4. Slow Network Case Study

Storage Recommendations

The storage subsystem presents more software, hardware, and network components that potentially interfere with backup performance. Disk systems are becoming increasingly prevalent as intermediary storage locations in backup processing, and as permanent augmentations to a tape storage system. Although SCSI disk is still regarded as the fastest, ATA and IDE protocols are also popular, and many ATA drives are being designed as tape drive replacements. Understanding the differences between the various drive protocols is important. Also, remember that storage exists in many locations within the backup infrastructure – not only at the application data store. The NetBackup client contains disk storage, and the NetBackup server hosts the catalogs that are used during backup and restore operations.

It is essential to understand the capabilities of each component of the storage infrastructure before assessing the optimum throughput. This includes the specification for each HBA and network connection – for example, whether it is SCSI or Fibre Channel. Again, free utilities are available to measure storage subsystem performance (bonnie; Bonnie++; and tiobench), and you should use the 70 percent rule to determine realistic throughput expectations.

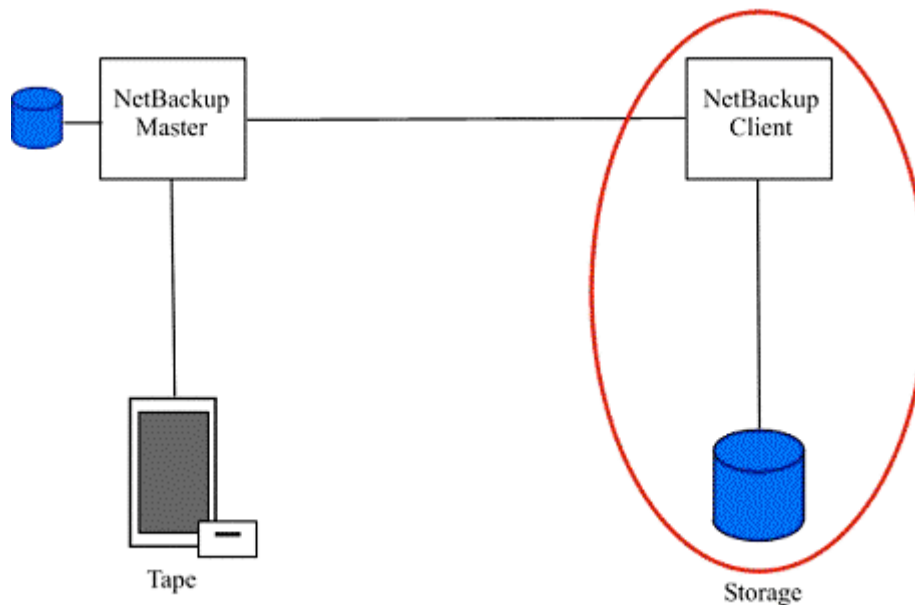


Figure 5. The Storage Subsystem Configuration

Tuning NetBackup

Testing the backup infrastructure gives an accurate assessment of the environment independent of NetBackup. The next step is to understand how NetBackup relates to this infrastructure.

Capacity Planning

Before making any attempt to address the NetBackup environment, it is necessary to understand the data being backed up. To determine how to configure NetBackup appropriately, you must answer the following questions:

- Is the data being backed up to a local or remote location?
- What are the characteristics of the storage subsystem and data path?
- How busy is the storage subsystem?
- How much data is being backed up?
- How much of the data being backed up will change on a regular basis?
- What type of data is being backed up – text, graphics, or databases?
- How many files are being backed up, and how compressible are the files?

Sizing the NetBackup Media Server

When sizing the NetBackup server, bear in mind that for this platform I/O performance is generally more important than CPU performance. Also, when choosing a server consider the potential for future CPU, memory, and I/O expansion.

The NetBackup Media Server requires 5 MHz of CPU capacity per 1 MB/sec of data movement. Gigabit Ethernet configurations should add 1 CPU per GbE card unless the cards contain their own CPU. This is a conservative estimate and mileage will vary depending on the server hardware and OS. Note that this calculation includes movement of data both in and out of the Media Server. For example, a system to back up clients over a network to a local tape drive at the rate of 10 MB/sec would need 100 MHz of available CPU power. This equates to 50 MHz to move data from the network to NetBackup Media Server, and 50 MHz to move data from the NetBackup server to tape.

Benchmark studies have found that the ratio of MHz to MB/sec increases as the number of CPUs decreases, and the ratio increases as the load on the CPU approaches 100 percent. When sizing the NetBackup Media Server, also bear in mind that the server OS and other server applications will be consuming CPU resources.

Memory usage

Memory is relatively cheap and, as far as the NetBackup server is concerned, more is always better. At least 512 MB is recommended (Java GUI). NetBackup uses shared memory for local backups. NetBackup buffer usage will affect this (see [Buffer Management](#)). Don't forget that other applications are also running.

Configuring Multiplexing and Multi-Streaming

Tape drives operate at their best when they receive a sustained stream of data that matches the throughput capacity of the drive. Continuous start-and-stop processing (the result of writing several small files) is inefficient, adds significant wear and tear to the drive mechanism, and reduces the lifespan of the hardware. Keeping tape drives streaming, however, can be a challenge.

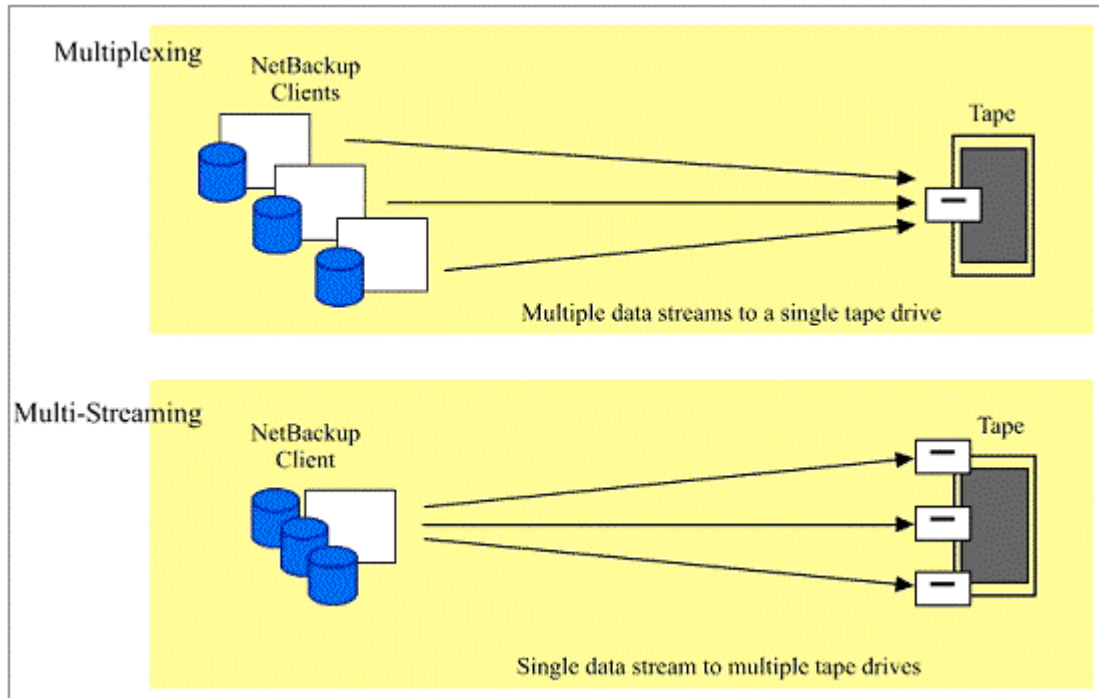


Figure 6. Multiplexing and Multi-Streaming to Tape

Multiplexing and multi-streaming provide powerful techniques for generating data streams to maximize the efficient use of tape and minimize wear and tear on tape drives. NetBackup supports concurrent use of both multiplexing and multi-streaming to tape.

Multiplexing allows multiple streams of data, from one or more NetBackup clients, to be simultaneously written to a single tape drive. This can help generate sufficient data to maintain streaming at the tape drive. Restoring data from a multiplexed tape requires all contents of tape to be read, both applicable and non-applicable, and this will slow the restore process.

Multi-streaming enables a single NetBackup client to send multiple data streams to one or more tape drives. Disk access is much faster than tape, and multi-streaming can be used to efficiently spread the backup load across many tape drives to speed the backup process. Each physical disk configured for multi-streaming must be accessed by no more than one streaming process at a time. If a single disk is the target of multiple streams, it will cause “disk thrashing” and significantly reduce the performance of the read process.

The optimal values for multiplexing and multi-streaming are achieved when tape drives are able to sustain streaming. The `NEW_STREAM` directive is useful for fine-tuning streams to ensure no disk subsystem is under- or over-utilized. Incremental backups often benefit from multiplexing, and staging the incrementals to disk before starting the multiplexed backup process is worth considering. Local backups typically benefit less from multiplexing and can be configured with lower values.

Compression

Compressing the backup data stream can reduce the amount of data sent over the network and minimize the size of the backup on tape. If a NetBackup client configuration suffers from a slow network connection, hardware or software compression can speed the backup process. If backup media usage is the primary concern, compression at the tape drive will reduce the amount of data being offloaded. Whether using tape compression or client-side hardware or software compression, there are some limitations to be aware of.

Compression at the client can be configured in both hardware and software but must never be activated for both at the same time. The algorithms used by the compression processes can conflict, resulting in a data stream larger than the original file. This will have the opposite effect on performance than what you desired. The same result can occur if both client-side and tape-side compression are activated.

Client-side compression creates additional processing overhead at the client. The overhead of compressing data increases the elapsed time of backup and restore processing, slowing the transfer rate between the client and the NetBackup Media Server. If the transfer rate is insufficient to maintain tape streaming, compression will result in additional wear and tear on the tape drives, although multiplexing multiple clients to a single tape drive can offset the problem.

Compression at the tape drive offloads processing overhead from client and server and is always preferable if the objective is to reduce the amount of data written to tape. If client-side software compression is used, the `bp.conf` `MEGABYTES_OF_MEMORY` entry at the client may help performance. The `bp.conf` entry `COMPRESS_SUFFIX` indicates files that the compression routines must bypass.

Buffer Management

NetBackup uses shared memory to buffer data between the network, or disk, and tape drive. Four of the NetBackup buffer management parameters can be modified: `NET_BUFFER_SZ`, the communications buffer setting; `SIZE_DATA_BUFFERS`, the size of each buffer; `NUMBER_DATA_BUFFERS`, the number of buffers; and `NUMBER_DATA_BUFFERS_RESTORE`, the number of restore buffers.

Default values for the `SIZE_DATA_BUFFERS`, `NUMBER_DATA_BUFFERS`, and `NUMBER_DATA_BUFFERS_RESTORE` parameters are changed by creating files in the NetBackup directory `/usr/open/netbackup/db/config`. Each file is named after the parameter it modifies and contains a single value that represents the new buffer setting.

NET_BUFFER_SZ

Changes to the `NET_BUFFER_SZ` parameter adjust the TCP/IP socket buffer size, which is used in data transfers between the NetBackup Media Server and UNIX clients. `NET_BUFFER_SZ` is modified by creating a file named `NET_BUFFER_SZ` in the `/usr/open/netbackup` directory on the NetBackup Master, NetBackup Media Server, and NetBackup client. These files contain identical values, representing the new communications buffer size, overriding the NetBackup UNIX default of 32,032.

On Windows systems, the optimal `NET_BUFFER_SZ` setting is 132,096, representing twice the network buffer size plus 1 KB (double the buffer size plus the TCP socket buffer). This value is modified through the Windows Client tab of the Client Properties menu.

SIZE_DATA_BUFFERS

The `SIZE_DATA_BUFFERS` parameter indicates how much shared memory NetBackup uses to buffer data between a disk and tape. The default values are 32 KB for non-multiplexed UNIX, 64 KB for multiplexed UNIX, and 64 KB, for Windows, regardless of multiplexing. To override the default setting, a value must be entered in the `SIZE_DATA_BUFFERS` file. This value must be a power of 2 and entered as an exact number of bytes - for example, 32 KB would be entered as 32768 (32 multiplied by 1024).

Changes to buffer values do not require NetBackup processes to be bounced. Modifying a buffer value is sufficient to cause the next backup run to use the new values. Restore processes use the same buffer values that were set at the time the backup was taken.

The I/O size of the backup tape is derived from the `SIZE_DATA_BUFFERS` value. This buffer setting may be restricted by the maximum allowable value at the tape device or file system. Network configuration parameters must also be taken into account when adjusting buffer size. The TCP/IP maximum transmission unit (MTU) value – for the LAN segment between the NetBackup client and Media Server – may require changing to accommodate adjustments to the `SIZE_DATA_BUFFERS` value.

NUMBER_DATA_BUFFERS

The default `NUMBER_DATA_BUFFERS` setting is 8 for non-multiplexed environments, 4 for multiplexed environments, and 8 for non-multiplexed restore, verify, import, and duplicate. These values are doubled for Windows NT environments.

NUMBER_DATA_BUFFERS_RESTORE

`NUMBER_DATA_BUFFERS_RESTORE` is a NetBackup 4.5 parameter, and is only applicable to multiplexed restores. The default value is 8 for a non-multiplexed image, and 12 for a multiplexed image. This buffer parameter can be helpful when tuning the restore of multiple concurrent databases.

Buffers and System Memory

Changing the size and number of NetBackup data buffers impacts available shared memory, which is a limited system resource. The total amount of shared memory used for each tape drive can be calculated using the following formula.

$$\text{Shared memory} = (\text{SIZE_DATA_BUFFERS} * \text{NUMBER_DATA_BUFFERS}) * \text{drives} * \text{MPX}$$

It is important to monitor the impact of modifications to the buffer parameters carefully. Not all changes have a positive effect. Although rare, changes to the buffer size have been known to slow backup processing and create problems during restores. Any change to buffer parameters must be thoroughly tested by performing both backup and restore processes.

Digital Linear Tape and the SIZE_DATA_BUFFERS

For Digital Linear Tape (DLT) drives, a `SIZE_DATA_BUFFERS` value of 65,536 has been observed to provide the best performance. Beyond this, performance improvements are gained from increasing `NUMBER_DATA_BUFFERS`.

Restoring Data From UNIX to Windows

Windows tape devices are unable to write block sizes larger than 64 KB. If backups are performed on a UNIX Media Server with a `SIZE_DATA_BUFFERS` value greater than 64 KB, they will be rejected by a Windows NT NetBackup Media Server. If importing or restoring images from a UNIX Media Server to Windows NT Media Server, a `SIZE_DATA_BUFFERS` value less than or equal to 64 KB must be used.

BPTM Log Contents

Analyzing data from the BPTM log indicates whether buffer settings must be modified. The wait and delay counters in the log help to determine where changes are needed and the impact the change has made. Figure 7 shows the BPTM entries that apply to buffer settings.

```
12:02:55 [28551] <2> io_init: using 32768 data buffer size
12:02:55 [28551] <2> io_init: CINDEX 0, sched bytes for monitoring = 2000
12:02:55 [28551] <2> io_init: using 8 data buffers
```

Figure 7. NetBackup BPTM Log Entries for Buffer Parameters

UNIX memory use is monitored using the `vmstat` utility. The `vmstat` scan rate indicates the amount of swapping activity taking place. SAR also provides insight into UNIX memory use. On Windows systems, the Windows Performance Monitor utility can be used to track memory utilization.

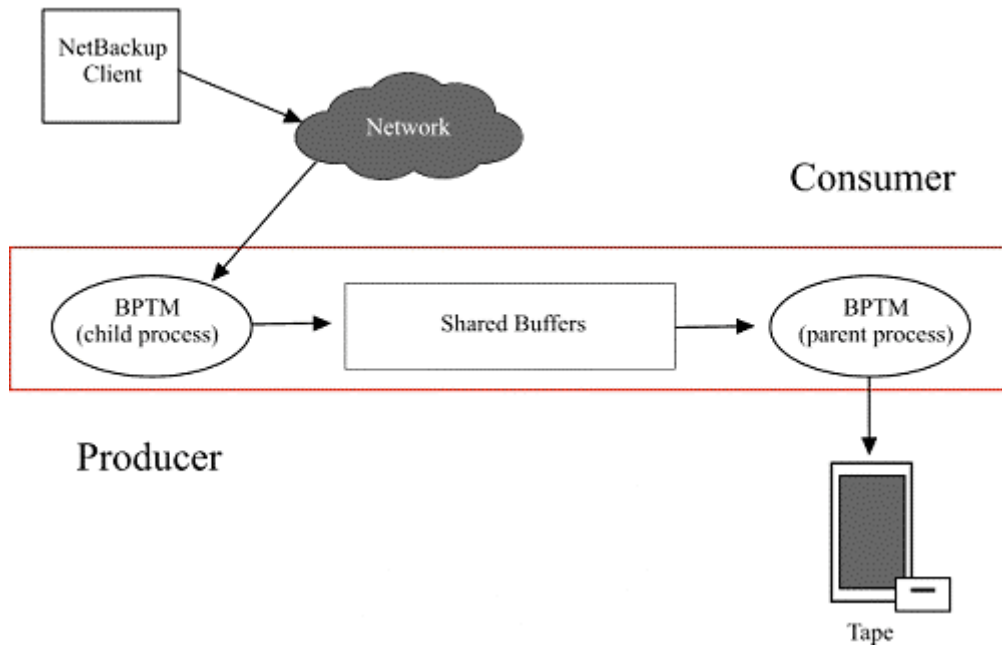


Figure 8. NetBackup Media Server Processes – Producer-Consumer Relationship

The goal in managing NetBackup buffers is to achieve a balance between data producer and data consumer (see Figure 8). The data producer (this is the NetBackup client during a backup process and NetBackup Media Server during restore) must have an empty buffer to write to. If it is not available, it will wait and cause a delay. The data consumer (this is the NetBackup Media Server during a backup and the NetBackup Client when restoring) must have a full data buffer to read from.

Entries in the BPTM log show the number of times BPTM waited and the exact number of times this caused a delay. A high wait for empty buffers indicates that the data producer (the BPTM child process) is receiving data from the source faster than it is being processed by the data consumer (the BPTM parent process). If this is the result of a multiplexed backup, reducing the number of plexes may alleviate the problem. Adding buffers is another alternative. If the BPTM log indicates that the parent process is waiting for full buffers, this means the tape drive is starving for data. In this situation, changing buffer values will not help, but adding multiplexing could help.

The actual delay time can be calculated by multiplying waits for empty buffers by 20 milliseconds and waits for full buffers by 30 milliseconds, and then adding these two numbers. For example, a child process wait count of 65,084 equates to a delay of 21.7 minutes (65,084 multiplied by 20 milliseconds).

The BPTM log delay values can also be viewed as a ratio. An ideal ratio for number of times waited to number of times delayed is 1:1.

Improving Restore Performance

Restore performance problems commonly result from one of four issues:

- Improperly set multiplexing
- Poor index performance
- Sub-optimal fragment size
- `MPX_RESTORE_DELAY` and `NUMBER_DATA_BUFFERS_RESTORE` settings.

The ideal multiplexing setting is one that is only as high as needed to continuously stream the tape drives. Improperly set multiplexing can result in more tape searching than is necessary. However, do not assume that multiplexing is always the cause of performance problems, because this is unlikely.

Performance of the disk subsystem hosting the NetBackup image catalog indexes significantly impacts restore processing. The disk subsystem must be configured for high-performance read processing to allow fast location of catalog entries. Indexing the catalog will also improve restore speed. Testing performance of the catalog disk subsystem determines whether or not indexes are slowing restores.

Fragment size affects how many tape marks are placed. The fewer the tape marks, the slower the recovery. If available, setting SCSI fast locate block positioning to 2048 MB may help. However, as with all other changes, this should be thoroughly tested.

NetBackup can perform multiple restores from a single multiplexed tape. Setting the `MPX_RESTORE_DELAY` to 30 seconds – a change that does not require processes to be brought down and restarted – will improve restore processing.

The `NUMBER_DATA_BUFFERS_RESTORE` parameter enables other NetBackup processes to stay busy while a multiplexed tape is being positioned during restore processing. However, the buffer will cause more physical RAM to be used and should be factored into overall memory usage calculations.

Disk-Based Storage

Storing backup data on disk generally provides a faster restore process. However, disk should never be considered a replacement for tape when used for long-term archival purposes.

Thousands of Small Backups

If a client produces thousands of small backup images, search performance can be improved by the `bpimage` command.



Figure 10. The `bpimage` Command Format and Output Directory Contents

The files `IMAGE_LIST`, `IMAGE_INFO`, and `IMAGE_FILES` must not be edited, because they contain offsets and byte counts used during the seek process. These files add 30 to 40 percent more data to the client directory.

Determining Read and Write Performance of Backups and Restores

The NetBackup error log – located at `/usr/open/netbackup/db/error` – records the data streaming speed, in kilobytes per second, for each backup and restore process. Searching the log for the string “Kbytes/sec” will locate a log record that looks similar to the example in Figure 11. Kbytes/sec will vary depending on many factors, including the availability of system resources and system utilization, but the data can be used to assess the performance of the data streaming process.

```

0996594975 1 4 4 calvin 4 0 0 calvin bptm successfully wrote backup id calvin_0996593762,
copy 1, fragment 1, 2593472 Kbytes at 2530.273 Kbytes/sec
  
```

Figure 11. Sample NetBackup Error Log

The statistics from the NetBackup error log show the actual amount of time spent reading and writing data to and from tape. This does not include time spent mounting and positioning the tape. Cross-referencing the information from the error log with data from the `bpbkar` log on the NetBackup client (showing the end-to-end elapsed time of the entire process) indicates how much time was spent on operations unrelated to reading and writing from and to tape.

Each stream of multiplexed data also has an entry in the error log (see figure 12).

```
0996596712 1 4 4 calvin 10 9 0 calvin bptm successfully wrote backup id calvin_0996595952,
copy 1, 1329238 Kbytes

0996597086 1 4 4 calvin 9 9 0 calvin bptm successfully wrote backup id calvin_0996595951,
copy 1, 1294917 Kbytes
```

Figure 12. Sample NetBackup Error Log for Multiplexing

The final entry in the log for the multiplexed backup shows performance data. Figure 13 shows the performance data for the multiplexed backup listed in figure 12.

```
0996597086 1 2 4 calvin 0 0 0 *NULL* bptm successfully wrote 2 of 2 multiplexed backups,
total Kbytes 2624155 at 456.274 Kbytes/sec
```

Figure 13. Sample NetBackup Error Log Showing Multiplexing Performance Statistics

Figure 14 shows the NetBackup error log entry for a restore. Figure 15 shows the same information for the restore from a multiplexed backup.

```
0996610860 1 16 4 calvin 11 0 0 calvin bptm successfully restored from
backup_id calvin_0996593762, copy 1, 1284736 Kbytes
```

Figure 14. Sample NetBackup Error Log Showing Restore Statistics

```
0996611717 1 16 4 calvin 12 0 0 calvin bptm successfully restored from
backup_id calvin_0996595952, copy 1, 1329238 Kbytes

0996612059 1 16 4 calvin 12 0 0 calvin bptm successfully restored from
backup_id calvin_0996595951, copy 1, 1294917 Kbytes

0996612059 1 2 4 calvin 0 0 0 *NULL* bptm successfully restored 2 of 2 requests, read
total of 2624155 Kbytes at 3981.422 Kbytes/sec
```

Figure 15. Sample NetBackup Error Log Showing Statistics for a Restore from Multiplexed Backup

Log Files

The NetBackup logging facility has the potential to impact the performance of backup and recovery processing. However, logging is usually enabled only when troubleshooting a NetBackup problem, and the short-term nature of the performance impact can often be endured. The performance impact can be determined by the amount of logging used and the verbosity level set.

Java User Interface Performance

In some NetBackup environments, the Java GUIs may seem to run slowly and will occasionally stop responding. Increasing the amount of memory allocated to the GUIs will resolve this problem.

The Java GUIs are initiated by executing either `/usr/opensv/netbackup/bin/jnbSA` or `/usr/opensv/netbackup/bin/jnpSA`. These executables are scripts that set the Java environment before invoking the user interface. Changing the memory allocation for each GUI requires you to edit the script and change the memory allocation value.

Each Java GUI script contains a line with the variables `-ms4m` and `-mx32m`. These values represent the memory allocation defaults for the script, equating to 4 MB minimum and 32 MB maximum.

To increase the memory available to the GUIs, change the variables as follows:

- To set minimum memory allocation to 32 MB, change `-ms4m` to `-ms32m`.
- To change the maximum memory allocation to 256 MB, change `-mx32m` to `-mx256m`.

Summary

Obtaining the best performance from a backup infrastructure is not complex, but it requires careful review of the many factors that can affect processing. The first step in any attempt to improve backup performance is to gain an accurate assessment of each hardware, software, and networking component in the backup data path. Many performance problems are resolved before attempting to change NetBackup parameters.

NetBackup offers plenty of resources to help isolate performance problems and assess the impact of configuration changes. However, it is essential to thoroughly test both backup and restore processes after making any changes to NetBackup configuration parameters.

Appendix

Operating System Settings

Minimum Solaris Kernel Parameters

The Solaris operating system dynamically builds the OS kernel with each boot of the system. For other UNIX operating systems, please reference the system administration procedures to determine how to rebuild the kernel.

The parameters below reflect minimum settings recommended for a system dedicated to running VERITAS NetBackup. If other applications are running on the server, such as the Oracle RDBMS, the values may need to be increased.

The `msgsys:msginfo_msgssz`, `msgsys:msginfo_msgmap`, and `msgsys:msginfo_msgseg` parameters became obsolete with Solaris 8. These variables have been left in place below to avoid error messages. Any values applied are ignored.

- `set msgsys:msginfo_msgmap=500`
The number of elements in the map used to allocate message segments.

- `set msgsys:msginfo_msgmnb=65536`
The maximum length of a message queue in bytes. The length of the message queue is the sum of the lengths of all the messages in the queue.
- `set msgsys:msginfo_msgssz=8`
The size of the message segment in bytes.
- `set msgsys:msginfo_msgseg=8192`
The maximum number of message segments. The kernel reserves a total of `msgssz * msgseg` bytes for message segments and must be less than 128 KB. Together, `msgssz` and `msgseg` limit the amount of text for all outstanding messages.
- `set msgsys:msginfo_msgtql=500`
The maximum number of outstanding messages system-wide that are waiting to be read across all message queues.
- `set semsys:seminfo_semmni=300`
The maximum number of semaphore sets system-wide.
- `set semsys:seminfo_semmns=300`
The maximum number of semaphores system-wide.
- `set semsys:seminfo_semmsl=300`
The maximum number of semaphores per set.
- `set semsys:seminfo_semmnu=600`
The maximum number of "undo" structures, system-wide.
- `set shmsys:shminfo_shmmax=10000000`
The maximum size of a shared memory segment.
- `set shmsys:shminfo_shmmin=1`
The minimum size of a shared memory segment.
- `set shmsys:shminfo_shmmni=100`
The maximum number of shared memory segments that the system will support.
- `set shmsys:shminfo_shmseg=10`
The maximum number of shared memory segments that can be attached to a given process at one time.

The `ipcs -a` command displays system resources and their allocation, and is a useful command to use when a process is hanging or sleeping to see if there are available resources for it to use.

TCP_CLOSE_WAIT_INTERVAL

The `TCP_CLOSE_WAIT_INTERVAL` parameter sets the amount of time to wait after a TCP socket is closed before it can be reused. The current value for `TCP_CLOSE_WAIT_INTERVAL` can be viewed using the commands shown below:

- For Solaris 2.6 or previous use the following command:

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

- For Solaris 7 or above use the following command:

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

- For HP-UX 11 use the following command:

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

- For HP-UX 10 use the "net tune" command instead of "nnd."

Running these commands will produce a large number. The default value for most systems is 240000, representing 4 minutes (240 seconds) in milliseconds. In most cases this value can be shortened to about 1000 (1 second) using the following commands:

- On Solaris 2.6 and previous versions:

```
ndd -set /dev/tcp tcp_close_wait_interval 1000
```

- On Solaris 7 and later versions:

```
ndd -set /dev/tcp tcp_time_wait_interval 1000
```

- On HP-UX 11:

```
ndd -set /dev/tcp tcp_time_wait_interval 1000
```

The `ndd` command changes the value of the `TCP_CLOSE_WAIT_INTERVAL` parameter immediately, without requiring a reboot. The setting will return to the default value after a reboot, however. To ensure the new value is set after each reboot the `ndd` command must be added to the appropriate TCP/IP startup script. On Solaris this script is in `/etc/rc2.d/S69inet`. On HP-UX 11 see `/etc/rc.config.d/nddconf` for configuration examples.

Recommended Network Interface Settings

Set the Network Interface Card (NIC) on all VERITAS NetBackup Clients and Servers to full duplex. Autosense has been known to cause network performance issues. Note that all routers and switches must also be set to full duplex.

The information in this section is from Sun Infodoc 16144 (from April 14, 2000) titled *How to Force the HME Card to Work at 100mb* (full-duplex).

If the auto negotiate does not work, then the 100-MB full-duplex mode can be forced to run at 100-MB full-duplex using the following guidelines.

When forcing one side of the link, it is important to force both sides. The speed and mode on the link partner (switch or hub) must be set to match (or link down), or excessive/late collisions and errors will result.

If using `/etc/rc2.d/S99...`, try the following:

```
ndd -set /dev/hme instance 0
nnd -set /dev/hme adv_100T4_cap 0
nnd -set /dev/hme adv_100fdx_cap 1
nnd -set /dev/hme adv_100hdx_cap 0
nnd -set /dev/hme adv_10fdx_cap 0
nnd -set /dev/hme adv_10hdx_cap 0
nnd -set /dev/hme adv_autoneg_cap 0
```

If using `/etc/system`, try the following:

```
set hme:hme_adv_autoneg_cap=0
set hme:hme_adv_100T4_cap=0
set hme:hme_adv_100fdx_cap=1
set hme:hme_adv_100hdx_cap=0
set hme:hme_adv_10fdx_cap=0
set hme:hme_adv_10hdx_cap=0
```

The commands sequence is important. The link is renegotiated when the interface is brought up with `ifconfig` or when the `nnd nnd adv_autoneg_cap` command is executed.

If adding commands to `/etc/system`, the system must be rebooted. For additional information, refer to the NIC guide and hme platform notes:

- 802-6022-10 SunFastEthernet Adapter 2.0 Installation and User's Guide
- 802-3970-10 Platform Notes: The hme Fast Ethernet Device Driver

Other Helpful Sun Infodocs include:

- 16728 – *How Does 100baseT Ethernet Auto-Negotiation Work and What are the hme and qfe Driver Defaults?*
- 18262 – *How Do I Troubleshoot 100Mb FastEthernet 802.3 Autonegotiation Problems?*
- 16070 – *How to Set hme Parameters for More than One hme Interface*
- 15606 – *How to Set Up an hme.conf file.*

Additional Web Resources

The following Web sites provide information on performance monitoring utilities:

- Iperf – <http://dast.nlanr.net/Projects/lperf1.1.1/index.htm>
- Bonnie – <http://www.textuality.com/bonnie>
- Bonnie++ – <http://www.coker.com.au/bonnie++/readme.html>
- Tiobench – <http://sourceforge.net/projects/tiobench>

VERITAS ARCHITECT NETWORK

VERITAS Software Corporation

Corporate Headquarters
350 Ellis Street
Mountain View, CA 94043
650-527-8000 or 866-837-4827

For additional information about VERITAS Software, its products, VERITAS Architect Network, or the location of an office near you, please call our corporate headquarters or visit our Web site at www.veritas.com.