COMPAQ

Backup and Recovery for Microsoft SQL Server

Compaq TechNote

Includes information on:

- Compaq Tape Drives:
  525-MB ACA,
  2/8-GB DAT,
  4/16-GB TurboDAT,
  and TurboDAT AutoLoader
- Data protection and recovery methods
- Benefits and limitiations on backup and recovery methods
- Local and remote file-based backup and recovery
- Benefits and drawbacks of SQL Server dump options
- Guidelines for choosing the best method for your environment

NOTICE

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

The software described in this guide is furnished under a license agreement or non disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state, or local requirements. Compaq does not warrant products other than its own strictly as stated in Compaq product warranties.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Compaq Backup and Recovery for Microsoft SQL Server

First Edition (March 1995)
Part Number 184489-001

Contents

Chapter 4
Off-Line Backup and Recovery

Chapter 5
On-Line Backup and Recovery

Chapter 6

Key Points for a Successful Backup and Recovery

Glossary

Index

Figures

Tables

Chapter 1
About This TechNote

This Compaq TechNote is intended to provide system integrators with valuable backup and recovery technology specific to the Compaq ProLiant Family of Servers and Microsoft SQL Server for Windows NT. TechNotes provide technical information based on Compaq integration testing and analysis on a specific integration topic.

Data backup and recovery is one of the most important aspects of maintaining a database server. The data contained in your database(s) is the most valuable asset of your company. System integrators consider it to be one of the most underestimated or undefined areas of the application server industry today. With relatively new operating environments, such as Microsoft Windows NT and Microsoft SQL Server for Windows NT, new tools and applications, including those in the area of backup and recovery, are still being developed.

This TechNote presents discussions and analyses of various backup and recovery solutions for your Compaq ProLiant server in a Microsoft Windows NT and SQL Server environment, from both the hardware and software perspective. It also provides you with solutions based on performance, cost, capacity, and functionality. Compaq encourages you to use the information in this TechNote to make a more informed decision about the protection of your data.

Chapter Organization

This chapter is intended to help you use this TechNote; therefore, a brief summary of each chapter is described as follows:

- ***Chapter 1*** -- Introduces the TechNote with a brief summary of each chapter, general information about conventions used, and other resources you could use to find more information on backup and recovery procedures.
- ***Chapter 2*** -- Discusses various data protection and recovery methods that are available with Microsoft SQL Server for the Windows NT environment, such as instance recovery, fault tolerance, and database backup and recovery.
- ***Chapter 3*** -- Provides an analysis of the key benefits and limitations of the various backup and recovery methods available today and discusses some guidelines to help you choose the best method for your environment.
- ***Chapter 4*** -- Discusses local and remote file-based backup and recovery, involving both disks and tapes as destinations and the performance considerations associated with each method.
- ***Chapter 5*** -- Discusses the benefits and drawbacks of various dump options supported by SQL Server, including single and parallel dumps to disk and tape devices, local and remote dumps, and dumps under SQL Server activity.
- ***Chapter 6*** -- Provides conclusions and key points of relevant information on backup and recovery technology available with Microsoft SQL Server for Windows NT and Compaq server products.
- ***Appendix A*** -- Provides information on how to order individual Compaq TechNotes, including pricing information, 1 (800) telephone numbers, and a list of the latest TechNotes available.

Reference Material

Use the following list of resources, which are used throughout this TechNote, for more information on backup and recovery information for Microsoft SQL Server:

- Compaq TechNotes:
  - *Configuring Compaq RAID Technology for Database Servers*
  - *Configuration and Tuning of Microsoft SQL Server for Windows NT on Compaq Servers*
- *Microsoft SQL Server Administrator's Guide*
- *Microsoft SQL Server, Backup and Recovery Guidelines*
- *Microsoft Windows NT Operating System Manuals*
- *Backup Exec for Windows NT System Manual*
- Other Backup and Recovery documentation available from Microsoft
- Third-party software documentation

Vendors

Listed in Table 1-1 are names and addresses of vendors used to perform the testing presented in this TechNote:

Table 1-1
Hardware and Software Vendors

Hardware Vendor: SysKonnect	Software Vendor: Backup Exec for Windows NT
SysKonnect 12930 Saratoga Avenue, Suite D-1 Saratoga, CA 95070 Main (408) 725-4650 Sales (800) SK2-FDDI Fax (408) 725-4654	Arcada Software 37 Skyline Drive, Suite 1101 Lake Mary, FL 32746 (800) 541-2220 or (407) 262-8000 United Kingdom Office: 44-628-777-277

Notational Conventions

This TechNote uses the following conventions to distinguish elements of text found within this document:

Table 1-2
Notational Conventions

Convention	Use
ENTER	Keys on your keyboard appear in boldface and uppercase, not a sequence of keys to press.
Enter	When instructed to enter information, type the information and press the **ENTER** key.
Key + **Key**	When you see a plus sign between two keys, hold down the first key while you simultaneously press the second key. For example, "Press the **CTRL** + **Z** keys" means to press the **CTRL** key while you simultaneously press and release the **Z** key.
Select *item* → *item*	Items separated by arrows indicate items you select in a sequence.
items of importance	Presents important or specific points of information. These items appear in italics in all chapters of this TechNote.
screen selections and *variables*	These items appear in italics in all chapters of this TechNote.
user input, screen display	Information you type exactly as it appears on the screen.
USER INPUT	Information you type exactly as it appears is shown in uppercase.

continued

Notational Conventions **Continued**

FILENAMES	Names of files appear in uppercase italic in the DOS and other environments.
PROGRAMS, COMMANDS, UTILITIES, DIRECTORY NAMES, and DRIVE NAMES	These items appear in uppercase in the DOS and other environments.
DISCLAIMER	Presents information in this TechNote with the purpose of comparing different configurations allowing the user to review and compare different methodologies and configurations.
! IMPORTANT	Presents clarifying or specific points of information.
NOTE:	Presents commentary, sidelines, or interesting points of information.

Chapter 2
Data Protection Concepts

When planning or implementing a comprehensive data protection system, fault tolerance often comes to mind. Fault tolerance, whether hardware-based or software-based, complements not replaces backup and recovery. The primary purpose of backup procedures is to store data on an auxiliary storage medium for long term retention and catastrophic failure recovery, for protection against accidental or malicious data deletions or modifications, theft, viruses, etc. Fault tolerance is protection against loss of data due to a hardware failure. For mission-critical systems, where data availability is crucial and any downtime due to a hardware failure is very costly, fault tolerance is a necessity.

Replication, on the other hand, is a form of fault tolerance at the server level. It involves two or more servers, which all contain a replica of the primary database, synchronized either real-time or near real-time. In the event of a primary server failure, all user requests are rerouted to a designated secondary server.

This chapter discusses various data protection and recovery methods. All of these methods are available to you with Microsoft SQL Server for the Windows NT environment. The data protection concepts discussed here include:

- Instance Recovery
- Fault Tolerance
- Database Backup and Recovery

Instance Recovery

Automatic data protection and instance recovery is where the system has the ability to preserve the effects of all committed transactions and ensures the database is in a consistent state after a recovery from any single point of failure. This single point of failure includes a hardware component failure, software component failure, power loss, and so on.

SQL Server provides this automatic data protection and instance recovery through its write-ahead log, called the *transaction log*, where SQL Server records changes before it modifies the actual data pages. In the case of a server failure and recovery, SQL Server uses the transaction log upon startup to *roll back* uncommitted transactions and *roll forward* transactions that were committed but not applied to the data files.

All data definition changes and all data manipulation changes are logged in the transaction log. Examples of data definition changes are creating or dropping tables, views, stored procedures, rules, and so on. Examples of data manipulation changes are inserts, deletes and updates, but not selects. Certain other events such as checkpoints and dumps are logged as well. All data definition changes, checkpoints, dumps, and so on, do not roll backward or forward, they are just logged in the transaction log.

This data protection scheme cannot be turned off and guarantees that data remains consistent at any point in time. Automatic data protection and instance recovery, in terms of the write-ahead transaction log, are inherent to SQL Server. For more detailed information about transaction logging, refer to the Microsoft SQL Server manuals and other publications related to SQL Server.

Fault Tolerance

Fault tolerance and database archiving should coexist on the same system. Fault tolerance, both hardware and software based, provides immediate protection against a hardware or a software component failure. If you implement hardware-based fault tolerance in conjunction with hot-pluggable drive subsystems, you have continuous data availability, maximum performance, and automatic data recovery. Database archiving allows for a long term data retention and recovery from operational errors and catastrophic failures.

For performance characteristics of various RAID levels of the Compaq SMART Array Controller and configuration considerations, refer to the TechNote *Configuring Compaq RAID Technology for Database Servers*.

The drive subsystem has the highest chance of failure of all components in the system. Here is where various levels of fault tolerance are available, ranging from no fault tolerance to fault tolerance for the entire drive subsystem. The higher level of fault tolerance, the higher the implementation cost. Many times a high level of fault tolerance can be an expensive solution. You must weigh the cost of implementing a good fault tolerant system against the cost attributed to any significant down-time. Sites requiring access to data with minimal down-time must implement hardware fault tolerance for the entire drive subsystem. Sites that are not mission-critical can implement a lower level of fault tolerance.

If you choose to implement fault tolerance on the entire drive subsystem, a drive failure will not interrupt any operations and will not cause any data loss. With the Compaq SMART Array Controller and the hot-pluggable drive subsystem, you can replace a failed drive and restore fault tolerance without taking the system off-line.

Your data is also safe with disk fault tolerance installed only on the transaction log volume along with systematic database and transaction log dumps. If you have a full database dump and prior transaction log dumps, you can easily restore your data in case of a failed data drive. This data drive recovery, however, requires the system to be off-line. If you lose a transaction log drive, fault tolerance protects this volume and the recovery process consists of replacing only the failed drive and letting the controller rebuild the new drive to full fault tolerance.

With no fault tolerance installed, you run a risk of losing transactions even in the case of frequent backups. If your transaction log drive fails, you lose any changes to your database that occurred after your last transaction log dump. In addition to running the risk of data loss, you will have to take the system off-line to perform the recovery.

For more detailed information on the backup and recovery procedures in various failure scenarios, refer to the *Microsoft SQL Server Administrator's Guide*.

Database Backup and Recovery

The ability to recover data depends entirely on the data protection scheme. This protection scheme can consist of various methods, such as implementing database and transaction log backups alone to combining these backups with fault tolerance or even taking advantage of replication.

Backup and Recovery Strategies

Database backup and recovery consists of a database administrator archiving data onto a secondary storage medium based on scheduled intervals such as, daily or weekly. In the case of a failure, the database administrator restores the data by using the archived data from the secondary storage. Secondary storage can be a remote backup server, a local hard disk device, a local tape device, or a local optical disk device.

You can accomplish database archiving in two ways:

- The database administrator can shut down SQL Server, thus closing all database and transaction log files, then perform a file-based backup of all SQL Server devices.

  This type of backup is called *off-line backup*: It is necessary to stop SQL Server to gain access to the files, since SQL Server keeps all database and transaction log files open with exclusive access as long as the database engine is running. Therefore, any attempt to backup these files while SQL Server is running will fail.

- The database administrator can use the SQL Server backup facility to dump the contents of the transaction log, the database, or both, while SQL Server remains running.

  This type of backup is referred to as *dynamic backup* or *on-line backup*. The SQL Server backup operation is referred to as a *dump* and the restore operation is referred to as a *load*.

Choosing the correct backup and recovery strategy involves a careful analysis of your environment. Some mission-critical environments can accept only one strategy: on-line backup and recovery. Others can employ either on-line or off-line backup and recovery. The following section presents some of the major characteristics of each strategy. The chapters found later in this TechNote provide more detailed information on database backup and recovery, especially in the area of performance.

Off-Line Backup Characteristics

To accomplish an off-line backup, you must shut down SQL Server. When you do this certain characteristics accompany this strategy. They are as follows:

- All user databases remain inaccessible for the duration of the backup operation. SQL Server opens all databases at startup and keeps them open whether they are being accessed or not. Therefore, you cannot use typical file backup utilities while SQL Server is running.

- All file backup utilities backup the files in their entirety. For example, you allocate a 100 megabyte database device, create a 100 megabyte database on this device, and store 50 megabytes of data in this database. The operating system detects 100 megabytes of data that needs to be backed up. However, in reality you are backing up 50 megabytes of data and 50 megabytes of empty database structures, increasing both the duration of the operation and the amount of the backup storage used
or needed.
- You must backup all database and transaction log files to perform the recovery. SQL Server keeps track of all user databases in the master database and time stamps all files. If you restore only selected files, SQL Server detects inconsistency between the files and the information in the master database. Likewise, to restore a database, you have to restore all database and transaction log files; thereby, keeping them all consistent.
- You cannot perform incremental backups (i.e., backups of changes since the last full backup) or partial backups (i.e., backups containing selected subsets of data, such as a particular database).
- Disk Input/Output (I/O) associated with file-based backup utilities typically occur in 64 kilobyte blocks, thus improving performance.
- Many file backup utilities, such as Backup Exec for Windows NT, support software data compression. Software data compression typically yields the fastest backup throughputs.
- Off-line backups can take advantage of verifying the data (i.e., backup with verify) on the tape after backup. This guarantees data integrity on the tape.
- With Backup Exec for Windows NT and the Compaq TurboDAT AutoLoader, you can group tapes together to increase the capacity. Groups of tapes appear to the host as one continuous tape with an increased capacity. The next tape in the group automatically and transparently loads after the current tape is filled.

On-Line Backup Characteristics

SQL Server-based database and transaction log backups have the following characteristics:

- A major limiting factor of the current version of SQL Server-based backup is performance. All read requests that the SQL Server backup facility generates are 2 kilobytes. SQL Server buffers these pages and writes them to the dump device in 60 kilobyte blocks. Similarly during the load operation, SQL Server reads from the dump device in 60 kilobyte blocks and writes pages to the disk in 2 kilobyte blocks. With this type of disk I/O, you can expect certain I/O throughput limitations. This TechNote analyzes the performance implications of SQL Server-based and file-based backups in greater detail in later chapters.
- SQL Server dumps a database or a transaction log while permitting access to all user databases. All processing in any database not being dumped continues normally. Any changes (e.g., insert, delete or update requests) to databases not being dumped remain unaffected.

  NOTE: Look-up (read-only) queries have no affect on the user data in this discussion.

  In the database being dumped, any changes to the data page already dumped occurs immediately. This change, however, is not included in the backup image. If a transaction intends to update a data page or an index page not yet backed up, SQL Server places this transaction on an internal queue, dumps that page before allowing any change to this page, then proceeds with the transaction. Therefore, the database image is the data captured at the instant the DUMP command is issued. Performance implications of this type of backup are discussed later in this TechNote.
- SQL Server backs up only the pages containing data — both data and index pages. As a result, unnecessary data is not backed up. However, during the load operation, SQL Server initializes all unused pages in addition to reloading all used data and index pages, thus consuming more time than the dump operation.
- On-line archiving allows for incremental backups (i.e., transaction log dumps) and partial backups (i.e., dumps of selected databases).

- On-line dumps can take advantage of hardware data compression, if it is available via the tape drive. However, software data compression is not available.
- Since on-line dumps allow backups while the database is active, it would not be efficient to perform a verify operation after the backup. Therefore, the on-line backups do not support the verify operation. SQL Server, however, reports any errors it encounters via the error log and the event log.
- During a restore operation, the damaged database must be dropped, recreated, and the most recent copy of the database dump is then loaded from the backup media. SQL Server locks the database being restored for the duration of the operation. However, all other databases remain open for access.
- SQL Server currently does not support the autoloading capabilities of the Compaq TurboDAT AutoLoader. SQL Server dumps to and restores from an installed TurboDAT AutoLoader, but regards this device as a 4/16-Gigabyte TurboDAT drive. For example, it prompts for the next tape, should one be needed, instead of automatically loading the next tape in the magazine.

Transaction Log vs. Database Archiving

Database dumps create complete images of both the database and the transaction log, take more time to perform than transaction log dumps, and consume more of the backup media. A recent database dump, however, might provide a faster data recovery as opposed to loading an older database dump and a number of transaction log dumps created afterwards.

For example:

Scenario 1: You create a database dump on Sunday, then perform transaction log dumps each weekday. If you experience a failure on Saturday, you have to restore the database image created on Sunday and five transaction log dumps created for each day of the week.

Scenario 2: You create two database dumps, one on Sunday and another on Wednesday and also perform transaction log dumps on the days in between. If your system fails on Saturday, you can restore the database image from your most recent database dump (i.e., Wednesday) and you only have to restore two transaction log dumps (i.e., Thursday and Friday).

Transaction log dumps, also called on-line incremental backups, create only images of the transaction log, take substantially less time to perform, and consume only a fraction of the backup media as compared to database dumps. Having a high number of transaction log dumps since the last database dump prolongs your data recovery.

NOTE: The transaction log must reside on its own physical device for the transaction log dump to take place.

The strategy involving these types of backups and their frequency depend entirely on your requirements. A careful analysis of your needs, along with the techniques outlined in the *Microsoft SQL Server Administrator's Guide*, will help you determine the plan that best meets your needs.

Storage Destinations

Many options are available for storing your backup images. You can back up your data to a local tape drive, a hard disk drive, or a local rewritable optical disk drive. Another alternative is to back up to a remote server, which can store the data on its tape drive, hard disk drive, or rewritable optical disk drive. This section discusses the advantages and disadvantages of each type of storage destination.

Local Backup vs. Remote Backup

Local backup consists of archiving your data to a storage device physically attached to the database server. This device can include a tape drive, a hard disk drive or a rewritable optical disk drive. You can also use a separate disk drive for temporary storage and move the image to a tape at a later time.

Remote backup consists of archiving your data to a storage device located on another server. This device can consist of a hard disk drive or a rewritable optical disk drive. SQL Server does not support direct dumps to a remote tape, even though you can dump your data to the shared disk drive of a remote server and then move it to a tape that is local to that server. The remote server can then act as a database backup server, to which multiple database servers dump their data.

With a local backup implementation, each database server requires its own backup storage. This storage is typically a tape drive, even though it might also be a separate disk volume. There is no additional network overhead generated, but users of this server will experience an increased response time due to the additional workload induced by the backup activity.

Using the same physical disk volume as your database or transaction log volume is not recommended for two reasons:

1. If your dump destination volume is the same as your database volume and you lose a disk on this volume, you have no way of recovering your data.
2. Isolating I/O generated by the dump activity from the transaction log and database I/O activity minimizes performance impact.

With a remote backup implementation, the cost of the backup server hardware and software can be shared among many database (and other types of) servers. You can, in some cases, even use an existing file server as your database backup server. You can alleviate additional overhead generated by increased network activity by scheduling backups for after-hours or by dedicating a separate, high-speed network link between the database server and the backup server.

Tape Backup vs. Disk Backup

Backups done directly to a tape device are convenient and relatively inexpensive. The capacity and the speed are dependent upon the tape hardware. The tape device must be physically attached to the database server to perform SQL Server dumps to tape. File-based tape backup utilities support archiving of remote data, but the backup software must be running on the same machine as the tape drive..

Backups done directly to a disk device (local or remote) are typically faster but are more costly than tape backups. You can consider dumping directly to a disk device and then moving the image onto a tape.

Backup Frequency, Scheduling, and Retention

The frequency of your database and transaction log dumps, scheduling of these dumps, and the retention period all depend entirely on your environment. Factors such as an acceptable amount of work that could be lost, if any, acceptable down-time due to a recovery from a failure, the volume of update transactions, and so on, all influence the backup frequency, scheduling, and data retention. SQL Server offers a "media retention" option, configurable through *sp_configure*, which allows you to prevent overwriting an existing dump too soon. Automatic scheduling is available via SQL Server or third-party software. Refer to the *Microsoft SQL Server Administrator's Guide*, other documents available from Microsoft, or any third-party software documentation for more detailed information.

Recovery Considerations

The recovery operation consists of restoring your data from the backup media. You must use the same utility to restore your data as you used to create the backup. If you used a file-based backup utility, you must use the same utility to restore your data. If you used SQL Server to dump the database or the transaction log, you must use SQL Server to reload your data. The same considerations that apply to backups also apply to restores.

Restoring Off-line Backups

Restoring your database from an off-line backup involves:

- Restoring all database and transaction log files.
- Restoring your master database file.

All SQL Server databases remain unavailable until all files are restored.

Restoring On-line Backups

Restoring your database from an on-line backup involves:

- Dropping the damaged database and lost devices, then recreating both.

  This creation of the database framework can be very time-consuming, depending on the size of your database. Consequently, this adds to the total time needed to restore your data.

- Loading the most recent database dump and all subsequent dumps of the transaction log. Do this after all database devices are created and an empty database is built.

The entire on-line restore procedure locks the database and disallows any use. Other undamaged databases remain on-line and accessible.

Recovery Success

The level of success in recovering all your data and the time required to do so depend on your backup strategy. The most important consideration is, if you cannot restore your data, your backups are worthless. Therefore, spend time and effort to test your backup and recovery strategy.

For more details on various SQL Server-based recovery scenarios, load requirements and other considerations, refer to the *Microsoft SQL Server Administrator's Guide* or the manuals available with your backup software. A further analysis of some considerations and performance characteristics are discussed later in this TechNote.

Backup Tips

The following information provides some helpful information on backups for disk drive dumps and database consistency.

Database File Size

When SQL Server dumps to a disk device, each subsequent dump automatically adjusts the size of the existing file to reflect the amount of data dumped. This did not happen in prior versions of SQL Server for OS/2, that is, if less data was dumped, the file size did not readjust.

For example, you initially dump 100 megabytes to a *DISKDUMP.DAT* file, copy it onto tape, but do not delete the file. Then, the next day, you dump a transaction log (25 megabytes of data) to the same dump device.

- With SQL Server for OS/2, the *DISKDUMP.DAT* file would remain 100 megabytes, even though it actually held only 25 megabytes of data.
- With SQL Server for Windows NT, the *DISKDUMP.DAT* file adjusts the size of the file to 25 megabytes.

Database Consistency Checking

Microsoft recommends running database consistency checking on the database before performing database dumps. This database checking includes the following files:

- *DBCC CHECKDB*
- *DBCC CHECKALLOC*
- *DBCC CHECKCATALOG*

If the database contains consistency errors, these errors will be included in the dump and in the database when it is reloaded. These consistency errors could prevent the database from being loaded. The execution of these consistency commands might, however, take extended periods of time. Refer to Microsoft's *SQL Server Backup and Recovery Guidelines* document for more information.

Chapter 3
Backup and Recovery Implementation

This chapter provides an analysis of the key benefits and limitations of the various backup and recovery methods available today for your Microsoft Windows NT and SQL Server platform running on Compaq ProLiant servers. It also discusses some guidelines to help you choose the best method for your environment.

DISCLAIMER: The performance numbers presented in this TechNote are included for the purpose of comparing different configurations. Review and compare the different methodologies and configurations and decide which configuration achieves the best throughput rates for your installation. The throughput rates can and will vary with the nature of data being backed up and with other variables. This is particularly true with file-based backups, using data compression. The type of data you are backing up will determine the effectiveness of the data compression algorithms and in turn affect the backup throughput. Highly compressible data sets will yield higher backup throughputs and vice versa. The "Data Compression" section in Chapter 3, provides guidelines for more accurate performance estimates with various types of data sets.

Product Evaluation

The following products were included in the performance evaluation for this TechNote:

Compaq tape drives:

- 525-Megabyte ACA Tape Drive
- 2/8-Gigabyte DAT Drive
- 4/16-Gigabyte TurboDAT Drive
- TurboDAT AutoLoader

NOTE: The TurboDAT AutoLoader is essentially the 4/16-Gigabyte TurboDAT Drive with the tape autoloading capabilities, and thus the performance considerations are the same.

Software products:

- SQL Server and its backup capabilities
- Windows NT Backup (included with Windows NT)
- Backup Exec for Windows NT (available from Arcada Software)

The products included in the evaluation were the only products available on the market at the time this TechNote was written. The GB/hour metric was chosen to represent backup throughput.

Tape Formats and Cartridges Supported

The following section discusses the tape formats and cartridges that you can use to backup Windows NT and SQL Server:

- The 525-MB ACA tape drive supports QIC (Quarter Inch Cartridge) tape formats and uses serpentine recording technology. This tape drive is a SCSI-1 device and does not support hardware data compression.
- The 2/8-Gigabyte DAT drives and the 4/16-Gigabyte TurboDAT support DDS (Digital Data Storage) and DDS-DC (Digital Data Storage-Data Compression, using Lempel Ziv algorithms) tape formats and use helical recording technology.

  The DDS-DC format is a superset of the DDS format and ensures backward compatibility with uncompressed tapes. The 4/16-Gigabyte TurboDAT drive and the 2/8-Gigabyte DAT drive are both SCSI-2 devices.
- The 4/16-Gigabyte TurboDAT drive is a DDS-2 drive and the 2/8-Gigabyte DAT drive is a DDS-1 drive. DDS-1 is the first specification of the DDS tape format and DDS-2 is the second specification. DDS-2 is intended to at least double the recording density and read/write tape speeds. The benefits of DDS-2 support are evident not only in increased capacity, but also in performance of the tape drive in the uncompressed mode. As stated earlier, a DDS-2 compliant tape drive should perform at double the speed of a DDS-1 compliant tape drive.

For example, the 2/8-Gigabyte DAT drive is capable of approximately 0.61 Gigabytes/hour throughput without data compression, whereas the 4/16-Gigabyte TurboDAT drive is capable of approximately 1.3 Gigabytes/hour throughput without data compression.

All Compaq DAT drives support hardware data compression. The capacity of the cartridge depends on the length of the tape and the compression used. For example:

- 60m DDS-1 cartridges can hold 1.3 gigabytes of uncompressed data
- 90m DDS-1 cartridges can hold 2.0 gigabytes of uncompressed data
- 120m DDS-2 cartridges can hold 4.0 gigabytes of uncompressed data

Data compression ratios, according to the vendors' specifications, can vary between 2.5 to 1 and 4 to 1, depending on the nature of data. If the data set is highly compressible and will compress in the 4 to 1 ratio, the cartridges can hold 4 times the capacity.

Table 3-1 summarizes the tape cartridge support of the tape drives evaluated in this TechNote:

Table 3-1
Tape Cartridge Support Matrix

Tape Drive	Tape Cartridges Supported
Compaq 4/16-GB TurboDAT Drive Compaq TurboDAT AutoLoader	60m DDS-1, 90m DDS-1 and 120m DDS-2 DAT cartridges (see Note below)
Compaq 2/8-GB DAT Drive	60m DDS-1 and 90m DDS-1 DAT cartridges
Compaq 525-MB ACA Tape Drive	DC6525, DC6320, DC6250, DC6150 and DC600A cartridges

NOTE: Using the 60m or 90m DDS-1 cartridges in the 4/16-Gigabyte TurboDAT drive will yield the same throughput as using the 120m DDS-2 cartridge. However, the capacity difference might influence the backup times if you run out of space on the cartridge and need to change tape cartridges.

The tape drive documentation for each product provides a complete matrix of supported formats and tape cartridges.

Compaq Tape Drive Support Matrix

Table 3-2 provides a quick overview of Compaq tape drives and associated device drivers. We have included only the currently available tape drives and only those drives that would likely be used in a database server. For further information, refer to the manuals provided with your Microsoft Windows NT operating system or the README file included on the Compaq Software Support Diskette for Windows NT (NT SSD).

NOTE: Always make sure that you are using device drivers from the latest Windows NT SSD available from Compaq. The diskette contains an automated SETUP program and detailed README information.

Table 3-2
Compaq Tape Drive Support Matrix

Tape Drive	Device Driver	Device Driver Location
Compaq 525-MB ACA Tape Drive	Wangtek 525, 250 drives	Microsoft Windows NT standard product
Compaq 4/16-Gigabyte TurboDAT Drive	4 millimeter DAT	Compaq Windows NT SSD
Compaq TurboDAT AutoLoader	4 millimeter DAT, Arcada supported SCSI-2 Medium Changer	Compaq Windows NT SSD, Backup Exec for Windows NT from Arcada Software
Compaq 2/8-Gigabyte DAT Drive	4 millimeter DAT	Microsoft Windows NT standard product

NOTE: Among the file-based backup utilities we have evaluated, only Backup Exec for Windows NT from Arcada Software currently supports the Compaq TurboDAT AutoLoader. It is necessary to install both the 4 millimeter DAT device driver and the Arcada supported SCSI-2 Medium Changer driver. The product diskettes provide the latest device driver.

You can install the appropriate device driver(s) by using the *Windows NT Setup* icon, *Add/Remote SCSI Adapters* and/or *Add/Remote Tape Devices* options. Refer to the manuals provided with the Windows NT operating system for more detailed information.

Data Compression

Data compression reduces the time and storage required to backup systems; however, this is at the expense of adding some processing overhead. Compression ratio and levels of support vary from one method to another and from one data set to another. Data compression is usually available through hardware (e.g., tape drive or tape drive controller) or backup software.

Hardware data compression, if available with the tape drive, is usually factory-enabled. In the data compression mode, data is always compressed when written to the tape cartridge; however, the drive reads both compressed and uncompressed tapes. You can turn data compression off by using a switch located on the tape drive or by using tape software that supports data compression control.

Backup software packages, which support software data compression, typically allow the user to interactively enable or disable data compression at both the hardware and software level. You can choose either type of data compression, but not both. Refer to the backup software documentation manual(s) for more information on software data compression.

The type of the data determines the effectiveness of data compression and directly impacts backup performance. Highly compressible data, which is characterized below, yields substantially higher backup throughput rates than data that is not compressible. Data that compresses well includes:

- Unallocated space within the database, that has never been used. This includes:
  - Space on a SQL Server device that has not been allocated to a database or a transaction log.
  - Space in a database or in a transaction log that is not allocated to an object. However, this does not include pages that were used previously but marked by SQL Server as “unused” or “free.”

Even if your transaction log is empty, that is, after a transaction log truncates, its pages might contain transaction log entries and are only *marked* as empty. Any file-based backup utility regards such marked pages as containing data. This does not apply to on-line dumps, since SQL Server "knows" which pages contain valid data and which do not and dumps them accordingly.

- Repeated patterns within the data set. The compression algorithms detect repeated patterns in the database (regardless of data type), such as repeated customer names, cities or states, phone number prefixes, credit limits, and so on, and compresses them at high ratios.

  Database tables or indexes are not the only repeated patterns within a data set. Transaction logs can also have highly repeatable patterns. For example, when transactions modify rows of data with the same or very similar columns (e.g., date and time stamps or any repeated values), two or more transaction log entries will have highly compressible data.

The Effects of Data Type When Using Data Compression

Figure 3-1 illustrates the influence of data compression and the data type on backup performance. These results were obtained from an off-line backup, using the Backup Exec for Windows NT utility. The information that follows describes the fields presented in Figure 3-1.

- *Unallocated Space* - represents a backup of an empty database. This database consists of initialized and never used pages only, and yields very high compression ratios and very high backup throughput.
- *Repeated Patterns* - represents a data set with a high percentage of repeated data and unused space, thus relatively high compression ratios and high performance.
- *Random Patterns* - represents a data set with a very low to zero percentage of repeated data and unused space, thus low or no data compression. The chart illustrates how the compression ratios and backup performance vary, depending on the nature of data.

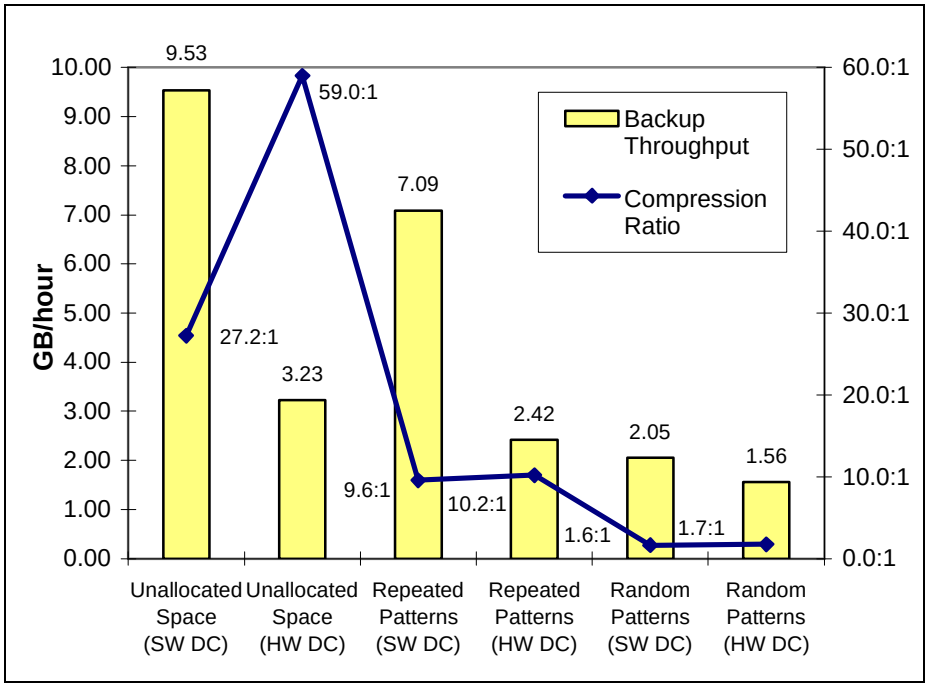

Figure 3-1. Data compression ratios and performance

In Figure 3-1 our test illustrates three data types with software (SWDC) or hardware (HWDC) data compression methods being used and from this we can conclude the following:

- Hardware data compression is typically more effective than software data compression, when both are operating on the same data set. This is an advantage for sites that want to conserve backup storage and are not pressed to perform backups within the smallest time window possible.
- Software data compression is typically faster than hardware data compression, since it can utilize a more powerful and faster system processor. Performance of software data compression is an advantage for sites that have a need to back up as fast as possible, with less concern over the backup storage consumed.

Optimizing the Use of Data Compression

To achieve the results presented in this TechNote, we used a dual Pentium/66 configuration. System processor utilization monitoring showed that the system had sufficient processing power to keep up with the load. However, when the processor utilization reached 90% or higher, tests showed that upgrading to a faster processor(s) or installing additional processor(s) increased performance.

> **!** **IMPORTANT:** It is important that with both hardware and software data compression, you keep the tape drive *streaming* at all times to achieve optimal performance. Streaming means the system can and does deliver enough information to the tape drive to keep it constantly writing, as opposed to the tape drive having to stop and wait for new information.

If the system cannot keep the tape drive streaming, you might achieve better performance without using any data compression. By not using data compression, data is written to the tape "as is." Since no compression occurs, enough data is delivered to the tape drive to keep it constantly busy. With data compression, data is analyzed, compressed, and only a fraction of the original amount of data is actually being written to the tape. If the compressed data stream is interrupted due to the host not keeping up, the tape drive writes out the current data, stops and waits for the data stream to start again. Once this occurs, the tape drive has to search for the end of data, reposition its write heads, then continue writing. This overhead is known as the head positioning overhead. Later in this TechNote this scenario is discussed with a SQL Server dump to the Compaq 4/16-GB TurboDAT drive

File-based backup software typically utilizes the CPU(s) heavily when software data compression is on and operating on highly compressible data. To achieve optimal off-line tape backup throughput, make sure you have a fast processor (at least a Pentium-based processor, or dual processors). Use the Performance Monitor to determine your CPU utilization level. Hardware data compression, however, does not utilize the CPU(s) at all and even the disk I/O imposes little overhead on the system processors.

When using the on-line backup method to a tape device, your backup throughput is limited by SQL Server and how fast it can perform 2 kilobyte reads from the disk subsystem. Hardware data compression increases the throughput, but the performance benefits are limited.

It is extremely difficult to predict the backup duration and throughput without knowing the compressibility of your data. Even when the compressibility is known, you can only guess the throughput and duration time for the backup. In highly volatile environments, where the data changes rapidly, it is even difficult to use past experiences with the same database as a guideline.

Consequently, for the remainder of this TechNote, we have chosen to present performance numbers with a database that has a very low percentage of repeated data and unused space, thus a low compression ratio.

NOTE: Our test database compressed approximately 1.71:1 with software data compression and approximately 1.85:1 with hardware data compression.

Our discussion, therefore, represents the "worst-case" scenario. A database with a higher compression ratio would yield higher backup throughputs.

Chapter 4
Off-Line Backup and Recovery

This chapter compares local and remote file-based backup and recovery, involving both disks and tapes as destinations and the performance considerations associated with each method.

Off-Line Backup Considerations

When considering an off-line backup method, be aware of the following:

- Windows NT Backup does not support multiple simultaneous backup or restore operations. You are limited to only one backup or restore at a time.
- The performance of the Compaq TurboDAT AutoLoader is equivalent to the performance of the Compaq 4/16-Gigabyte TurboDAT Drive.
- Currently only Backup Exec for Windows NT from Arcada Software supports the TurboDAT AutoLoader. Refer to the "Product Evaluation" section in Chapter 3 for further information.
- Backup Exec for Windows NT supports one running copy of the utility per tape device. Therefore, you can increase the backup throughput by utilizing multiple tape devices simultaneously.
- Memory consumption of the file backup utilities is typically very small. Compaq measured the consumption of Windows NT Backup to be about 2.15 megabytes and the consumption of Backup Exec for Windows NT to be about 3.3 megabytes, with the software data compression enabled.

  NOTE: Compaq measured the memory consumption by using the Performance Monitor, object Process, and counter Working Set.

- All file-based backup utilities tested perform disk reads in 64 kilobyte blocks. Therefore, the sequential read performance of the off-line backup is much better than the sequential read performance of the on-line (SQL Server-based) backup. The on-line backup uses 2 kilobyte disk reads and the speed of your backup hardware (e.g., a tape drive) has a bigger influence on the overall performance of the backup operation.

Single Tape Device Backup

During an off-line backup to a single tape device, backup throughput will in most cases be limited by throughput of the tape device. With remote backups to a tape device, the network might influence the backup throughput, especially if the network utilization is high. Data compression can increase backup throughput, particularly when the data is compressible and the system is capable of streaming the tape drive. As mentioned earlier, it is important to monitor the CPU utilization when using software data compression because you want to eliminate any performance restrictions at the CPU(s).

Figure 4-1 provides relative performance information on selected Compaq tape drives with:

- No data compression
- Software data compression
- Hardware data compression, when applicable.

The throughput represents a backup of a populated database and its corresponding transaction log to a single tape device, using Backup Exec for Windows NT without the verify operation.

NOTE: The 525-MB ACA tape drive does not support hardware data compression. All tape drives were installed at the database server.

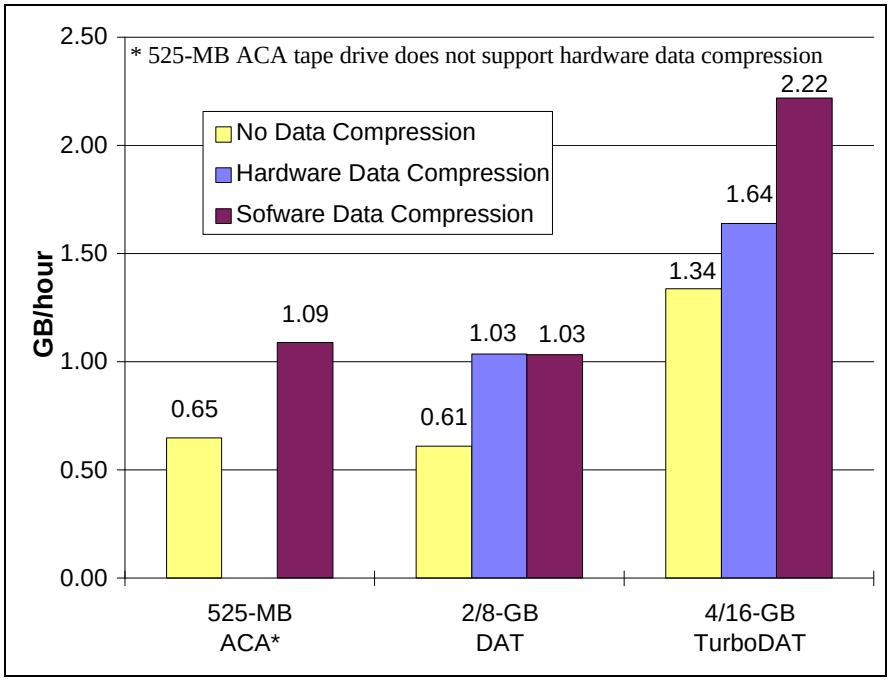

Figure 4-1. Off-line performance of Compaq tape drives

Multiple Tape Device Backup

Employing multiple tape devices simultaneously can greatly improve backup throughput. For maximum throughput, Compaq recommends using one tape drive for each logical disk volume you are backing up. Attempting to back up the same logical disk volume (i.e., two different database files) to multiple tape drives introduces random disk I/O instead of sequential disk I/O thus limiting the disk performance.).

The Compaq tests performed for multiple tape backups were with two 4/16-Gigabyte TurboDAT drives on a single Fast-SCSI-2 Controller, each backing up a separate logical volume using two running instances of Backup Exec for Windows NT. During testing Compaq did not find any significant benefit by configuring these two tape drives on two separate Fast-SCSI-2 Controllers. However, if you use more than two tape drives simultaneously, multiple controllers might be beneficial. When you run multiple copies of Backup Exec for Windows NT with software data compression, the possibility of CPU performance restrictions increase. To eliminate this restriction, sometime known as "bottlenecking," make sure your system has enough CPU power to perform multiple simultaneous backups.

Off-Line Single vs. Multiple Backup with a Local Tape Drive

Figure 4-2 represents a comparison between utilizing a single tape backup (the "serial" denotation) and multiple tape backups (the "parallel" denotation), performed with tape drives installed locally at the database server.

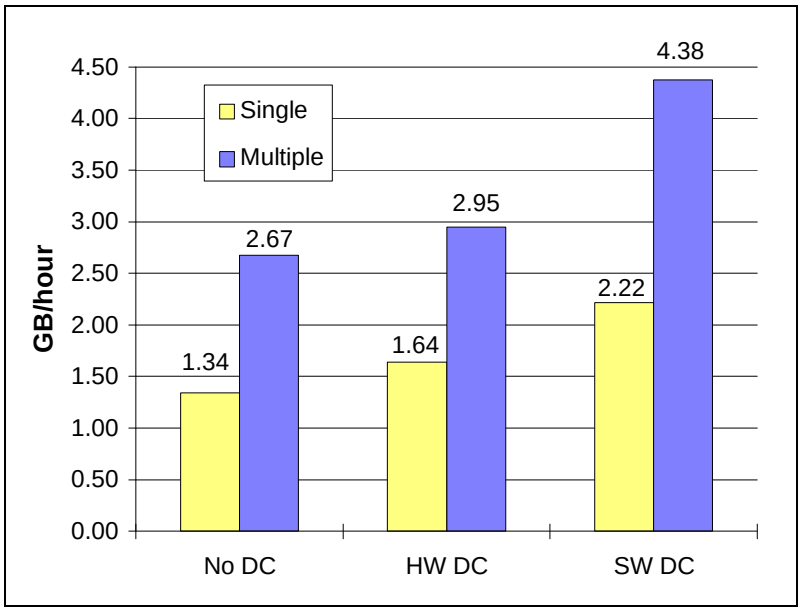

Figure 4-2. Off-line: single vs. multiple backup (local tape drive)

Off-Line Single vs. Multiple Backup via Ethernet

Figure 4-3 represents a comparison between utilizing a single tape backup (the "serial" denotation) and multiple tape backups (the "parallel" denotation), performed with tape drives installed remotely and utilizing an Ethernet network. The relatively low scalability with software data compression and multiple backups is due to the tape drive not streaming. .

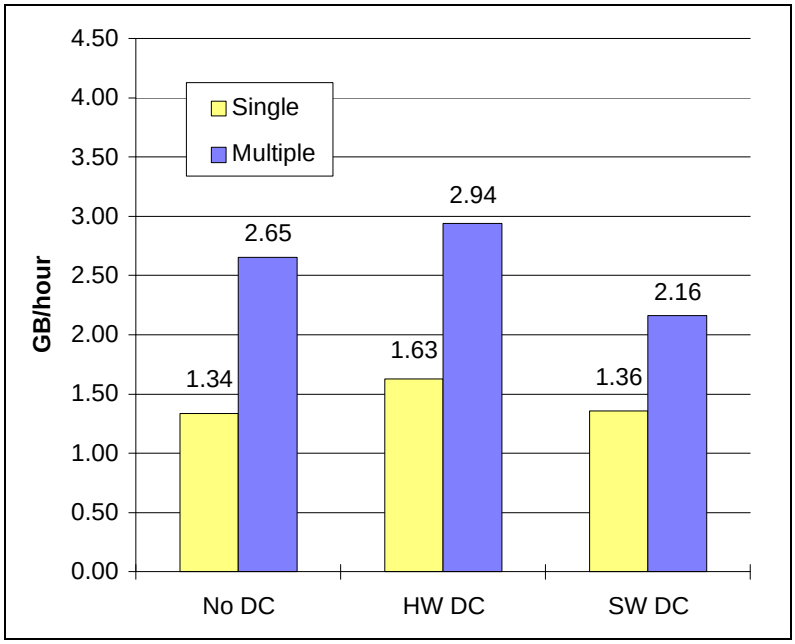

Figure 4-3. Off-line: single vs. multiple backup (remote tape drive via Ethernet))

Off-Line Single vs. Multiple Backup via FDDI

Figure 4-4 represents a comparison between utilizing a single tape backup (the “serial” denotation) and multiple tape backups (the “parallel” denotation), performed with tape drives installed remotely and utilizing an FDDI network.
.

NOTE: Compaq used SysKonnect SK-NET FDDI-FE EISA 32-Bit FDDI LAN adapters and cabling to connect the database server and the backup server.

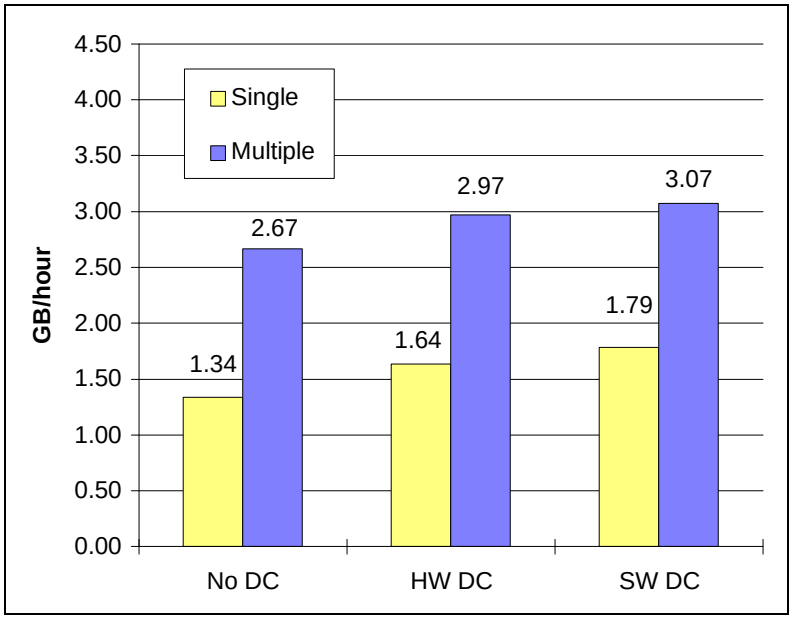

Figure 4-4. Off-line: single vs. multiple backup (remote tape drive via FDDI)

Multiple backups to tape drives can significantly improve backup performance, especially when you are able to preserve sequential disk I/O as discussed earlier in this chapter and keep the tape drive(s) streaming.

With multiple backups you can achieve:

- Almost 100% scalability when multiple backups are performed locally
- Almost 100% scalability when multiple backups are performed remotely with no data compression
- Approximately 70-80% scalability when multiple backups are performed remotely with data compression

When you perform multiple backups across the network, the network bandwidth and the amount of traffic on the network can influence backup throughput and multiple backup scalability. Compressing your data might increase the backup throughput, as long as the tape drive is kept streaming. Software data compression can provide significant improvements in backup throughput on high-speed networks with fast throughput rates and large packet sizes, such as FDDI.

Disk Backup

You perform off-line backups to disk devices by using the operating system COPY or XCOPY commands. Compaq testing did not find any variances in performance by using the COPY command over the XCOPY command.

Figure 4-5 represents a comparison between a single disk backup (the “serial” denotation) and multiple disk backup (the “parallel” denotation), performed both locally to a dedicated disk volume(s) and remotely, utilizing a dedicated Ethernet or FDDI network..

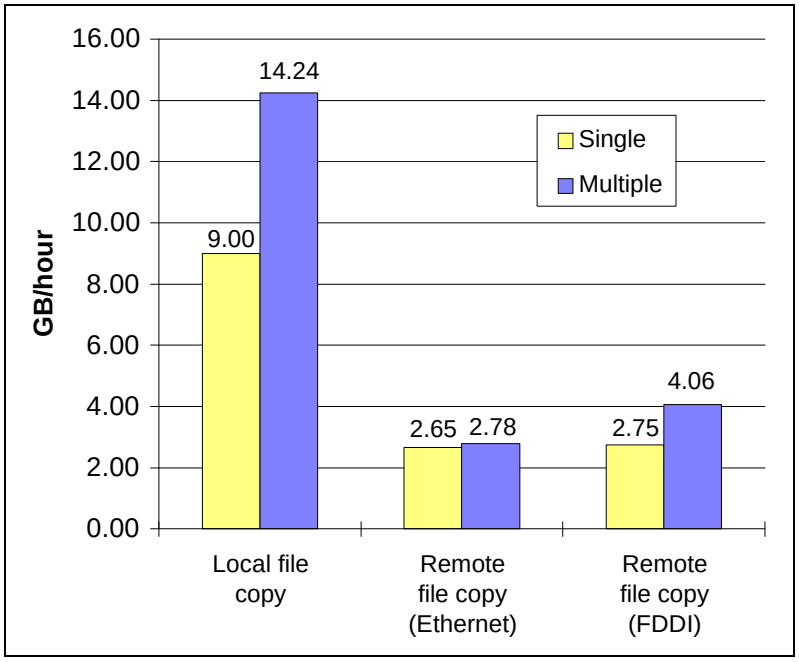

Figure 4-5. Off-line: single vs. multiple backup (local and remote disk)

The fastest method of backing up database files is to copy them locally to a dedicated disk volume. You can increase performance by copying two or more files simultaneously, as long as you preserve sequential disk I/O at both the source and destination points.

A faster network with remote disk backups, such as a dedicated network, a network with low or no traffic, high-speed network, high-speed network interface controllers, can improve the backup throughput.

Be aware that fault tolerance at the destination point can influence the throughput of disk backups. Distributed Data Guarding (RAID 5) will perform slightly slower than Disk Striping (RAID 0) or Disk Mirroring (RAID 1).

Backup with Verify

Backup with verify ensures the data writes to the backup device correctly and that the data is readable. You can achieve this by comparing data at the destination point with the original data at the source. This concept easily applies to both tape-based and disk-based backups. When performing tape backups with utilities like Windows NT Backup and Backup Exec for Windows NT, support is provided for backup with verify. If you are performing disk backups, you can always perform a file comparison.

The length of the verify operation varies with the type of data compression used and the type of data being manipulated. Compaq found that the verify operation usually takes the same amount of time as the backup in the following cases:

- No data compression used with tape backups.
- Data compression is used with tape backups, but the data is not compressible.
- File comparison is used.

However, if the operating system(s) can cache the files after they are copied or backed up, you might experience faster compare times.

The verify operation takes less time than the backup operation when you use data compression on compressible data. The duration depends entirely on the compression ratios. However, when using the data verification operation, the hardware data compression is faster than software data compression.

It is inevitable that the verify or compare operation prolongs the duration of the whole backup. However, remember that your backups are only as good as the data at the backup destination. If your data becomes corrupted or you cannot access your data, your backup is no good. By taking the time to verify the integrity of the backup image, you are protecting yourself against data loss.

Recovery

This section provides supplemental information on recovery methods found in sources such as the *Backup Exec for Windows NT System Manual* and *Microsoft Windows NT Operating System Manuals*. The information presented in this TechNote provides additional considerations and performance information not found in these publications.

Off-Line Recovery Considerations

The following are some functional considerations you should be aware of when restoring a database after a failure.

- You have to restore all database and transaction log files, including your *MASTER.DAT* file, to keep them all synchronized. SQL Server time stamps all files and detects an inconsistency between the files and the information stored in the master database.
- The backup software might precede the restore operation by cataloging the information on the tape, if the catalog information is lost. A tape catalog is information maintained by the backup software about each tape backup set and its contents and it is typically stored on the disk along with other files. Some software, including Backup Exec for Windows NT, provides you with an option to include the tape catalog on the tape, thus eliminating the need to regenerate a lost catalog by reading the entire contents of the tape.
- You can save time by restoring the full contents of a tape without having the backup software generate a tape catalog. The backup software will not, however, be able to show progress in the progress indicator, provided that one is available.
- To successfully restore the database and transaction log files, make sure all files are restored in their original place, for example, the same logical drive and directory. You can, however, successfully restore files on a logical drive with a different file system or on the same logical volume with different physical characteristics (such as more or less disk drives, different types of drives, different RAID, and so on).
- Some backup software, including Backup Exec for Windows NT, allows you to verify the files after restore. However, this prolongs the recovery process.

Database Restore Performance

The following chart illustrates database and transaction log restore time relative to its corresponding backup. Shorter bars represent shorter backup and restore times, thus better performance. These results do not include the following activities, which prolong the restore process:

- Hardware diagnosis, repairs, or replacements
- Operating system recovery (if necessary)
- SQL Server software restore, including creation of default *master* database (if necessary)
- Cataloging (if necessary or desired)

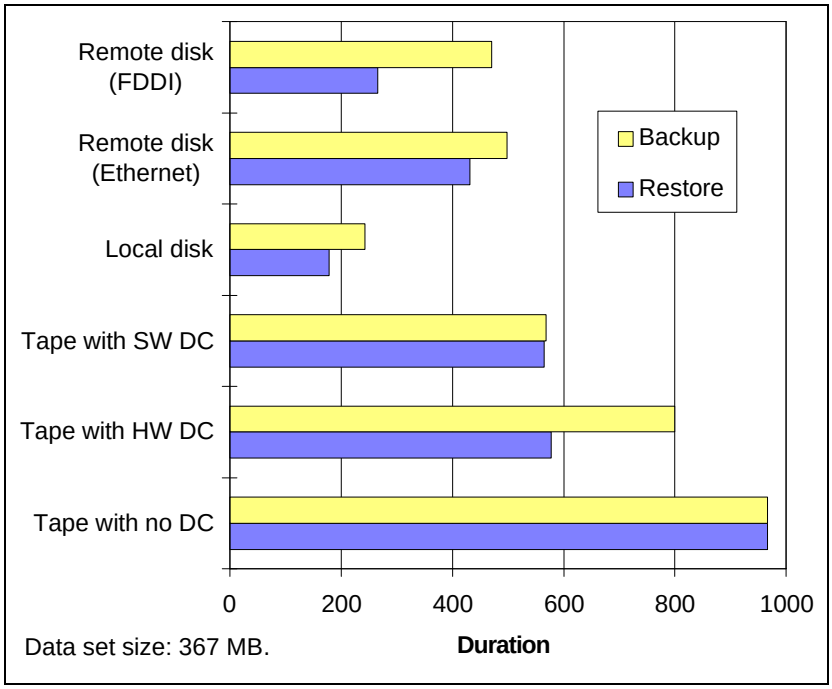

Figure 4-6. Off-line: backup vs. restore

Based on the results shown in Figure 4-6, you can make the following conclusions:

- The restore process typically takes less time than its corresponding backup process; in the worst case it will take equal amount of time.
- When you backup and restore to disk, factors such as the file system or the configuration of the disk subsystem at the source and at the destination influence the backup and restore durations.
- Local disk restores and restores over an FDDI network provide best performance results. The restore process improves when you employ multiple tape drive techniques - by restoring from two or more sources.

! **IMPORTANT:** The results in Figure 4-6 should not be compared with the results from the chart presented in Figure 5-6 in Chapter 5, "On-Line Backup and Recovery." These results were obtained with data sets of different sizes. Therefore, it would be misleading and incorrect to compare the two.

Chapter 5
On-Line Backup and Recovery

This chapter discusses the benefits and drawbacks of various dump options supported by SQL Server, including single and parallel dumps to disk and tape devices, local and remote dumps, and dumps under SQL Server activity.

To measure the maximum throughput that SQL Server can achieve during a dump for a given configuration, we performed a database dump to a NULL dump device in the Compaq Integration Lab. The throughput measured approximately 2.25 gigabytes/hour. You can increase this throughput slightly if you configure your system to perform multiple simultaneous dumps efficiently. For example, you can configure the databases and the disk subsystem where
no two databases residing on the same logical volume will be dumped at the same time. Such a configuration would preserve sequential reads on each logical volume during multiple simultaneous dumps, yielding maximum disk performance.

On-Line Backup Considerations

When considering an on-line backup, be aware of the following:

- SQL Server reads all pages from the disk devices, not from data cache. As a result, a larger SQL Server data cache will not improve the performance of database dumps.
- SQL Server reads all pages individually (2 kilobytes per read), buffers them and writes 60 kilobyte blocks to the dump device.
  - If you are dumping one database at a time, the speed of your backup operation will depend primarily on how fast your disk subsystem can perform single-threaded sequential reads of 2 kilobyte blocks.

- If you are loading one database at the time, the limiting factors on performance will be the backup device and the single-threaded sequential writes of 2 kilobyte blocks. You might increase the backup performance by using a faster disk subsystem (e.g., faster drives, faster controller, controller with read-ahead capabilities), or by dumping multiple databases simultaneously.

  NOTE: Multiple simultaneous database dumps require the same number of dump devices as the number of simultaneous databases being dumped. These dump devices, of course, can be disk dump devices or tape dump devices. Additionally, each tape dump device must have its own tape drive.

  Additional performance information on dumping multiple databases simultaneously is provided later in this TechNote.

- The database being backed up, as well as other databases, remains open for use. Subsequently, the backup activity and queries against the database server conflict with each other, resulting in a performance degradation of all databases during the entire backup operation.
- The disk dump device is overwritten with every dump. Therefore, you must rename the file or move it to another destination, such as tape, before you issue another SQL Server dump.
- You have an option to rewrite the contents or to append to existing data when SQL Server dumps to a tape.

Monitoring Transaction Log Usage

Preventing the transaction log from filling up is crucial for continuous operation of SQL Server. The transaction log will fill up when entries continue to accumulate in the log and no dumps have been performed during the same period of time. Eventually, all free pages in the log segment will fill up. Once the transaction log fills up, the corresponding database becomes unavailable for any insert, delete, or update operations as well as any other activity that generates a transaction log entry (e.g., dumps, checkpoints, object creations).

When the transaction log fills up and SQL Server is unable to record a checkpoint in the transaction log, you have to perform a dump of the transaction log, without making a backup copy of it, by using the following command:

```
DUMP TRAN <database> WITH NO_LOG
```

If this situation occurs, make sure to follow up with a full database backup. For more details and important considerations regarding the use of this command, refer to the *Microsoft SQL Server System Administrator's Guide*.

There are a number of ways you can monitor the transaction log usage and prevent the transaction log from filling up. Some of the suggested methods are described below:

- Monitoring Transaction Log Usage with DBCC

  To do this, issue the following command:

  ```
  DBCC checktable(syslogs)
  ```

  SQL Server updates the appropriate system tables and reports the following:

  - Total number of data pages in the *syslogs* table
  - Total number of rows in the *syslogs* table
  - Space used on the log segment in megabytes and in percent
  - Space free on the log segment in megabytes and in percent

  NOTE: The time to complete the *DBCC checktable(syslogs)* command depends on the size of your transaction log and space used.

- Monitoring Transaction Log Usage with the Performance Monitor

  To do this, you can:

  - Select *Object SQLServer-Log, Counter Log Space Used (%)* and *Instance < database(s)>*.
  - Use the *Alert* option to generate administrative alerts triggered on a predefined transaction log usage, then have the administrator dump the transaction log upon reception of this alert. Or you can automate this procedure by creating a batch file to dump the contents of the log.

- Truncating Transaction Log on Checkpoint

  You can issue the following command to have the transaction log automatically truncated at every occurrence of a checkpoint.

  ```
  sp_dboption <database>, 'trunc. log', true
  ```

  Use this option only if, after a catastrophic data loss, you do not need to recover transactions which followed the last full database backup.

Estimating Transaction Log Dump Size

The transaction log dump size can be estimated using the SQL Server Administrator utility by choosing the following sequence of options: *Manage Databases Manage Database Properties*. The output includes log space used in kilobytes.

You can also use the following command:

```
DBCC Checktable(syslogs)
```

The output of this command includes the total number of data pages in the *syslogs* table and the amount of space used on the log segment in megabytes.

For example, take the number of megabytes used on the log segment:

Data pages (syslog table) X 2048 byte (page size) = Approximately transaction log dump size

Always add 5 to 10% of the result to compensate for possible inaccuracy of the estimate.

Estimating Database Dump Size

The database dump size can be estimated with the SQL Server Administrator utility by choosing the following sequence of options:
Manage Databases
Manage Database Properties.

The SQL Server Administrator can estimate database space usage and report reserved space, data space used, log space used, index space used, unused reserve, and unused log space, all in kilobytes. The SQL Server Administrator can also add data space used, log space used, and index space used.

The Dump Database command causes SQL Server to include a dump of the transaction log, to recover transactions that were in progress at the time of the dump. Be sure to always include the size of the transaction log in your estimate. Also, add an extra 5 to 10% of the result to compensate for inaccuracy of the estimate.

Single Tape Device Dump

Figure 5-1 represents results of SQL Server dumps to the 525-MB ACA tape drive, the 5.0-GB DAT drive and the 4/16-GB TurboDAT drive. Note that the 525-MB ACA tape drive is slightly faster than the 5.0-GB DAT drive, but does not support hardware data compression and lacks the capacity required by
most environments. The 4/16-GB TurboDAT drive provides the highest backup throughputs.

Data compression increases capacity of the backup media and, in most cases, improves backup performance. However, the results with the 4/16-GB TurboDAT drive show the opposite. If the system cannot keep the tape drive streaming, data compression can result in lower backup throughput as compared to no data compression as explained earlier in Chapter 3 under "Backup and Recovery Implementation." Environments that are limited by SQL Server's single-threaded sequential reads of 2 kilobyte per request are prime candidates to create such a condition. The test environment created here at Compaq was not able to provide enough data to the tape drive to keep it streaming; therefore, in our test, data compression also showed lower throughput.

It is difficult to detect or predict whether or not the tape drive is streaming. The streaming threshold naturally varies with the compression ratios, throughput of the disk subsystem, speed of the tape drive, bus bandwidth and other factors. With some "louder" tape drives, such as the 525-MB ACA tape drive, you can audibly detect the streaming condition of the drive. Some DAT drives, such as the 4/16-GB TurboDAT drive, feature a status indicator, which displays *Writing*, *Ready*, and *Reading* conditions. However, experimentation and actual measurements seem to be the best indicators.

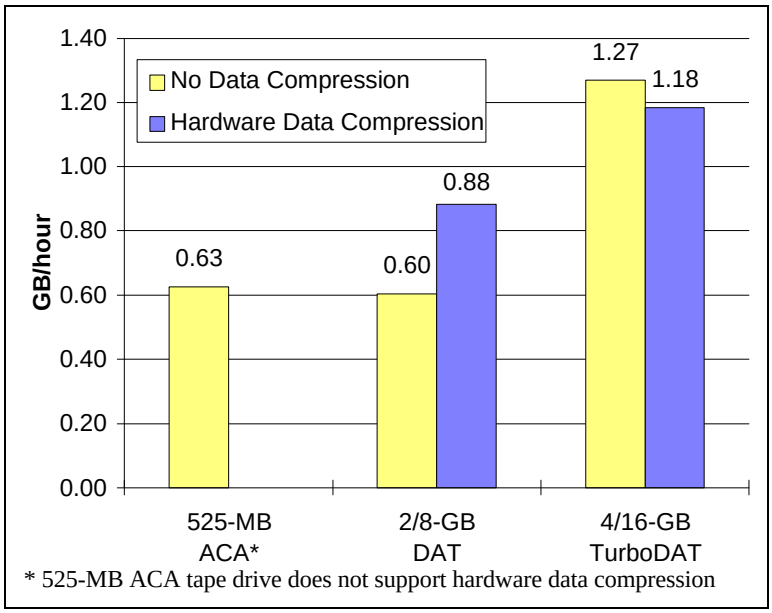

Figure 5-1. On-line: single tape dump with and without data compression

Multiple Tape Device Dumps

When employing multiple tape devices simultaneously to perform parallel (multiple) database dumps, be aware of the following considerations:

- You can perform one dump (transaction log or database) per tape device. To dump two databases simultaneously, you need to have two tape devices installed. These devices can coexist on the same controller. The same applies to load (recovery) operations.
- With SQL Server performing all reads in 2 kilobytes blocks, the performance gained by employing multiple tape devices are limited. All I/O to and from dump devices is performed in 60 kilobyte blocks. This limitation is explained later in this chapter.
- SQL Server upon startup allocates three internal 60 kilobyte buffers dedicated to dump and load operations. When all three buffers are consumed, as with three simultaneous dump or load processes, SQL Server does not allocate any additional buffers and all I/O to and from these additional dump devices occur in 2 kilobytes. As a result, there is no benefit in performing more than three simultaneous dumps or loads.

Figure 5-2 shows Compaq's results from dumping two very similar databases, single and multiple, using the 4/16-GB TurboDAT drives. With no data compression used, single dumps yielded 1.22 GB/hour throughput, whereas multiple dumps yielded 2.42 GB/hour throughput, approximately a 99% increase. With hardware data compression, multiple dumps yielded 2.02 GB/hour throughput, whereas single dumps yielded 1.11 GB/hour throughput, approximately an 82% increase. Multiple dumps would have provided only a fraction of the total possible throughput had the two databases been residing on the same logical volume.

Figure 5-2 also includes single and multiple dumps to the NULL devices for comparison. These dumps represent maximum attainable backup performance of the system and are limited by the performance of the drive subsystem.

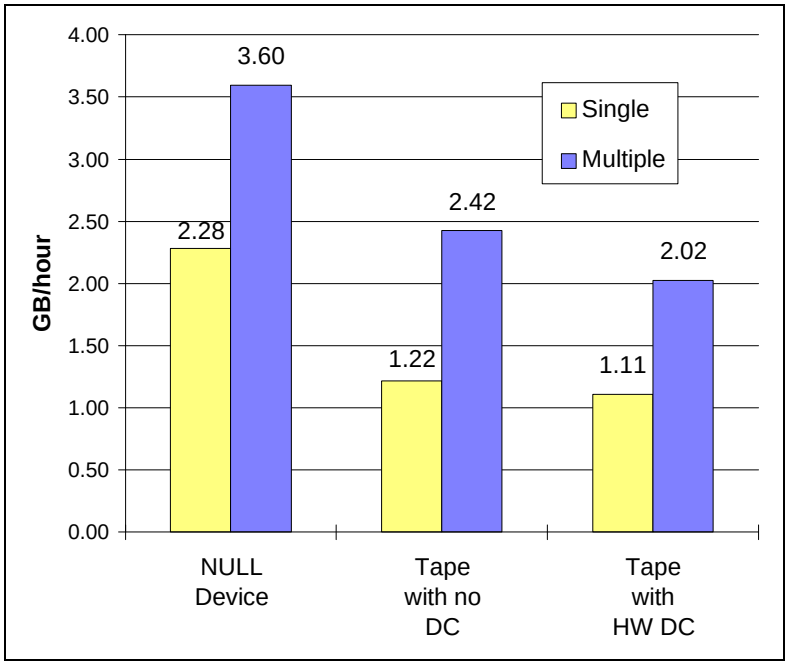

Figure 5-2. On-line: multiple tape dumps

To effectively employ multiple simultaneous dumps to a tape, you must:

- Put your databases on separate logical volumes to prevent SQL Server from causing head contention.

  Two databases residing on the same logical volume with striping (RAID 0) cause random I/O, yielding only partial throughput gains. The throughput gained, in this case, depend primarily on the disk subsystem speed and configuration. Compaq recommends experimentation and actual measurements to determine whether or not the throughput improvements justify any investment into an additional tape device.

  NOTE: Distributed data guarding (RAID 5) would also produce similar results.

- Dump only two databases residing on the same mirrored (RAID 1) volume.

  The SMART SCSI Array Controller can effectively keep the reads sequential on each set of mirrored drives. In other words, the controller uses one set of these mirrored drives to satisfy read requirements of the first dump and uses the other set of mirrored drives for the second dump.

Furthermore, the dump devices should be physically separated to eliminate contention. This is inherent with tape drives. You can only perform one dump per tape device. However, with local and remote dumps to disk, you can achieve the best performance if you have a separate disk volume or destination per dump.

Single Disk Device Dump

Figure 5-3 provides performance test results achieved with various single device dumps to a local and remote disk.

SQL Server dumps yielded the following results:

- When dumped to a dedicated local disk volume, throughput was about 2.15 GB/hour.
- When dumped to a remote server disk using a 10 Mb/s Ethernet network, throughput was about 1.35 GB/hour.
- When dumped to a remote server disk using a 100 Mb/s FDDI (Fiber Distributed Data Interface) network, throughput was about 1.88 GB/hour.

  NOTE: Compaq used SysKonnect SK-NET FDDI-FE EISA 32-bit FDDI LAN adapters and cabling to connect the database server and the backup server.

Figure 5-3 provides a comparison of the maximum theoretical throughput of SQL Server dumping to a NULL device, which was approximately 2.28 GB/hour in this particular configuration.

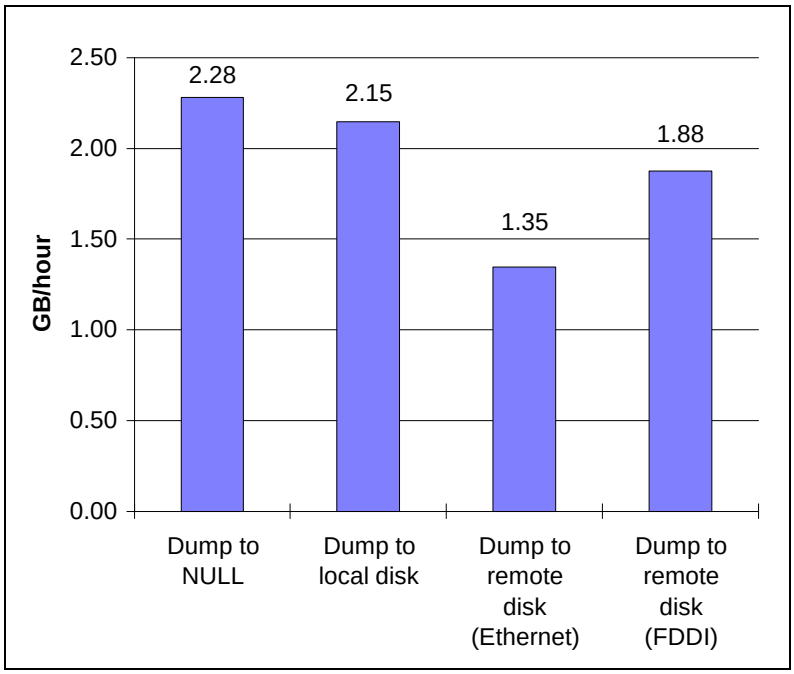

Figure 5-3. On-line: single disk dump

Multiple Disk Device Dumps

Figure 5-4 provides performance test results achieved with dumping two very similar databases, single and multiple, to a local and a remote disk.

Both databases resided on separate logical volumes to preserve sequential I/O. Multiple dumps would have provided only a fraction of the total possible throughput had the two databases been residing on the same logical volume. The same principle applies to the destination devices, which should be kept physically separate.

Figure 5-4 provides the comparison of single and multiple dumps to the NULL devices. SQL Server dumps yielded the following results:

- A single dump to a NULL device provided approximately 2.28 GB/hour of throughput.
- A multiple dump to two NULL devices provided approximately 3.60 GB/hour of throughput.

These dumps represent maximum attainable backup performance of the system and are limited by the performance of the drive subsystem.

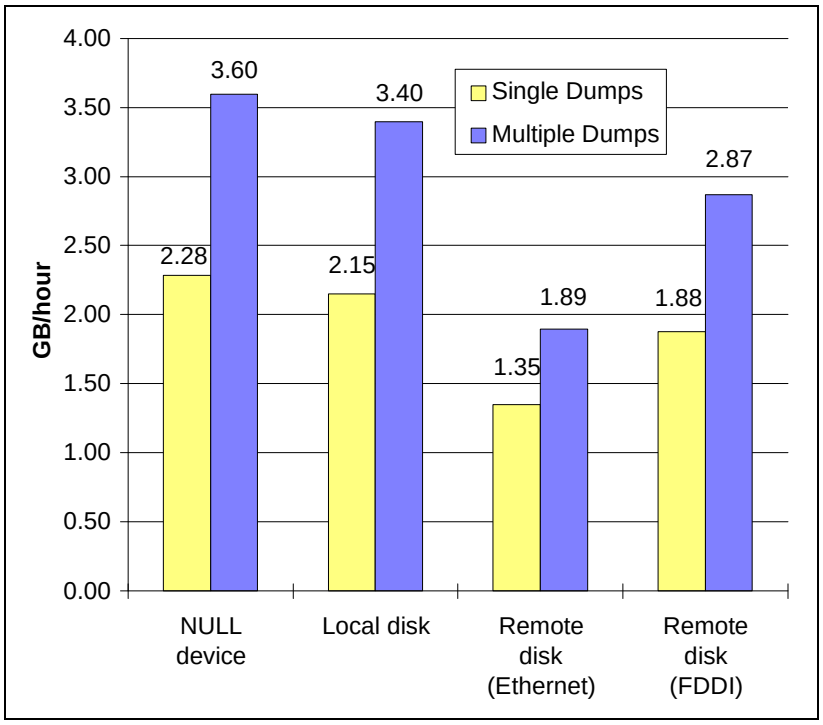

Figure 5-4. On-line: multiple disk dumps

SQL Server Dump with Database Activity

Performing on-line backups of a database during user activity requires certain considerations. This section discusses those considerations with regard to the dynamics as well as the performance of a SQL Server dump under user activity.

SQL Server Dump Dynamics Under User Activity

When you perform a dump of a database under user activity, several things happen with SQL Server. They are listed in sequential order as follows:

1. SQL Server executes queries sent by the users at a throughput of *x*-transactions-per-second.
2. SQL Server receives a dump database command.
3. SQL Server issues an immediate checkpoint to flush all *dirty* pages to the disk.
4. When the checkpoint finishes, SQL Server starts dumping the contents of the database (one 2 kilobyte page at a time) to the destination device (in the case of a tape drive, the tape is rewound first). SQL Server continues processing user queries (including updates); however, the throughput is reduced to *y*-transactions-per-second for the duration of the dump.
5. If a user transaction causes an update to a page not yet backed up, SQL Server backs up the old value of this page prior to changing its contents, then allows the transaction to complete. Any transaction updating a page already backed up, proceeds immediately. Thus, SQL Server ensures the backup image is consistent with the state of the database at the time the dump command was issued.
6. When SQL Server finishes the dump procedure, it resumes processing at the original throughput (*x*-transactions-per-second).

Figure 5-5 represents this sequence graphically.

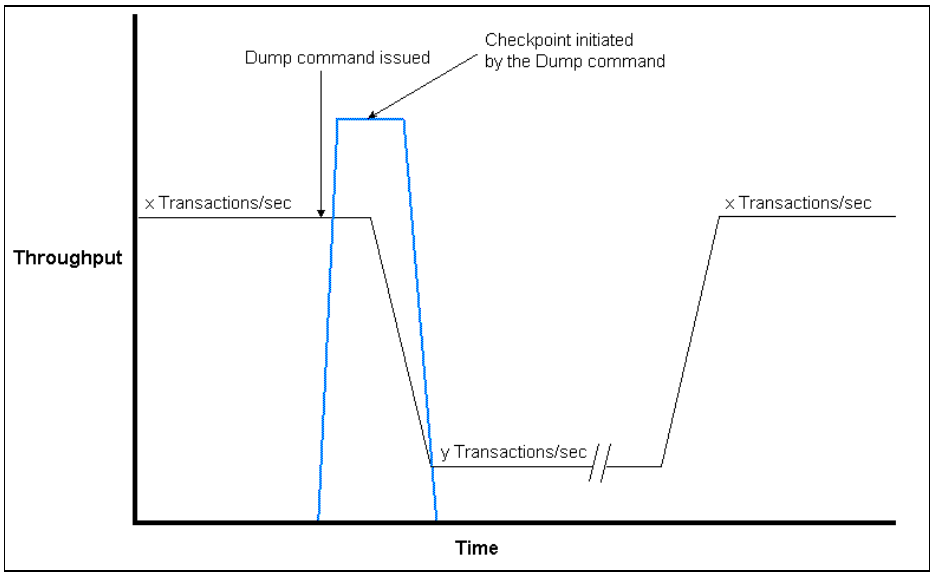

Figure 5-5. A SQL Server database dump under activity

Performance of a SQL Server Dump Under User Activity

Compaq engineered a lab test environment to measure the performance of SQL Server dumps under user activity. The load level at the server was controlled by introducing a *think time* or a slight delay for our artificial users between submitting transactions.

Compaq conducted the following tests, sequentially, to measure the performance of SQL Server dumps under user activity.

1. To establish a baseline, we measured the maximum throughput in transactions-per-second (or a 100% load at the server) by allowing users to submit transactions as fast as they could with no induced delays.

2. Next, we introduced delays to achieve 25% and 75% of the maximum transactions-per-second throughput rate (25% and 75% load or activity at the server).
3. Finally, we measured throughput (both user transactions-per-second and GB/hour backup throughput) during a database dump to a Compaq 4/16 GB TurboDAT drive, with the SQL Server symmetric multiprocessing performance option turned OFF and ON.

   For more information on the the SQL Server symmetric multiprocessing performance option, refer to the *Microsoft SQL Server Administrator's Guide* available from Microsoft.

Dump of an Active Database

This test scenario represented a dump of an active database, meaning a database under user activity, at 0%, 25%, and 75% load at the server, tested with and without the SQL Server symmetric multiprocessing performance option.

With a dump of an active database, as summarized in Tables 5-1 and 5-2, the user's transactions-per-second throughput had significantly decreased during the entire dump operation. The throughput measured was only 5 to 8% of the maximum throughput. Furthermore, the backup throughput has decreased from the maximum throughput by 10% with a 25% load at the server and by 20% with a 75% load at the server.

Table 5-1
Active Database Under Load -
Symmetric Multiprocessor Performance Option OFF

Load at the Server Before Dump	Load at the Server During Dump	Dump Throughput
No user load - system is idle	No user load - system is idle	1.55 GB/hour
25% of the maximum throughput	7% of the maximum throughput	1.39 GB/hour
75% of the maximum throughput	8% of the maximum throughput	1.25 GB/hour

Table 5-2
Active Database Under Load -
Symmetric Multiprocessor Performance Option ON

Load at the Server Before Dump	Load at the Server During Dump	Dump Throughput
No user load - system is idle	No user load - system is idle	1.55 GB/hour
25% of the maximum throughput	5.6% of the maximum throughput	1.40 GB/hour
75% of the maximum throughput	7% of the maximum throughput	1.29 GB/hour

As explained earlier in this TechNote, any transaction updating a page that has not already been backed up is placed on an internal queue. The transaction stays in the queue until SQL Server checks it and determines which page needs to be dumped next. Once the transaction is released and SQL Server dumps the page, the transaction is allowed to complete. This is believed to be the bottleneck observed during our testing.

Dump of an Inactive Database

This test scenario represented a dump of an inactive database, meaning no user activity but another database in the system is active at the same time, again yielding 25% and 75% load at the server.

With a dump of an inactive database, the user transactions-per-second throughput has not changed considerably. However, the backup performance drops significantly as shown in Table 5-3.

Table 5-3
Inactive Database Under Load

Load at the Server Before Dump	Dump Throughput
No user load - system is idle	1.55GB/hour
25% of the maximum throughput	0.73GB/hour
75% of the maximum throughput	0.30GB/hour

NOTE: For an inactive database as shown in Table 5-3, the state of the Dedicated Multiprocessing Performance option didn't seem to influence the backup throughput.

As represented in Table 5-3, the dump thread competes with other SQL Server threads processing user transactions against another database. Since bottlenecks are not imposed on any of the user transactions, SQL Server schedules them equally with the thread performing the dump operation. SQL Server can have as many worker threads as there are users (up to a certain user limit). Once this limit is reached, user connections are shared by a worker thread; therefore, the dump thread receives only a fraction of the execution time, resulting in low backup throughput.

Tables 5-4 and 5-5 summarize our results during a backup of an inactive database.

Table 5-4
Dump of Inactive Database Under Load -
Symmetric Multiprocessor Performance Option OFF

Load at the Server Before Dump	Load at the Server During Dump	Dump Throughput
No user load - system is idle	No user load - system is idle	1.55GB/hour
25% of the maximum throughput	23-25% of the maximum throughput	0.73GB/hour
75% of the maximum throughput	73-75% of the maximum throughput	0.30GB/hour

Table 5-5
Dump of Inactive Database Under Load -
Symmetric Multiprocessor Performance Option ON

Load at the Server Before Dump	Load at the Server During Dump	Dump Throughput
No user load - system is idle	No user load - system is idle	1.55GB/hour
25% of the maximum throughput	23-25% of the maximum throughput	0.72GB/hour
75% of the maximum throughput	73-75% of the maximum throughput	0.30GB/hour

From our results we can conclude that a SQL Server dump of an active database greatly impacts the user transactions-per-second throughput and user response time, whereas the impact on backup performance is relatively low. However, a SQL Server dump of an inactive database significantly affects the backup throughput, whereas the user transactions-per-second throughput degradation is insignificant.

Recovery

The *Microsoft SQL Server System Administrator's Guide* provides detailed information as well as examples of different recovery scenarios, such as:

- Recovering from media failure
- Recreating lost devices
- Recreating and reloading lost databases
- Restoring and reloading the master database

This section of the TechNote provides supplemental information, such as additional considerations and a performance analysis of the recovery process not found in the *Microsoft SQL Server System Administrator's Guide*.

On-Line Recovery Considerations

Following are some functional considerations you should be aware of when loading a database after a failure:

- Make sure you create a database equal to or larger than the original database size. You will not be able to load a database into a smaller frame structure, even if only a portion of the database is used.
- If you are reloading the database to another server, the code page and sort order must match the original installation. To move data between servers with different code pages and/or sort orders, use the BCP utility or the SQL Transfer utility.
- Create the database in the same way as the original one, with respect to its physical characteristics, such as device allocation, log space allocation, segment allocation, etc. The best method to ensure identical physical characteristics is to save the SQL scripts used to initially create your database(s) and rerun them prior to loading the database(s).
- When recovering a database from a database dump and a series of subsequent transaction log dumps, all dump images must be successfully loaded in sequence.

  If, for example, you load from a database dump and another user updates a record in this database before you have a chance to load the next transaction log dump, the transaction log load will fail. When this happens, you are forced to start over and reload from the last database dump. Obviously, the update transaction occurring after the first load will be lost. To prevent this from happening, start SQL Server in a single user mode.
- If the disk subsystem is the performance bottleneck within the system, the Compaq SMART SCSI Array Controller with the Array Accelerator enabled can improve performance of the recovery process.

  One of the Compaq tests performed with a remote database load across an FDDI network showed a performance improvement of 27% in load throughput with the Array Accelerator enabled, as compared to the Array Accelerator disabled.

Other important considerations can be found in the Microsoft SQL Server manuals and the *Microsoft SQL Server, Backup and Recovery Guidelines* document, available from Microsoft.

Database Load Performance

The purpose of this section is to provide information on the on-line database load performance, as compared to the database dump performance. Figure 5-6 shows database load times relative to database dump times, with shorter bars representing shorter dump and load times, thus providing better throughput. The results shown in Figure 5-6 do not include the following activities, which prolong the recovery process:

- Any hardware diagnosis, repairs or replacements
- Operating system recovery, if necessary
- SQL Server software recovery, including the master database, if necessary
- Any database or transaction log device recreation, if necessary
- Database creation

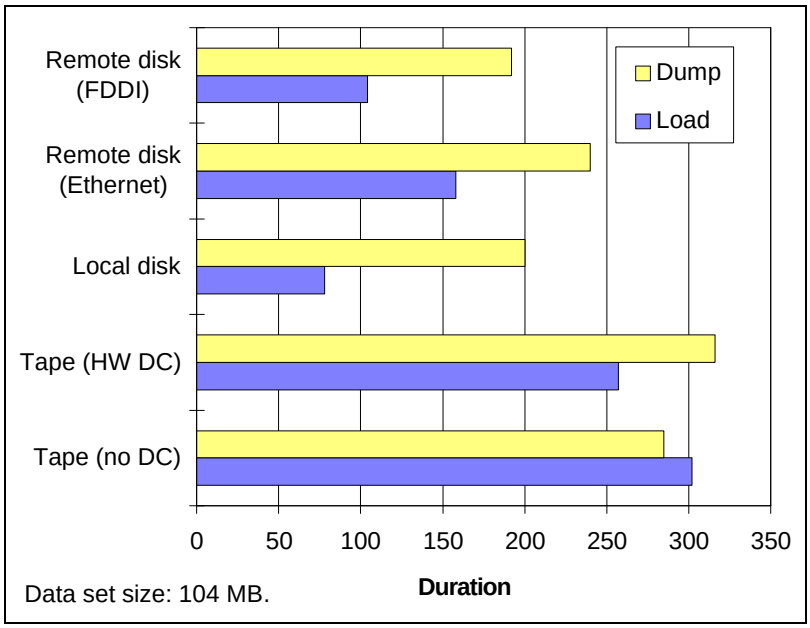

Figure 5-6. On-line: backup vs. restore

From the results shown in Figure 5-6, we can conclude the following:

- The actual load process typically takes less time than the dump process because faster devices are used, but usually is preceded by hardware repairs, software recovery, database creation, etc.
- Local disk and remote disk dumps as well as loads over an FDDI network provide best performance results.

! **IMPORTANT:** The results in Figure 5-6 should not be compared with the results from the chart presented in Figure 4-6 in Chapter 4, "Off-Line Backup and Recovery." These results were obtained with data sets of different sizes. Therefore, it would be misleading and incorrect to compare the two.

Chapter 6
Key Points for a Successful Backup and Recovery

This TechNote communicates relevant information on backup and recovery technology available with Microsoft SQL Server for Windows NT and Compaq server products. The key points presented in this TechNote are:

- Determine the type of backup strategy: off-line or on-line. If you can afford shutting SQL Server down to back up your transaction log(s) and database(s) within an acceptable time interval, investigate the options available with an off-line backup. Otherwise, implement an on-line backup.
- Use data compression to increase the tape cartridge's capacity and backup throughput. However, make sure that the system is able to stream the tape drive. If the system is not capable of streaming the tape drive and the capacity increase with data compression is of secondary concern, turn the data compression off to achieve better throughput.
- The nature of your data will determine the effectiveness of data compression. It will also directly impact your backup throughput.
- Use multiple backups simultaneously to increase your throughput. You will achieve the highest throughput improvements and greatest tape drive scalability if you can preserve sequential I/O. Otherwise, you will achieve only partial improvements.
- Preserve sequential I/O whenever possible. You can do this by accessing one file per logical volume at a time.
- Dedicated networks and high-speed networks, such as FDDI, can greatly improve remote backup throughput.
- Make sure your data is adequately protected by appropriate disk fault tolerance and a sound archiving strategy. Make sure you test this strategy thoroughly.

Each method discussed in this TechNote has its own benefits and limitations. A careful analysis of your requirements will help you determine the best method for your environment.

Additional Considerations

There are a number of additional factors not discussed in this TechNote that can influence the performance of backup and restore operations. Some of these factors are:

- Rewinding tape
- Changing tapes
- Searching for the end of the backup set, when applicable (appending new data)
- Writing additional data onto the tape, such as registry and/or catalog information
- Waiting on open files, when applicable
- Verify after backup
- Cataloging the tapes
- Backing up across the network when the network utilization is high (available network bandwidth)
- Diagnostics and replacement of the failed hardware
- Installation of the software, such as the operating system, SQL Server and/or the backup software, if necessary
- Creating and/or formatting of lost partitions, if necessary

For more information on these topics, refer to the *Microsoft SQL Server Administrator's Guide*, the *Microsoft Windows NT Operating System Manuals*, or the documentation manuals that came with your tape drive.

Glossary

Backup with Verify	A procedure that guarantees that the data writes to the backup device correctly and the data is readable; therefore, ensuring the integrity of your backup image.
Database Backup and Recovery	The ability to recover data by using a protection scheme which can consist of implementing database and transaction log backups along with fault tolerance and even replication.
data compression	A method used in backup and recovery to compress data either through the use of hardware or backup software.
disk backup	A method of archiving your data directly to a local or remote disk device.
Disk Mirroring	A fault tolerance method that uses 50 percent of drive storage capacity to increase data reliability by storing a duplicate of all user data. Half the drives in the array are duplicated or “mirrored” by the other half.
Disk Striping	No fault tolerance method used.
Distributed Data Guarding	A fault tolerance method that “stripes” data and parity across all the drives in the configuration to ensure the uninterrupted availability of uncorrupted data. It allocates a small percentage of the drive array storage to protect users from data loss.
dump	The SQL Server backup operation.

Ethernet	A local area network cabling and signaling scheme capable of linking up to 1024 nodes in a bus network. Ethernet provides a data transfer rate of 10 megabits per second.
FDDI	See Fiber Distributed Data Interface.
Fault Tolerance	Protection against loss of data due to a hardware or software component failure with methods such as drive mirroring, data guarding, and distributed data guarding.
Fiber Distributed Data Interface (FDDI)	A dedicated high-speed networking technology that has technical features like fault tolerance which are preferred for mission-critical systems.
Instance Recovery	An automatic data protection and recovery method where the server has the ability to preserve the effects of all committed transactions and ensures the database is in a consistent state after a recovery from any failure.
load	The SQL Server restore operation.
local backup	A method of archiving your data to a storage device that is physically attached to the database server.
Media Retention	A SQL Server option, configurable through *sp_configure*, that allows you to prevent overwriting an existing backup too soon.
NIC	See Network Interface Controller.
Network Interface Controller (NIC)	A controller that acts as the communications interface between a personal computer and a network.

off-line backup	A backup procedure where the database administrator must shut down SQL Server, closing all database and transaction log files before performing a file-based backup of all SQL Server devices.
on-line backup	A backup procedure which uses the SQL Server backup facility to backup the contents of the transaction log, the database, or both, while SQL Server remains running.
parallel backups	A multiple tape backup.
RAID 0	See Disk Striping.
RAID 1	See Disk Mirroring.
RAID 5	See Distributed Data Guarding.
random patterns	A data set with zero to very low percentage of repeated data and unused space which produces low or no data compression.
remote backup	A method of archiving your data to a storage device located on another server.
repeated patterns	A data set with a high percentage of repeated data and unused space which produces relatively high compression ratios; therefore, high throughput.
tape backup	A method of archiving your data on a tape device that is physically attached to the database server.
tape catalog	Information maintained by backup software about each tape backup set and its contents. It is typically stored on the disk along with other files.

transaction log	SQL Server's write-ahead log, where SQL Server records changes before it modifies the data pages.
unallocated space	Represents unused space like an empty database that produces very high compression ratios and very high backup throughput.

Index

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

V