

NETWORK PROFESSIONAL'S LIBRARY

OSBORNE

"Storage networking represents the most promising opportunities in IT today for those equipped to embrace it. This book contains a complete overview of the technologies and architectures needed to successfully plan and implement storage networks."

—Dr. Paul Borrill, Chairman Emeritus, Storage Networking Industry Association



SNIA
Storage Networking Industry Association

Building Storage Networks

- In-depth coverage of network storage technologies, such as network backup, RAID, mirroring, SCSI, volume management, caching, and more
- Explains SAN and NAS applications, including Fibre Channel, NFS, CIFS, fabrics, loops, SAN file systems, storage pooling, and serverless backup
- Explores new technologies, including Internet storage and the InfiniBand™ serial I/O channels

MARC FARLEY
STORAGE TECHNOLOGY CONSULTANT

PREFACE BY **THORNTON MAY**, CAMBRIDGE
TECHNOLOGY PARTNERS
FOREWORD BY **JOHN MCARTHUR**, IDC
COMPUTERWORLD
COMPUTERWORLD.COM

Petitioner has added a romanettes to page (i)-(vi). Otherwise, it leaves the original page numbering.

DROPBOX, INC., ET AL.
EXHIBIT 1007

(i)



Building Storage Networks

Marc Farley

Osborne/McGraw-Hill

Berkeley New York St. Louis San Francisco
Auckland Bogotá Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations or book distributors outside the U.S.A., or to arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact Osborne/McGraw-Hill at the above address.

Building Storage Networks

Copyright © 2000 by The McGraw-Hill Companies. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

4567890 DOC DOC 019876543210

ISBN 0-07-212050-9

Publisher

Brandon A. Nordin

Associate Publisher and

Editor-in-Chief

Scott Rogers

Senior Acquisitions Editor

Gareth Hancock

Project Editor

Carolyn Welch

Acquisitions Coordinator

Tara Davis

Technical Editor

Paul Massiglia

Copy Editors

Kathy Hashimoto

Luann Rouff

Proofreader

Mike McGee

Indexer

Valerie Perry

Computer Designers

Jani Beckwith

Jean Butterfield

Illustrators

Lyssa Sieben-Wald

Beth Young

Brian Wells

Robert Hansen

Series Design

Peter F. Hancik

This book was composed with Corel VENTURA™ Publisher.

Information has been obtained by Osborne/McGraw-Hill from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Osborne/McGraw-Hill, or others, Osborne/McGraw-Hill does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from use of such information.

**This book is dedicated
to my wife Gretchen
and my children Laura and Bret.
May we continue to share
our experiences, talents,
and blessings for
a long, long time.**

ABOUT THE AUTHOR

Marc Farley started working with network storage technology in 1991 at Palindrome Corporation as a systems engineer and has been in the network storage field ever since working in business development, product management, and strategic marketing at Hitachi, Crossroads Systems, and ConvergeNet. He is an independent consultant to the storage networking industry providing marketing and educational services for companies developing SAN and NAS technologies.

AT A GLANCE

Part I Introduction to Network Storage

- ▼ 1 The Network Storage Landscape 3
- ▼ 2 Piecing Together a Path for Storage I/O . . . 35
- ▼ 3 Charting the I/O Path from Application
to Storage in Client/Server Networks 73

Part II Established Applications in Network Storage

- ▼ 4 Disk Mirroring for Data Protection,
Availability, and Performance 105
 - ▼ 5 Enhancing Performance with Caching
and other Techniques 141
 - ▼ 6 Boosting Availability and Performance
with RAID and Disk Subsystems 175
 - ▼ 7 Network Backup: The Foundation of
Storage Management 229
-

Part III Network Access to Storage

- ▼ 8 Comparing SAN and NAS Technologies . . . 285
- ▼ 9 SAN Designs 301
- ▼ 10 Building Better Backup Systems with SANs . . 349
- ▼ 11 Fibre Channel Networking Technology
for Storage Networks 379

**Part IV Network Attached Storage Devices and Other
Advanced Topics**

- ▼ 12 Plug and Play Storage with NAS 421
- ▼ 13 Distributing Intelligence and Data Sharing
in the SAN 453
- ▼ 14 Accessing Data on S/390 MVS Mainframes
from a SAN 483
- ▼ 15 Clustered Storage and I/O Directions 507
- ▼ 16 Storing and Retrieving Data on the Internet . . 529

CONTENTS

Foreword	xix
Acknowledgments	xxv
Introduction.	xxvii
Preface	xxix

Part I

Introduction to Network Storage

▼ 1 The Network Storage Landscape	3
The Changing Role of Data As a Business Asset	4
Establishing the Relative Worth of Data	5
A Flood of Fluid Data	7
An Overview of Traditional Storage Options	13
Open Systems Client/Server Storage	13
Data Storage on Open Systems Servers	16
Data on Mainframe Systems	19

SCSI: The Reigning I/O Channel Technology for Open Systems Servers	21
I/O Channel	21
The Origins of SCSI	22
SCSI Limitations	22
New Storage Connectivity Options That Extend the I/O Channel	24
Network Attached Storage	25
Fibre Channel	27
Storage Area Networks	28
Summary	32
▼ 2 Piecing Together a Path for Storage I/O	35
Identifying the Physical I/O Building Blocks	36
System Memory Bus	36
The Host I/O Bus	38
Host I/O Controllers	44
I/O Bus and Network Connectivity	50
Storage Devices and Subsystems	53
Media	57
SCSI Bus Integration Variables	57
Connecting to the SCSI Bus	58
The Logical Components of the I/O Channel	63
Application Software	63
Operating System	63
File Systems and Database Systems	66
Volume Managers	67
Device Drivers	69
Combining Hardware and Software into an I/O "Stack"	70
Summary	70
▼ 3 Charting the I/O Path from Application to Storage in Client/Server Networks	73
The I/O Path for Local Storage	74
Local I/O on Workstations	74
A Detailed Examination of the Local I/O Path	74
Local I/O on Network Servers	85
Theme and Variations on the Local I/O Path	88
Client/Server I/O	90
I/O Redirection	90
Server-Side Network I/O	93
Implementing Device Virtualization in the I/O Path	98
Locations in the I/O Path for Virtualization	99
Summary	101

Part II

Established Applications in Network Storage

▼ 4	Disk Mirroring for Data Protection, Availability, and Performance	105
	Safeguarding Your Data with Disk Mirroring	106
	The Mechanics of Disk Mirroring	107
	Selecting a Disk Mirroring Solution	112
	Software Mirroring	112
	Mirroring in an External Disk Subsystem	114
	Host I/O Controller Mirroring	116
	Increasing Data Protection with Dual I/O Controllers	117
	Maximizing Performance with Disk Mirroring	118
	Avoiding Performance Problems when Mixing Tape Drives with Disk Mirrors	119
	Increasing I/O Performance with Disk Mirroring	122
	Planning Your Mirrored Configuration	126
	Mirroring External Disk Subsystems	127
	Taking a Snapshot of Your Data	128
	Mirroring in a Campus Environment	131
	Mirroring over a WAN	134
	Comparing Storage and Network Transmission Speeds	137
	The "Big Pipe," "Little Pipe" Problem	139
	Summary	140
▼ 5	Enhancing Performance with Caching and other Techniques	141
	Much Ado about Caching	142
	Cache Hits and Cache Misses	143
	The Difference Between Caches and Buffers	144
	Performance Expectations of Disk Caching	145
	Algorithms Used in Disk Caching	146
	Disk Cache Components	154
	Optimizing Disk Cache by Location	155
	Independent Caches in Storage Networks	161
	Configuring Cache	162
	Solid State Disks	164
	Advantages and Uses of SSDs	165
	Limitations of SSDs	165
	Nonvolatile SSDs	165
	Tagged Command Queuing	166
	Using Intelligent Processors in Disk Drives	167
	Effectiveness of Tagged Command Queuing	168

Importance of the I/O Path in System Performance	168
Balancing Processor and I/O Performance	168
Summary	173
▼ 6 Boosting Availability and Performance with RAID and Disk Subsystems . . .	175
Three Reasons for RAID	176
RAID Capacity and Manageability	177
Scaling Up with Capacity	178
Management Advantages of RAID	179
RAID for Performance	180
Performance Through Striping	180
RAID Reliability and Availability Advantages	184
Data Reliability Through Redundancy	184
Power Protection	185
Hot Spares and Hot Swapping	190
The Internal I/O Paths in a RAID Subsystem	192
Organizing Data in RAID Arrays: Extents, Strips, and Stripes . . .	195
Array Management Software	196
Applying Parity to Striped Data	202
Creating Check Data with the XOR Function	202
Parity in Parallel Access RAID	205
Parity in Independent Access RAID	205
A Comparison of RAID Levels	210
The RAID Advisory Board	210
RAID Level 0: Striping	211
RAID Level 1: Mirroring	211
RAID Level 2: Parallel Access with Specialized Disks	211
RAID Level 3: Synchronous Access with a Dedicated Parity Disk	213
RAID Level 4: Independent Access with a Dedicated Parity Disk	213
RAID Level 5: Independent Access with Distributed Parity	215
RAID Level 6: Independent Access with Double Parity . . .	217
Combining RAID Levels	220
Objectives of Multilayered RAID Arrays	221
Striping and Mirroring Combined—RAID 0+1/RAID 10 . . .	221
Locating RAID Functionality in the I/O Path	224
Setting Fault Tolerant Standards: the RAID Advisory Board (RAB)	227
Summary	228

▼ 7	Network Backup: The Foundation of Storage Management	229
	Analyzing the Pieces of Network Backup	230
	Hardware Components	230
	Media Components	237
	Software Components	241
	Working with Backup	250
	Types of Backup Operations	251
	Backing Up Live Systems	253
	The Special Case of Image Backup	256
	Restoring Data	257
	Integrating Restore with File Systems and Databases	257
	Types of Restore Operations	257
	Importance of Media Management to Restores	260
	Backing Up and Restoring Security Data	261
	User and File System Security Components	261
	Tape Rotation	266
	The Necessity of Tape Rotation	267
	Commonly Used Tape Rotation Models	268
	Problem Areas for Backup and Recovery	273
	Six Reasons Why Backups Fail	273
	Six Reasons Why Restores Fail	275
	Challenges of Managing Backup	277
	Measurable Problems in Backup	277
	Summary	280

Part III

Network Access to Storage

▼ 8	Comparing SAN and NAS Technologies	285
	I/O Unbound	286
	Two Approaches: SAN and NAS	287
	Differentiating NAS from SANs	289
	Cleaving at the File System	289
	NAS and SAN Soft Machines	290
	Hardware Divisions Between NAS and SAN	291
	NAS and SAN Protocol Differences	292
	Multiple Protocols in a Storage Network	295
	Visualizing NAS and SAN Implementations	298
	Moving Forward: Use of the Term "SAN"	300
	Summary	300

▼ 9	SAN Designs	301
	Open Systems Storage Retrospective	302
	Bus-Attached, Subservient Storage	303
	Scalability Limitations of Subservient Storage	304
	SAN Architectures and Topologies	316
	Point-to-Point SANs	316
	Switched Fabric SANs	316
	Loop SANs	317
	Placing SANs in the I/O Path	318
	Variations of Basic SAN Configurations	322
	Creating Solutions and Relieving Systems Management	
	Burdens with SANs	327
	Major Applications of SANs	328
	Backup	328
	Storage Pooling	328
	Data Sharing	331
	I/O Pathing	332
	Data Movers	342
	Remote Storage	343
	The Challenges of Building SANs	345
	An Anarchy of Access to Shared SAN Storage	345
	Compatibility	345
	Summary	346
▼ 10	Building Better Backup Systems with SANs	349
	Reviewing the Problems with SCSI Bus-Attached Backups	350
	Challenges of Network Backup	350
	Backup over the SAN	353
	Centralized Backup over Fibre Channel	353
	Three Stages of SAN Backup Development	355
	First Stage: LAN-Free, Virtual Private Backup Networks	355
	Stage 2: Integrating Media and Devices	362
	Stage 3: Serverless Backup	366
	Combining Serverless with Integrated SAN Backup	371
	Subsystem-Based Backup	372
	A Brief Overview of Subsystem-Based SAN Backup	372
	Virtual Tape	374
	Virtual Tape and Physical Tapes	375
	Future of Backup in Storage Networks	377
	Summary	377

▼ 11 Fibre Channel Networking Technology for Storage Networks	379
Distinguishing Fibre Channel Networks from SANs	380
Revisiting SANs	381
A Brief History of Fibre Channel	381
The Physical Fibre Channel Network	382
The Role of Ports in Fibre Channel Networks	382
Cabling	383
The Logical Side of Fibre Channel	386
The Structure of Fibre Channel Networks	387
Fibre Channel Network Topologies	387
The Syntax of Fibre Channel Communications	392
Nodes and Ports	394
Class of Service	404
Class 1 Service	404
Class 2 Service	405
Class 3 Service	406
Addresses and Naming in Fibre Channel	408
Network Name and Address Elements in Fibre Channel	408
Loop Initialization	410
Zoning in Fibre Channel Switches	412
Protocol Layers and Serial SCSI in Fibre Channel	413
FCP Mapping Independence	413
Storage I/O on Ethernet	414
Summary	416

Part IV

Network Attached Storage Devices and Other Advanced Topics

▼ 12 Plug and Play Storage with NAS	421
The Roots of Network Attached Storage	422
Integrating Storage as an Appliance	423
Characteristics of NAS Products	423
Applications of NAS Technology	427
Implementations of NAS Appliances	430
Potential Disadvantages of NAS Products	434
Network File Systems and Protocols	441
The Role of Desktop Client Systems	443
CIFS Emulation for NFS Servers	446

A New Spin on Network Storage: NASD and OBS	449
OBS	449
The NASD Project	450
Summary	452
▼ 13 Distributing Intelligence and Data Sharing in the SAN	453
Packing Power in Intelligent Back-End Storage Subsystems	454
Storage Pooling and Volume Management	455
Sharing Data in Intelligent Back-End Storage Subsystems	456
The Value of Data Sharing	459
Managing a Single Source of Data	459
Space Allocation for Data Sharing	460
Resolving Locks and Semantic Differences	464
Lock Management for Network Storage	465
Caching Shared Data	466
Cache Coherence	467
Installable File Systems	470
File Systems for Network Storage	471
Summary	481
▼ 14 Accessing Data on S/390 MVS Mainframes from a SAN	483
The Big Picture of I/O on Big Iron	484
Mainframe I/O Processing	484
ESCON: the S/390 Storage Network	489
From ESCON to FICON	490
Open Systems Connectivity to Mainframe Storage Networks	492
Exchanging Data Across Platforms	493
Data Transfers Between Mainframes and Open Systems	497
Examining ESCON/SAN Data Transfers	498
The Future of Sharing Data Across Platforms	503
Summary	505
▼ 15 Clustered Storage and I/O Directions	507
Clustering Servers	508
A SAN Is a SAN Is a SAN, Isn't It?	508
Reasons for Clustering	509
Characteristics of Clusters	510
Inter-Cluster Communications	516
Using SAN Technology for Cluster Communications	517

An Introduction to the InfiniBand I/O Channel	521
Expectations for InfiniBand Technology	522
Integrating InfiniBand with Storage Networks	525
Summary	527
▼ 16 Storing and Retrieving Data on the Internet	529
Internet Infrastructure Storage	530
Scaling to Server-Side Requirements	530
Client-Side Web Caching	539
Storing Data on the Internet	543
Five Reasons for Internet Storage Growth	544
Storage, Storage Everywhere	546
Internet Storage Without Browsing	547
Web-Based Storage	549
Comparing Internet File Transfer Technologies	553
Internet Backup Software and Services	556
Integrating the Internet with Storage and I/O	560
Possibilities for Internet Storage Integration	560
Enabling Technologies for Internet Storage Integration	563
Summary	565
Index	567

SUMMARY

Disk subsystems and RAID are the primary vehicles for creating server-class storage. As storage capacities continue to increase with continued pressure to improve performance and achieve 24×7 availability, it is likely that disk and RAID subsystems will be the mainstays of server storage for many years to come.

Today's RAID implementations are based on striping, mirroring, and XOR-based parity. As the cost of disk drives continues to decrease, it is possible that the cost arguments for using parity RAID will become less interesting and that mirroring-based solutions will become the accepted redundancy technique. While the discussions of XOR computations in this chapter may be interesting, there is little doubt that they add a fair amount of complexity to the picture, especially for error recovery.

It is hoped that the characteristics of the various RAID levels presented will provide insight for readers looking to match storage system performance and protection with the requirements of their applications. The subject of tuning RAID subsystems by adjusting stripe depth was not included in this book, as it is a topic that is fairly implementation- and application-specific. But the reader should be aware that beyond matching the best fitting RAID level to an application, the adjustments in stripe depth may be the next most important parameter to optimizing performance.

RAID is used in all server storage environments, including traditional SCSI storage, networked attached storage, and storage area networks. As storage networking becomes more popular over time, people will appreciate how the I/O path becomes more accessible and flexible, which makes it easier to change and restructure storage components in the path without causing major interruptions in daily processing.

Most of the remainder of this book deals with the networking of storage and what that means to systems professionals. However, in the next chapter we will explore one more fundamental storage technology—backup and recovery.

CHAPTER 7



Network Backup: The Foundation of Storage Management

With so much data being placed on open systems servers, it is important to understand how this data is protected from disasters. Unfortunately, there are many threats to data that cannot be easily prevented that will destroy data and the systems that access it. If that happens, the only viable solution is to restore the systems and data. Several techniques for preserving data availability have already been discussed in previous chapters. Disk mirroring enables data to be recovered from another disk or subsystem. RAID enables data to be accessed when a single disk drive fails.

These techniques protect against equipment failures, but they do not protect against defective system users. In fact, users constitute the single biggest threat to data by deleting, changing, and otherwise abusing data. In many cases, it is easier to restore a previous version of a file than it is to work with a file that has been mangled by someone who wanted to "try something different." Backup is a well-known safeguard against all kinds of unexpected data loss, and it is generally considered irresponsible systems management to not perform daily backups.

The realization that data might need to be restored someday is not necessarily a very profound thought. A lot of backup hardware, software, and tapes have been sold over the years. Unfortunately, the challenges of backup and recovery in network environments are great, and many customers experience chronic difficulties in managing backup for their open systems servers. This is somewhat remarkable, considering that network backup is probably the second most prevalent network storage application behind file serving.

This chapter examines the technologies used in network backup today and some of the reasons why backup is a problem for many organizations. Disaster recovery is also covered as an important related topic. The discussions in this chapter set the stage for Chapter 10, which examines ways that storage networking technology can be applied to solving the difficult problems of network server backup and recovery.

ANALYZING THE PIECES OF NETWORK BACKUP

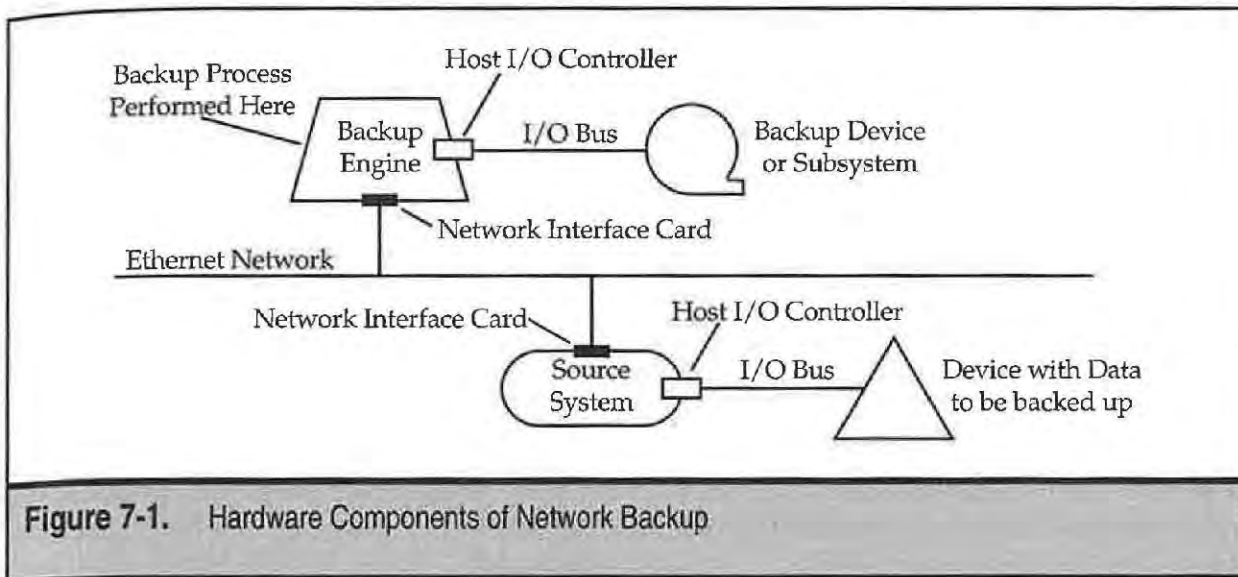
There are several different technologies used in network backup and recovery. In general, they can be broken down into three classes:

- ▼ Hardware
- Media
- ▲ Software

We'll look at each of these components in the sections that follow.

Hardware Components

We'll first describe the physical hardware that is commonly used in network backup (see Figure 7-1). This section will introduce terminology that is used throughout this chapter and the rest of the book.



The various hardware components are:

- ▼ Backup engine system
- Source system
- Network and network interfaces
- ▲ Backup devices and subsystems

Backup Engine System

The *backup engine system* is the computer where the primary controlling software for backup runs. This is typically where all of the management functions reside, including device operations, backup scheduling, media management, database/catalog processing, and error handling. The backup engine system is characterized by the type and capabilities of the processors, the system memory available and its internal system bus architecture.

The engine system also has a storage I/O bus, such as a parallel SCSI bus for conducting data transfers to and from tape drives and subsystems to the backup engine. This I/O bus could be the same bus used for attaching disk drives to the backup engine system, but tape drives are often isolated on a separate bus to allow optimal disk performance, as discussed in Chapter 4. Of course, where there is an I/O bus, there is also a host I/O controller also. For tape drives used with network backup installations in 1999 this is almost always a SCSI host I/O controller.

Source System

The *source system* is the computer having data that needs to be backed up. Sometimes this is also referred to as the *client system*. However, using client/server terminology for backup can be confusing, as backup clients are often network servers. Therefore, this chapter attempts to avoid using the word client when discussing network backup. The

term "target system" is also used sometimes to identify the source system. However, the word "target" is part of the SCSI syntax that identifies a device or subsystem on the bus, including the backup devices or subsystems where backup data is being written to. Therefore, to avoid confusion, we'll use source system, or simply source, to refer to the systems being **backed up**.

That said, the term *target system* is used in this chapter to identify the system where data is being restored to by the network backup system.

Source systems traditionally store their data on locally attached devices and subsystems, connected through a host I/O controller of some kind. The characteristics of the local storage and I/O buses on source systems can have a profound cumulative effect on network backup.

In 1999, most network backup is managed over ethernet networks. FDDI is also used, sometimes as a dedicated backup network. Relatively good performance can be achieved over ethernet using fast ethernet, also called 100BaseT. Performance can be further enhanced by network segments that are dedicated to backing up traffic. These segments provide optimal bandwidth and isolate backup traffic from normal production network traffic. The prevailing network protocols used are TCP/IP for Unix and Windows NT environments and IPX/SPX for NetWare environments. Where there are ethernet networks, there are ethernet ports on systems. Ethernet ports can be provided by ethernet network interface cards (NICs), as well as by ethernet processing chips that can be integrated into system boards. There are many other implementation options possible, such as using multiple ports for parallel data transfers or redundant links.

Backup Device and Subsystem Technology

The foundation of most backup and recovery systems is a tape device of some kind. As mentioned previously, sometimes these devices are referred to as *target* devices. This can be confusing as when talking about backup and restore operations. For example, the term target system is used in this chapter to identify the system on the network where data is being restored. In that case, there can be both a *target device* on the backup engine system where the data is being read from as well as a *target system* where data is being restored to. In general, the term target has not been used much to avoid confusion in this book. However, the reader should be aware that the terms are often used in practice.

Some of the technology variants used in tape products are discussed in the following section. The most common tape technologies in use for network backup in 1999 are DLT and 8mm. These technologies are likely to be applicable to the network backup market for several years, as they both have technology road maps showing regular increases of capacity and performance every two years.

Tape devices have a mode of data transfers referred to as streaming. *Streaming* refers to running at full speed with the drive's buffers being continuously filled by the host I/O controller as data is being written to tape. Streaming speed for tape drives is much faster than nonstreaming speeds where the tape mechanism must be continually stopped and restarted. This slows down the overall process and causes extra wear on the tape and mechanism. Therefore, it is desirable to structure a backup system that can stream a backup drive to attain optimal performance.

Tape heads are the tiny devices that recognize fluctuations in the magnetic signals that occur as the head and tape move in relationship with each other. In linear tape technology, the tape is passed over a more or less stationary head in a perpendicular orientation, similar to a music cassette tape player. Helical scan tape technology uses a different geometry, where the tape head is situated in a high-speed rotating drum that moves diagonally across the tape surface.

PHYSICAL TAPE FORMAT Between the tape head and the I/O external interface of the device are several electronic circuits, including the read and write channels. Heads are usually optimized and dedicated to either writing or reading. It is possible for heads to share channel circuitry, however multiple heads work best if they each have their own channels to avoid interference. Therefore, tape drives with multiple read and write heads will typically also have multiple channels, which increases the complexity of the transmission circuitry, the overall part count, and the cost of the drive.

Tape drives and cartridges are typically developed together as inseparable technologies. New tape drives with new designs usually have larger capacities and faster performance than previous tape drive designs. Part of the design of the drive/cartridge combination is its media format, or the specification of the physical and electrical characteristics of how data is written to the media.

The physical format of the tape drive specifies the size of a standard unit of data, its orientation on the tape, the location of tracking information, error correction data, and the position and frequency of location markings for locating data on the tape. Design changes to heads in tape drive products often alter the geometry and configuration of the heads as well as making incremental changes to the format. While it is possible for new designs to take existing formats into account, it is virtually impossible for the improved design and its associated format to be achievable by previously existing drives. That's why new tape drives can usually read existing tapes made on older drives while older drives cannot read the tapes written by newer drives. While the same phenomenon occurs with disk drive technology, it's not a problem because the media in disk drives is not exchanged with other disk drives.

In general, there is an astounding lack of compatibility among various physical and logical tape formats in the market. This situation has frustrated IT professionals for many years, who have had to discard existing tape equipment because it was no longer compatible with newer technology products. Unfortunately, this situation is a by-product of the evolution of tape technology in a competitive market. Tape manufacturers have to be able to produce higher-capacity, higher-performance tape drives in order to stay in business.

The very real problem with this situation is how to read data from tapes that have been stored for many years when the drive technology has been long-since obsolete and is no longer available to read it. Backward compatibility should not be assumed in future generations of products. For that reason, IT organizations need to plan for the availability of equipment needed to read tapes into the future. This is a much easier issue to discuss than to follow through with, as there may not be convenient services or facilities to adequately store old tape drive equipment.

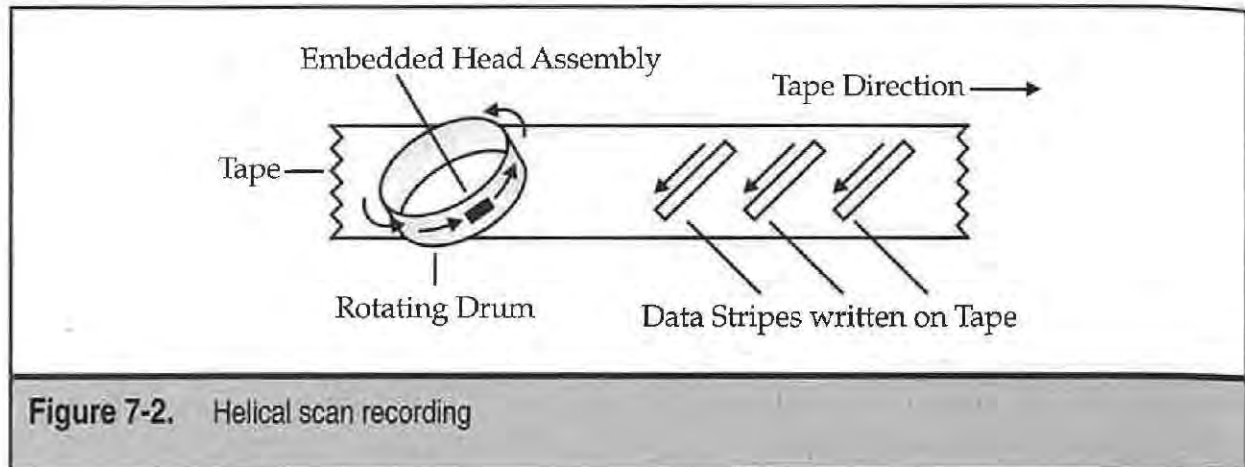


Figure 7-2. Helical scan recording

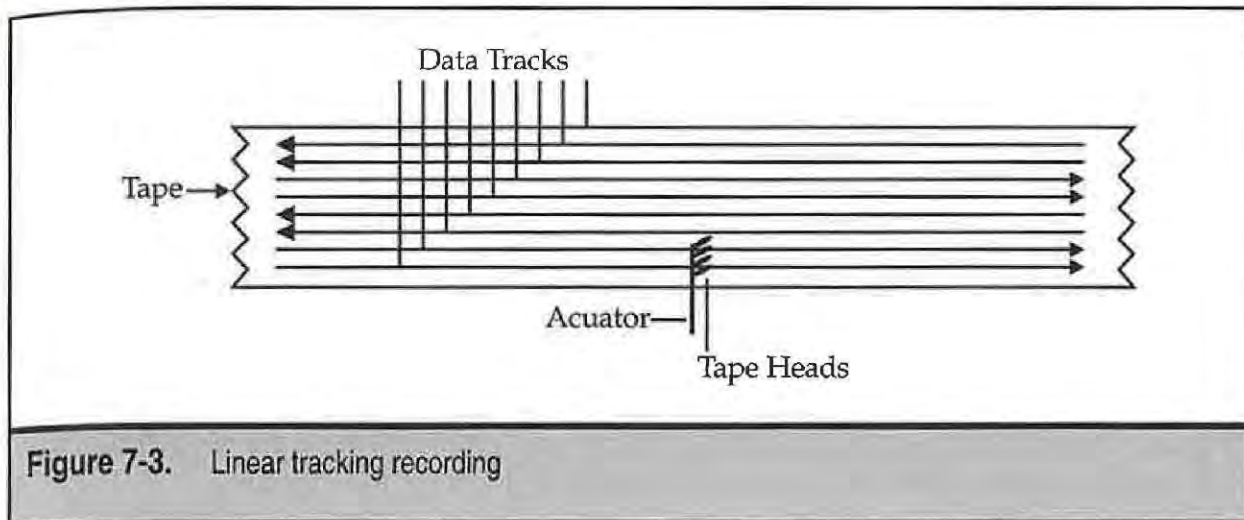
HELICAL SCAN TECHNOLOGY Helical scan tape drives use the basic mechanism that has been developed for video tape recorders. The quality of this technology varies from relatively expensive and high-accuracy 19mm professional studio gear to the relatively inexpensive and reasonably accurate high-volume 8mm and 4mm tape technologies used in camcorders and DAT (Digital Audio Tape) recorders, respectively.

The general concept of helical scan tape is that the heads are embedded in a rotating drum that spins at high speed, at an angle relative to the tape's edges. As the tape is passed in front of this moving drum, the heads read and write data in a pattern of diagonal stripes. Typically, helical scan drives have two read and two write heads, which can be positioned in a variety of geometries. The transmission channel for helical drives uses radio frequency transmissions to transfer between the spinning drum and the I/O circuitry of the drive. Figure 7-2 shows the relationship between the head, drum tape and stripes of helical scan recording.

One of the advantages of helical scan tape technology is the fact that it is used widely in the film, video, broadcast, music, and home electronics industries. As a result, the cost of parts for a helical scan tape drive can be lower than comparable linear recording technology due to the high-volume manufacturing efficiencies achieved with camcorder production.

LINEAR RECORDING TECHNOLOGY Linear recording technology is similar in nature to common audio tape recording. The head is held stationary while the tape is passed over it. Often several tape heads are stacked next to each other to read and write data from adjacent tracks. (The word "track" in this context refers to a defined strip on the surface of the tape that is parallel to the edge of the tape and maintains its distance from the edge.) Stacking allows linear tape drives to support many tracks at once—from 9 to 128 in late 1999.

The head stacks are connected to servo motor mechanisms and actuators that change the relative position of the heads to enable a track to be read or written. Like audio tapes, with their A and B sides, linear tape drives reverse their direction at the end of the tape and start working in the opposite direction. Some linear tape drives use a *serpentine* tracking system with several reversals. With the ability to use both directions of the tape, searches for data on tape can be accomplished much faster. Figure 7-3 shows the relationship between the heads, actuators and tape in linear track recording.

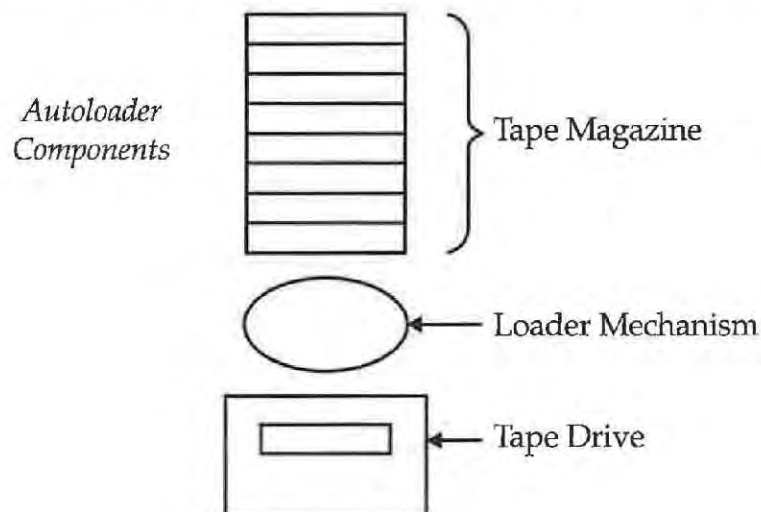


Among the varieties of linear tape technologies are Quantum's DLT, IBM's Magstar, Tandberg's MLR, QIC, and Travan.

Backup Subsystems

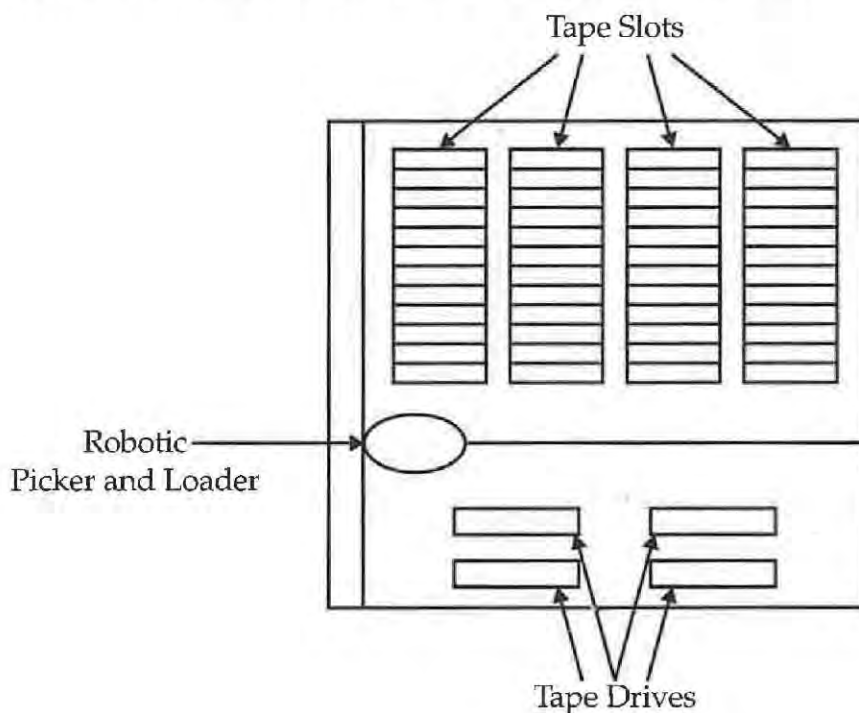
Just as disk drives can be placed in JBOD and RAID subsystems, tape drives can also be combined with other technology to create subsystems that provide a variety of functions geared to simplifying backup management.

TAPE AUTOLOADERS Tape autoloaders combine an individual tape drive with a tape robot mechanism that can automatically select, grab, remove and insert tapes under either programmatic or manual control. Typically, autoloaders can hold between four and twelve tapes and be small enough to fit inside a full-height 5.25-inch drive bay. An autoloader's tape drive and "picker" robot are typically enclosed inside the external case of the autoloader. The diagram below shows the basic components of a tape autoloader.



Autoloaders are a good solution for situations where backup commonly extends beyond one or two tapes. In networks where the storage capacity is not large, autoloaders can be used to hold a week's worth of tapes without needing to add or remove any. Because they hold a relatively small number of tapes, they have limited value for managing large amounts of media.

TAPE LIBRARIES Tape libraries refer to any tape automation products that are physically larger than, and hold more tapes than, an autoloader. They often have more than one tape drive, but that is not a requirement. Tape libraries can be quite large and hold thousands of tapes, although libraries of this size are rare for open systems network backup. Common library configurations have two to four tape drives and can hold 20 to 80 tapes. There are many possible configurations and geometries used in library systems. The diagram below shows the basic components in a typical tape library.



Libraries can use bar code readers to identify tapes as they are inserted. When a large number of tapes are inserted, the ability to use bar code recognition can save a great deal of time. Normally, the bar code information is a sequence of numbers that is applied to the spine of the tape cartridge as a sticker. The bar code number is independent of the name used by the backup software application, however most library-enabled backup software packages provide a way to correlate bar code information with a more intuitive name.

TAPE MIRRORING The concept of a tape mirror is simple—the data that is written to one drive is written to the other, just as with a disk mirror. There are several products in the market that sell two tape drives that are designed to work together as a mirrored pair. One of the major advantages of disk mirroring is the ability to have copies of data on site

and off site simultaneously. The main drawback to tape mirroring is the fact that two tapes are created with the same name, which can cause media management problems.

TAPE RAID Just as tape mirroring is similar to disk mirroring, tape RAID is similar to disk RAID. The primary benefit of tape RAID is performance. When data is striped across multiple drives, performance can be much better than that of a single drive. However, media failures in tape RAID are much more of a problem than they are with disk, due to the relative high number of errors that occur and the performance impact of stopping and starting multiple sequential access devices to perform RAID recovery procedures. An additional problem people have experienced with tape RAID is remembering which tape goes in which drive. While it sounds easy enough, this can be surprisingly difficult, especially during restore operations.

Media Components

For backup and recovery, the buck stops at media. Neither hardware nor software, media is the least expensive and most critical part of the backup system. The entire subject of media and media management is one of the least understood aspects of network backup. Media management becomes increasingly confusing as the capacity of online and stored data increases and the number of media parts needed to store new and historical data increases along with it.

The Media Cartridge

Network backup media is almost always packaged in a cartridge mechanism, commonly called a "media cartridge" or "tape cartridge." It's a subtle distinction perhaps, but the media is the actual magnetic material that receives and retains the signal while the cartridge is the integrated case with reels, guides, bearings, and all the other small parts that physically support the media.

A media cartridge can be identified by holes and other indicators molded into its shell. These indicators communicate to devices the technology generation of the media and the device used to read and write data to the cartridge. For instance, an indentation in the cartridge can be detected by a device that adjusts its operations to match those of an older technology device to provide backward compatibility.

CARTRIDGE MEMORY For the most part, media cartridges are unintelligent, and lack the characteristics of electronic devices. They provide no intelligence or memory to impart information about the contents or condition of the media inside. Instead, this type of information can be written on the media where it can later be accessed by loading the media and reading it. However, there are times when a cartridge-based memory can be useful, such as when identifying tapes in a multicartridge library.

Sony has developed a cartridge memory technology that forms the basis of its Advanced Intelligent Tape technology, or AIT, which contains a programmable, nonvolatile memory for holding information provided by media and storage management software. While the benefits of this type of memory could be valuable to customers by saving time

during backup and restore operations, backup software vendors are not exercising it much today. It's a technology catch-22: the implementation of the technology probably won't occur until the demand for it increases first, but the demand isn't likely to increase until the implementation can show real benefits.

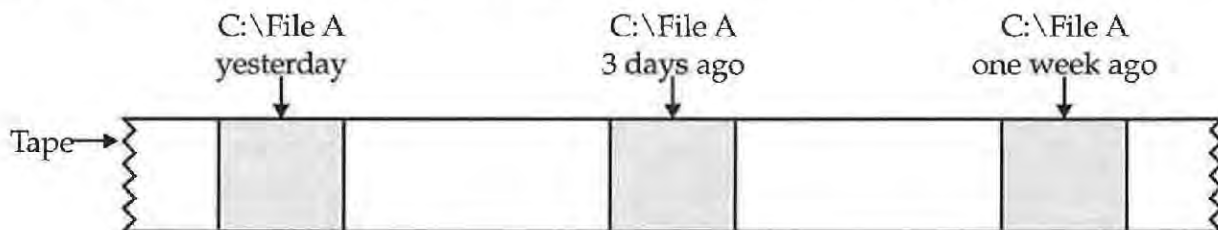
Tape Media

Tape media is practically synonymous with backup in network backup for several reasons:

- ▼ Tapes provide good performance for reading and writing long streams of data typically encountered during backup and restore operations.
- Tapes provide excellent capacity compared with rotating magnetic and optical media.
- Tapes are relatively small and can be easily stored and transported.
- ▲ Tape is cheaper per megabyte than other media, such as rotating magnetic and the various forms of optical media.

USING SEQUENTIAL MEDIA Tapes are accessed sequentially. In other words, data is written to tape as sequential blocks starting at the front of the tape and filling the tape until it reaches the end of the tape or the operation stops.

Sequentially accessed media differs considerably from random access media, such as a rotating disk drive platter. To begin with, tapes are an off-line media and do not depend on having a single copy of a file that every user or application accesses. For example, disk file systems typically do not allow different versions of the same file to have the same name and be in the same directory. Tape storage, on the other hand, can store many copies and versions of a file in multiple locations on the tape. As files are updated on disk and copied onto tape, the complete new version from the same directory location is written to a new tape location. Existing versions of the file stored on the same tape are not updated with new information—the previously written file version is left intact. The drawing below shows how multiple complete versions of files are stored on tape.



Tapes work well for storing large files. As a sequential access media, tape does not spread data evenly across the media surface in small chunks like disk storage typically does; instead, it simply writes it in large contiguous blocks on tape. Therefore, it is well suited for copying extremely large files that would be scattered all over rotating disk platters. While the access time to any random block on a tape is extremely slow, it is capable of streaming speeds in the range of 5 to 15MB/s using technology available in 1999.

The demise of tape has been predicted for decades as other media has been developed and destined to take its place. However, tape technology continues to prove itself as an excellent combination of performance, capacity, and price. So far, optical media has not been fast enough nor does it provide the capacity and cost benefits that allow it to compete with tape.

Disk drives have also been used for backup, but they have historically been too expensive and are not easily removed or transported as dictated by disaster recovery requirements. Both these limitations are becoming less of an issue over time as disk drive pricing declines to less than \$.01/MB and network storage provides the ability to make remote electronic copies of data that circumvent the need to physically move tapes.

Media Sets

Although tapes offer an excellent combination of capacity and performance, sometimes their capabilities are not adequate for the job at hand. In these cases, multiple tapes can be used together as a linked set commonly referred to as *media sets*, or sometimes *media pools*. Media sets enable concurrent and/or sequential backup operations to run on multiple tape drives to achieve the performance and/or capacity required. As these tapes represent a single repository of backup data that may need to be restored together at some time in the future, it's important to be able to schedule sets together as part of the same backup or restore operation. Just as you would not typically cut and splice pieces of tape together from different data tapes to create a merged backup tape, you should also avoid mixing tapes together from different media sets.

Media Magazines

Tape autoloaders and libraries store tapes in internal slots that are sometimes referred to as elements. Usually tapes are removed and inserted one at a time; however, many automation products use magazines, which are multislot tape cartridge containers that can be removed and inserted. Magazines provide a way to handle tapes in groups or sets for the purposes of storing them.

TAPE FORMULAS Data tapes are made by means of combining several different materials together to form a single composite ribbon. A great deal of specialized chemical technology is involved in creating data tapes. For example, the largest American producer of data tapes is the 3M Corporation, which is primarily a chemical manufacturing and technology company. For that reason, the tape *formula* is an important technology for backup and recovery systems.

Most tape is made up of the following layers:

- ▼ A backing layer, which blocks magnetic signals
- A base layer made out of mylar for strength and flexibility
- A binding layer for holding magnetic materials in place
- A magnetic layer, which provides the actual storage function
- ▲ A tape coating, which protects the magnetic layer and provides lubrication

The backing, base, and binding layers are important for the longevity of the media. A failure in any of these components can result in the tape literally falling apart. However, problems with tapes can be unknown for many years until the tape has aged and been exposed to environmental influences.

The magnetic layer gives the tape its signal strength characteristics, also called the tape's *output*. Typically, output is expressed as some range of negative decibels (-dB). The greater the output level, the stronger the signal. The output measurement of tapes tends to be proportional to the frequency of the signal; as frequency increases, so does the signal strength, until a point is reached where higher frequencies can no longer be supported. The important point of output levels is that higher output measurements may mean it can be easier to read and restore data from a high-output tape than a low-output tape.

Output is a function of the coercivity of the magnetic material particles. In a nutshell, *coercivity* indicates how readily magnetized a tape is. A tape that is more easily magnetized typically records a stronger signal. Coercivity is determined by the density of magnetic particles in the tape and their magnetic properties. The more magnetic particles in the tape, the more chances there are to write a signal.

Finally, the coating on the tape determines how well the tape will wear over many uses. The coating protects the magnetic layer from small dirt particles that can remove or eclipse the magnetic layer. The coating also helps maintain a constant speed by reducing friction. This is extremely important, especially in helical scan tape drives.

The two leading tape formulas in 1999 are MP++ (metal particle ++) and AME (advanced metal evaporated). MP++ is based on a successful tape formula that has existed for many years and has continued to be improved. AME is a relatively newer formula developed for the video recording industry that has an extremely durable and smooth coating. In general, MP++ is used for linear tape drives and AME is used for helical scan drives.

Caring for and Storing Tapes

Manufacturer specifications for tape commonly claim the ability to read data for 30 years. Such claims are based on accelerated tests under optimal conditions and should not be used to base a realistic tape storage policy. It is much more realistic to think about tapes as lasting for five years, possibly ten if they are carefully stored and maintained. As an advanced chemical and materials technology, tapes require consistent care to maintain its composition and the integrity of the data written to it. Tapes degrade over time due to oxidation of the metal particles within and aging of the various layers, resulting in cracking and gaps in the magnetic layer. Here are a few maintenance tips to remember about tape:

- ▼ Maintain low humidity.
- Maintain cool temperatures.
- Periodically re-tension tapes.
- Limit the number of passes.
- ▲ Keep tape heads clean.

Humidity is tape's worst enemy because it causes tapes to oxidize. Oxidation adversely affects the magnetic and physical properties of the magnetic layer in the tape. Therefore, one should avoid storing tapes in a humid environment.

The backing and binder of tapes stretches and changes shape as the temperature increases. This causes the data signal on tape to stretch, which makes it more difficult to read correctly. If possible, tapes should be maintained at consistent, cool temperatures in the range of 40 to 60 degrees Fahrenheit or 5 to 15 degrees Celsius.

Tape stiffens with age if it is not used. Stiffness causes cracks and leads to the breaking apart of the various layers in the tape. Therefore, tapes should be periodically *tensioned*, preferably every six months, if they are not being actively used at least once a year. Tape tensioning is a common operation found in most backup applications that basically unwinds and rewinds the tape. Archived tapes are at the most risk for loss due to tape stiffness because they are typically stored away, out of sight, and not expected to be used again, except for reading data if it is ever needed.

Tape heads read and write data when the tape comes in contact with the tape head. This causes wear and tear on both the tape and the tape head. A *tape pass* occurs each time a head and tape read or write on the same section of tape. While the data might be different, the physical section of tape is the same. Tapes sometime specify the number of passes allowed before degradation occurs (usually in the range of 2,000), and some backup software packages count the number of tape passes that occur. A tape pass should not be confused with a tape usage. For instance, a section on a tape could have several tape passes occur on it in the course of a single backup operation. This makes counting tape passes extremely difficult, because it is virtually impossible to track each section of tape uniquely. Still, software manufacturers' tape-pass counts are better than nothing and serve as a metric for when tapes should be "retired" from active backup duty.

Tape materials shed when they are used, particularly when they are initially used, and coat the tape head. In addition, airborne dust and other contaminants stick to tape heads. The more stuff there is on the tape head, the less strong the signal is recorded on tape. It doesn't take a rocket scientist to figure out that a poorly recorded signal may be harder to read than a signal that is recorded well. For that reason, tape heads on tape drives should be regularly cleaned to reduce the effect of "junk" on the heads. Most manufacturers specify cleaning approximately every 30 hours of operation. A weekly cleaning schedule is a good idea.

NOTE: Tapes last longest if they are kept in cool, dry conditions, are used in drives that have regular head cleanings, are not overused, and are tensioned twice a year.

Software Components

The software components of network backup and recovery are discussed in the following sections. While hardware determines the maximum transfer rates and performance of a backup system, software determines how efficiently the hardware can be used. The

terminology and concepts presented here are general and are not intended to represent specific products.

Job Scheduler

All backup software products have some form of job scheduler. This is the component that determines what information is to be backed up, and when the backup operation should start. It also usually determines what device to use and which piece(s) of media to use. Job scheduling can be done interactively through an interface where the administrator selects all the variables, and it can also be automated. For the most part, backups tend to be automated and restores tend to be done manually.

Media Manager

Backup data is written to tapes. As these tapes contain corporate assets, it makes sense to manage their use and storage policies. In fact, media management is one of the most im-

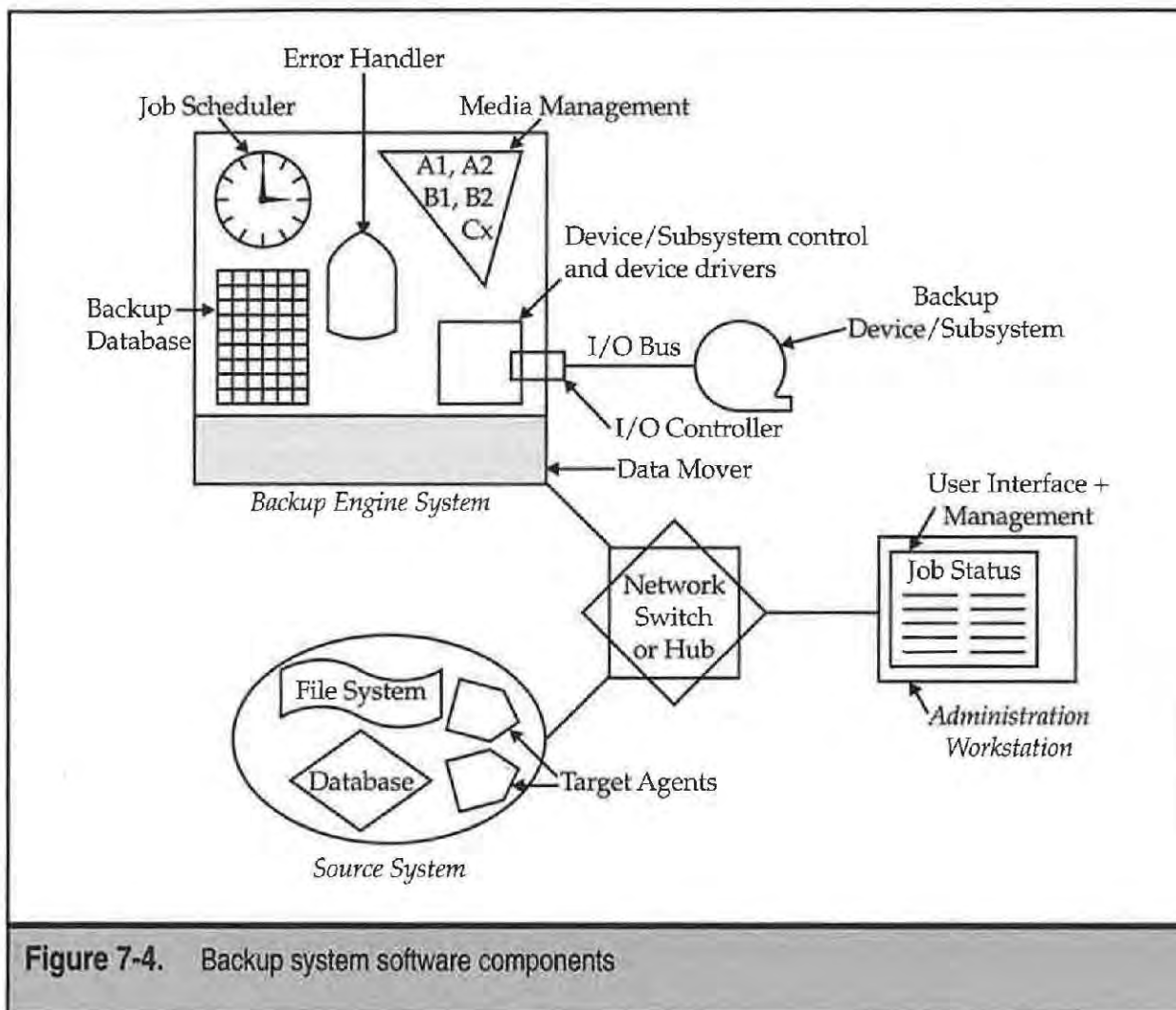


Figure 7-4. Backup system software components

portant aspects in any network backup product. There are often many tapes to manage in support of several different backup goals. Tapes that are not used in their correct sequence can result in the inability to restore a complete server. With many tapes available within the grasp of hurried administrators, it is easy and commonplace for mistakes to be made selecting media for backup operations. Media management can help reduce these mistakes.

One function of a media manager is to ensure that usable tapes are available that allow upcoming backup operations to complete successfully. A related function provided by media managers is keeping an accurate inventory of tapes loaded inside an automated autochanger or library. Other functions can include determining which tapes to move off site for disaster recovery protection, which tapes to move back on site for backup operations, which tapes to discard due to error counts or overuse, and which tapes are duplicates of other tapes.

Source System Agent

Network backup is based on the idea of a backup engine communicating with a source system on a network and copying data from it across the network to backup media loaded in a locally attached tape drive. The software component running on the source system that enables this is usually called a backup *agent*. Agents are typically developed for particular operating system platforms or databases and take advantage of special programming interfaces for determining what data to back up.

For the most part, backup agents are designed to work with a particular backup engine and are not standardized. The protocols and exchanges between engines and agents are highly specialized to fit the particular processing sequence of the engine. While standardization for backup agents is highly desired by the market, it is not likely to occur with existing network backup architectures, due to the necessity of backup vendors adjusting their sizeable code bases. However, storage networks may develop standards for these interfaces that will achieve broader engine/agent interoperability.

Network Data Mover

The data mover is the function that oversees the copying of data from the source system to the engine and eventually to tape. In traditional network backup, this typically resides on the engine system, although there can be distributed data movers that “push” data from source systems to the engine as well as remote device servers that provide transfer to some other system than the engine.

Device/Subsystem Controller

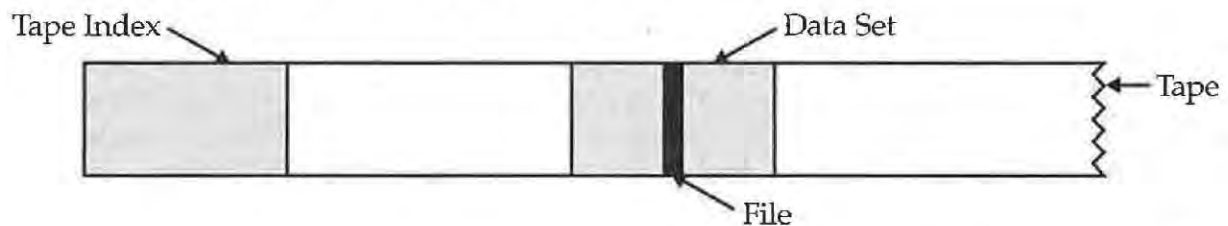
The device controller is the software component that manages the detailed operations of the physical storage device or subsystem that writes to media. Usually this is part of the backup engine, but it can also reside on a source system or other system in the network and have its operations controlled remotely by the backup engine. This is called *remote device support*.

Logical Tape Format

In addition to the physical tape format discussed previously that is determined by the drive manufacturers, there is also a logical format that is imposed over it by backup software. The logical format creates the higher level structure that is used by the backup software to organize its data. Logical formats determine the size and location of:

- ▼ Header information, including the electronic tape name or label
- Index information and location "markers" to quickly locate backup data sets
- Backup data sets, including data that may be interleaved
- ▲ Error encoding and statistics

The primary requirement for a logical tape format is locating data correctly on tapes. One of the most common methods for locating data is to keep an index at the beginning of the tape that identifies where data sets are on tape. A *data set* is a file or group of files copied together to tape as part of a single backup job. For example, files copied from a specific disk volume on a server could form a data set; in contrast, an individual file that is manually backed up would be written as a single data set. Individual files are accessed for restores by first reading the index, then locating the data set, and finally locating the file within that data set. There are many ways to structure the information and access to data on tape. One possible structure is shown below.



The index section in a logical tape format is usually a fixed-length block of tape that can accommodate many data set records. When new data sets are written to tape, the index is updated to reflect the presence and location of the new data set. When data is overwritten as tapes are reused in later backup operations, the index is also overwritten to indicate the new data sets. As a result, the index is constantly being used by both backup and restore operations. This is typically the section of tape that has the highest number of tape passes and wears out the fastest.

There are several prevailing tape formats in use. The Unix TAR (Tape Archive and Retrieval) format is used by several vendors. The MTF (Microsoft Tape Format) is commonly implemented on products that back up and restore Windows systems. NetWare backup systems tend to use SIFF (Systems Independent Data Format). Legato systems have a tape format called OpenTape, which is available for other companies to use but has been basi-

cally ignored by the other backup vendors. For years CA/Cheyenne had its own proprietary tape format, but it has begun to use MTF in its Windows-based products.

The use of multiple tape formats on different platforms means that it can be difficult if not impossible to read a tape on one platform that was created on another one. Readers are advised to inquire about cross-platform tape format compatibility.

Backup System Database or Metadata Catalog

Despite common perceptions, the real heart and soul of any network backup system is its internal database, sometimes called its metadata or catalog. For the most part, the word "catalog" implies a collection of records of specific backup operations such as what was backed up by those operations and what tapes were used.

A *backup system database*, on the other hand, implies some type of relational database technology that allows views of backup information based on multiple indices, including such things as filenames, file date and timestamps, tape names, pool names, date of backup, and file ownership information. The actual views allowed by any backup system will vary considerably depending on its ability to back up and store the *metadata* about the data being backed up.

The database is central to the backup process for its ability to support the various functions of the backup program, particularly restores. By providing several views, administrators can quickly search for particular files or database tables to restore without having to scan records of backups to find what they are looking for. This can save hours and even days, depending on circumstances.

However, this ability is not free. In order to have a database, the data must first be collected and then entered into the database. Where backup is concerned, these databases tend to be very large, as the number of objects being backed up tends to be large. For instance, if there are 500,000 files on a server, there have to be at least 500,000 records in the backup system database to store records for each file. Then, as one adds metadata categories that are desired, it's easy to see where a full backup of all 500,000 files creates a pretty large database operation. The transaction processing required to create these records about what was backed up can take an hour or more. Multiplying the number of servers by ten or more, it becomes obvious how large a task the metadata processing is for backup.

The situation becomes even more interesting when each entity being backed up has a record for each of the copies available on media. Let's say there were 15 records for each file on the system. The size of the database now starts becoming an issue. Most backup systems have a facility for removing backup records from their databases when the size of the database becomes a problem—usually this is apparent in sluggish backup performance and extended database processing cycles. The deletion of a database record does not mean the data on a tape cannot be restored, but it does mean it will take much longer.

Network backup systems provide the ability to perform database maintenance by deleting, or purging, older records from the database. While this helps manage the size of the database, it also means that it can be difficult finding data on tapes that no longer have database entries for them. Therefore, database maintenance tools that provide a detailed selection of records can be valuable, especially when there are tapes being stored for extended periods of time to meet data retention and audit requirements.

The key issue here is the number of entities being backed up and tracked individually. A large database that backs up relatively large (50MB) tables as a single entity can be much easier to manage than a small file server with lots of small (10KB) files.

User Interface

Network backup software is like any other software product in its requirement to support a graphical user interface, or GUI. A GUI controls most of the other functions in the software, including the configuration of the source agent.

Error Handling System

All backup software products have some way of communicating errors—some better than others. The general idea here is that there needs to be some way of knowing when parts of backup failed. The structure and scope of the error reporting system can be a real problem for the same reason database processing is: the size of the system practically guarantees errors will occur on a daily basis. The challenge is understanding which errors need to be treated as important and which are unimportant. For example, a file that is backed up but does not have its access date metadata backed up could be considered an error. The question is, what level of alert or reporting should be generated to draw attention to this? In reality, there are bound to be several such errors every night backup runs. Trend analysis for backup file system errors would be valuable, but it does not exist yet. Maybe it will yet in our lifetimes.

It is important to review the error logs of backup regularly to try to find problems and get them fixed. While this is somewhat painstaking work, it can identify critical system shortcomings that need attention. Too often, a serious backup problem is not uncovered until a restore is needed and then everybody finds out that some of the data can't be restored because it hadn't been backed up for the last 30 days due to some anomaly or bug.

Higher-Level Management

Closely related to the notion of an error handling system is a global management system that can manage multiple backup systems by providing centralized administration and/or reporting. The idea here is not necessarily a high-level network management system such as HP's Openview, Tivoli's Management Framework, or CA's Unicenter. While these products will provide some reporting capabilities, it's not clear that these consoles will be able to present the details required for successful management of backup. It makes more sense to use a backup-specific management system that provides detailed reporting

and management of the logical components of backup systems such as consolidated media and database management and maintenance. Several such management systems are under development by Veritas, Legato, and others.

The sections that follow contain several examples of common network backup systems that can be built from the components discussed in the previous sections.

Typical Network Backup

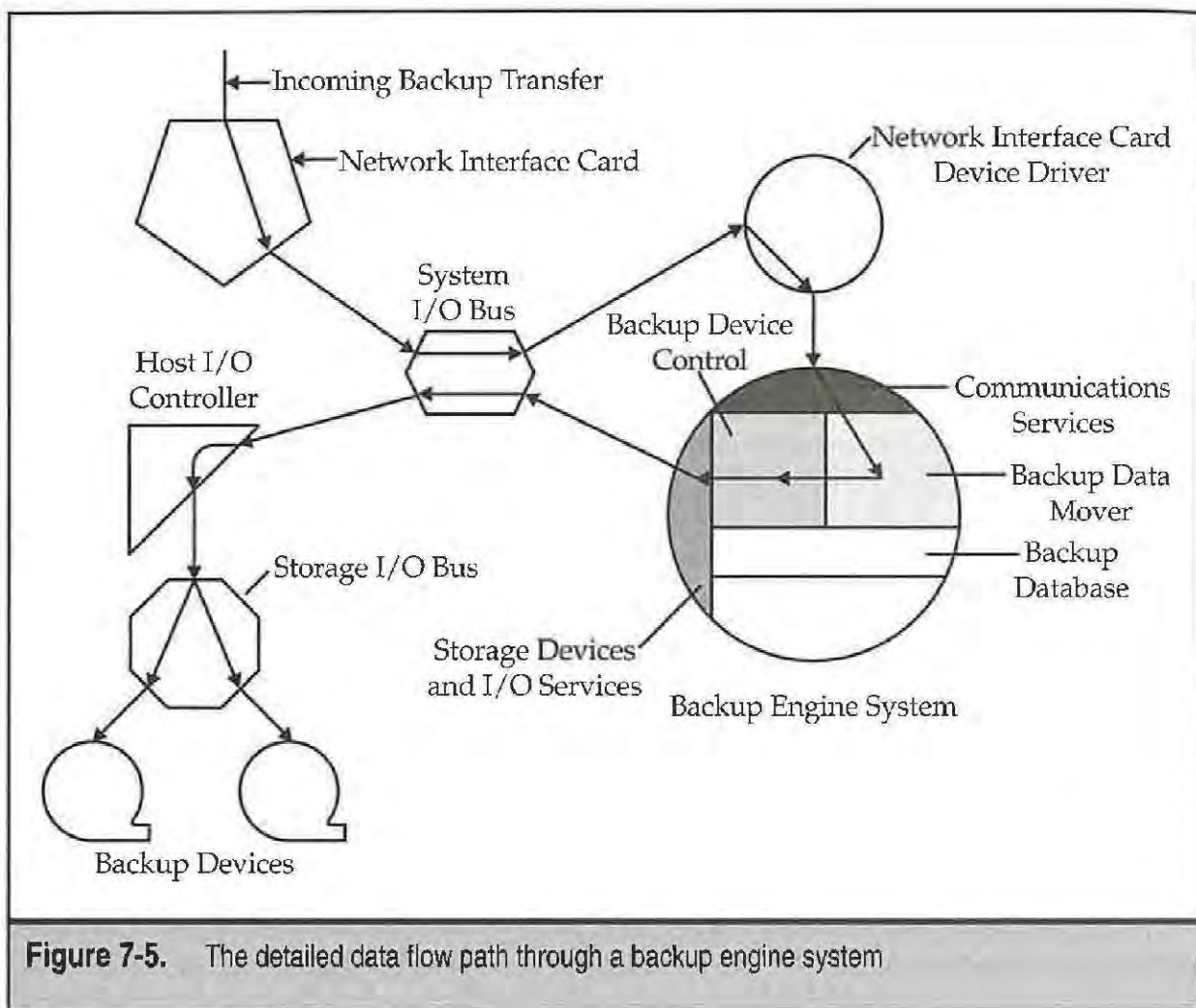
The most common network backup system used today is similar to the one shown in Figure 7-4. It is a client/server backup model where the backup engine establishes network connections directly to an agent on source systems and manages backup operations through those links. This is typically done over a TCP/IP network, but it can also be done using IPX/SPX for NetWare environments.

Direct network backup can quickly saturate an ordinary client/server network due to the heavy traffic loads generated. 10BaseT ethernet links are not adequate for most server backup operations. 100Mb links through either 100BaseT or FDDI will deliver much better results. It is important to remember that other network traffic can be adversely affected while backups are running. Two backup sessions can take most of the usable bandwidth of a 100baseT connection.

The source system typically has a *backup agent* running on it that responds to commands and inquiries from the backup engine. This backup agent works with the unique environment of the source system and interacts with any backup interfaces and processes on that system. Most often, this agent functions as a remote procedure of the engine's software. In other words, it runs as part of the engine's operations, but on the source system.

When the engine begins backing up the source system it communicates to the agent and begins accessing the file system or database using the programming interfaces they provide. The results of the scan determine what data needs to be backed up on the source. This information is transmitted back to the engine, where it is given to the data mover to start transferring data. The data mover copies the files over the network and hands it off to the backup device control system and it is eventually written to a tape in a tape drive. When the data is completely written to tape, the database can be updated to reflect its successful backup. Figure 7-5 shows the data transfer through the backup engine system starting at the networks interface card all the way to the backup device(s).

This diagram is similar to Figure 3-14 (see Chapter 3), which shows the data path for a client accessing a file on a server over the network. In that drawing, the backup engine would assume the role of the server and the source system would assume the role of the network client. When one sees the whole picture laid out like this, it's not hard to see why backup over a network is slower than backup over an I/O bus. For starters, a 100Mb (10MB) network has a theoretical performance maximum equal to a fairly slow and outdated I/O channel. This not only makes it difficult to get the job done within the allotted time, but also compounds matters by not delivering data fast enough to stream tape

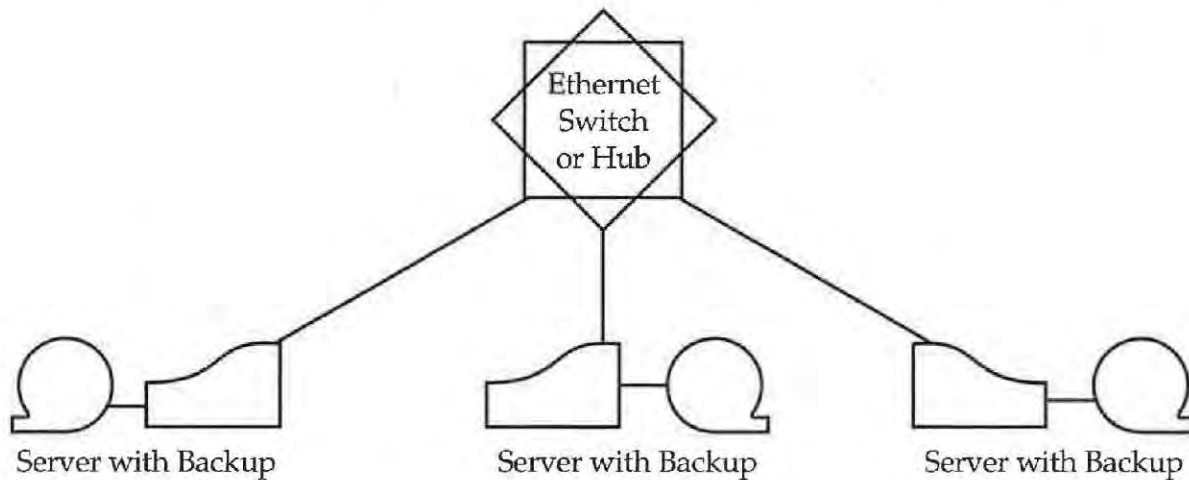


drives. The inability for the network to deliver data comparable to the speed of the backup equipment is the primary shortcoming of network backup systems.

We will now look at a few techniques that attempt to circumvent this performance problem.

Local Backup to an Attached Tape Drive

The most simple network backup is not really network backup at all—it's local server backup of network servers. This method installs all components on the server: engine, source, data mover, and everything else. The main reason for running local backup is to get faster performance from a locally attached tape drive on an I/O bus. If the server is large enough and backup window small enough, it might be necessary to back up the server using parallel backup streams to locally attached drives. The diagram below shows three servers in a network, each with their own local backup.



The primary problem with single server backup is the lack of management integration to oversee its various functions. Local backup equates to at least one tape drive per server, one backup software package per server, and multiple tapes for each server. This is considerably more to manage than a single large network backup system. The additional workload of the entire backup engine will compete with other resources on the source system and impact its performance.

Backup with Remote Devices on Source Systems

To achieve the performance of locally attached devices with the manageability of a single system, one can attach tape drives to large source systems having more data than can be backed up over the network. But instead of operating the tape drive with a backup engine running on the system, the backup operations are managed remotely by a backup engine somewhere else on the network. Figure 7-6 shows a network backup system using this kind of *remote device support* on source systems.

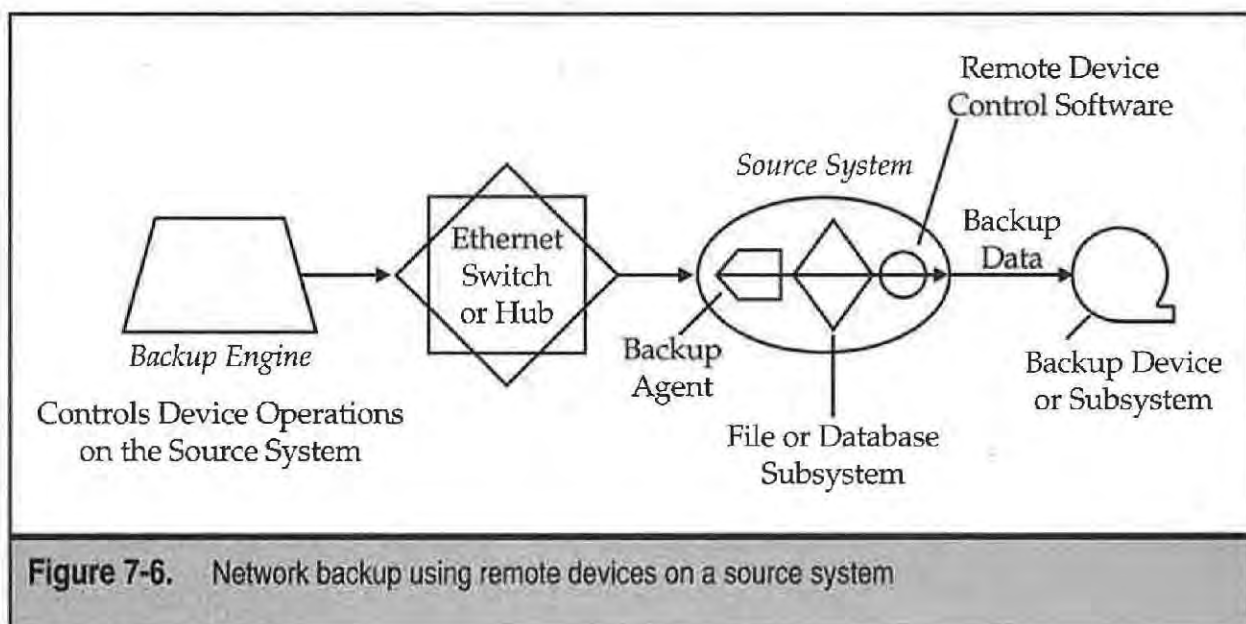


Figure 7-6. Network backup using remote devices on a source system

Remote device support provides excellent performance for large source systems that have too much data to be backed up over the network. In addition, the management effort is easier than implementing a separate single server backup system. Media management, however, is almost as difficult as it is for multiple single server backup systems.

Network Backup with Parallel Data Transfers

Another technique used to improve backup performance is parallelism. The basic idea is that a backup engine can be connected to the network using multiple NICs or multiport NICs that have external connections for more than one network connection. Backup engines supporting concurrent network backup sessions between the engine and source systems can implement two approaches to parallel operations.

DEVICE PARALLELISM The first type of parallelism is referred to as device parallelism or multiple device streams. The idea is that data from one source system can be routed to a dedicated device in a one-to-one relationship. This way, a backup engine can back up multiple sources simultaneously to an equal number of devices.

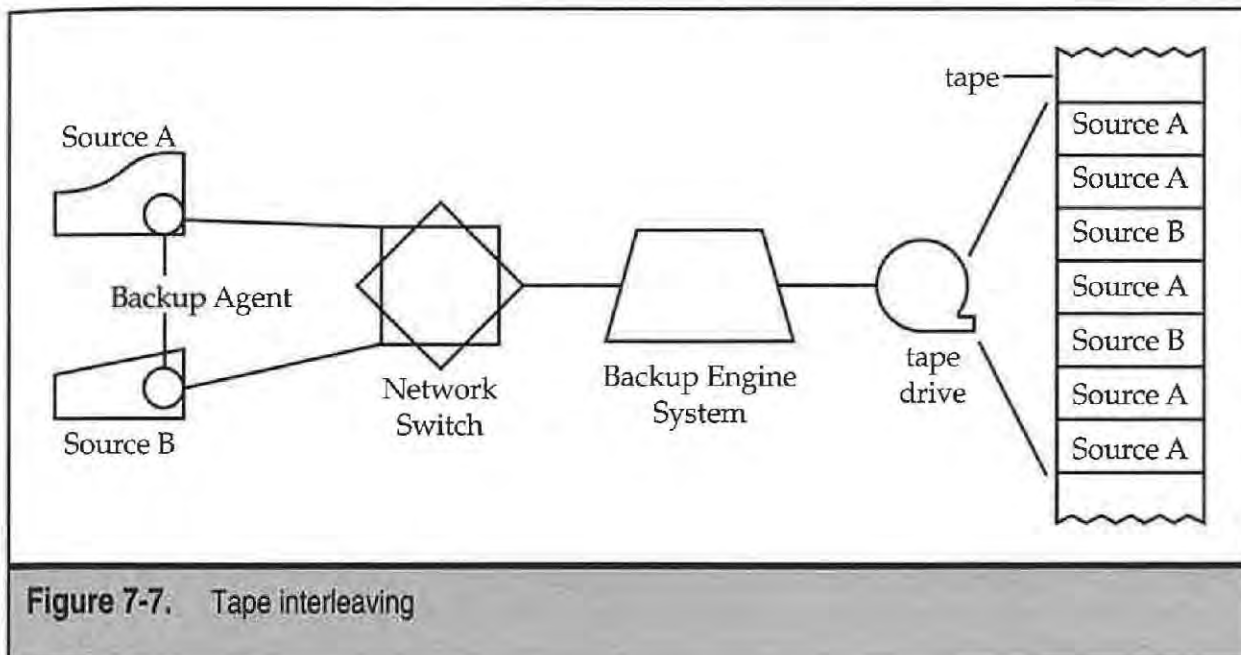
Device parallelism provides a way to get more “bulk” backup done, but it does not help stream the tape drives or speed up the backup of any source system. The primary advantage is the ability to finish the entire backup faster than processing one session at a time.

INTERLEAVED DATA The other type of device parallelism uses a technique called *interleaving*. Interleaving combines the data from multiple sessions with multiple sources onto a single tape. The result is the ability to achieve streaming speeds to a tape drive by combining data from several sessions.

Interleaving data cannot speed up the performance of backup on any one source, but it does make the overall backup process finish faster. The combination of device parallelism and interleaving allows backups to finish faster yet—if the networking capabilities of the backup engine can support the additional traffic. Figure 7-7 shows two sources being backed up simultaneously by a backup engine that is interleaving their data on a single tape drive.

WORKING WITH BACKUP

There are several variables in backup operations that are used in the normal course of network backup. This section explores some of these variations and explains how and why they are used. As mentioned before, backup and restore operations normally are run in close coordination with the server’s file system or database. The backup engine communicates with the source agent, which in turn performs its work on the source system’s file system or database.



Types of Backup Operations

There are a few basic backup operations that are performed for various reasons. Usually, the selection of the backup operation depends on the importance of the data, the amount of time available to run backups, the day of the week or month, and other variables. In general, there are four fundamental operations to consider:

- ▼ Full backups
- Incremental backups
- Differential backups
- ▲ Selective, or on-demand, backups

Each of these will be briefly discussed in the following sections.

Full Backup

Full backups copy the complete contents of a disk volume, or logical disk. In other words, a full backup would be used to back up the C: drive, D: drive, etc. of a system. The term "full backup" can be applied to the server and include all its assigned logical volumes, or it can be applied on a volume-by-volume basis. In that case, a complete server backup would be comprised of several volume full backups.

The main reason for performing full backups is to provide the easiest means to restore a disk volume. With the entire contents of the volume on a single tape, or group of tapes, the restore process is straightforward and easy to understand. Full backups are typically performed on weekends when there is adequate time to finish the job without interrupting the work of end users.

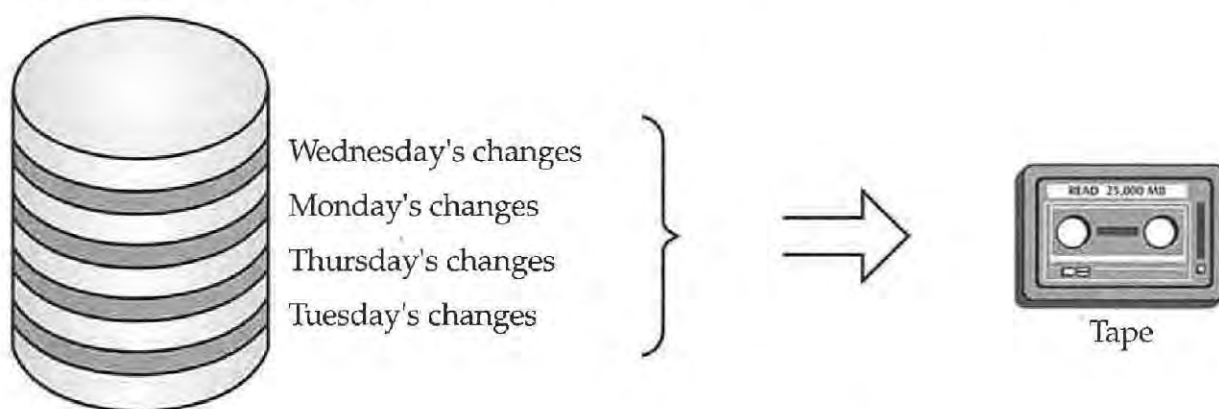
Incremental Backup

Incremental backups copy newly changed data that has been created or updated since the last backup operation. The main advantage of incremental backup is to perform the shortest possible backup operation. When incremental backups are used, the restore process requires data to be restored from the full backup and all incremental backups run since the last full backup.

This does not mean that a different tape needs to be used for full and incremental backups. The same tape can be used or a different tape could be used every day if desired, depending on capacity and off-site storage requirements.

Differential Backup

Differential backup copies all new data that has been created or updated since the last full backup. For instance, if a full backup were performed over the weekend, at the end of the business day on Monday the differential backup would be the same as an incremental backup. However, on Tuesday the incremental backup would contain all the data that would have been copied on the incremental backups on both Monday and Tuesday. By Thursday, the differential backup would contain the incremental data from Monday through Thursday like this:



The main purpose of differential backup is to limit the number of tapes needed for a full restore to two. In reality, the number of tapes can exceed two if the amount of data on a server exceeds the capacity of a tape. But the point is the same—to limit the number of tapes involved in a restore.

On-Demand Backup

The backup operations listed above normally run as automatic operations that follow a defined schedule established by the job scheduler component of the backup application. However, administrators also need to be able to backup data on demand as situations dictate.

Backing Up Live Systems

Many server systems today, especially web servers, have 24x7 availability requirements and can allow only the shortest amount of time possible for backups—if they can allow any time at all. In some cases, the backup system must be able to copy data from the server while it is running and providing services to end users. In Chapter 4 on mirroring, some mention was made of the value of brute force operations in storage management. Backup is another area where there is little substitute for brute-force performance and capacity advantages. No matter how clever your algorithms are for managing backup operations, the ability to finish backup processing faster is almost always a significant benefit. So, where backup is concerned, it is hard to go wrong by purchasing the fastest and highest-capacity tape equipment. One of the trickiest parts of server backup is ensuring that the data being backed up is valid and has integrity. At the root of the problem is the nature of multiuser server systems where many users may be accessing the server's data at the same time the backup system is attempting to make a copy. If there are updates occurring to files or database records at the same time the backup process is copying it, it's possible that the backup copy may contain sections that were copied before the data was updated, with the remaining sections being copied after the data was updated. In other words, the data copied for backup represents two different versions in unequal halves. When that happens, the data copied to tape is inconsistent and unusable.

There are two basic ways to deal with this problem:

- ▼ Cold backups
- ▲ Hot backups

We will discuss these two types of backup in the following sections.

Cold Backup

Cold backup is a backup operation that makes backup copies of data to tape while the server is not accepting new updates from end-users or applications. This completely alleviates the problems of a concurrent update while a backup copy is going on. Cold backups work very well and are often used for backing up parts of databases.

Of course, the problem with cold backups is that they make the server unavailable for updates while the backup process is running. If the cold backup runs for six or eight hours, this could be a serious problem for a server with 24x7 availability requirements.

Hot Backup

The availability issue with cold backups initiated the development of hot backups. The idea of hot backups is that the system can be backed up while users and applications are updating data. There are two integrity issues that crop up in hot backups:

- ▼ Each file or database entity needs to be backed up as a complete, consistent version.
- ▲ Related groups of files or database entities that have correlated data versions must be backed up as a consistent linked group.

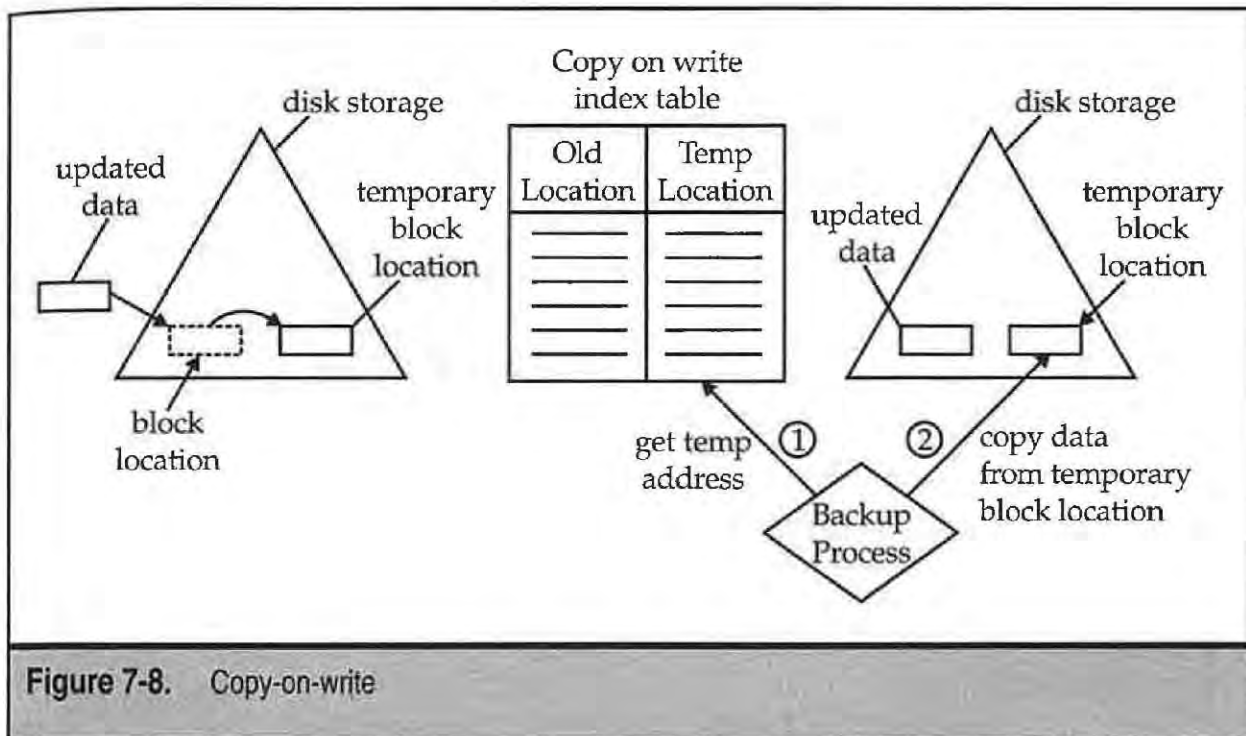
The simplest solution is to ignore the problem altogether and hope the backup data won't be needed for a restore later. This approach is not recommended for fairly obvious reasons: if you don't want to restore data later, then why bother doing it at all? Another solution is to not backup any files that are being used by other applications while backup is running. This way, there is no risk of backing up a file while an application is updating it. However, it is also possible that some important data files that are always opened by an application will never be backed up. Also, if this approach is used, the backup system should attempt to open the file for exclusive write/update privileges to ensure applications don't update it while it is being backed up.

COPY-ON-WRITE One of the most promising technologies for hot backup is called *copy-on-write*. The idea of copy-on-write is to copy old data blocks on disk to a temporary disk location when updates are made to a file or database object that is being backed up. The old block locations and their corresponding location in temporary storage are held in a special bit-map index, similar to a cache index, that the backup system uses to determine if the blocks to be read next need to be read from the temporary location. If so, the backup process is redirected to access the old data blocks from the temporary disk location. When the file or database object is done being backed up, the bit map index is cleared and the blocks in temporary storage are released to be used again. Figure 7-8 shows the basic components of a copy-on-write process.

Copy-on-write functionality has been implemented most often for database backup. Many database companies have copy-on-write functionality built into their products to enable 24x7 operations of their databases. This capability must be matched with a backup software product that takes advantage of this facility. The copy-on-write backup modules are typically more expensive compared to the other software components of a backup system. Oracle, Sybase, Informix, IBM, and Microsoft relational database products all have the capability to provide copy-on-write capability for hot backups. BMC software also has a product suite called Data Tools that is used to manage the copy-on-write capabilities of Oracle and Sybase databases.

Copy-on-write also can be used for hot backup of file systems. The same principals apply for file system backup as they do for database backup.

SOFTWARE SNAPSHOT TECHNOLOGY A similar technology to copy on write is referred to as a *snapshot*. The concept of a snapshot was introduced in Chapter 4 as a way to make a



third copy of data on a mirrored disk. Snapshots can also be created in software to provide a point-in-time image of the file system or database, which can be used to obtain a complete copy of data for backup purposes.

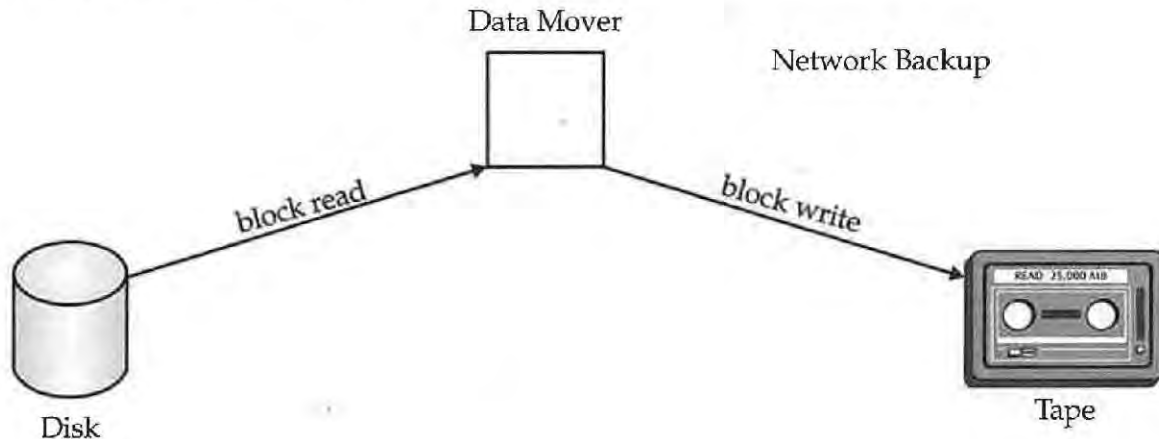
Software snapshots work by keeping a duplicate copy of the file system's or database's block assignments on storage. At any point in time, the "view" of the file system or database is determined by those block assignments. So, if one wants to take an image of the system at any point in time, all one needs to do is preserve the access to those blocks. That is what a software snapshot does. It freezes a view of the block assignments for the system, or some subset of the system, such as a directory or database table. Backup systems can then use a snapshot view of the system for reading blocks during backup.

Software snapshots require new blocks or updated blocks to be written in locations that are not referenced by an existing snapshot block assignment. If new data overwrites a block assignment from a previous snapshot view, that snapshot view loses its integrity. Therefore, there is an inference that software snapshots require sufficient free space on storage to hold all new data as well as the old data. The amount of free space necessary depends on the activity in the file or database system and how much new data is being generated in relation to the existing old data.

An example of a system with built-in software snapshots is the WAFL (write anywhere file layout) file system used by Network Appliance in the Filer products. One of the primary principals of the WAFL file system is that data can be written anywhere there are free blocks in the system. Free blocks in this sense refer to blocks that are not claimed by the current block assignment mapping or some number of snapshot view block assignments. The system provides the ability to maintain multiple snapshot versions of the file system.

The Special Case of Image Backup

While backup operations usually incorporate file system or database facilities for identifying and copying data, it is also possible to copy the contents of a storage device on a block-by-block basis. This approach is known as *image backup* because it makes an image copy of the physical block contents of a storage device or subsystem. The diagram below shows the process of an image backup where an entity referred to as a data mover transfers block data from a disk drive to a tape drive.



Normal restore operations can re-create a complete system with all the data it has, covering several storage devices and subsystems. Image restores, on the other hand, are targeted at individual devices or subsystems. Image backup has the advantage of having brute-force performance for both backup and restore operations. When the data is copied from the backup device(s) to the recovery device(s), the recovery device(s) can be disconnected and reconnected to any system with a compatible file system structure. This theoretically allows a system with multiple storage devices to have its devices restored as images in parallel and then connected to the replacement system.

The primary difficulty in image backup is the problem of restoring individual files and database tables. As the image copy is a block-by-block copy without any information from the file or database system, there is no easy way to identify the blocks containing any individual files or database tables. Even with the knowledge of which blocks correspond to a certain file, these blocks are practically guaranteed not to be located in contiguous sections of tape. Therefore, an index is required that maps files or database objects on tape by their tape block locations. Even so, a restore operation for an individual file or object would necessarily jump from place to place on tape, taking much more time than a similar restore operation from a file-oriented backup tape. It can be done, but is somewhat slower than the block-copy backup operation.

Another weakness of image backup and restore is that empty blocks are copied, which is not very efficient. An image backup takes the same amount of time whether the storage is 10 percent full or 95 percent full.

It is possible to combine the concept of image backup with file system backup. To accomplish this, each file is copied by first retrieving the block locations for the file from the file

system and then moving those blocks to backup media. Notice that the file copy is not performed through the file system interface, but through a device-level block copy operation.

Stac Electronic's Replica and Columbia Data Products' Snapback backup products are based on the concept of performing image copies of files. These products provide a way to mount images of disk storage volumes that are written sequentially on tape. Although the access to files on these tapes is considerably slower than it is on disk, it does give the user a simple file system interface for accessing files on backup tapes. A related technology is Legato's Celestra, which is based on making device-to-device copies in a SAN environment. Celestra is discussed in greater detail in Chapter 10.

RESTORING DATA

Backup and restore are obviously closely linked processes. Data cannot be restored that has not first been backed up. While backup is a prerequisite for restore, one should not expect restores to be a simple inverse process of backups. Restores are much more complicated than backups and tend to give administrators more problems.

Integrating Restore with File Systems and Databases

Backup operations write data to devices and media that are controlled by the backup engine and not accessed by user applications. While the reading of backup data is usually done using the source system's file or database system, the actual writing of backup data onto backup media is independent of the file or database system it was copied from.

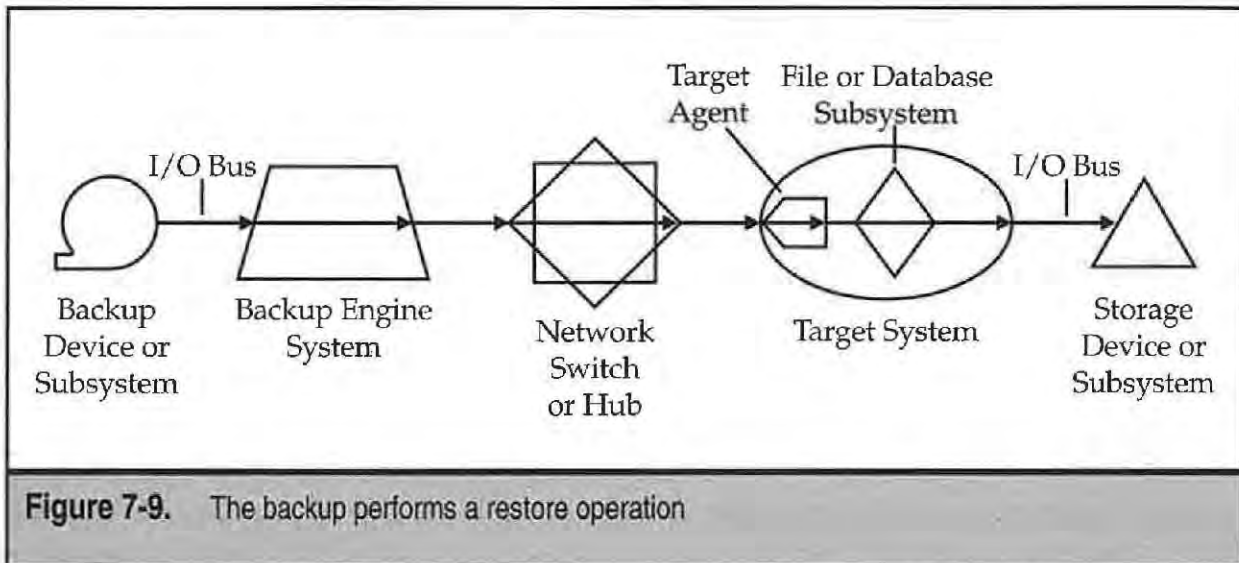
Restores, on the other hand, create data in the file or database system. The term *restore target* is used here to differentiate a system that has data restored to it from a system that had its data backed up. They are often the same system, but they can also be completely different systems. Just as the source system has an agent for handling the details of system security as well as file systems and databases, the target system also typically uses an agent to ensure data is transferred correctly onto its storage devices or subsystems (Figure 7-9).

Types of Restore Operations

Just as there are common backup operations most sites are familiar with, there are also common restore operations that are called for most often by storage and network administrators. Whereas most backup operations are performed automatically and unattended, most restore operations are manual where an administrator selects the data entity to restore and possibly also the media to restore from. The common types of restore operations are:

- ▼ Full restores
- File or table restores
- ▲ Redirected restores

We'll discuss each of these briefly in the following sections.



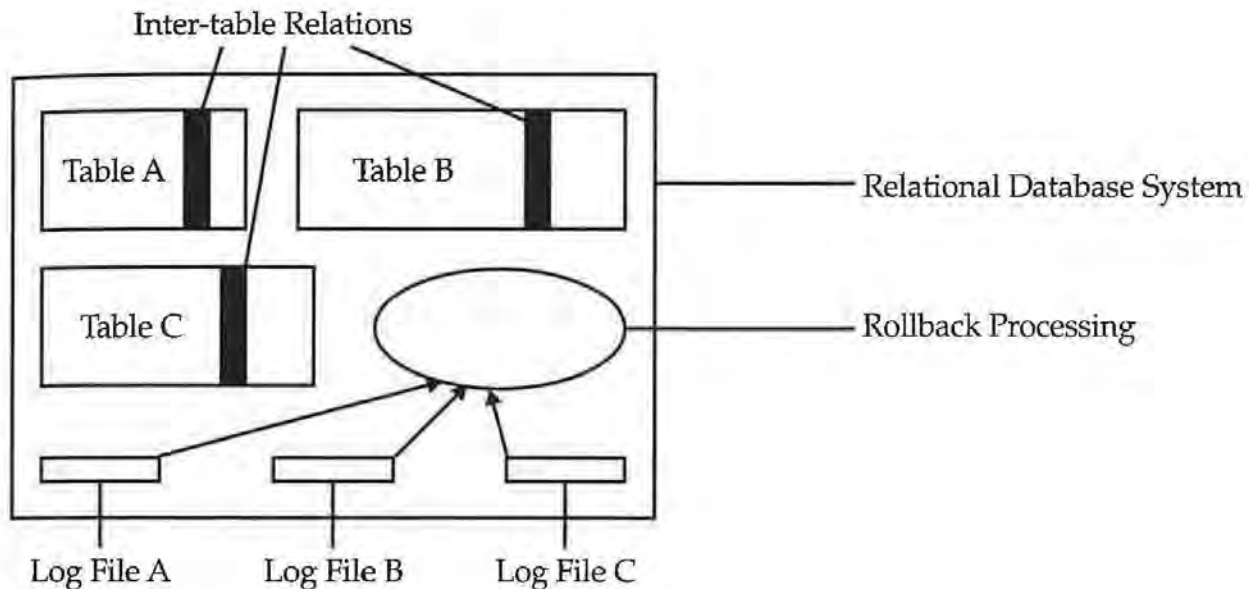
The Full Restore Process

The only true single-pass, full restore operation is an image restore. The idea of an image restore is that a small, specialized operating system is loaded on the target system (system being restored from backup data) that has the required device drivers to perform all necessary disk, tape, and network operations. These can include disk partitioning, block copy, and network transfer operations. Notice that image restores can only work if the disk partitions being restored to are the same size as they were previously. Image restores can be accomplished by a tape drive attached directly to the target system, or by a backup engine connected to the network that manages tape drive operations.

Otherwise, there is a sequence of events that needs to be followed to provide accurate full restore operations over the network. Typically, the server's disk partitions and logical volume assignments must be made first before reloading the target system's operating system. When the operating system is running, the appropriate backup agent can be loaded to provide restore services for the backup engine. Then the file system directory structure, files, user security information, and other system resources, such as the Windows registry or the NDS (NetWare Directory Services) tree, are also restored.

The situation for databases is usually considerably more complicated and may require the expertise of the database administrator to ensure the database can be brought back to a functioning system. Databases typically have more interdependent data components that depend on the organization of the database. In general, these database can be restored without a lot of difficulty if the databases are shut down in an orderly fashion. However, if the restore is being done to recover from an unexpected shutdown or failure, the various components probably need to be restored in the proper sequence. Careful attention needs to be paid to the database log files to identify any partially completed write operations and determine if they should be rolled back to ensure a consistent whole database image. Readers interested in more detailed explanations of database recoveries are

advised to read their database manuals and other books on the subject. The drawing below shows some of the basic software components involved in database backup and recovery.

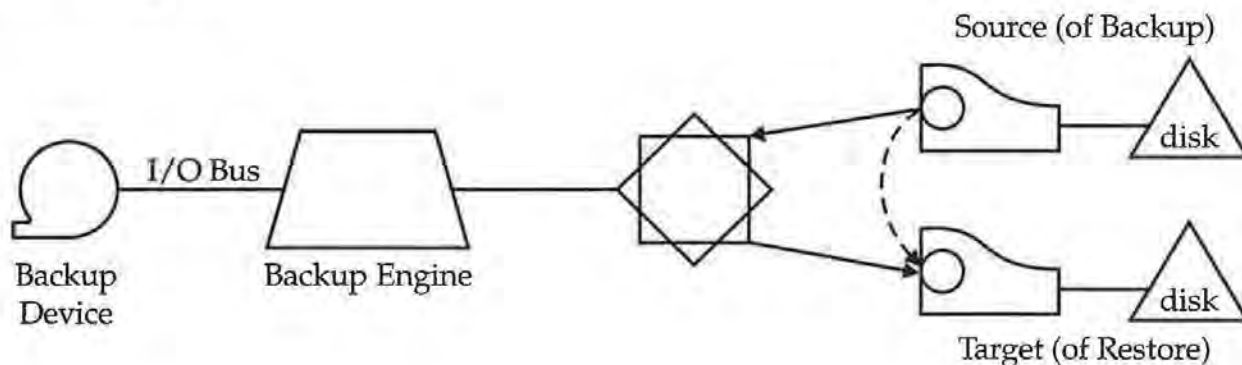


Restoring Files, Tables, and Other Objects

Another common restore operation is for an individual file, table, or other system entity. In this case, the administrator uses the backup system database or other system control mechanism to select the item to be restored. If a tape autoloader or library is used, the restore process may be able to continue without the administrator's attention; otherwise, the correct tape will need to be inserted in the tape device or subsystem.

Restoring Data to Another Location

Another restore operation is when data is restored to a system or file system location that is different than the one it was copied from. Sometimes this is referred to as a redirected restore. A redirected restore is commonly done when tape is being used to exchange data between systems.



Importance of Media Management to Restores

Backups will write data to almost any tape that is available, even if it's not necessarily the most optimal tape to write to. The goal of backup is typically to copy as much of the data that needs to be backed up. Restores, on the other hand, cannot allow this kind of ambiguity. A tape either has the data that is required or it does not. Correct identification of tapes is essential for network restore operations. When there are 50 to 150 tapes to choose from, as is the case for an average network backup system, a trial-and-error method of locating tapes is not very efficient.

Tapes need external labels if they are going to be quickly identified for restores. Hopefully, the external labels of the tapes accurately and adequately reflect the contents of the tape. This might seem obvious, but incorrect labels are a common mistake that can cause a great deal of frustration. Many times, this situation does not occur because the tape was mislabeled but results from how the tape was used for backup operations. For example, a tape with a label of Wednesday that is used on Sunday will not contain the expected data. This simple example underscores the importance of monitoring the use of backup tapes to ensure they are being used in their proper sequence.

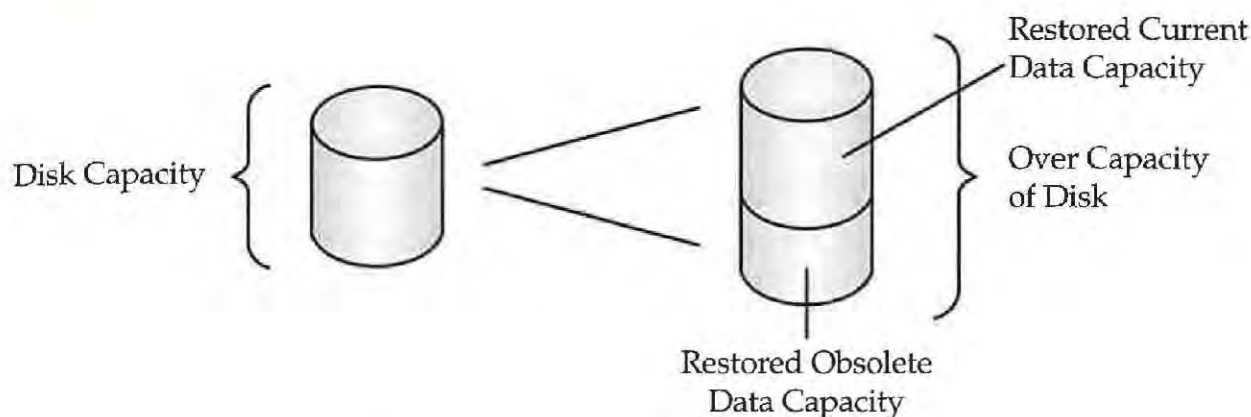
If tapes are not labeled correctly or if the labels cannot be understood, tapes should be inserted into a tape drive and read with a backup/utility program to view the tape header and tape directory information written there. Although it takes time, one can usually determine which tapes are needed for restoring by carefully comparing the header and directory information on the tapes.

The use of tape automation in autoloaders and libraries can help alleviate these kinds of problems. Backup software products usually have the ability to force autoloaders and libraries to learn the contents of tapes stored in them. These tape inventory operations read the electronic labels on tape and correctly identify it, regardless of its external label.

Many tape libraries implement bar code readers to speed the tape inventory process. Instead of loading each tape, the bar code reader simply reads the bar code and links it to the name used by the backup system. Bar code readers can save a great deal of time, especially with libraries that contain many tapes. However, it is somewhat dangerous to rely completely on bar code labels, as they usually don't mean much to human beings. Therefore, it is a good idea to use both bar code and human-readable labels.

Restoring Outdated Data

Whatever the reason for mistakenly using the wrong tapes, the results can be problematic. When the wrong tapes are used, the potential exists to restore data that no longer exists on the server or exists in some other location. For instance, a large application's data files may have been on one volume a month ago but been moved to another volume two weeks ago. If a tape is used that contains backup data that is over a month old, it is possible that large amounts of data will be restored to both volumes—and overfilling them, as shown on the following diagram.



BACKING UP AND RESTORING SECURITY DATA

There are many books written on the important topic of system security. This book is not explicitly concerned with this topic, because it is not a part of typical storage and I/O operations. However, backup and recovery technology is expected to be able to restore complete file systems along with the ability for end-users to access data, just as they did before the disaster. This section looks at the various system entities providing access control and the ways they can be backed up and restored correctly.

User access control is a common feature of all server systems. Access control enables administrators to establish access privileges and exclusions for specific users and groups of users regarding specific data objects. User access is an important component in the overall system security implementation on any system.

Most access control schemes can be broken down into two system components:

- ▼ The user trying to gain access
- ▲ The thing that can be accessed—usually files and directories

Users are typically managed through an administration application where they are created, deleted, and have their passwords set. Along with users, system administrators can also create groups that users can belong to. Groups allow administrators to assign the same access control to many users at once. In addition, sometimes users are assigned a set of security characteristics that further define their ability to access data.

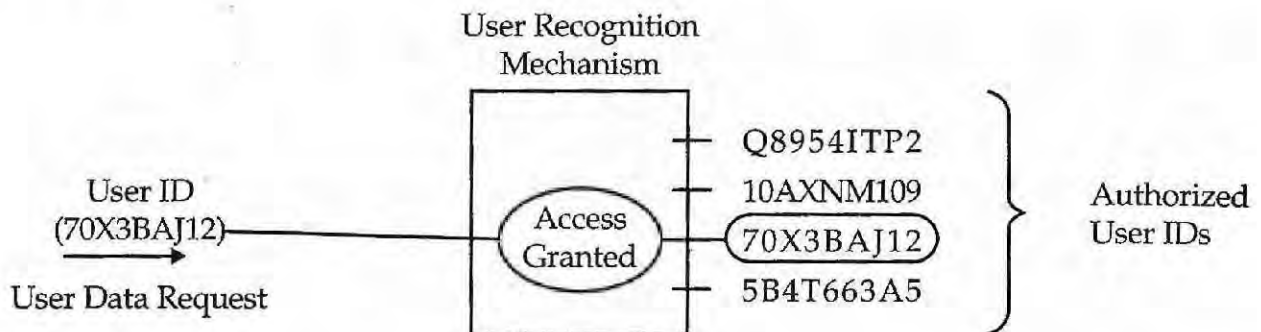
User and File System Security Components

Users access files. Whether it is to run an application or to open a file from within that application, users access a file in order to get their work done. The access to these files can be established at the directory (or folder) level and also at the file level. Directory-level security is applied to all the files within a directory and allows a user with access to the direc-

tory to access all the files it contains. File-level security is applied to an individual file. A user with access at the directory level may not be able to access an individual file in that directory if it has file-level security specifying more restricted access.

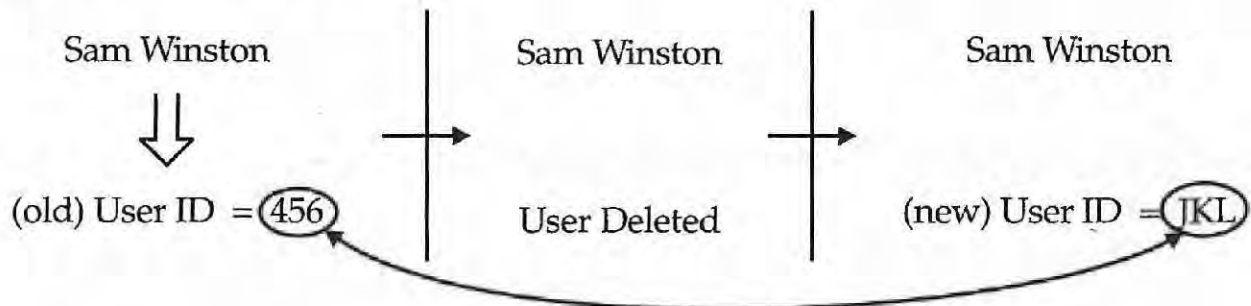
Caller ID as an Analogy for User IDs

It is important to understand that security is established using both user and file system entities. Security operations are analogous to using Caller ID, where a user calls for a file and the system checks the user's ID to see if it recognizes the caller. If there is a match between the file system's authorized users and the user requesting access, then the user is allowed to access the file. If there is no match, the user is not granted access. The figure below illustrates.



System user IDs are created for internal system use to represent users as objects. Numerical representations are much easier to manage than alphanumeric names. Unlike a person's phone number, system user IDs are typically not known to their owners—they are probably not even known to system administrators. However, they are known to the file or database system. It's like having an unlisted telephone number and the person you call can recognize your number but you don't know it yourself.

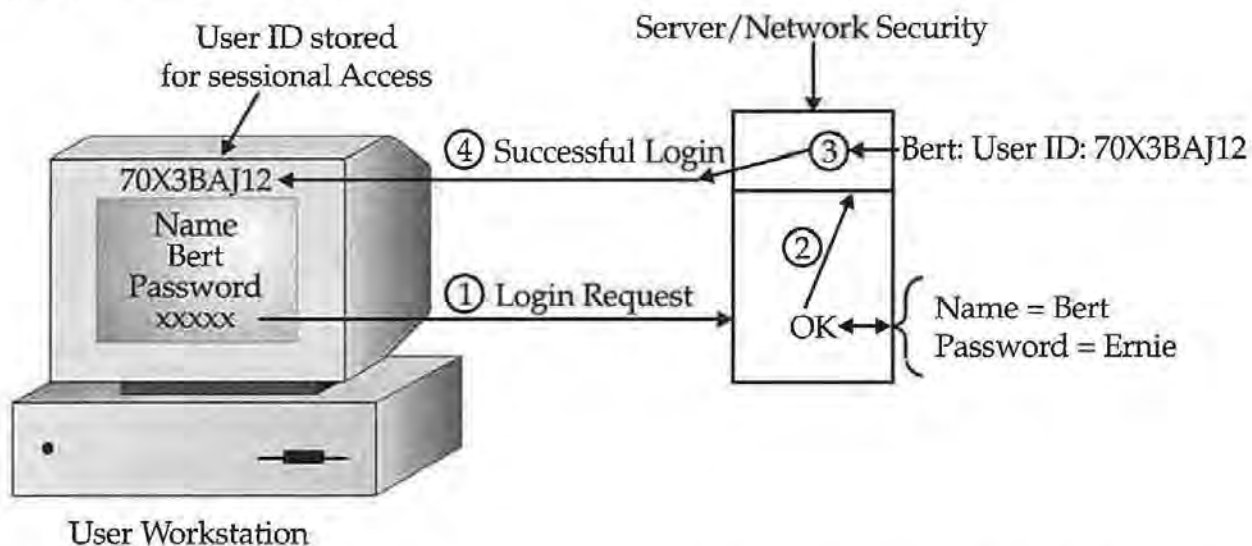
Security systems generate user IDs randomly. For that reason, a user that is created once, then is removed from the system and created again will have a different user ID number assigned the second time.



This means that users known to a file system entity by their user ID numbers will not be recognized by the entity after they are deleted and re-created. The users (or administrator, actually) will have to make their number known to the entity again by joining a group that is known, or by having their access assigned explicitly.

User Security Data

The login process authenticates users by password or some other means and assigns a user ID to a user's communication session. This login process uses a special user security data structure that maps the user's visible login ID to the system's internal user ID. This user security data structure is a required component for establishing correct access to file and database resources.



For the purposes of this chapter, it is important that this data be able to be backed up and restored. If it is not, users and their respective random-generated user IDs will have to be re-created and the caller ID system will be broken.

File System Metadata

When a user attempts to access a file system object, its system "calls" the host file system on behalf of the user and makes a request for the object. For the file system to respond, it has to be able to compare the caller's user ID with the user ID numbers it recognizes. That means there has to be someplace to look up known user IDs. Typically this is done within data structures contained in the file system structure itself, although it can also be done in a separate system file. This is part of what comprises the *metadata* of the file system object.

Access control lists, or ACLs, are commonly used in file system security implementations. ACLs are lists of user numbers and group numbers that are authorized to access the contents of the file system object.

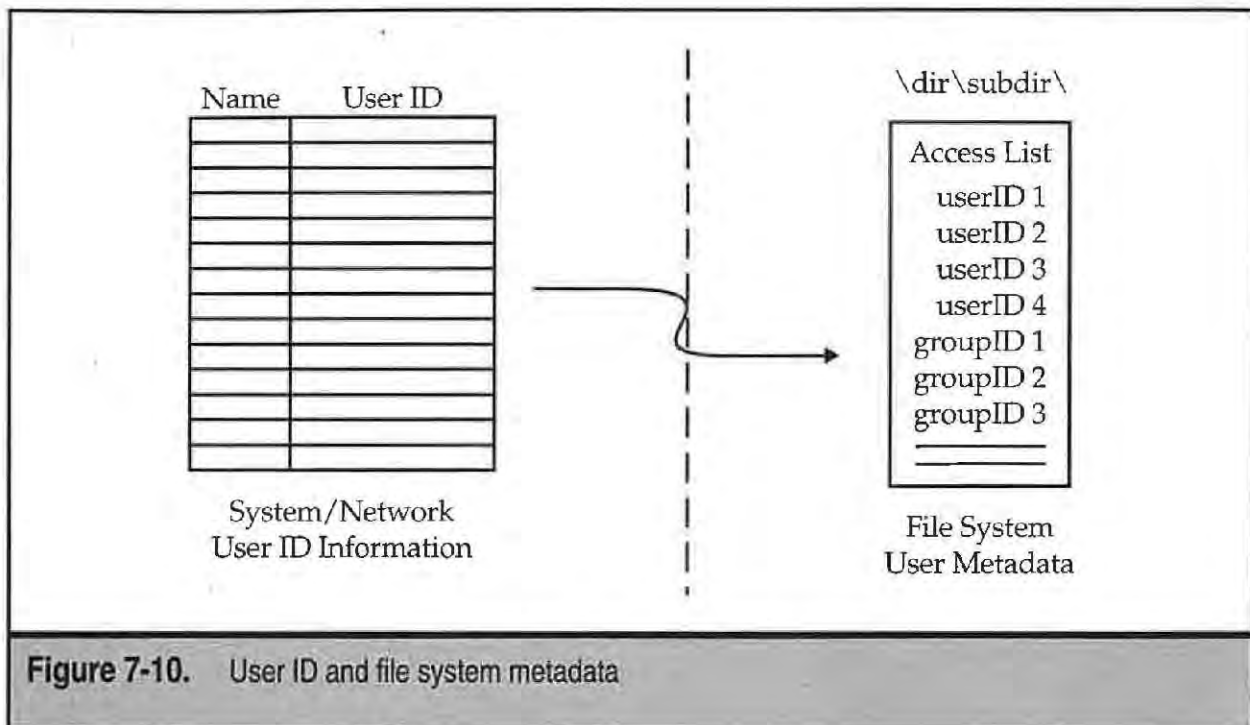


Figure 7-10. User ID and file system metadata

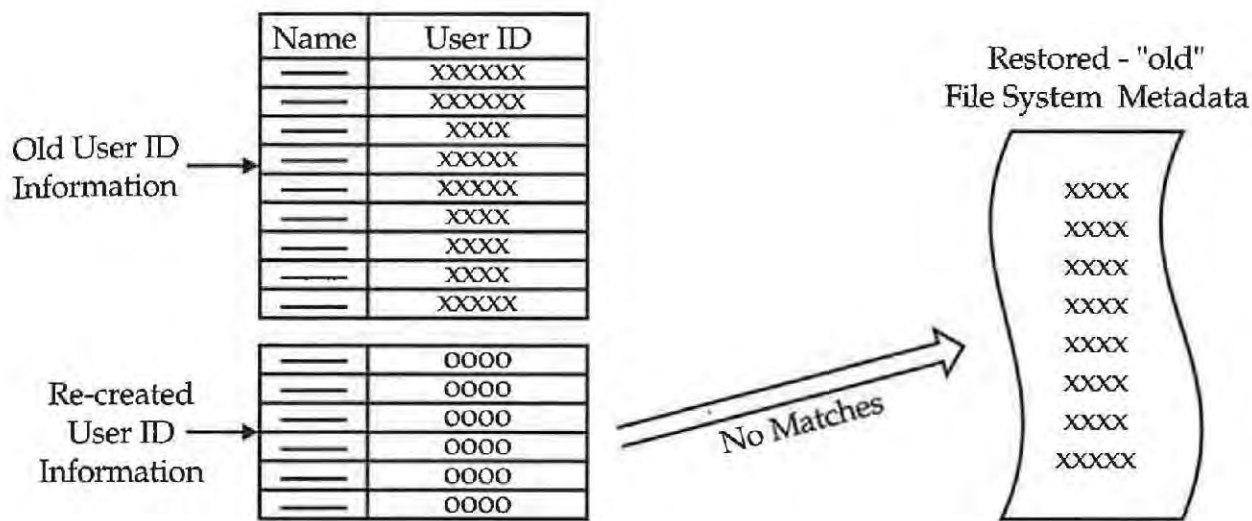
User Security Data and File System Metadata as a Single System

So, there are two independent data structures that are used to implement file system access. There is the user security data structure used to assign the internal user IDs during user login processing, and there is the file system metadata that determines which user IDs can access objects in the file system. These two data structures are obviously very closely linked and form two halves of a single access system. Figure 7-10 shows these two data structures.

Maintaining File System Access During Disaster Recovery

So, it should be somewhat evident that this security and access system requires both parts. The most interesting thing about this, for the purposes of this chapter, is what it means for disaster recovery situations. The type of disaster considered here is a total disaster where the complete system has to be rebuilt from scratch. The two parts of the security/access system can function without each other but do not work properly as a single system. In fact, the file system needs to be thought of this way also. For instance, restoring files and user security data—but not metadata—means that new metadata will need to be created by system administrators by manually reassigning users to file system objects.

Similarly, consider a restore process where the file system and its metadata were restored correctly but the user ID information could not be. This time, the administrator has to manually re-create each user and assign them manually to file system objects. This security disconnect is illustrated in the following.



This means that the two components of the security/access system need to be backed up and restored as a matched pair. While most network backup software products provide this capability for the various platforms they protect, it is up to system administrators to ensure that it is being done correctly.

Network User Security Systems

There are basically three types of user ID security systems used in servers today:

- ▼ Local
- Domain
- ▲ Networking

The first of these, local user ID security, has been described in the sections above. Every network server operating system has some form of local security available to it. An example of this type of system was the Novell NetWare 2.x and 3.x bindery files. The NetWare bindery was a database system that contained records of all the user IDs for a single server. The bindery system was fairly straightforward and easy to work with on a single system. However, it was not very effective in helping administrators establish management over multiple servers with the same user base.

DISASTER RECOVERY IN DOMAIN CONTROLLER NETWORKS The second type of user ID system uses the concept of a domain controller. A domain controller typically is based on the idea that access to data on several servers can be managed with a centralized user security system. In a domain controller scenario, each server's file system has its own user recognition data held within the file system, but the user ID data is stored by a system called the domain controller. After users log into the domain controller, they use their domain user ID to make calls to the various servers they access. The domain user ID is then used by each individual server to compare against their file system metadata.

One of the advantages of using domain controllers is that there is less user ID information to back up and restore. Typically, one system is defined as the domain controller and one or two others are backup or secondary domain controllers. It is important to ensure that the domain and backup domain controller's domain user ID data is being backed up and can be restored. Other than that, user ID information does not have to be backed up from the other servers.

When a server that is not a domain controller is restored, it does not need to restore user ID information. As long as the server is part of the domain, it does not need to supply its own user ID data. Of course, it is important that the complete file system metadata be restored.

DISASTER RECOVERY IN DIRECTORY SERVICE NETWORKS The last type of user ID security system makes use of a directory service. A great deal has been written and discussed about directory services in recent years. When the directory service spans a wide area connection, its becomes an extremely useful disaster recovery tool.

Directory services are typically specialized database applications that contain information about various resources in the network and the users/applications that request access to them. These directory service databases are periodically replicated through the network to ensure each server has up-to-date information about network resources. Similar to the domain server concept discussed above, when users log into the directory service they are authenticated and given network user IDs to access resources in the network. Unlike domain services, directory services can be globally accessed in wide area network locations around the world.

So, when a disaster occurs in a directory service network, it is possible to recover all user ID information through the usual process of replicating the directory service database from other systems in the network (Figure 7-11). Therefore, there is no disaster recovery process needed for the network user ID information as long as the server is receiving the information from other servers in the network. Replicating network user ID information in this manner is generally preferable to restoring the directory service database from tape.

NOTE: An exception to this could occur when the network connection between servers is slow and the size of the directory database is large. In that case, it may be preferable to restore a local copy of the database from tape and then synchronize the restored directory database with the copies on other servers.

TAPE ROTATION

Organizations often want to keep several different versions of backup copies to satisfy business requirements. For example, a business may want to keep copies of data from the end of each month for business, legal, or audit purposes. The structure of tape names, their operations, and frequency of use is known as *tape rotation*, or tape scheduling.

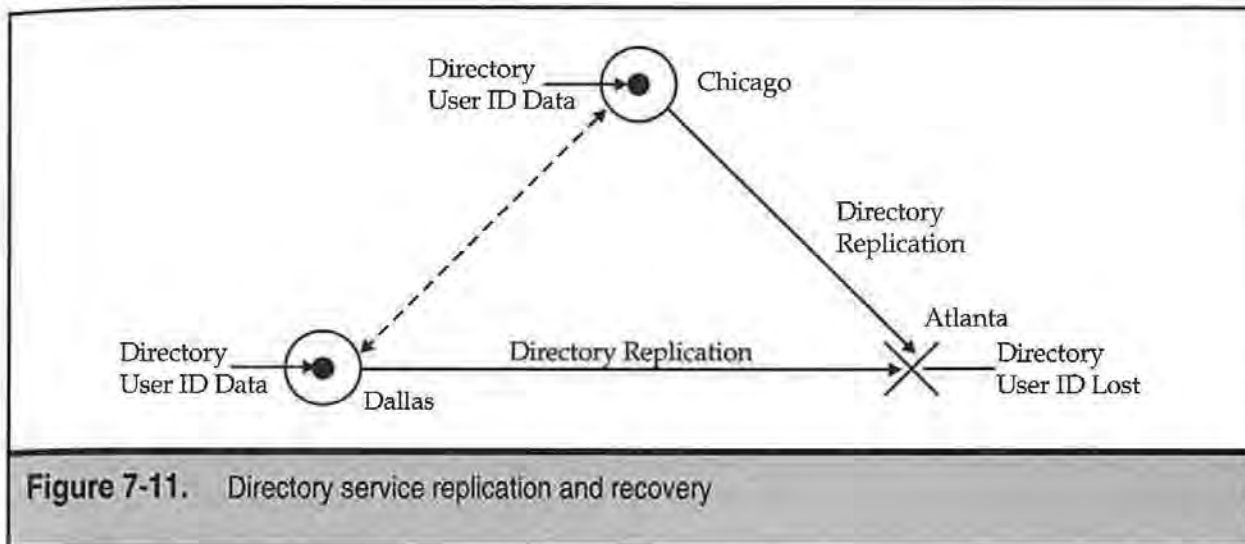


Figure 7-11. Directory service replication and recovery

The Necessity of Tape Rotation

The implication of backup is that there is a requirement to be able to restore data at some future time. Unfortunately, for whatever reason, human beings tend to have some difficulties thinking about the past and the future at the same time. This creates some confusion where tape backup and tape rotation is concerned.

The core concept of tape rotation is ensuring restore integrity. Tape rotation is needed, in part, because backup operations are not really concerned with the integrity of the restores—just the integrity of the backup. For example, when backups start there is a clear picture of where the data is; it then does its best to ensure that data is written to whatever tapes are available to work with.

Restoring data is an inverse process. In restores, one knows exactly where the data needs to go, but it is usually more difficult figuring out where it's coming from. For that reason, a structured approach is needed to be able to identify where data is on various tapes to enable an administrator to easily direct restores.

This section explores several options for tape rotation. It's important to ensure the media has the expected data. Surprises concerning tape contents are usually of the unpleasant variety. One of the best ways to avoid such surprises is to implement a well-defined tape rotation scheme. The opposite, a poorly implemented tape rotation scheme, is practically guaranteed to result in lost backup data that cannot be restored later.

Reusing or Recycling Tapes

One of the basic ideas of tape rotation is that tapes will be reused on a periodic basis. While tapes are relatively inexpensive for the capacity they provide, they cost enough that most people don't want to use them only once and then place them on the shelf forever. More important, if tapes were only used once, the backup engine's database information would become huge and require constant maintenance.

From a wear and tear perspective, most tapes today are made to be able to last over 2,000 passes over the head assembly. This number is not as high as it might appear, as several tape device operations will cause a tape pass to occur even though it is not apparent to the administrator. But the fact remains that tape can be used many times and most administrators like it that way. When tapes are reused or inserted again for backup operations, the data that is on them will likely be overwritten, destroying the previous historical data.

Matching Business Cycles with Rotation Schedules

Typically, tape rotation matches business cycles to enable historically relevant versions of backup data to be created. In other words, the goal is to keep backup data that accurately reflects the status of the business. This takes into account the completion of regular periodic business cycles, such as the consolidation of sales and inventory data as well as the preparation of financial documents. These business cycles are processed differently according to the organization's requirements, but it is common that they are done quarterly, monthly, biweekly, or weekly. Often, businesses require these periodic backups to be archived for historical record preservation as well as for audit purposes.

Enabling Efficient Backup Operations

Another reason for tape rotation is to implement efficient daily backup operations. As it turns out, certain media schedules allow more efficient backup operations to run than others. For example, if a simple two-tape rotation schedule is used where each tape is used on alternate days, it will probably be necessary to run a full backup every day to ensure complete restores. Other rotation schedules, as discussed in following sections, allow smaller and faster incremental backup operations.

Commonly Used Tape Rotation Models

There are several types of rotation models used and many variations on them. In practice, it is unlikely that any one rotation model will fit the needs of the business precisely, due to subtle variations in business and data processing cycles.

Weekly Rotation

Weekly tape rotation is one of the simplest schemes available. Basically, one tape or group of tapes is used for an entire week while other tapes are stored off-site. The rotation cycle begins with a full backup operation on the first night's backup. On subsequent nights, incremental backups are run and appended to tape or written to additional tapes. Restores are relatively simple, as the number of tapes containing backup data is relatively limited. Weekly rotation works best when rotation days occur on days when the backup window is longest. For most organizations, this operation will be started on a Friday night and allowed to run over the weekend, if necessary.

There are several variations that can be used with the weekly rotation model. Two sets of tapes could be alternated or a group of tapes could be created for each week of the month, including months with five rotation days.

In many respects, the weekly rotation scheme is an excellent choice. It can be adjusted fairly easily to accommodate special circumstances such as holiday weekends and end-of-quarter archiving. More important, it is fairly easy to understand and communicate to system administrators and off-site courier services. However, the exposure for data loss due to a site disaster is six days. For example, a disaster that occurs on the sixth day after rotation could destroy all the data on tape that had been created in the previous six days.

Daily Rotation

To reduce the exposure of catastrophic data loss, a daily rotation model could be used where different tapes are used every day. This requires the ability to complete full backups every night, something that cannot be done in many locations. However, if nightly full backups can be done without disrupting normal operations, the protection of server data is excellent. This is a very simple rotation to manage and communicate with administrators and service providers.

Monthly Rotation

Another variation on daily and weekly rotation is a monthly rotation. Although it is not often used, it can be successfully implemented when matched with systems that do not generate a great deal of new data. The general idea of monthly rotation is to make a full backup at the end or beginning of the month and incremental or differential backups on the rest of the days of the month.

Before thinking that a monthly rotation will work in your environment, there are some important considerations. If the daily operations during the month are incremental backups, restores could require the use of many tapes. If you append the daily incrementals to the end of the previously used tape(s), a tape failure could result in the loss of a whole month's worth of data. On the other hand, if different tapes are used every day for incremental backups, the restore process could use 20 or more tapes, which can certainly be problematic—just in assembling the tapes for the restore.

For those reasons, daily differential backups are recommended with monthly rotations. This way, only two tapes are needed for restores—the monthly full and the last daily differential. The implication here is that the amount of new data created during the month is somewhat limited, so that the differential backup does not run too long. In either case, whether or not incremental or differential backups are used, it is a good idea to make copies of the daily backups to maintain both local and remote copies. With only one set of tapes covering an entire month, it's likely that too much data could be lost in a site disaster or that end-user requests to restore individual files would take far too long to service.

Systems that create a lot of new data should not use a monthly rotation. Large amounts of new data will result in more tapes than are easily managed for both backups and restores.

Grandfather, Father, Son

The grandfather, father, son rotation model, also called GFS for short, is a combination of the three rotation models above. The basic idea is that there are tapes assigned to days of the week, weeks in the month, and months in the year.

In the GFS model, full backups are typically performed every weekend and incremental or differential backup are performed on the other days of the week. On either the first or last weekend of the month, the full backup is written to a monthly tape; on every other weekend, the full backup is written to a weekly tape. Daily tapes are used on non-rotation days of the week. Table 7-1 shows the tapes one could expect to use with a GFS rotation model.

The GFS rotation scheme is popular for good reasons. For starters, it is easy to understand and manage. More important, it has built-in retention of data that keeps weekly full backups available for a month, and monthly full backups can be kept available for as long as desired. Table 7-1 shows four monthly tapes, but there is no reason why four monthly tapes need to be used. The number of monthly tapes depends on how long you wish to keep copies of data. There could be three monthly tapes, one for each month in a quarter, twelve monthly tapes for each month in a year, or a never-ending number of monthly tapes, if they are never rotated. This makes it very easy to match a wide variety of data retention requirements.

An important point to consider when keeping monthly tapes for extended periods is the impact it can have on the backup system database. The size of the backup system database can become quite large and have a significant impact on the performance of the backup system. For that reason, it is a good idea to periodically delete the database records for older monthly backups, even if the tapes are not going to be rotated and overwritten. It is not likely that these tapes will need to be used again, but if they are, the data can still be restored—it will just take more effort and time.

Using Rotation Models to Balance Backup Performance and Restore Performance

As it turns out, the amount of time spent during restores that results from rotation scheduling is inversely proportional to the time it takes to run backups. In other words, rotations that enable more efficient backup operations result in longer restore operations.

Efficient backups involve backing up only the necessary changed data, and ignoring data that has already been written to tape(s). While the backup operation can proceed

Monthly	Weekly	Daily
January	First weekend	Monday
February	Second weekend	Tuesday
March	Third weekend	Wednesday
April	Fourth weekend	Thursday

Table 7-1. Tapes Used in a GFS Rotation

very quickly, it can result in many tapes being needed for the restore. To demonstrate the idea, we'll use an extreme example in which a system was backed up once after it was installed, and from that point on, only the changed data was backed up.

Consider such a perpetual incremental backup system that runs over nine weeks. On the first day, the full backup operation uses five tapes. Each day after that, a new tape is used for an incremental backup. By the end of nine weeks, 67 tapes have been used. If this system ever needed to be restored, it might be necessary to use all 67 tapes. The actual number of tapes required to restore depends on how often new files are created and whether or not newer versions are created later. The worst-case scenario occurs when a new file is created every day that becomes permanent data and is never updated but needs to be restored as part of the complete system.

While the fastest possible backups are accomplished with only incremental operations, the most direct restore operations occur as a result of running full backups. Full backups that fit on a single tape work best if tape capacity is sufficient, but whether it is one tape or several, the important point is that the system was written as a consistent set of data that can be restored in a single operation. Restore operations that require the administrator to pick through tapes looking for the best combination are time consuming and lead to mistakes.

Another reason for wanting to limit the number of tapes for restore is to prevent a tape failure from making it impossible to restore data. In the example given above with perpetual incremental backups, the failure of any one tape could prevent the server from being completely restored.

So, one of the goals of tape rotation should be to balance the performance and efficiency of backup with the ease and reliability desired for restores. That was the motivation for the invention of differential backup operations. There are some backup products that allow data sets on various tapes to be merged to create a set of tapes that have the complete system image on it. For instance, they would use the data set from the last full backup and merge it with incremental data sets from other tapes and days of the week. This way, a composite tape or tape set can be generated that provides all the data needed for a complete restore image. The reader should be aware that this is a fairly sophisticated feature not supported by all backup software products.

Marking Deleted Files

Files that once existed on the system but were later deleted should not be restored. It is very difficult to go back after a restore operation and remove files that should not have been restored. While this might seem like a nuisance, it is actually much worse than that: if deleted files are incorrectly restored, it can fill the disk volume and result in not being able to restore data that is needed. Then, it is necessary to stop the restore process and remove files that are not needed. When the pressure is on and people are anxiously awaiting their data, this is much easier said than done.

This is the basic problem with the perpetual incremental approach, where changed data is backed up continually every day. In general, this is a problem for any tape rotation schedule, but some more than others. As a rule of thumb, you should expect to restore

some files that have been deleted from your system whenever a full server restore is performed. The number of files can be approximated as the number of files deleted since the last full backup operation was run. Full backups are typically used by backup software to determine the "baseline" state of the system. Restore operations will restore all the data from the last full backup and then add newer files created since that time. Files deleted after that full backup operation will likely be restored improperly.

Off-Site Tape Rotation

Backup tapes should be kept off-site for some period of time, enabling a system restore if a disaster wipes out the organization's systems. When such a disaster strikes, this supply of off-site tapes is the only means to restoring functioning systems. Obviously, if an organization depends heavily on data processing for its business, such as an Internet business, the ability to re-create recent data from off-site tapes is extremely important.

There are services in most metropolitan areas that can be used to deliver and store tapes in off-site storage vaults. These services were developed years ago primarily for the data on mainframe systems, but they can also be used for data tapes from open systems backup. The selection of a service and the verification of their practices could be an important part of a disaster recovery plan.

Duplicating and Mirroring Tapes for On- and Off-Site Protection

Some businesses have requirements to have all their data available online and offline at all times. As discussed previously in the discussion on threats to data, the most common threat is a malfunctioning end-user that deletes or changes a file so badly that it is easier to restore a previous version than it is to work with the existing file. This causes a real problem for system administrators who want to be able to restore all their data from off-site tapes and who also need to be able to restore individual files for their end-users. Obviously a tape that is off-site for a week does not help the end-user very much who needed their file yesterday.

The most common solutions to this problem are to use mirrored tape drives, make copies of backup tapes, or run two separate backup operations covering the same data. Mirrored tape drive subsystems are available from several manufacturers, as is tape-mirroring software. Similar to disk mirrors, tape mirrors make two tape drives appear as one virtual drive.

Duplicating tapes is fairly simple to understand. Take a tape that has been written to and make a media-to-media copy of it before sending one of the tapes off-site. Most backup software products provide a function for making this kind of copy.

The third alternative, running two backup operations, is much more problematic. It involves managing an additional backup process, which not only takes more administrative effort, but also doubles the load on the network. More important, it probably won't deliver the desired effect, due to the way archive bits are reset by backup system. Even though backup systems can behave differently with respect to archive bits, there are still a number of things that can go wrong, especially the human factor that has to try to make sense of it all. For that reason, it is not recommended.

Maintaining Rotation Order

When tapes are in tape drives and backups are running, there is an excellent chance data that is on them will be overwritten with new data. For the most part, this is the way it should work. However, if the wrong tapes are put in the drive, it is possible that data that is needed for some other purpose could be overwritten and lost. For instance, a tape that was supposed to have been sent to an off-site storage facility may have been left behind in a tape drive. When backup runs again, it is possible that the tape will be overwritten with new data, making it impossible to complete a full restore from off-site tapes. For that reason, it is important to maintain the order of rotation. While it may not seem like such a big deal at the time the tape is used, it could become a major problem in the future if the data is needed but no longer exists.

Making Things Easier and More Accurate with Tape Automation

Media management, particularly tape rotation, can be simplified with the use of tape automation equipment. For starters, automation makes it much less likely to have human error screw up operations (someone using the wrong tape, for instance). In addition, multitape restorations can be much easier if administrators no longer have to search for each tape; the tape subsystem can do it for them. Finally, tape automation provides a good place to store tapes. They are safer in an autoloader or library than they are stacked in a cube or in a desk drawer.

PROBLEM AREAS FOR BACKUP AND RECOVERY

Problems with network backup and recovery are common. Some of them are operational, some are environmental, and others have to do with the technology of networking. In this section, we'll review some of the typical problems affecting the success of backups and restores.

Six Reasons Why Backups Fail

Let's now examine backup-specific problems and discuss ways they could be avoided.

Inadequate Oomph!

One of the most common problems with backup systems is trying to do too much with too little. Often the requirements for backup are overlooked and not given adequate analysis. Backup is a big job that involves heavier network and I/O traffic than any other application. If it is not analyzed as such, the backup system is likely to be inadequate.

One way to prevent this from happening is to analyze the network backup system as the primary application in the network. In other words, try to understand the complete network as existing for the purpose of performing backing. What would you do differently if that were the case? This is covered in more detail at the end of the chapter.

Insufficient Management Resources

Without sufficient oversight of the backup process there is an excellent chance that nobody will know when backup is not working correctly. It also means that administrator errors are more likely to be made. It's common to underestimate the administrative effort needed to ensure backups are reliable. A yearly audit of backup processing should help determine if adequate resources are being used.

Insufficient Capacity and Time

If the media and devices you are using do not have adequate capacity to complete the backup job, there will obviously be some amount of data that does not get backed up. Automated tape changers and libraries can overcome most capacity problems. It is necessary to monitor tapes in autoloaders and libraries to make sure tapes are available for the next expected operation, including the rotation schedule.

Ample capacity can't help, however, if there is not enough time to finish backups before they have to be stopped. Some organizations require backup operations to be completed by a certain time, such as the start of the next business day, to ensure end-user response times. Therefore, backup operations may have to be stopped prematurely by an administrator. Obviously, data that doesn't get backed up can never be restored.

Using a dedicated network for backup may alleviate this problem. Otherwise, it may be necessary to look at other alternatives, such as those discussed in Chapter 10 for solving backup performance problems.

Hardware Failures

Occasionally, things go wrong with devices. Over the last several years this situation has improved a great deal with newer tape and media technologies and users' understanding of how to use and maintain their devices. It's important to realize that a tape drive is a highly sophisticated electromechanical device that is exposed to dust, dirt, and airborne contaminants. In other words, tape equipment works a lot better when it is placed in a controlled environment with controlled humidity and dust. Tape heads require cleaning. They become dirty through external dust and also from material that sheds from tapes. The more dust and dirt there is on the heads, the more likely it is to have errors. Weekly cleaning with nonabrasive head cleaners are recommended.

Improper termination of the SCSI bus causes data transfer errors. Although the problem is the bus, termination problems appear to the backup software as either hardware failures or media failures. Terminating resistors are needed on both ends of the bus. Host I/O controllers have electronic terminators and most devices use external terminators. The terminator must match the characteristics of the bus, 8-bit or 16-bit and single-ended or differential.

Media Failures

Tapes exist in a challenging environment. They are wound and unwound in tight spools and handled by complex drive transport mechanisms where they are exposed to external el-

ements and come into contact with rollers and capstans inside the tape drive. They are used and often reused over and over. Eventually they wear out. Care and maintenance of tapes is discussed earlier in this chapter in the section called "Caring for and Storing Tapes."

Network Problems

Network backup creates huge data transfers that can flood a network, causing a variety of congestion-related problems, such as slow performance and transfer failures. It's not unusual for network routers, hubs, and switches to become congested and drop packets. For most data processing applications, dropped packets cause the data to be transferred again, which degrades performance but maintains the connection. When the level of network traffic is heavy enough, as can occur during backup operations, it is possible that retransmissions can also have packets dropped. If packets continue to be dropped, the network communications session will time-out and the backup data transfer will fail. It may be necessary to consider a dedicated backup network or the SAN remedies discussed in Chapter 10.

Six Reasons Why Restores Fail

In this section, we discuss the most common reasons restore operations fail. In general, restore operations are much more difficult than backups, because restores create data as opposed to simply making copies. Whenever data is created, there are several operations and system checks that can make things go haywire.

Lack of Practice, Planning, and Testing

Restores often fail because administrators are unfamiliar with the necessary procedures. Therefore, the first rule of disaster readiness is to practice recovering data. Keep in mind that there are many types of restore operations and that they don't all work the same way. Here are a few exercises to help prepare for an unexpected restore scenario:

- ▼ Restore an entire server.
- Restore a disk volume, or logical disk drive.
- Restore a directory to a different location in the directory tree.
- Restore a directory to a different server and file system.
- Restore an empty directory.
- ▲ Restore a file to a new location in the directory tree.

Tapes Have No Data on Them or Cannot be Read

Data can't be restored if it doesn't exist on tape. This happens when:

- ▼ Backups fail or are stopped prematurely
- Tapes are used out of order and had their data overwritten

- Tapes wear out
- ▲ Tapes are destroyed by heat, humidity, dust, and other threats

If you experience a tape failure during a restore, stop the operation and clean the heads on the tape drive. It is possible that dirt deposits on the tape head have temporarily reduced the ability of the tape drive to read the data. Sometimes it is necessary to clean the heads twice.

Tapes with extremely important data on them that need to be read but cannot, due to the effects of contaminants and humidity, can be taken to facilities that have special techniques for baking tapes and making them readable again. The services are costly without any guarantees that data can be recovered.

NOTE: Re-tensioning tapes (unwinding and rewinding them again) periodically extends their shelf life. If possible, try to do this every six months, but at least once a year.

Data on Tape Is Corrupt or Incomplete

Another related problem occurs when the data on tape can be read just fine, but the data itself is corrupt and worthless. This can happen because of some flaw in the backup system, but it is more likely to be caused by data being updated as the backup is running. This is particularly problematic for database systems, due to the amount of data they have and the requirements for high availability. An adequate live or hot backup product is needed to ensure data integrity.

File System and System Configuration Changes

Most network backup systems default to restoring data back to its original location in the network, server, and file system. If the directory structure in a server has been reorganized, the name of a server changes, or other network-location information changes on a server, it is possible that data will be much more difficult to restore.

Sometimes people will use a system crash as an opportunity to upgrade their server—especially if they have been limping along with an underpowered server for some time. So, they build a whole new server with new disks, a new operating system, new disk volumes, and new users and passwords. New operating systems often have new backup system requirements and may not work with an older backup software package. It is much easier to rebuild the crashed server with its old OS and logical disk volumes, then restore its data before moving ahead with a system upgrade.

One additional note regarding configuration information: backup systems can provide a password for backed up data that requires the correct password to be entered before the data can be restored. This can be a very bad idea for obvious reasons.

Software and Equipment Is Unavailable or not Working

Physical and logical tape formats could be incompatible with the equipment available for restoring data. It is important to plan to have compatible systems, host I/O controllers, cables, terminating resistors, drives, and software to be able to read off-site tapes.

Capacity Overflow

In the earlier section on tape rotation, the problem of restoring more data than expected was discussed. More is definitely not better when it comes to restore operations. If too much data is restored, disk volumes will fill and the restore will halt or even crash the server.

Challenges of Managing Backup

The assertion was made previously that both backups and restores fail because planning and supervision were inadequate. This section attempts to explain why this might be so.

Harsh Realities of Backup Administration

In the context of the IT organization and the career aspirations of IT workers, there are some aspects of backup operations that need to be honestly assessed.

- ▼ Backup is a cost that nobody *really* wants to pay for.
- Backup is boring to manage, which results in mistakes.
- Backup problems are complex and difficult to troubleshoot.
- Managing backup is usually not a desired assignment. It's rarely noticed, except when things go wrong.
- Backup completeness sometimes is not verified, or it is inadequately verified.
- Backup administrators may be slow to ask for help while they try to figure out problems on their own.
- ▲ Errors are sometimes treated as "one-off" events and are not investigated adequately.

These items set the stage for an "accident waiting to happen." One warning sign is the belief that there aren't ever any problems with backup. The danger is that problems with backup can and will go undetected because they do not impact any other process in the facility. Where backup management is concerned, the easiest path to take is to believe some anomaly occurred that hopefully won't repeat itself. It's ironic that a backup system that is not working creates fewer problems than one that is—and is getting in the way of other applications.

There certainly are many organizations that place a high priority on backup. However, it can be difficult to maintain watchfulness over it due to the amount of resources it takes. The fact is, an administrator spending time on backup is not helping with the backlog of other tasks. Also, backup requirements are not static and change as the network changes. It is not a case where the backup problem can be solved once and for all—it often returns with completely different symptoms.

Measurable Problems in Backup

Some challenges for network backup are measurable. Here are several key items that must be considered when putting a backup plan in place:

- ▼ Backup window
- Largest backup source
- Amount of total data
- Amount of updated data
- Network transfer rate
- Speed of backup devices
- ▲ Backup engine performance

We'll look at each of these briefly in the following sections.

Backup Window

The backup window is the amount of time there is in a day or evening to get backups done. The backup window is often thought to be useful for cold backups, but it also applies to hot backups. Just because a server can be backed up without downing the server does not mean that it ought to be. The performance impact of backup can still limit the hours that backups can run.

There are basically two backup windows to consider. One is the nightly backup, which is usually performed overnight between normal business days. This is typically a backup window of three to eight hours and is used primarily for incremental or differential backup operations. The other backup window is the window to complete weekly or periodic full backups. It's not unusual for this backup window to extend over 24 hours, sometime, lasting from Friday evening until late in the day on Sunday. The backup system needs to have the capability to work within the constraints of both backup windows.

The technologies for hot backup discussed previously in this chapter—copy-on-write and snapshots—are designed to alleviate the problems of 24×7 computing and make it possible to accommodate the "impossible shrinking backup window."

Largest Backup Source

Network backup usually covers the majority of backup requirements in a network. Unfortunately, it tends to be deficient when backing up the largest backup source in the network. In this case, *backup source* refers to one of the storage components on a source system that can be uniquely managed by the backup system. For example, a logical disk volume or database system are backup objects. It is not unusual to find one particular backup object that is significantly larger than the rest, containing important data that must be protected by backup. Unfortunately, network backup systems often are not fast enough to get the job done. In such cases, sometimes the only solution is to use a dedicated backup system installed directly on the system containing the backup object with high-speed tape drives. In some cases, virtual tape subsystems where disk drives masquerade as tape equipment may be needed. Virtual tape is described in Chapter 10.

Amount of Total Data

The amount of data is not a difficult concept by itself. The total amount of data is significant for determining the dynamics of full backups. It is also significant when trying to plan for recovery requirements. Notice that this is not the total capacity of storage, but the amount of data residing in storage.

Amount of Updated Data

It is much more difficult to determine the volume of data that is updated on a daily basis that needs to be backed up. Rough estimates can be made by means of file system scanning or searching operations to find all the files that have new date and time stamps from some arbitrary time during the day to some arbitrary stop time, say from 6:00 A.M. to 6:00 P.M. Current backup records probably give a much better idea of the actual day-to-day backup loads. The amount of updated data is used in the calculations above to determine how many parallel backup processes need to be established to finish incremental or differential backup operations on weekdays.

Network Transfer Rate

As mentioned previously, network backups can drive network utilization to saturation and cause severe performance degradation. There are many variables to consider when projecting the impact of backup on a network. However, with careful thought and planning it is possible to make some realistic estimates. For the most part, daily backups are the main problem, because weekdays have short backup windows that restrict how long backups can run. Weekend backups usually have much more time to finish and do not put the same amount of pressure on administrators.

One should anticipate running parallel backups over the network. Some quick estimates can determine how many parallel backups will be needed. To start with, 1.5MB/s will work as a reasonable estimate for a 100BaseT network segment. This number is realistic if one considers that a 100Mbit network is comparable to a 10MByte link. Therefore, a single 1.5MB backup session is using 15 percent of the total (not usable) network bandwidth.

Since we are considering daily backups, we start with the average amount of updated data. Then, if we estimate the network transfer rate of 1.5MB/s, we can calculate the number of sessions needed by dividing the amount of data by the product of the backup window and the transfer rate.

$$\text{Updated data} / \{(\text{Backup window}) \times 1.5\text{MB/s}\} = \text{parallel sessions needed}$$

When making these calculations, it's necessary to convert some commonly used units to others. For instance, transfer rates measured in seconds need to match a backup window that is normally measured in hours or minutes.

As an example, consider a network backup scenario where the amount of updated data on network servers is 12GB and the backup window is one hour. Using the above equation and normalizing units, the actual calculation would be as follows:

$$12,000\text{MB} / \{60 \text{ min} \times (90 \text{ min/MB})\} = 2.2 \text{ sessions}$$

From the network's perspective, there is no such thing as a partial session. So the number of concurrent transfers for this scenario is three. With transfer rates of 1.5MB/s, per session, three sessions would mean 4.5MB/s are travelling over the network. However, this is hardly realistic for 100Mb ethernet networks. While two network segments could theoretically share this load, it will probably be necessary to use three independent physical 100BaseT network ports in the backup engine to support this traffic load over three independent 100Mb links.

Speed of Tape Devices

Even though the network is the most likely bottleneck for backup transfers, it is not necessarily the only one. Most network backup transfers are much slower than an average tape drive's capabilities, but interleaving data, dedicated local backup, and remote device operations can generate data transfers to tape that exceed their performance capabilities.

Tape drives are capable of maintaining streaming speeds if data can be supplied fast enough. It is normal for tape device manufacturers to publish the sustained throughput specification of their tape drives assuming that a 2:1 data compression ratio can be achieved. For instance, 8mm helical scan drives that list their transfer rates as 6MB/s have a noncompressed transfer rate of 3MB/s. The transfer rate attained has more to do with how compressible the data is than any I/O technology.

Device parallelism in the software can be an important contributor to the success of finishing backups on time.

Backup Engine Performance

The ability for the backup engine to make the transfer from network I/O to storage I/O is significant for network backup. This dictates a system with an efficient host I/O bus, such as the PCI bus, as well as using high-speed adapters. In order to conserve the number of slots in the backup engine, a multiple-port NIC should be considered.

As discussed previously, the database processing requirements for backup can be challenging. For that reason, the more processing power and system memory in the backup engine, the better.

SUMMARY

This chapter examined the technologies and processes of network backup. As one of the stalwart applications of systems management, backup is a part of most data processing installations. The chapter introduced the basic hardware, media, and software compo-

nents of backup before discussing the types of backup operations commonly used. The topic of backing up live data was discussed and the concept of copy-on-write, which will be discussed in more detail in Chapter 10, was introduced. Restore operations, media management, and tape rotation were also discussed, including the relationship between them. Finally, at the end of the chapter, the various trouble spots for network backup were discussed. As the amount of data increases, these problems become increasingly difficult to manage.

This part of the book was spent discussing traditional network backup management techniques and technologies as they are used today in TCP/IP network environments. The remainder of the book is devoted to the study of new storage networking technologies, such as SANs, fibre channel, and network attached storage. Chapter 10 is spent studying how these new technologies can be employed to solve some of the serious problems facing backup as these systems continue to expand.



CHAPTER 12



Plug and Play Storage with NAS

In Chapter 8, network attached storage, or NAS, is described as a type of turnkey product that provides network I/O services to client systems. This chapter looks in much more depth at what NAS products are about and how they work.

First, a short history is given of how NAS came to be, followed by an investigation of the technology and its applications. Systems management functions are also analyzed with particular notice paid to backup of data stored in a NAS server. Along the way, the differences between client systems and the network file protocols they use, NFS and CIFS, are examined and compared, especially in relation to the locking mechanisms they use.

The chapter closes with a look at two very interesting technologies: the Object Based Storage (OBS) project and the Network Attached Secure Devices (NASD) research currently under way at Carnegie Mellon University and several organizations. Both these technologies have the potential to radically change the way network attached storage is used, but both of them are also much more in the research stage than in the development stage and might not be commercially available for several years.

THE ROOTS OF NETWORK ATTACHED STORAGE

Not long after people began networking open systems computers they began looking for ways to provide access to files on other machines over the network. This idea gave birth to the notion of the file server—a special-purpose machine that other systems used to store and retrieve their files. File servers were typically designed to accommodate much larger storage capacities than desktop systems and were bolstered in other ways too, such as through the use of disk mirroring, dual power supplies, and multiple network interface cards. Over time, file servers have evolved into sophisticated, highly functional systems with integrated RAID storage subsystems and other availability features.

In 1985, Sun Microsystems introduced NFS, which was designed to provide conductivity for Unix desktop systems to Unix servers and to remotely access files on them. At the same time, there were several attempts to develop similar capabilities in the PC networking world. IBM, 3Com, Microsoft, and many others developed network operating system products.

Amidst all this competition from industry giants, the company that successfully propelled the concept of network serving in the PC arena was Novell, with its NetWare software products. For several years, Novell NetWare was the predominant network operating system, or NOS, and succeeded in popularizing the market for server systems in networks comprised mostly of PC systems.

In 1999, NOS technology has been studied in depth with a great deal written about it in many books. It is not the goal of this book to discuss NOS products but instead to look at NAS as an emerging derivative technology that has its roots in NOS technology and markets.

INTEGRATING STORAGE AS AN APPLIANCE

If NOS technology has already advanced so far, why is there a need for something else? The answer, in a nutshell, is that the level of integration needed to deploy NOS solutions requires a fair amount of administrative resources. The basic function desired in most cases is simple—a place to retrieve and put data files—but the integration and installation work involved can be inordinately large.

At some point, the question is asked, “Why should something that seems to be so simple require so much work?” Viewed from another perspective, the question is, “Why should it cost so much to hire a consultant to set up a system when all I’m really going to do with it is store files?” Another valid question is: “How can I maximize the system to utilize all its resources to its primary application—file sharing?” Or, finally, an associated thought is “Why should I use a general purpose system for file sharing with lots of capabilities that will never be used and may even negatively impact its file serving performance?”

In response to these questions, a small and growing industry has been forming around the notion of server-appliances that are specially designed to deliver storage functionality over a network. In a nutshell, NAS provides network-level plug and play storage capability. NAS products are typically designed to do one job very well at the expense of ever being used for other functions. The fact that there is a nascent market for storage appliances is a testimony to the success of the NOS market. NAS is a natural extension of the large NOS market discovering niche market technology to exploit.

Characteristics of NAS Products

There are three basic characteristics common to most NAS products. The first is a specialized operating system and/or application system that is dedicated to providing specific storage services. The second is an optimized hardware configuration that provides better cost or performance than standard off-the-shelf servers. The third characteristic of a NAS product is the preconfiguration of data network components that allow the NAS machine to be installed and administrated on an existing network with relative ease.

Specialized Storage System

One of the distinguishing features of most NAS products is the use of the specialized operating system and/or applications system to provide the storage function. Sometimes these specialized operating systems are called micro-kernel operating systems. A *micro-kernel* operating system typically has only the communications and storage services necessary to provide its special storage application. In other words, most of the capabilities of a desktop or server operating system are not supported. A NAS product does not necessarily have to be built on top of a micro-kernel operating system, however. There are several NAS products based on Linux, where the functions in the Linux OS were removed to provide a smaller, faster, specialized version for running the NAS function.