

# 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)-(v). Otherwise, it leaves the original page numbering.

DROPBOX, INC., ET AL.  
EXHIBIT 1013

(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        |
| <b>▼ 2 Piecing Together a Path for Storage I/O . . . . .</b>                                     | <b>35</b> |
| 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        |
| <b>▼ 3 Charting the I/O Path from Application to Storage in Client/Server Networks . . . . .</b> | <b>73</b> |
| 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<br>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        |
| <b>▼ 6 Boosting Availability and Performance with RAID and Disk Subsystems . . .</b> | <b>175</b> |
| 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<br>Parity Disk . . . . .           | 213        |
| RAID Level 4: Independent Access with a Dedicated<br>Parity Disk . . . . .           | 213        |
| RAID Level 5: Independent Access with<br>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<br>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 24x7 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 16



## Storing and Retrieving Data on the Internet

This final chapter looks at the dynamics of storage technologies and applications involving the Internet. Like most aspects of the Internet, there is no telling how or where technology changes will occur, but it is fairly certain they will. Some technologies, such as FTP (file transfer protocol), have been around for many years and may continue to survive through widespread use and continued innovation. Other technologies, such as WebDAV, are relatively new and have not been used much yet, but hold a great deal of potential for future developments.

The chapter starts with an analysis of storage networking technologies as an infrastructure component for the Internet. Then the chapter moves toward an exploration of the Internet as a storage facility. After a brief analysis of the dynamics driving storage applications on the Internet, you can find an analysis of some of the leading technologies and applications used to store and retrieve data over the Internet. The last section of this chapter highlights some of the more interesting technologies that are likely to be key components of future Internet storage integration work.

## INTERNET INFRASTRUCTURE STORAGE

While there are several interesting possibilities for storage applications on the Internet, the ability of storage and I/O technologies to keep pace with the requirements of rapidly growing web sites is an essential ingredient for many IT organizations. This section examines the deployment of storage and I/O technology for server-side and client-side environments.

### Scaling to Server-Side Requirements

Most web servers do not have enormous amounts of data stored on them. For instance, a web server with 50GB of online disk capacity is considered to be a fairly large web server in 1999. Of course, there are always exceptions; and the largest web servers have terabytes of data. There is also an important difference between a web server and a web site. It's fairly normal for a web site with huge amounts of data to be built from many smaller servers. Still, in late 1999, the overwhelming majority of web sites have less data on them than internal network servers.

What web sites may lack in data volumes they more than make up for in traffic. Busy web sites can have millions of file accesses, also called *server hits*, per day. High traffic sites can easily become I/O bound as the number of requests exceeds the ability of the I/O infrastructure to supply them.

### I/O Loading in Web Servers

Internet companies with high-traffic sites are not eager to discuss performance problems in public, but I/O operations are recognized as system bottlenecks by several leading Internet companies. I/O performance is particularly difficult as sites attempt to provide

customized information to thousands of users simultaneously. For instance, the news items a user is presented might be cached in memory, but the profile that determines the presentation of news has to be retrieved from disk storage. More difficult is the task of providing online personal information that a user will want to access on demand from any location, such as appointment information, phone numbers, and web-based e-mail. These information items have to be stored and accessed for each user of the system. As the number of users can be in the millions for such free services, the potential for I/O congestion is very real.

## Search Engines and Storage

Internet search engines work by reading information in from Internet sites and building a large text-indexed database that represents the content. The many different types of search technologies are beyond the scope of this book, but the point is that an enormous amount of information is collected and made available. After the database index has been generated, it is typically copied into a disk cache to provide the fastest performance. Heavily used search engines such as Yahoo!, Excite, Alta Vista, Lycos, and others would be unusable if the performance of their engines was gated by disk I/Os of the search engine index. The cache used for storing the index is normally host-resident cache, but it could also be subsystem-resident cache.

Unfortunately, the size of these index files can exceed the maximum installable memory of web servers by many gigabytes. For instance, a server with a 12GB memory limitation cannot store a 120GB search index. In these cases, the index is spread over many systems, and inquiries are distributed to the appropriate system. A complete copy of the database file and table structure may be stored on devices connected to each index server (see Figure 16-1).

This kind of indexing implementation works, but it has some significant cost drawbacks. First, the amount of money spent on multiple index server systems is high, especially if one considers the need to fill them with the maximum amount of memory possible. In addition, the database file that the index references has to be copied onto multiple storage subsystems. As this number of subsystems increases, the amount of time to do the copying gets bigger and bigger, the likelihood that errors will occur also grows, and the difficulty synchronizing index updates across multiple servers becomes a real headache. Finally, the requirement to duplicate all the equipment for fault tolerance doubles the cost of this implementation.

**IMPLEMENTING VOLUME MANAGEMENT AND SSDS FOR SEARCH INDEX I/O** A storage-centric solution, as opposed to a server-centric solution, could be used to reduce the cost and make the system more manageable. For starters, the system cache memory in each index server that is used to store the index segment could be replaced by a larger block of memory in a disk subsystem or a solid state disk device (SSD). This assumes that there are subsystem caches and SSDs large enough to hold the index file, which is not necessarily a valid assumption. Therefore, it may be necessary to use volume management software to

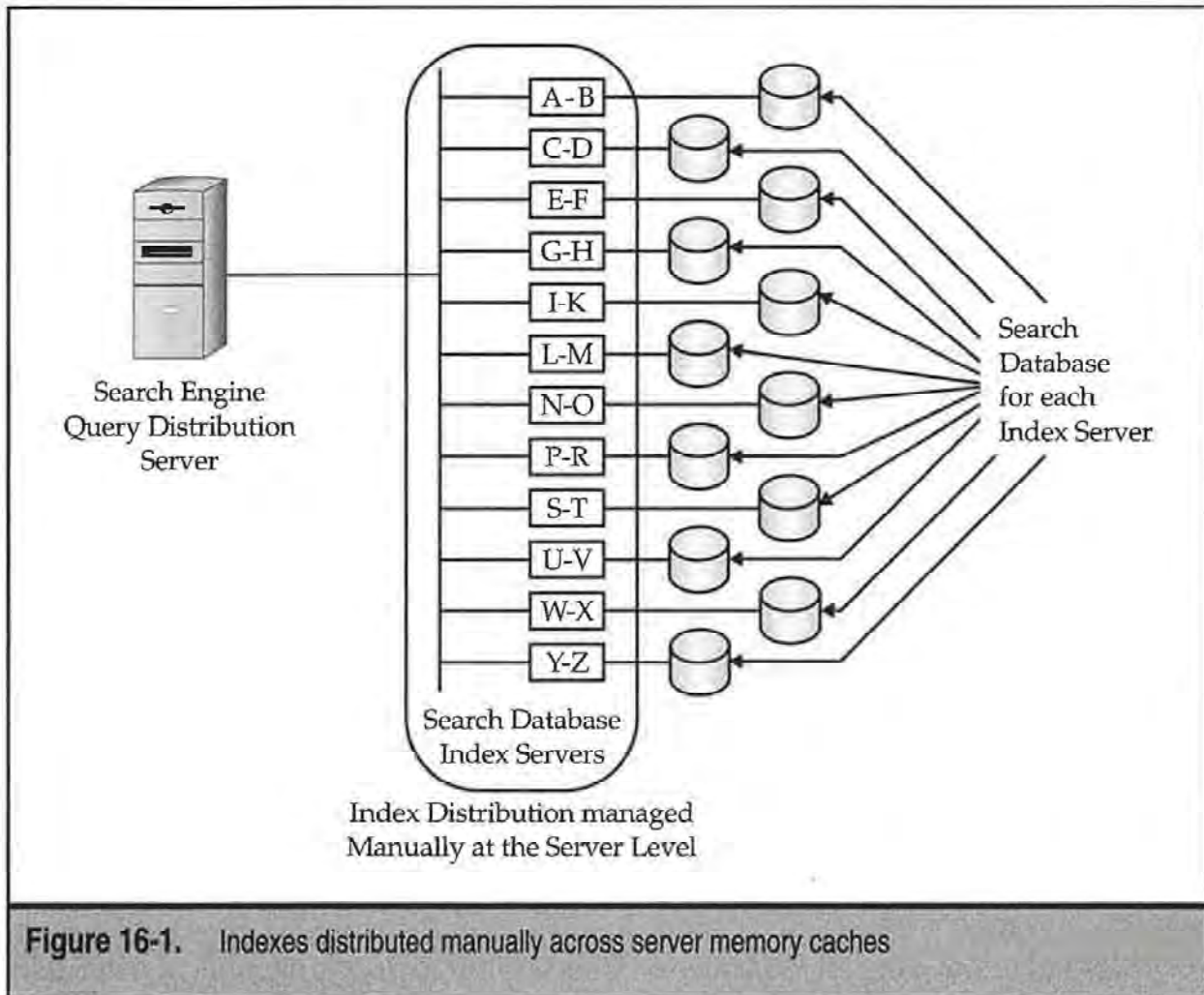


**Figure 16-1.** Indexes distributed manually across server memory caches

concatenate multiple SSDs and create a single large virtual SSD. This virtual SSD could hold the entire search index file and provide excellent performance. Figure 16-2 illustrates this volume manager/SSD approach and shows the I/O path involved when connecting five such SCSI SSDs of 24GB each to create a single virtual 120GB index volume.

This type of storage-centric approach eliminates the need for multiple index servers. As a by-product, it also eliminates the need to have a copy of the database stored on separate disks or subsystems for each index server. Not only that, but as the size requirements increase, additional SSDs can be incorporated by the volume manager software.

#### **USING AN ARRAY-STRUCTURED NETWORK FILE SYSTEM FOR SCALING SEARCH ENGINE I/O**

This still leaves a situation in which a single index server is both a potential bottleneck and a single source of failure. An alternative approach, and one that is hypothetical in 1999, is to use an array-structured network file system, as discussed in Chapter 13, to store

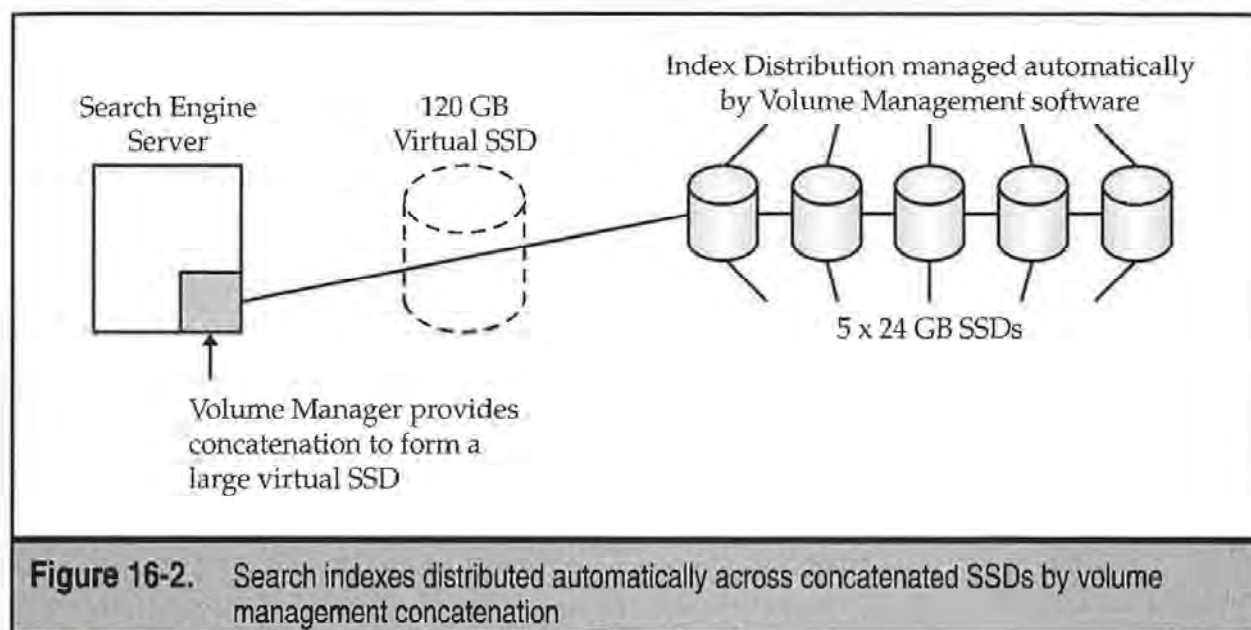


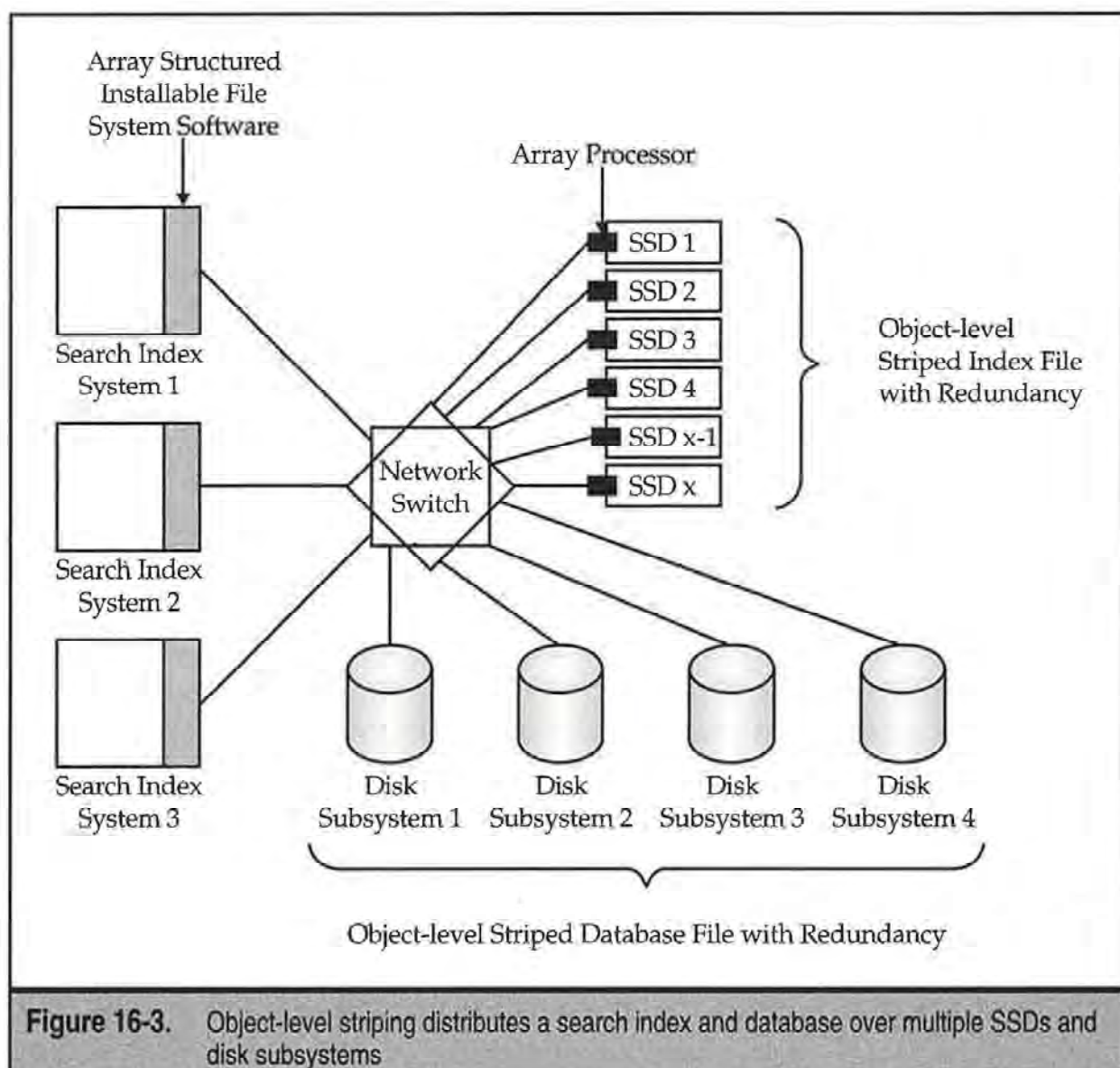
**Figure 16-2.** Search indexes distributed automatically across concatenated SSDs by volume management concatenation

the index file across multiple SSDs. This way, multiple index servers could access the same index file simultaneously over the storage network in response to user search requests. The subsystem containing the SSDs would need to incorporate I/O array processors to interact with the installable file system component in multiple index servers. The non-index database files could be stored in a similar fashion on distributed disk subsystems, also with I/O array processors. The end result would be a large, scalable network I/O system that easily adapts to increased user loads and ever-increasing search databases and indexes. Figure 16-3 shows a web site storage network in which an installable file system resides in search engines that have access to an object-striped file in an array of SSDs. The database file that the index file relates to is also striped in this example.

## General Requirements for Web Server I/O

Search engines are an interesting, but special case of web servers. There are many more transaction and HTTP servers in the Internet than there are search engine servers. This section discusses a variety of network storage technologies and approaches that could be employed to provide higher-performance, scalable, and more reliable Internet storage.

Performance requirements vary a great deal, depending on the type of traffic and the type of data. Most web servers do not have the I/O performance requirements that search engines do. For example, streaming audio servers do not need to provide responses immediately, but once they start streaming data, they need to be able to maintain a sustained rate.



The primary requirements for web site storage are to provide the following:

- ▼ Flexibility
- Scalability
- ▲ Availability

### Stable Storage in the Midst of Rapid Server Changes

Web sites can grow and change quickly. While web site architectures do not change daily, they can change a couple of times a year. Managing a web site can be an intense and wild ride for a successful e-commerce company. The requirement to add or upgrade servers

and reorganize applications across those servers is fundamental to managing large web sites. The trick is to make changes without any resulting downtime.

The storage infrastructure for these sites should support server changes while minimizing configuration changes to storage. In other words, the storage network should isolate the configuration of storage from the configuration of servers.

## Growing with Application Needs

As users and business data grow on a web site, the storage infrastructure to support it needs to scale along with it. For most web sites, this is not as big a problem as it might appear, because web servers often do not store as much data as other business servers and systems. For example, 100GBs of storage is considered a fairly large web system in terms of storage capacity. By comparison, a 100GB LAN-resident file or database server is not considered anything to take note of.

## Mirroring and RAID for Data Availability in Web Sites

Web sites are typically 24/7 operations. There aren't any good times to have a web site respond with *404 file not found* errors due to storage failures. The storage infrastructure for a web site should be able to provide high data availability. Disk mirroring or parity RAID should be used to provide availability of data in case of disk failures. Some points to keep in mind include the following:

- ▼ RAID 3 provides almost no benefit, as the Internet connection between the web site and the user cannot keep pace with RAID 3's streaming data delivery.
- RAID subsystems that provide independent access to member disks, such as RAID 5, provide the best sustained transfer rates for high volume web servers.
- RAID 5 write penalties are typically minimal due to the very high read/write ratios in typical web sites.
- Transaction processing servers with higher write percentages can benefit from RAID 1 or RAID 0+1 to eliminate the RAID 5 write penalty.
- ▲ Larger numbers of small disks in an array increases the number of disk arms and can increase system throughput in high-volume web servers.

**NAS FOR WEB SITES** Network attached storage provides a simple and flexible mechanism to access storage resources. The general idea is to implement a site file server, or servers, with a NAS product that provides file I/O for all other systems in the site. As production web servers are changed, the applications running in them can use a consistent and common file access mechanism. Managing systems in the site is made easier by not having to worry about the storage resources for each individual server. Instead, site servers can be configured for minimal storage resources to support system booting and operating system functions, while the storage resources for the site can be managed through a single NAS system.

Figure 16-4 shows a before/after picture of a web site. The before picture shows a 3-system web site, all accessing their files from a NAS system. The after picture shows the same site with 12 servers, each of them accessing the same NAS server. The NAS server has grown in both storage capacity and in network capacity through the addition of two additional network connections.

**SANS FOR WEB SITES** In the discussions that follow, the term SAN refers to block-protocol I/O technology, such as fibre channel, to differentiate it from file I/O technology as implemented in NAS systems.

SANs provide separation of storage resources from web site systems. Site systems can be easily replaced and upgraded without having to unplug or move storage resources. Notice, however, that web applications cannot be re-deployed across web systems without copying data from one server disk subsystem to another. If an application is moved from one server to another, it is likely that zone assignments, LUN masks, or port configurations will need to be changed to allow the application to continue to access its data. In any event, the changes required can be done logically, and do not involve moving physical connections or equipment.

SAN storage pools have the potential to save web site managers a great deal of worry and time. Storage pools that are established on application boundaries provide a centralized approach to managing the storage resources for a web site. As new needs and applications are discovered and implemented, storage capacity can be "carved out" of installed, but unallocated, storage. Figures 16-5 and 16-6 show the before and after views of a pool of vir-

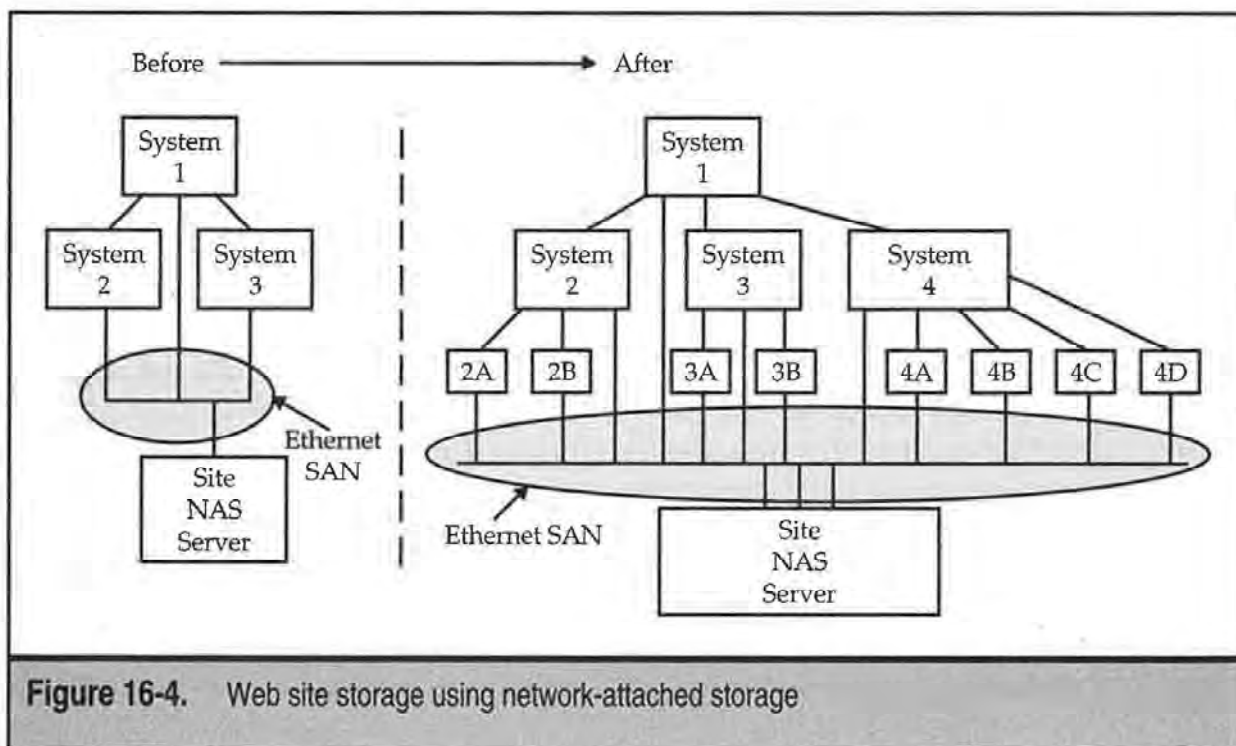


**Figure 16-4.** Web site storage using network-attached storage

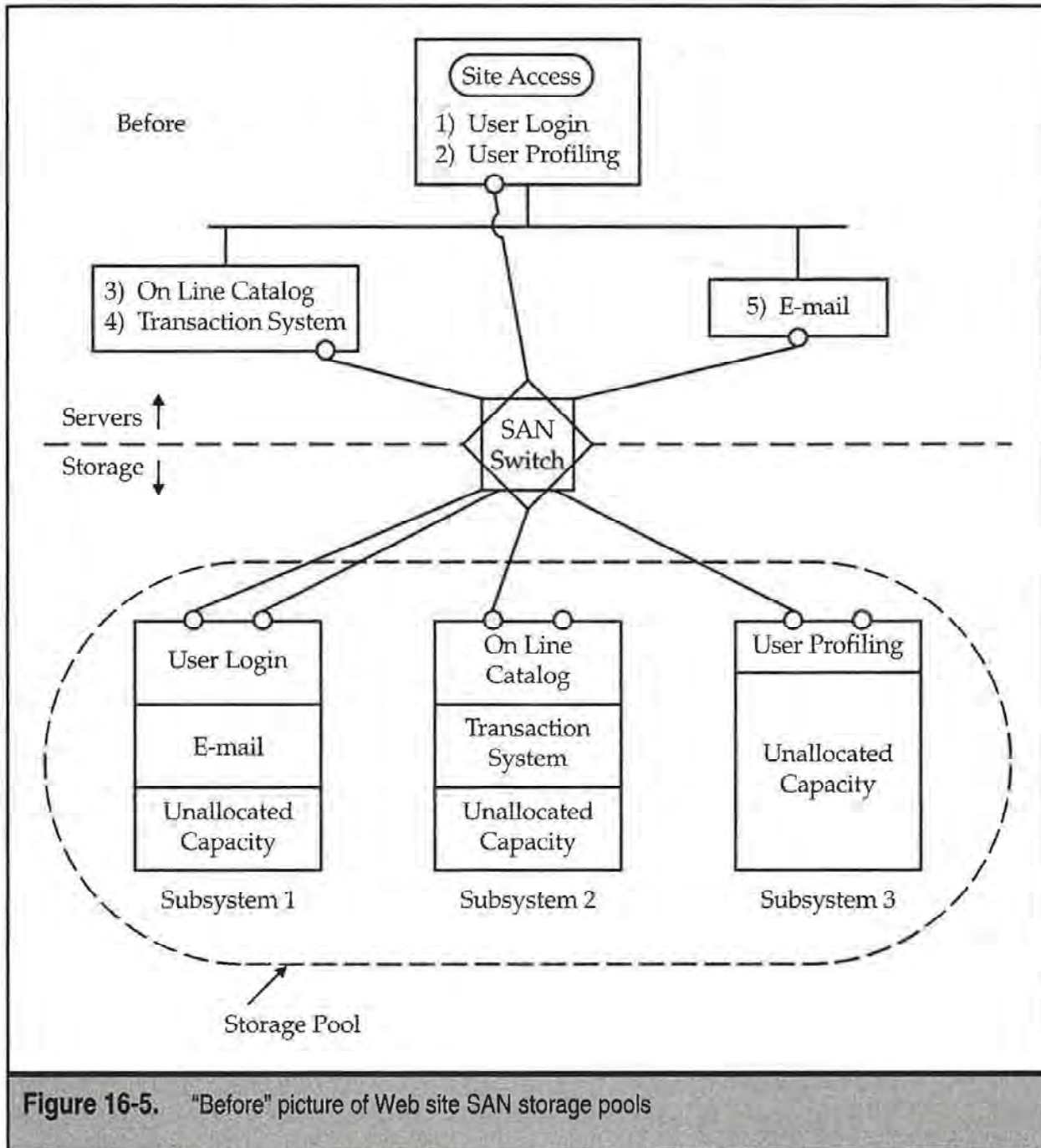


**Figure 16-5.** "Before" picture of Web site SAN storage pools

tual drives assigned by application, as opposed to server boundaries. The storage pool in this case is implemented across three similar dual-port/dual-controller storage subsystems.

Looking at Figures 16-5 and 16-6, the *before* picture shows three servers with five applications spread across three pooled storage subsystems. Each subsystem has a significant amount of unallocated storage capacity representing either empty disk drives or empty drive bays. While storage capacity was allocated by application, the port assignments on subsystem ports are assigned by server.

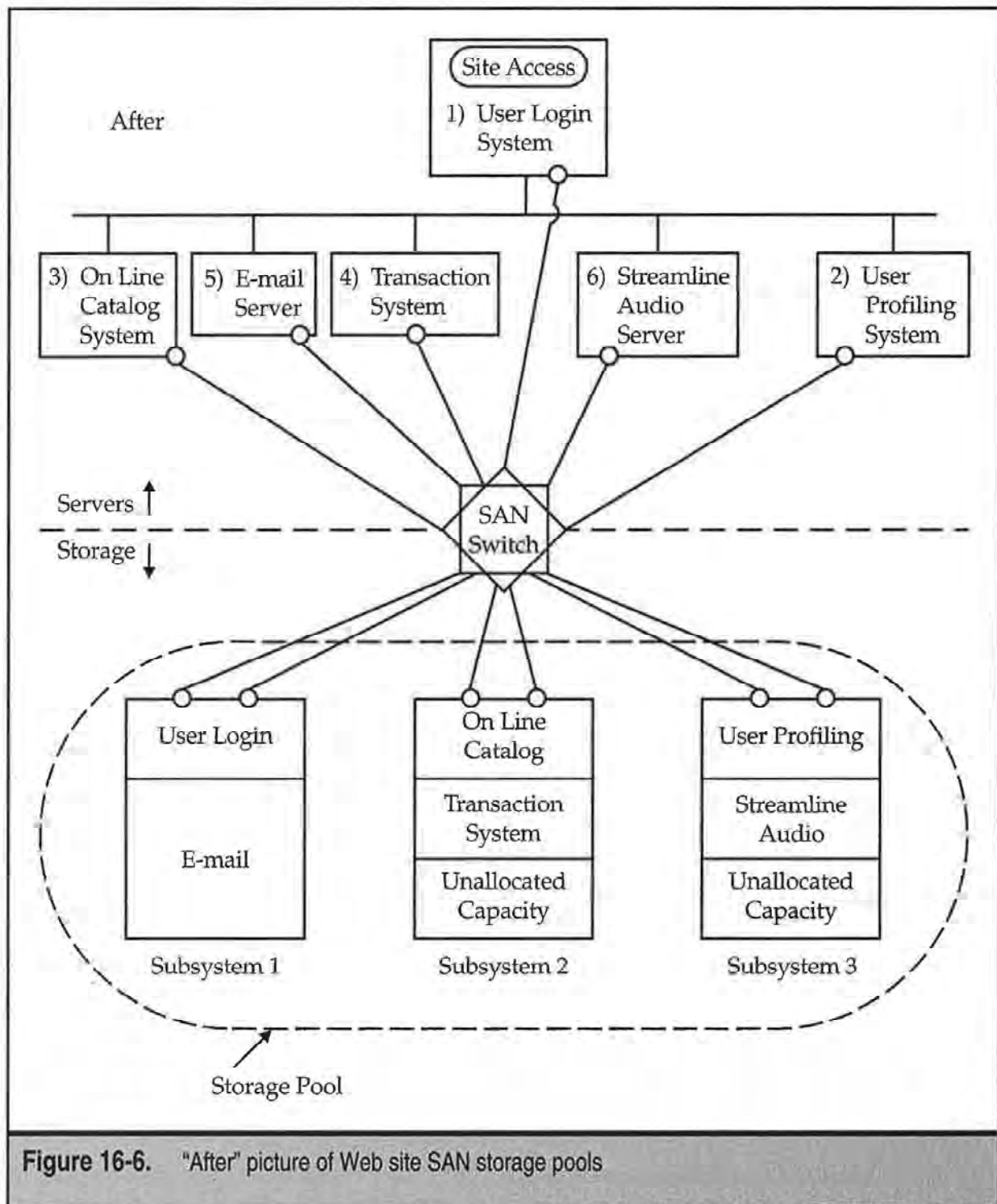


**Figure 16-6.** "After" picture of Web site SAN storage pools

The *after* picture shows each application being moved to its own dedicated server, and a new application, streaming audio, implemented on a sixth server. The e-mail storage has been expanded to use the remaining unallocated space on subsystem 1, subsystem 2 did not change, and subsystem 3 added streaming audio data and still has some remaining unallocated capacity. Each subsystem supports a connection to two site servers.

**DIFFERENCES BETWEEN SAN AND NAS WEB SITE IMPLEMENTATIONS** While both SAN and NAS technology can be used to provide storage services for web sites, consider some significant differences:

*NAS does not support all applications* Database applications sometimes require block I/O and do not work well with file I/O provided by a NAS server. In this case, either bus-attached SCSI or SAN technology is needed. SANs support all applications.

*SANs require storage administration for each server* Each server in a SAN requires storage to be configured and installed. In addition, each server requires some amount of storage management, such as backup. By contrast, a NAS solution centralizes all storage management on a single file server.

*NAS and SAN solutions can be single points of failure* A single NAS server that fails or is powered off for maintenance or upgrades can shut down an entire web site. The same thing could happen if a single multiport disk subsystem that provides all storage services for the web site fails.

*Capacity maximums are lower in NAS products* All products have limited capacity. In the case of a NAS product, the physical configuration limitations determine the capacity of a single integrated file system. Typically, two NAS products cannot be combined as a single named or addressable entity under a single file system. SAN disk subsystems also have capacity limitations, but they can be combined by storage management applications to create much larger virtual devices managed by a single file system.

*SAN storage is not inherently highly available* NAS products, as comprehensive systems, can implement fault-tolerant software, hardware, and I/O components in a single fault-tolerant solution. HA storage solutions for SANs can be achieved by integrating multiple components. This is changing as storage vendors offer HA packages for their products, but in general, it is not the case in late 1999.

*SANs can accommodate specialized technologies more easily* SANs can utilize virtually any kind of device. NAS products, on the other hand, typically qualify each device used as part of their solutions. This effectively restricts certain devices from being used with NAS implementations.

*Read ahead caching in NAS products is more effective* Because NAS products have their own integrated file system, read ahead caching is 100 percent accurate. By contrast, block I/O caching in SANs should only be expected to be 40 percent accurate. For longer data files such as graphics and streaming audio/video, this can be a big advantage for NAS storage. However, the maximum amount of cache available in SAN storage subsystems can be larger than the maximum amount of cache in NAS products (in late 1999). This is an implementation issue that depends on the cache memory capacity provided by manufacturers.

## Client-Side Web Caching

Web caching provides faster access to Web content by storing data close to users who access it. The idea of web caching is similar to disk caching as discussed in Chapter 5. Data is transferred across the Internet and stored on the disk storage of server systems, caching devices, or other specialized products. Users requesting frequently accessed Internet data can have their requests satisfied from the cache instead of a remote web server in another part of the world. Figure 16-7 shows an example of a web cache that provides fast

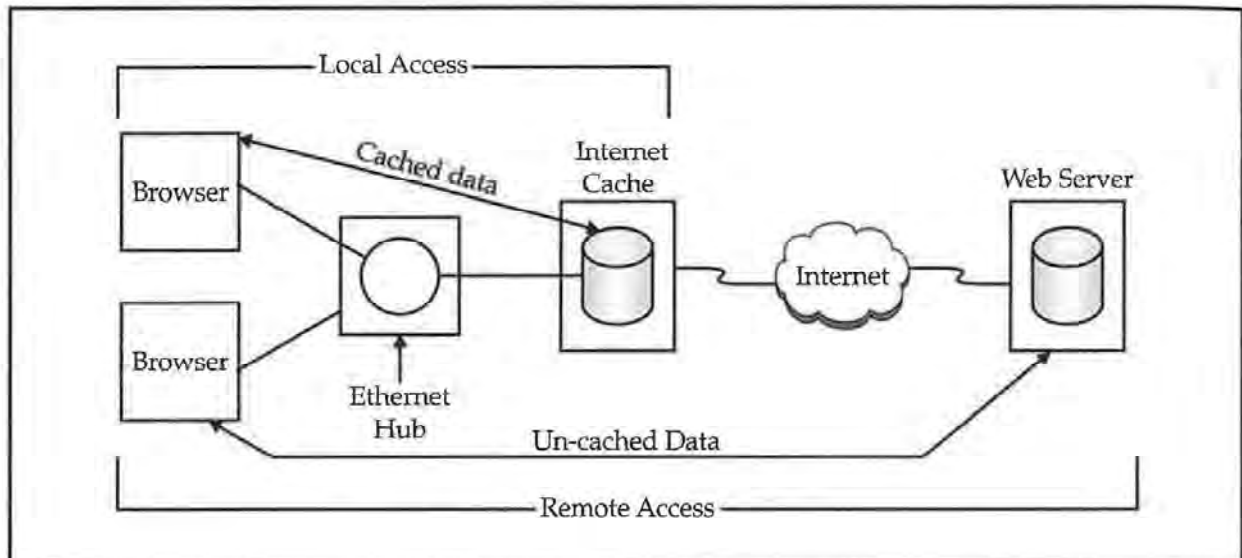


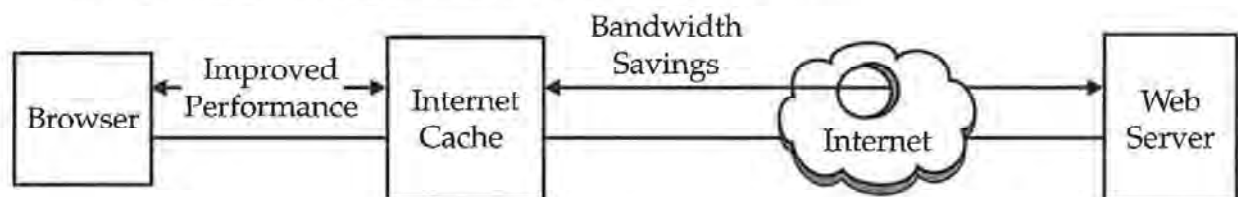
**Figure 16-7.** Internet caching reduces distance and time to data

access to data from a distant server, and contrasts it with data access over the Internet to a web server.

One way to think about Internet caching is the exchange of storage capacity for Internet bandwidth. Requests serviced out of cache have two benefits:

- ▼ Response time is better
- ▲ Less Internet bandwidth is used, which saves money and resources

These concepts are depicted in the following diagram.



While Internet bandwidth appears to be continually improving, this doesn't mean that there are not bandwidth problems in the Internet. Also, while local bandwidth may be improved by technologies such as DSL, it is possible that such services will have the effect of significantly boosting local traffic and causing congestion in the Internet's backbone. Web caching is intended to alleviate the bottlenecks inherent in the Internet's backbone structure.

## Internet Structure and Network Access Points

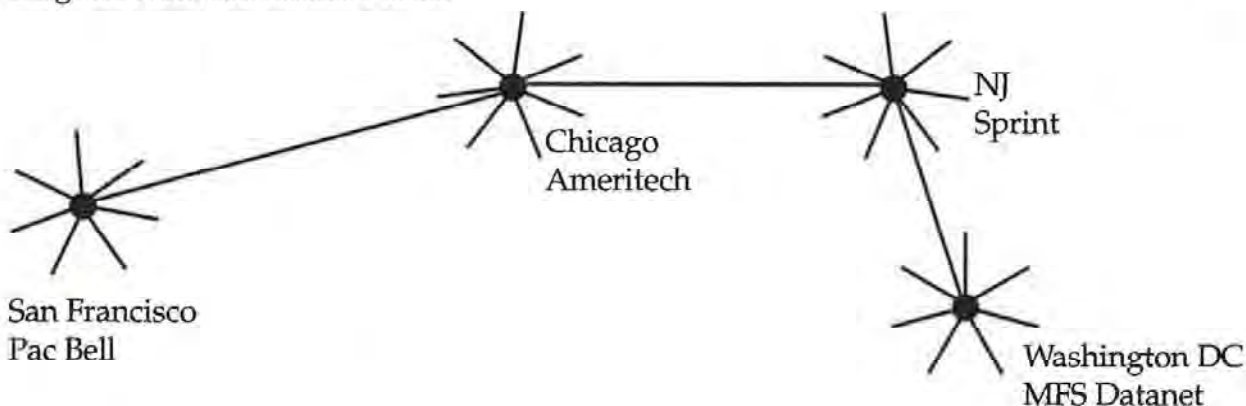
Understanding the web's structure begins with understanding how traffic is routed between the various networks that are operated by major ISPs. Users who use a local ISP in

Miami can have interactive sessions with a business server in Seattle. This seems like magic to most of us, but it results from an evolving structure that provides a mechanism for routing and exchanging data across different networks. Obviously, these network operators have to cooperate in order for Internet subscribers to access everything on the World Wide Web.

The NSF (National Science Foundation) established four NAPs (Network Access Points) for exchanging data among different network vendors. These NAPs are operated by four different telecommunications companies:

- ▼ Sprint NAP—Pennsauken, NJ
- PacBell NAP—San Francisco, CA
- Ameritech Advanced Data Services (AADS) NAP—Chicago, IL
- ▲ MFS Datanet (MAE-East) NAP—Washington, D.C.

Original Network Access Points



A high percentage of Internet traffic flows through these points on the way to their destination, establishing built-in bottlenecks in the Internet backbone. Alternative backbone networks are being built to address the issue, but they may not be able to keep up with the demands for bandwidth from a large and growing user community. As long as bandwidth is an issue in the backbone, it is likely that caches will be used to improve performance and lower bandwidth costs.

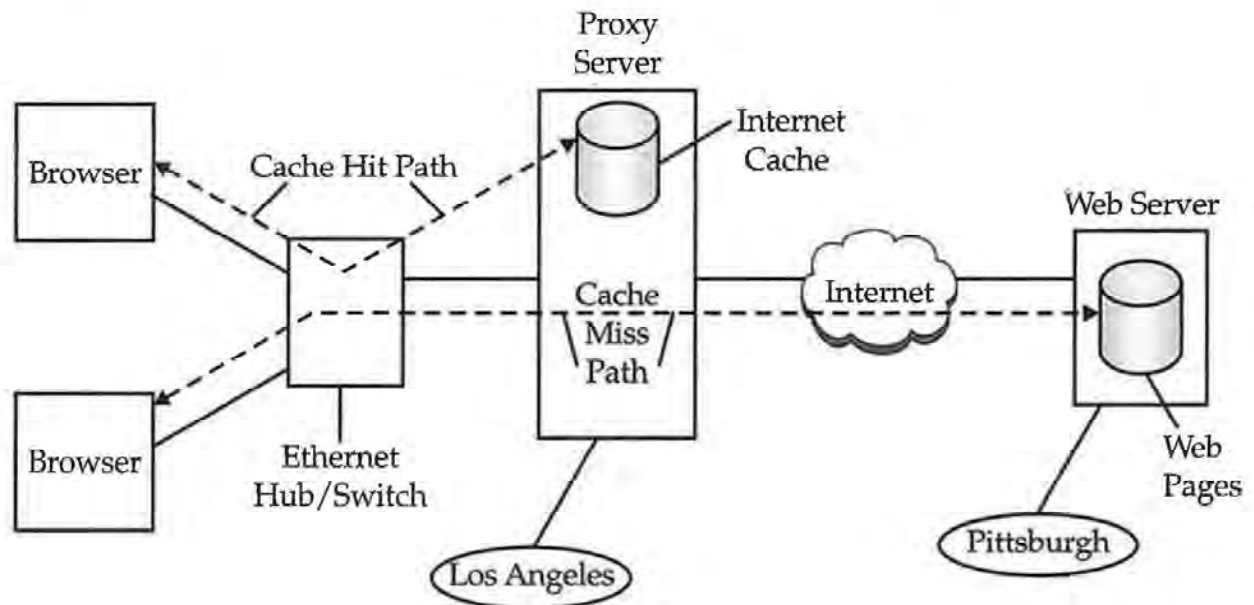
## Variations in Internet Caching

Internet caching is implemented in two basic types of products:

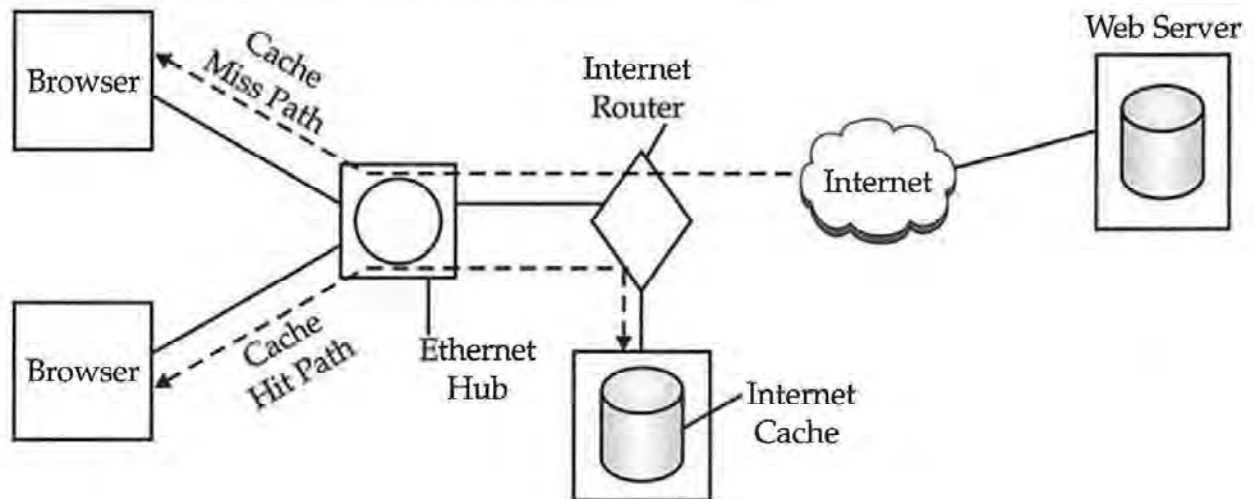
- ▼ Cache application software running in proxy servers
- ▲ Network caching products

*Proxy server caches* are software packages implemented in web servers, called *proxy servers*. Client systems send Internet requests to proxy servers, which forward them to web servers on behalf of local clients. A proxy web server is an obvious good place to provide an Internet cache and can be done without spending a lot of money on specialized hardware or systems. The main weakness of the proxy server approach is the complete

dependence of clients on the proxy server for Internet access. The proxy server is directly in the path between clients and the Internet. A failure or performance problem with the proxy server affects all access to the Internet. Also, configuration and troubleshooting of client/server communications can add administrative overhead. The function of an Internet proxy server is graphically depicted as follows:



Network caching products have no direct connection to clients, but instead work by "sniffing" Internet traffic and determining what data to cache. These types of products can be implemented as either complete caching systems or as software that runs on a specified platform. And a product can be implemented in the path between clients and web servers or it can be implemented as a look-aside cache that has requests routed to it from an Internet router that is aware of the cache's presence. The look-aside cache technique can be graphically shown as follows.



## Storage Subsystems for Internet Caching

The design of the I/O subsystem in an Internet cache can have a significant impact on its performance. Following are a few items to consider:

- ▼ Expandability and maximum capacity
- Block size on disk
- ▲ Availability and serviceability

For the most part, the performance of the I/O subsystem for an Internet cache is not likely to be a big problem. Most of today's I/O controllers are capable of providing performance far in excess of 100BaseT ethernet networks. Even a cache with multiple 100BaseT connections is not likely to have a problem with I/O performance. It isn't necessary to spend too much time thinking about such performance issues as whether or not the RAID 5 write penalty will be a problem or how much memory cache is needed.

It is more important to relate the capacity of the cache with the bandwidth efficiencies that can be gained, ensuring that enough capacity is available to have an impact. One way to do this is to allocate a certain amount of Internet cache storage for each user. The amount to allocate can depend on the type of data requested. For instance, multimedia files can benefit the most from being cached locally, but they require larger amounts of storage capacity. Considering the relative cost of disk drives, there is not much reason to pinch pennies on the disk storage used in an Internet cache.

For example, if each user is allocated 100MB of cache storage, it only takes 10GB of disk storage to support 100 users. That also means 500 users could be supported by 50-GB of online available capacity.

The ability to configure the block size of the cache storage is an important consideration. Large block sizes are typically used to support large, sequential access, which typifies the type of I/O operations performed over the web. The type of disk fragmentation that is typically viewed as a deterrent to using large block disk formats is not really a big problem for an Internet cache machine.

The availability of the Internet cache's storage takes on different levels of importance depending on the nature of the organization's business. For instance, an ISP providing web caching will likely want a disk solution that can withstand individual device failures and still continue to operate. A business that uses web caching to provide improved response time for its workers may not care as long as it doesn't prevent workers from accessing the web.

## STORING DATA ON THE INTERNET

The influence of the Internet is pervasive through the entire computer industry, including the storage and I/O industries. There is little doubt that the rush to publish and provide information over the Internet is contributing to an enormous demand for storage

capacity. But the relationship between the Internet and I/O is much more than the fact that web servers need a place to store their data. People have been storing and sharing data on the Internet for decades already and will continue to do so at even greater levels in the future.

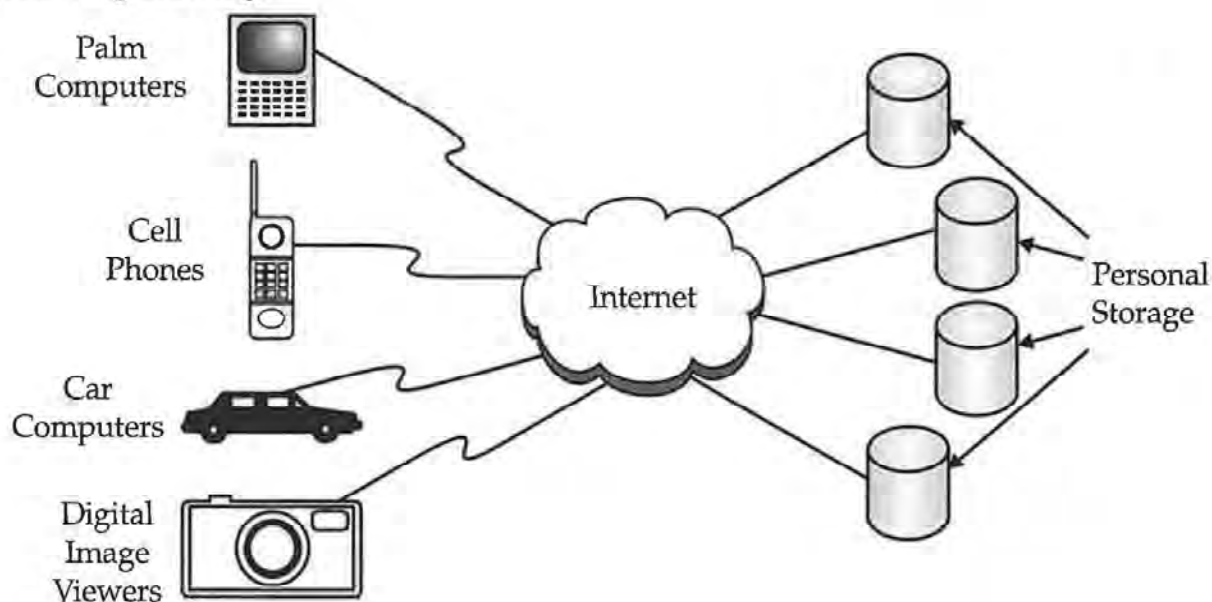
## Five Reasons for Internet Storage Growth

Many things contribute to the need to store and share data on the Internet. Five of the most important ones are as follows:

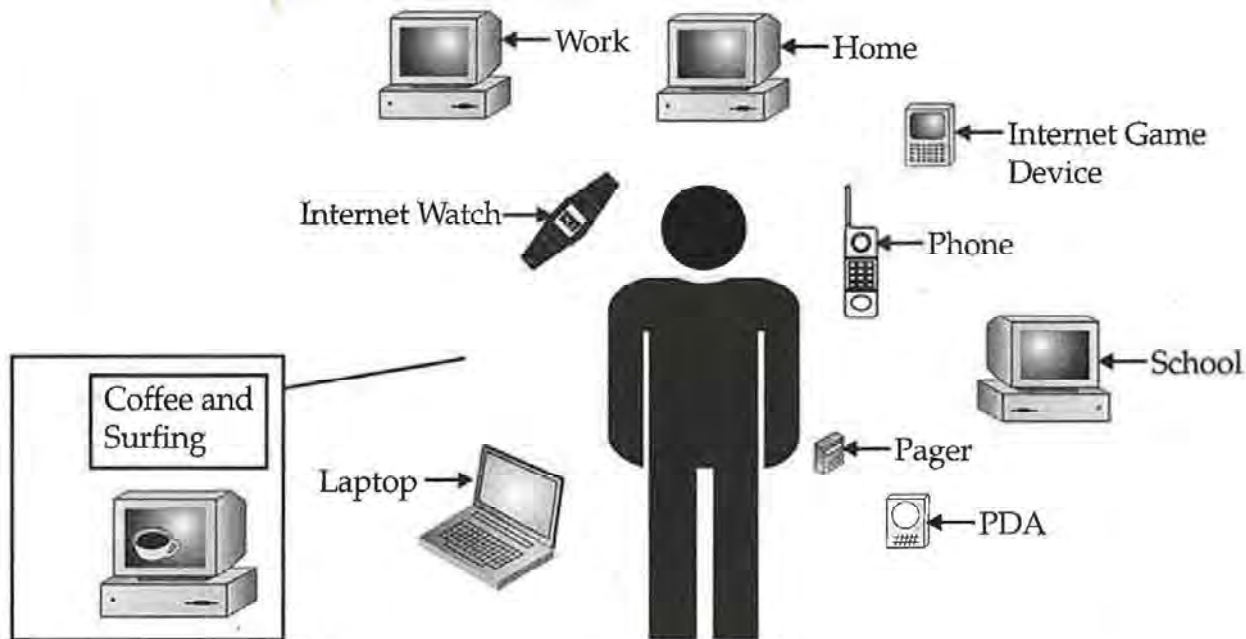
- ▼ Increasing number of users
- Increasing numbers of access equipment and devices
- Increasing number of access points
- Increasing breadth of Internet applications
- ▲ Increasing bandwidth and link performance

The number of users on the Internet is growing at a phenomenal rate and will continue to increase as more and more technology is integrated with Internet technology. This is certainly not a surprise to readers of this book.

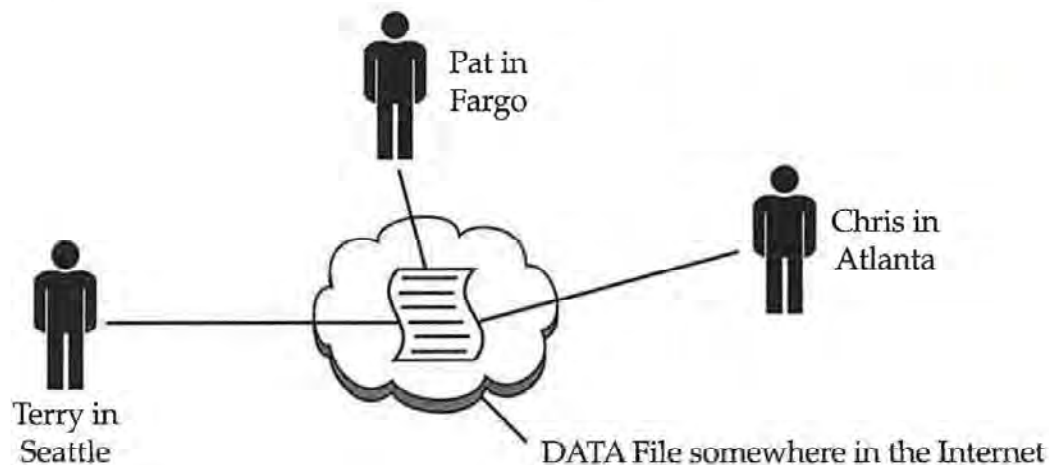
A great deal of technology being developed allows people to access the Internet from almost any location using small handheld appliances such as cellular phones and palm computers. While these devices offer ultimate portability, they do not have the ability to store large amounts of data or share that data with others. It is highly likely that users of these products will use the resources of the Internet to store and retrieve data. The partial universe of systems and devices needing Internet storage are shown in the following drawing.



People can access the Internet from computers at work, at home, in their cars, at friends and relative's houses, at airports, hotels, printing centers, and Internet cafes—almost any computer anywhere can have public access to the Internet. This means it is possible for an individual to work almost anywhere and store and retrieve his or her data. With so many places to access data, it follows that one of the best places to store data is a repository on the Internet. In fact, it seems almost obvious that Internet storage will be used as the central repository for mobile electronic workers, and as the optimal storage facility for synchronizing data. Data synchronization is certainly not a new application, as many end-users have been using it to manage different versions of files on laptops and desktop machines. With an Internet-wired world, the increasing number of machines/devices a person uses makes it even more important to synchronize and manage multiple generations of data files. Internet services that provide these functions already exist and will likely become more prevalent in the future. The following diagram shows some of the multiple access points individuals will use to access data on the Internet.



The ability of remote workers to cooperate on projects is one of the hallmarks of the information age we live in. Internet storage enables people to work together in virtual workgroups across dispersed organizations or locations. New collaborative applications designed to facilitate virtual workgroups will utilize centralized Internet data repositories. The following illustration shows a small team of workers in separate locations sharing data over the Internet.



High bandwidth carriers using large fiber-optic networks and satellite transmission are increasing available bandwidth. Other technologies such as cable modems and DSL (Digital Subscriber Loop) are providing higher link speeds for small business and home users. This not only makes it faster to download streaming multimedia data and large graphics files, it also helps provide transparent storage functions with acceptable performance. For example, DSL transmissions offer performance relatively equivalent to I/O operations on diskette drives.

## Storage, Storage Everywhere

The amount of available storage space on the Internet is staggering. Or perhaps it is more accurate to say that the amount of storage that has been *promised* to subscribers is staggering. The difference in these two statements is that most of the storage capacity that has been promised to Internet users goes unclaimed, and so it has not been implemented yet. This might seem strange, but it makes perfectly good economic sense in the "everything is free" Internet business model that prevails. Free storage doesn't actually cost an ISP (Internet Service Provider) anything if it doesn't get used.

Most ISPs provide subscribers with the option to have their own home page or web site on the Internet. While many people have created small web sites, the overwhelming majority of web users are perfectly comfortable accessing the Internet without creating their own web site. The additional effort of creating a web site is considered to be either not worth the effort or too challenging. As part of the free web site offer, most ISPs also offer subscribers 5, 10 or 25MB of data storage to hold the files for their personal web site. It is not generally well understood among the masses of Internet subscribers that all that capacity can be used to store any kind of file whatsoever, not just web HTML and related files. This translates into an astounding amount of storage capacity available to users, which could be used to store practically any kind of data they desire. This unused storage capacity includes businesses that have their own web sites.

To estimate the amount of available "virtual" storage that has been promised to Internet subscribers, multiply the number of Internet users by some reasonable amount of storage capacity in MB. For instance, 20 million user accounts can be multiplied by

5MB per user to estimate that 100,000GB of storage capacity have been allocated to individuals for storing files. Of course, if most people start using their free Internet storage capacity, the situation could change very quickly as ISPs scramble to provide the capacity that they have promised their end-users. Not only would it be difficult to provide all that capacity quickly, but the ability to access it in a reasonable amount of time would be a monstrous challenge and result in many unhappy customers.

## Internet Storage Without Browsing

For many, the World Wide Web is synonymous with the Internet. However, for the purpose of using the Internet for storage, it is a good idea to maintain a clear distinction between the Internet and the World Wide Web. The World Wide Web allows people to access all kinds of information from almost any place in the world; however, web users have not been using it much to access the information most important to them—their own.

Instead, people have been storing and retrieving data on the Internet using FTP (File Transfer Protocol), years before web browsers existed. FTP was developed as a command-line interface that can be run on systems—including PC DOS systems—lacking sophisticated graphics capabilities. It is an extremely flexible and mature utility that works very well for both storing and retrieving data.

Another Internet facility called *Gopher*, named after the mascot of the University of Minnesota—the “golden gopher”—was also used to store and retrieve data on the Internet prior to the advent of the World Wide Web. Gopher provided a graphical representation of directory structures on Internet servers and allowed users to navigate their directory trees and select files for downloading. Considering that these directory representations were actually directory files, one can see how Gopher anticipated the linking capability within documents that served as the impetus for HTTP. Today, Gopher functionality is integrated into web browsers, although its use is relatively rare.

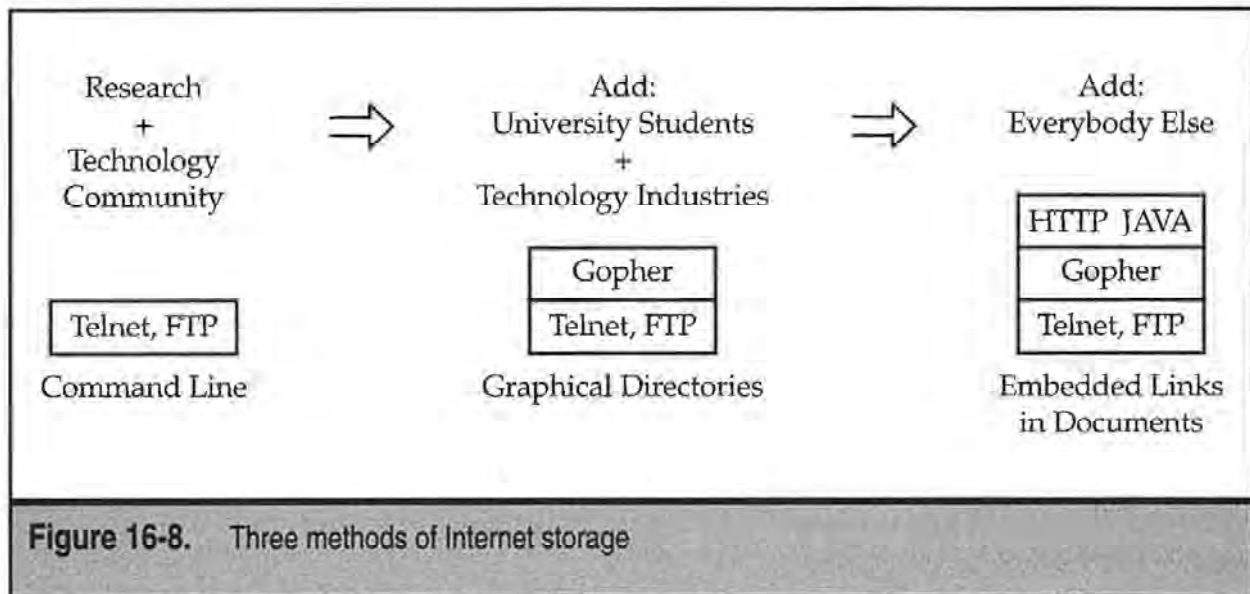
Finally, data can be stored and retrieved using the HTTP (Hypertext Transfer Protocol) used by web browsers to access web files from embedded links. The ability to store files is included in HTTP, but its use as a storage mechanism in lieu of FTP has been somewhat slow to be adopted.

Figure 16-8 shows the three methods of storage described in the preceding sections. On the left is the pre-Web FTP method used to upload and download files. The center is occupied by FTP and Gopher. The right side adds HTTP to create the World Wide Web, in addition to providing FTP and Gopher capabilities.

## Storing and Retrieving Data with FTP

Referring to Figure 16-8, the common facility across all three application environments is FTP. Considering its place in history, it is also the most popular mechanism for storing files on the Internet.

FTP is a client/server architecture. FTP client software runs in almost any system accessing the Internet. FTP server software can run in servers supporting other processes,



such as HTTP web services and e-mail services. FTP communications start when the FTP client establishes a control session with the FTP server using the Telnet protocol. Once the control session has been established, the client can initiate multiple file transfers to or from the FTP server. Each transfer session requires a separate TCP/IP connection.

**WHAT YOU NEED TO STORE YOUR DATA ON THE INTERNET** To access an FTP site, you need the following:

- ▼ The server name of an active FTP server
- A valid password, if it is a secure FTP site
- ▲ FTP client software or an FTP-capable application

There are both *secure* FTP sites and *anonymous* FTP sites. Secure FTP sites require a password to access them. Anonymous FTP sites allow anybody to download files from them, but typically do not allow any file uploads. Anonymous FTP sites do not require a password, but it is customary to use your e-mail address as a password.

FTP server names usually follow Internet naming conventions and begin with the letters ftp. For instance, the anonymous FTP site at ftp.knowareinc.com is an example of an FTP site that allows you to download an FTP program that creates a virtual drive on a Windows machine.

Readers interested in establishing or experimenting with personal storage capabilities on the Internet can contact their ISP and request an account for a personal web site. The ISP should be able to state whether there are any charges, and will create an FTP user account on an FTP server. After receiving the web site account information complete with server name and password, it is possible to read and write data from the FTP site.

**FTP APPLICATION AVAILABILITY** Most operating systems sold today come equipped with a text-based FTP program for those users familiar with FTP's command-line interface. However, the need to perform FTP operations from a command line is almost unnecessary. There are many excellent FTP software packages to choose from, and most are available over the Internet through software download sites. For example, [www.tucows.com](http://www.tucows.com) had 46 FTP programs available for Windows 95/98 systems at the time this was written.

However, it is not necessary to have an FTP program to be able to store and retrieve data from an FTP site on the Internet. Applications such as Microsoft's Office 97 applications provide access to files stored on FTP sites. For example, Microsoft PowerPoint presentations can be uploaded to a private FTP site from within the application and accessed later from a remote site. This makes it possible to retrieve files while traveling that may have been left behind at home or at the office. It is possible to sit down at virtually any Internet coffeehouse or Kinko's Copy Center that has Windows systems and Internet access and access your files. Notice that this does not require a browser. This access is provided completely through a user application; the system does not even need to have a browser installed to work.

## Web-Based Storage

Just as browserless FTP transfers are based on a client/server model, file transfers over HTTP are also based on a client/server process. Therefore, any new function has to be implemented in both client and server components. This co-development can take years, especially in highly competitive areas where many technologies and capabilities are involved. New functions sometimes do not become useful until a clearly defined standard is formed to provide interoperability between different vendors' products.

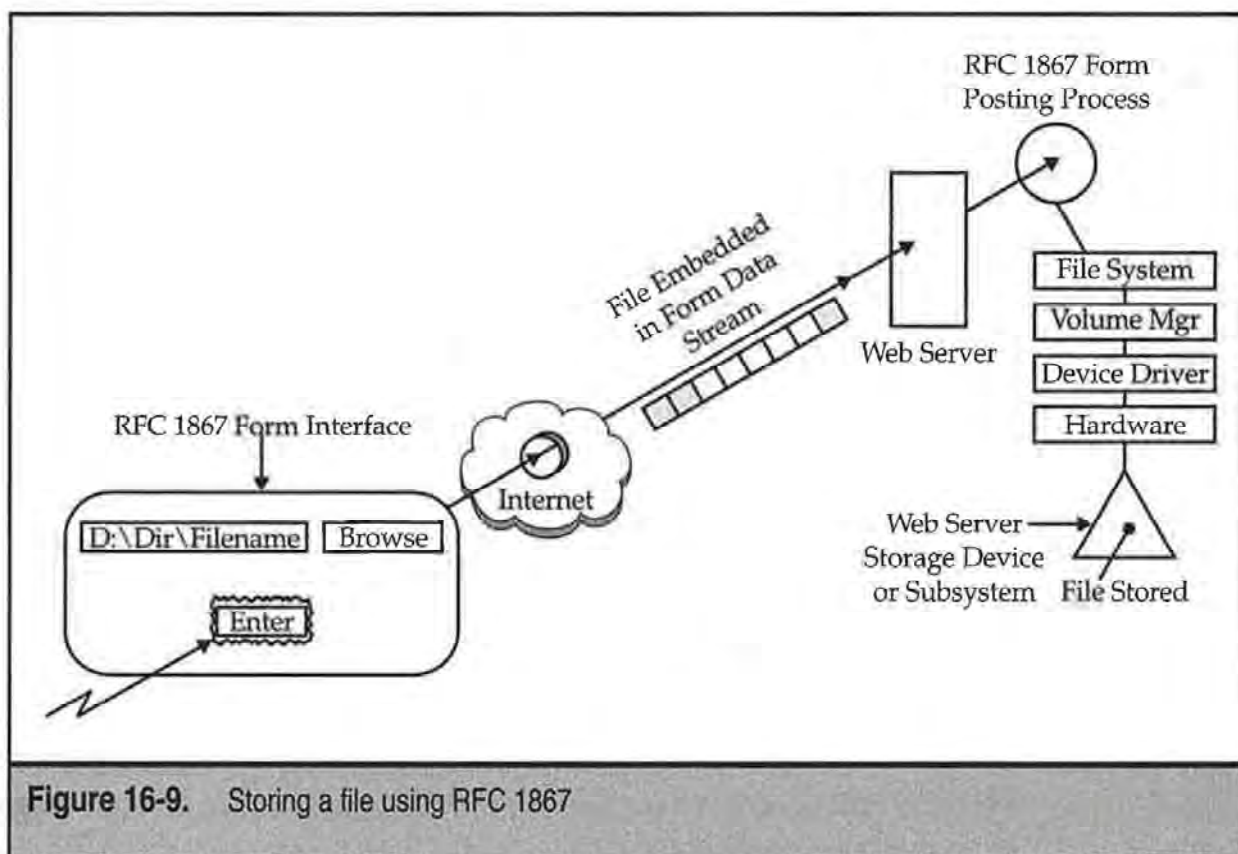
In the case of the World Wide Web, this scenario takes on an interesting twist because one of the leading web servers in the market is the Apache web server. Apache was developed as an open source technology with ongoing development by a legion of unaffiliated software developers. Because the Apache server software is so widely installed, it is very difficult for any vendor to create *de facto* Internet functions based on proprietary technology. New Internet standards cannot really become standard until they are included in the widely installed, open source products.

## RFC 1867

Internet technology has obviously undergone significant development over the last several years. One successful development has been the creation and adoption of RFC 1867 into the HTML specification, which specifies the use of HTML *forms* for uploading files from clients to web servers using HTTP (Hypertext Transfer Protocol) as the transfer protocol. Whereas FTP was developed years before browsers or the World Wide Web existed, RFC 1867 is browser-based technology designed to provide functionality for web applications or services. RFC 1867 was originally proposed in 1995 by two Xerox researchers, Larry Masinter and Ernesto Nebel, and since its introduction has been implemented broadly in browser and web server software.

The significant aspect of RFC 1867-based storage is that the facility is provided as an HTML web page, as opposed to being a separate client-side application. The file transfer begins by presenting the user with a form to fill out on a web page in his or her browser. The filled form contains the name and location of the file(s) they want to transfer. The actual transfer is accomplished by *posting* the form to a web server with the file identified as a field in the form. In other words, the file information is carried inside, or *tunneled* within the form data. A corresponding function running on the web server receives the form, identifies the file(s), and stores it according to the design and implementation of the application or service. Figure 16-9 shows the process of uploading a file to an Internet storage location using RFC 1867.

Server software for RFC 1867 is typically implemented as CGI (common gateway interface) scripts on Unix web servers or through ASP (active server page) integration in Microsoft web server products. Readers interested in developing integrated uploading and storage into their services or online applications can find out more about this at <http://www.15seconds.com/focus/upload.htm>. Companies such as Software Artisans (<http://www.softwareartisans.com>) develop software tools for web developers and administrators to facilitate the integration of RFC 1867 file transfer functionality into their web services or applications.



**EXAMPLES OF RFC 1867 WEB STORAGE APPLICATIONS** Readers interested in seeing how file uploads and accesses are managed using RFC 1867 may want to visit the following web sites offering free web storage. Each site has a different approach to storing and sharing files on the web, and also provides other functions. This list is certainly not intended to be exhaustive or show the complete breadth of RFC 1867 storage functions, but is intended to give the interested reader an opportunity to gain firsthand experience with it. In addition to the following sites, several of the online e-mail services use RFC 1867 for attaching files to e-mail messages.

▼ <http://click2send.com>

Click2send subscribers create storage boxes on their web site. Files that are uploaded to a subscriber's storage box are assigned expiration dates. The upload interface includes additional forms to fill in that generate e-mail messages for others who can access the site and download the file. The e-mail that is sent to the end-user includes the URL for the web site, as well as the user and password information needed to access the file. This is an example of a file exchange system that first transfers the file and uses e-mail as a notification process, instead of attaching a file with an e-mail message.

▼ <http://www.freedrive.com>

Freedrive doesn't make any excuses for its no-frills approach. They tell you to expect regular marketing e-mails in exchange for storing up to 20MB of files for free on their site. Their RFC 1867 interface provides the ability to select several files to upload simultaneously.

▼ <http://www.jointplanning.com>

The JointPlanning web site provides Internet "groupware" for collaborative projects. Functions including file uploading and downloading, common calendars, and e-mail reminders of project milestones or to do's. Group members share a common groupname and password to share data.

▼ <http://www.visto.com>

Visto provides a virtual briefcase function for subscribers, which includes file upload/download, calendar, e-mail, an address book, and a daily planner. Subscribers can share files with nonsubscribers by creating a guest web site that others can access by logging into Visto's web site using an assigned guest name and password.

## **Web-Based Distributed Authoring and Versioning, or WebDAV**

One of the problems with any storage resource is managing the data in it after it gets put there. Unfortunately, the tools for managing Internet-resident data are not very useful today, compared to the tools available for managing data in LANs and SANs. Indeed, it appears that the Internet is not used as widely as it could be for storing data. One problem discussed previously is the confusion and lack of knowledge among users as to how data

gets deposited in the Internet in the first place. But beyond that, there is great uncertainty about how to move it, copy it, edit it, or delete it after it is initially put on the Web.

**MANAGING DATA WITH WEBDAV** In response to the lack of manageability of Internet/Intranet data on web servers, a group of industry volunteers led by Jim Whitehead, a Ph.D. candidate at the University of California, Irvine, have developed new protocol extensions to HTTP 1.1 called WebDAV, for Web Distributed Authoring and Versioning.

According to the charter of the WebDAV working group, WebDAV is intended to "define the HTTP extensions necessary to enable distributed web authoring tools to be broadly interoperable, while supporting user needs." From a network storage perspective, this says that WebDAV will deliver the ability to write and read shareable data over the Internet. WebDAV contains the underpinnings to create shared, concurrent access to distributed Internet file systems.

As a protocol, WebDAV can be used to carry embedded management information about data stored at an Internet storage repository. This information, expressed as XML (extensible markup language) data, can interact with a wide variety of functions in storage servers, including policy-based management agents for data and storage management applications. In addition, WebDAV is designed to support distributed and dispersed workgroups that cross regional and organizational boundaries. The ability to create distributed software development environments is a direct target of the technology.

WebDAV is defined in RFC 2518, which can be viewed at Carnegie Mellon University's web site at <http://andrew2.andrew.cmu.edu/rfc/rfc2518.html>, as well as other locations. An active Internet Engineering Task Force (IETF) is working on the technology with volunteers from IBM, Microsoft, Xerox, and Novell, among others. Their work can be reviewed at <http://www.ics.uci.edu/pub/ietf/webdav>. This web site has a list of client and server applications that have implemented the WebDAV specification.

**EMBEDDED CONTROL IN WEBDAV** XML is used by WebDAV to convey *properties* associated with Internet data, such as the creation and transmission of descriptive metadata. WebDAV properties can be manipulated by the WebDAV protocol. An associated protocol called the DAV Searching and Locating protocol (DASL), can search for WebDAV data based on its XML properties. WebDAV also provides the ability to remotely establish and change access control information for data stored on Internet servers. This could obviously pose a security challenge, but it also opens the possibility for distributed project administration.

**USING WEBDAV WITH OFFICE 2000** In the preparation of this book, a Microsoft Office 2000 client was used to connect to a server at <http://www.sharemation.com/>. The web folders features of Office 2000 is a client-side implementation of WebDAV that represents server-side directories and files as graphical icons of folders and files, the same way they are represented in Windows' *My Computer* view of local devices. Using Office 2000 and the sharemation site as a client/server implementation, it was possible to write and read the manuscript for this chapter from within Microsoft Word, as if it had been stored on a local drive. No browser or FTP program was needed.

Internet speed affects everything, including book writing. Readers coming to this section of the book may already know a lot about WebDAV and will not need to know much more. But if not, the basic requirements for using WebDAV with Office 2000 applications are as follows:

1. Establish an account on a web server with WebDAV support. [WWW.Sharemation.com](http://WWW.Sharemation.com) is an example. This server account will probably require a user ID and password.
2. Create a web folder for Office 2000. This is done within the *My Computer* window in which devices and printers are created and managed. Web folders are installed as an icon, along with printers and devices, when Office 2000 is installed.
3. During the creation of the web folder, you will be asked for the Internet location of the web folder; you need to give it the complete URL for the web server, including your individual directory. For instance, at the Sharemation site, the URL given was <http://www.sharemation.com/~username>.
4. You will be asked for the user ID and password for the account while Windows validates the existence of the web folder.
5. Once the folder has been created, it can be used from within Office 2000 applications.

**THE FUTURE OF WEBDAV** Today, WebDAV is an unproven technology with relatively few implementations. The scope of the work defined in WebDAV appears to be fairly large and could result in significant changes to web storage, applications, and security. In other words, WebDAV contains some big ideas that could take several years to develop. Therefore, it is somewhat difficult to predict how quickly the technology will become adopted.

## Comparing Internet File Transfer Technologies

This section compares the transfer technologies discussed above. While there are always implementation differences between products in a class, it is possible to make some broad generalizations.

### HTTP Versus FTP Protocols

One of the primary differences between transfer technologies is the transfer protocol used, FTP or HTTP. The FTP protocol is used in FTP utilities and servers, and the HTTP protocol is used by RFC 1867 and by WebDAV.

Performance is roughly equivalent, with neither protocol having any inherent advantages. Both are implemented on top of TCP.

**ADVANTAGES AND DISADVANTAGES OF FTP** The primary strength of the FTP protocol is the wide availability of utilities and products that use it. Most FTP programs today use a

Windows Explorer interface, but a few use a virtual device approach that creates an icon on the desktop that supports drag-and-drop functionality. FTP transfers are even supported directly within some applications, such as Microsoft Office.

FTP supports direct access to data. The ability to freely access data from anonymous FTP sites has been one of the great enablers of the Internet for many years. Software developers in particular make great use of FTP in exchanging and distributing programs and code samples. The direct access to a remote file system through FTP provides a level of control that is not always possible when working through a browser or HTML interface.

However, as an Internet protocol that existed prior to the World Wide Web, FTP lacks some capabilities that have become standard ways of working on the web. For instance, FTP client and server operations and administration tend to be manually intensive and lack the sophisticated automated tools that have been developed for web sites and browsers.

Also, some of the most important transfer technologies are not supported by FTP. For instance, the FTP standard does not include encryption, although it can be implemented as a nonstandard extension. FTP traffic is often blocked by firewalls, and FTP data is not cached in proxy servers or Internet cache products, although it could be argued that not caching FTP data is a security advantage.

**ADVANTAGES AND DISADVANTAGES OF HTTP** The primary strength of HTTP for file transfers is its development as a web technology that takes advantage of new web capabilities. For example, HTTP transfers can be accomplished through firewalls, and they take advantage of web caching technology.

But the key to HTTP transfers is the ability to integrate file transfers with services and applications. For instance, HTTP transfers can include additional information as part of the upload process that could determine how the uploaded data is stored, identified, and shared. Beyond file transfers, HTTP uploads have the potential to be used for loading information into databases.

The downside to HTTP file transfers currently is the lack of support for the protocol in file transfer utilities and applications. For the most part, the integration of HTTP into web services or applications does not allow users to pick and choose their own file transfer tools; instead, they use the tools or methods provided by the service or application. This situation is likely to change over time, but in 1999 it is still a shortcoming with HTTP as a file transfer protocol.

## **Comparing FTP, RFC 1867, and WebDAV Access and Management**

The three types of file transfers, FTP, RFC 1867, and WebDAV all have different access characteristics that effect their transparency and how much of an interruption or barrier they may cause to users.

**FTP FILE ACCESS AND MANAGEMENT** FTP file access is mostly nontransparent. With the exception of virtual device FTP clients, a separate FTP program has to be started to read or write files. Beyond starting the application, users need to be able to log in to the FTP

site, if it is not a transparent FTP site, before they can access files. The entire login process, with the associated problems of remembering passwords and navigating the remote file system, is a long way from being transparent.

For users familiar with FTP and who access their sites fairly regularly, this is not a major problem. Many FTP utilities provide the ability to store access profiles for commonly accessed sites that include username and password information. Such utilities can have a user connected to a web site in as little as three mouse clicks. However, users unfamiliar with FTP can spend hours trying to get to this point. Virtual device FTP clients are somewhat easier to use, but they still require a working knowledge of FTP to set them up in the first place.

End-users have very little management control of their data in FTP sites. If they want somebody to share it, they can do so by giving them the same user ID and password that they use. There is no ability to establish an access hierarchy or read/write permissions.

**RFC 1867 FILE ACCESS AND MANAGEMENT** RFC 1867 transfers are by definition browser functions. This means a browser must be invoked to store and retrieve data. In addition, it means the web site and page must be accessed before the file transfer can start. Just as with FTP, this probably means a login process must be used to access the site. For instance, a subscriber wanting to upload a file on the Visto or Joint Planning web sites cannot access the web page for uploading files without first logging in. This is at least as much of an interruption as starting an FTP utility and logging in. It is possible to store login information in locally stored *cookies* if the service provides them. If login data is stored in cookies, the login process can be done transparently by the browser and bypass any user login activities. Users need to determine whether or not they want to store login information regarding private data storage within their systems.

Once the upload page is accessed, uploading files could not be easier. The user does not have to worry about choosing or accessing a directory. Downloads are easily accomplished by scanning a listing of stored files and clicking on the one wanted.

In addition, some of the RFC 1867 services provide the means to share data with different groups of users and to allow read-only privileges. This way, subscribers can function as administrators of the data they upload. The data management functions provided by an online storage service are determined by its developer (and the tools they use) and are not specified by any standard.

**WEBDAV FILE ACCESS AND MANAGEMENT** Of the technologies under consideration, WebDAV is the most difficult to discuss, due to its relative newness and lack of commercial implementations. However, some significant characteristics can be mentioned.

WebDAV access is designed to be transparent to the user. Realistically, it will probably be necessary to log into a WebDAV server first to authenticate the user. Once that step has been accomplished, the remote storage web site is accessed like an extension of the local system. This kind of transparency should enable users to work almost unencumbered by the data's location.

Managing data stored on WebDAV servers remains to be seen. The specification provides for XML embedded data, which could be used for many management features. At

this point it appears that these management features could be driven either by client- or server-side implementations. The fact is, the management of data over WebDAV will likely be one of the primary factors differentiating competing products and services.

## Internet Backup Software and Services

One of the earliest Internet-based storage applications to arise since the emergence of the World Wide Web is system backup and recovery. It makes a certain amount of sense to vault data to an Internet site, considering the chances are pretty good that the web site will not be destroyed by the same disaster that strikes a system in the office or at home.

For the most part, Internet online backup has been implemented as client/server, browserless technology using FTP as a transfer protocol. Unlike tape backup applications in which tape format compatibility is an ongoing concern, it is unlikely that the data written to a particular online service would be moved to another online service. Therefore, very little attention is paid to the storage format used on such sites.

Although the technologies are similar, the application challenges posed by web backup are almost the opposite of the problems posed by Internet data sharing. In general, backup data should be private and not accessible by others, unless they are chartered to retrieve it. This can include a single user who may have copies of data files on both a desktop system and a laptop computer.

### Characteristics of Internet Backup

Several services are available for online backup, providing a variety of functions, including such things as the ability to make CDs of backup data and ship them to subscribers and mix online backup with actual tape backup. The scheduling capabilities and user interfaces of the various products vary significantly. In general, there is a graphical user interface that provides point-and-click selection and exclusion of specified files and directories, and a scheduler that allows unattended backup operations to commence in the middle of the night.

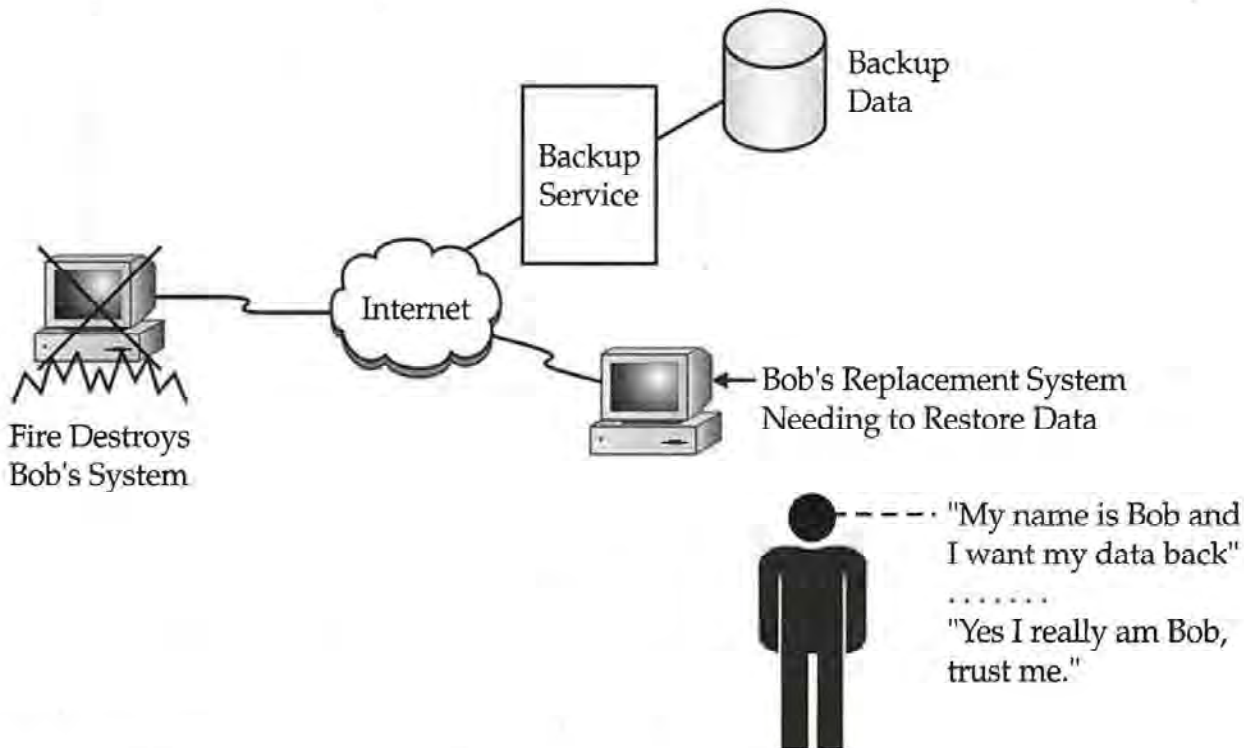
While all the online backup products have differentiating features, there are a few basic similarities among the various Internet backup products:

- ▼ Electronic distribution of client software
- Free trial offers and capacity-based pricing
- Local encryption at the client side
- Ability to move many files in a single operation
- Performance limited by Internet line speed
- ▲ Partial file backup technology

Internet backup applications are mostly sold and distributed over the Internet. Product updates can be managed with complete transparency to the end-user. In fact, some-

times the installation of the product is not completed until the web site is accessed for the first time. For the most part, Internet backup is not a free service, whereas file transfer and storage as discussed previously often is. However, most products offer a 30-day trial period. This should be plenty of time to adequately determine if the product will work for most environments.

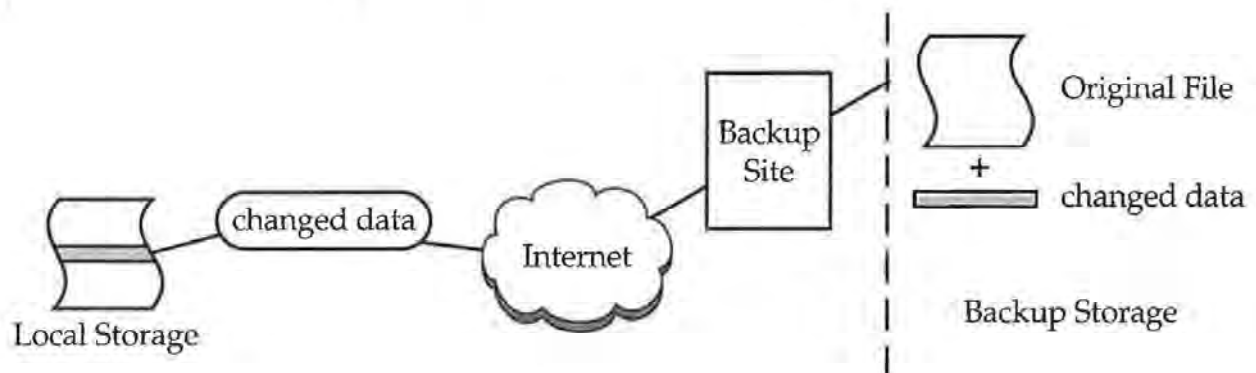
Subscribers are asked to provide an encryption key, or sometimes an encryption phrase, that is used to encrypt files before they are uploaded. It is up to subscribers to remember their user ID and encryption key to be able to restore or download files. This poses some interesting problems for disaster recovery. For example, a fire that destroys a small office is likely to destroy any written records containing account numbers and encryption keys. In that case, the software can be downloaded again and re-installed; the subscriber must then contact the technical support department of the backup service, which will give them their vital access info. It's not clear how such a "phone trust" system can be secure, but the practice is widespread among backup services. The process is illustrated here.



One of the advantages of Internet backup is the ability to transfer large numbers of files in a single user operation. When one has a large number of files to upload or download, backup provides the most service for the least amount of administration overhead. Unfortunately, the performance of Internet backup can leave a lot to be desired. A 56Kb modem cannot come close to the speeds of backing up to a locally attached tape or removable media device. DSL and cable modem speeds at approximately 150Kb/s are roughly equivalent to backing up to a diskette drive.

Compression technology can be used to increase the performance of Internet backup file transfers. In general, this can have the effect of doubling backup performance, but it depends on the data being transferred. Novastor's online backup product indicates the level of compression as a percentage of original file size while it is backing up files. It is interesting to see how certain files such as PDF documents are mostly uncompressible, while other types of files, such as HTML files, are highly compressible. So, compressing files during backup can be useful, but it may not provide the anticipated results.

For that reason, Internet backup services have implemented the ability to transfer partial files, instead of entire files. This means that the client software has to have the ability to recognize which parts of a file have changed, and only send those parts to the web site for storage. It also means the server software has to be able to integrate the partial file uploads with the existing file versions they now have. While this is not impossible, it is not necessarily trivial either. Besides capturing the changed data in the client system, it is also necessary to establish a consistent method for subdividing files and identifying any changed sections. Partial file transfers and storage are depicted in the following diagram.



The two technologies used to provide partial file backup are called *delta block* and *binary patch*. Delta block technology employs a method of parsing files into small block ranges. A checksum is calculated for each block range. When backup starts, new checksums are generated for each block range and compared with previous checksum values. If there are differences, the updated block ranges are transferred.

Binary patch has a similar way of identifying changed data in a file, but instead of transferring the data in block sections, it uses a technique similar to those used for application software for installing "patches." It appears that both techniques work well to improve backup performance. Delta block technology may have some advantages in the ability to recreate multiple historical versions of a file, but this ability is not highly touted in the existing Internet backup products.

Servers used for the "back end," or web side of these backup sites have specialized application software that provides the security and file management for backup data. In addition, the equipment used at these sites also differs considerably. Some use site mirroring

across geographic boundaries and others use regular tape backups and then store their backup tapes off-site.

The following is a short (and incomplete) sampling of backup sites on the Internet. Readers are encouraged to test these products to gain a better understanding of Internet backup and recovery capabilities.

▼ <http://www.tbackup.com/>

@backup provides a straightforward and simple-to-use online backup utility. One of the strengths of this service is the focus on backup scheduling, which encourages the subscriber during the installation process to establish a regular schedule for Internet backups. This philosophy establishes their Internet backup service as a management discipline, rather than just an Internet application.

▼ <http://www.compaq.com/smb/oneservices/Backup.html>

Compaq uses a product from Novastor Technologies called Novanet Web Online Backup for the end-user system software. As a company that has been developing workstation software for many years, Novastor successfully leveraged their backup expertise to the area of Internet backup. It is a direct, easy-to-use service that includes complete status information of online backup operations, including the compressibility of each file as it is being transferred. Compaq's online backup service is one of several services that provide online storage capabilities.

▼ <http://www.connected.com/>

Connected online backup (COB) has a very complete and informative user interface. Their product is easy to install and use. Besides providing Internet backup of single systems, this service also has licensing options for small businesses and corporations. Connected online backup is a professional service with a professional systems approach to Internet backup.

▼ <http://www.i-filezone.com>

One of the first companies to provide Internet backup, Atrieva has changed its approach and now positions its service as a file access service. Unlike most other Internet backup products, Atrieva is browser-based, but it does not use an RFC 1867 forms interface, instead using Java applets. To use the Atrieva system, the user must respond to Java access warning dialogue boxes that grant local access rights to the applet. Files are uploaded and downloaded using a point-and-click interface in the browser.

Atrieva storage is not managed like a backup application, but the process is similar. It allows multiple versions of the same file to be stored, and displays them all in a single view, listing their date/time stamps and size. This allows different team members to access the same document and update it with individual edits and changes, and each team

member can access any of the file versions. Different service levels are provided, as they are for other Internet storage services based on a fee paid subscription structure.

## INTEGRATING THE INTERNET WITH STORAGE AND I/O

Storage and I/O technologies are most effective when they are transparent to users. In general, users don't want to think much about how their data is stored and retrieved, they only want it to be available when they need it—with minimal requirements placed on them to remember operating procedures, names, locations, and passwords. This final section of the book speculates on how these requirements may be met by combining Internet and storage technologies so that users can have easy and secure access to their information when they need it, wherever they happen to be.

### Possibilities for Internet Storage Integration

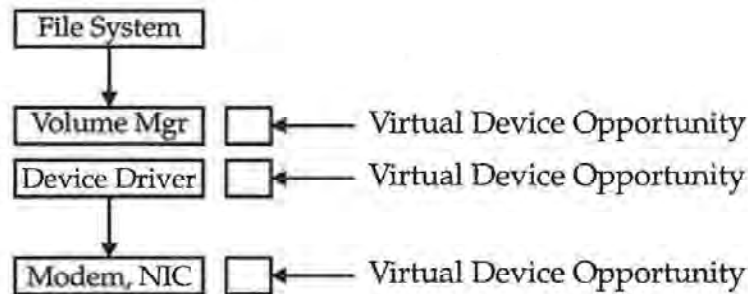
Where the Internet is concerned, there are many ways to do things and a seemingly endless supply of creativity. So, guessing about the most likely progression of technology integration is a gamble, but a risk worth taking if it stimulates progress. It is hoped that the ideas expressed in this section will be interpreted as a brainstorming exercise, as opposed to an Orwellian prediction.

#### Client System Level Integration

This section contains a brief examination of two techniques used to achieve transparent storage operations on client systems:

- ▼ Device emulation
- ▲ Installable file systems

**INTEGRATING STORAGE AND I/O THROUGH INTERNET EMULATED DEVICES** Device emulation provides a mechanism for one system hardware or software component to substitute for another one. The general idea of emulation is to provide a virtual device that behaves like an actual device and continue performing work as usual during normal I/O operations. Internet device emulation extends that concept to include Internet servers and services as part of the emulation process. Internet device emulation, therefore, should work the same as any local storage device, with the exception that data is transferred to a remote storage repository on the Internet. The device virtualization would necessarily be performed locally in software or in a hardware component, such as a modem or a set-top box. Some of the points along the I/O path where Internet device virtualization can be provided are shown here.



**FTPNETDRIVE** An example of a product that provides Internet device emulation through software is a product called ftpNetDrive from KnoWare, Inc. Interested readers may want to explore further by contacting KnoWare directly. Their web site is <http://www.knoware.com/>. ftpNetDrive virtual drives appear in the *My Computer* window and also integrate with Windows Explorer. Both virtual drive icons and Explorer windows accept drag-and-drop operations. Sharing data through ftpNetDrive requires all users to share the same user ID and password.

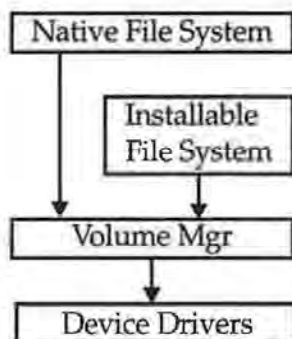
## Internet Emulated Device System

Figure 16-10 shows a basic software and hardware structure for implementing an emulated device for Internet storage. The system in Figure 16-10 has two types of devices: one physical and one virtual. Each device shown has a corresponding device driver and I/O hardware—a host controller for the physical device and a NIC for the emulated device.

Figure 16-11 shows another perspective: that of an FTP-based virtual device with the ability to access three different storage repositories on the Internet. This drawing also shows the relative presence of the device driver and the Telnet, FTP, and TCP/IP protocols.

## Integrating Storage and I/O Through an Installable File System

The concept of an installable file system was introduced in Chapter 13. The value of installable file systems is their ability to provide direct access for Internet file resources and to present those resources transparently to users and applications. In many respects, an installable file system is similar to a virtual or emulated device, except the installable file system integrates at a higher point in the I/O path. The relationship of an installable file system to the native file system is depicted here.



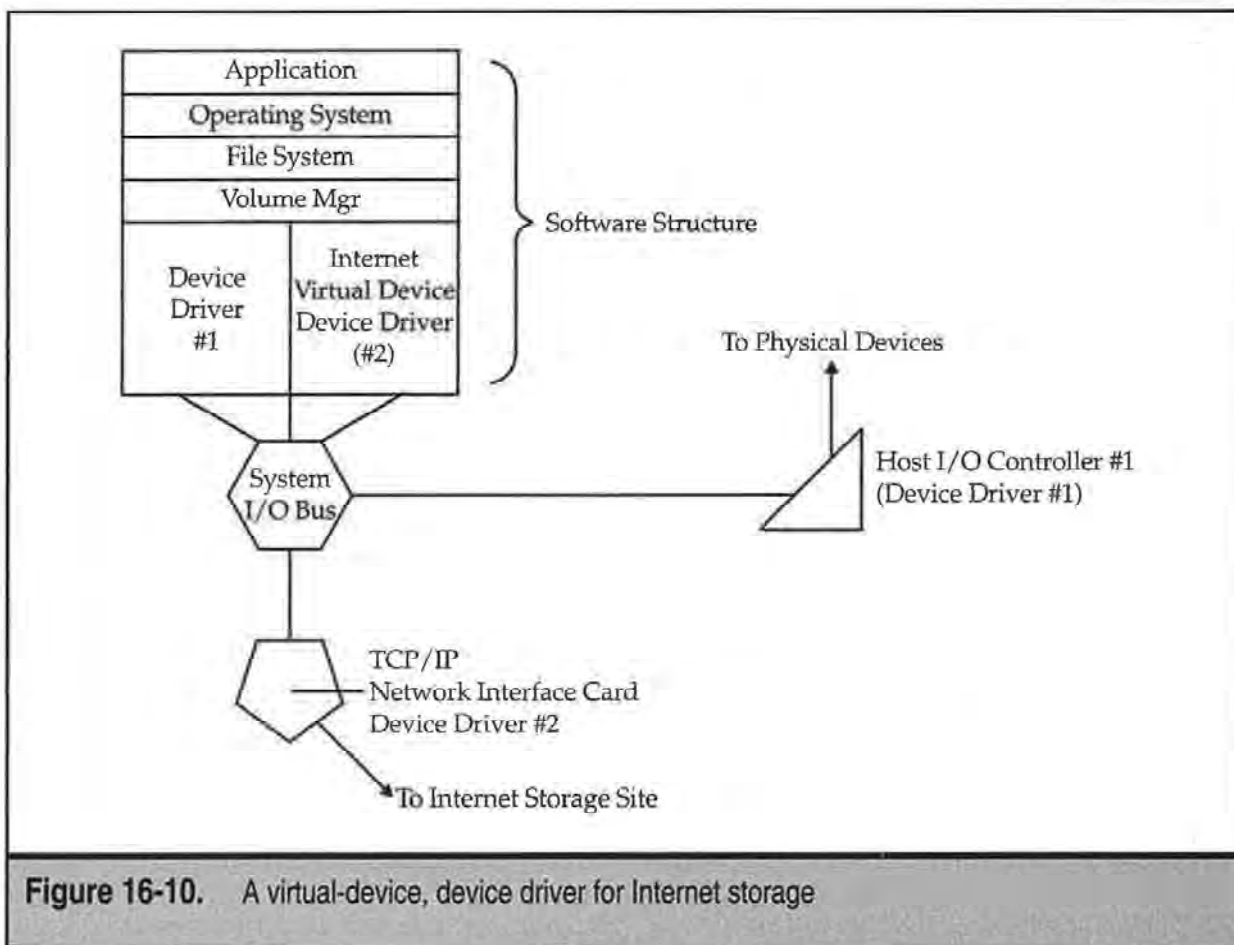


**Figure 16-10.** A virtual-device, device driver for Internet storage

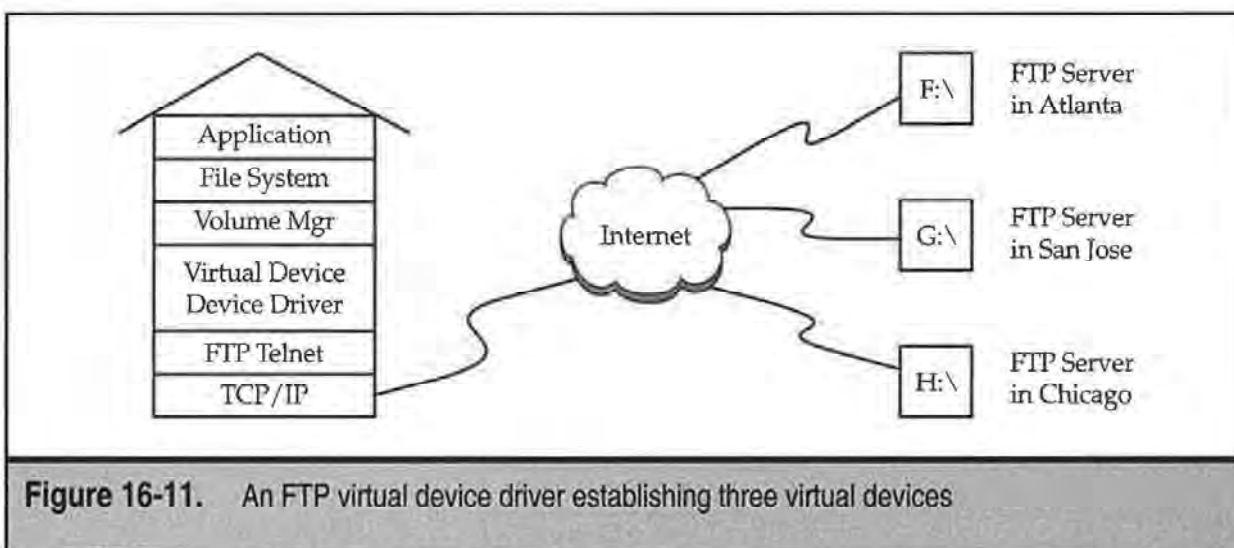


**Figure 16-11.** An FTP virtual device driver establishing three virtual devices

The goal of an installable file system is not to replace a system's native file system, but to augment it by providing the ability to access data in a different way, or to access data that cannot be accessed by the native file system. Considering the variables involved in Internet storage, an installable file system could be a very good approach.

The difficulty in using installable file system technology is teaching end-users something new. One of the advantages of device emulation is hiding the complexity of new technology from users, allowing them to continue to work in ways they always have. For that reason, any new installable file system implementations need to be easy to understand and use in order to succeed.

**WEB FOLDERS** The web folders feature in Microsoft Office 2000 was discussed previously when WebDAV was introduced in this chapter. Although the web folders feature is installed with Office 2000, the functionality is integrated through the operating system. Web folders are created and managed in the Windows' *My Computer* window, without needing to go through Office 2000 applications.

The WebDAV implementation used in web folders provides an installable file system on the Internet. Basic operations on files and directories through web folders work the same as they do on the Windows native file system. For instance, double-clicking on an icon for a file in a web folder not only downloads the file, but it also launches the application that created it.

## Enabling Technologies for Internet Storage Integration

This section examines additional technologies and explores how they might be used to provide Internet storage integration.

**USING JAVA AND ACTIVE X FOR STORAGE APPLICATIONS** In general, the question of using Java or Active X for Internet storage is an interesting one. Both provide the ability to download executable programs to a user's system and perform work. While this could be a real boon to providing consistent storage management in a large organization, it obviously poses some serious risks for data. The prospect of a malevolent Java virus applet infiltrating an organization and wiping out data is not a pleasant one.

For that reason, Java was developed specifically *not* to provide access to local files. Active X, on the other hand, allows access to local files to enable the integration of local and Internet data. Perhaps in light of competitive pressures from Active X, or due to requirements from software developers, Java has become more usable for transferring data to Internet sites. Java applets post dialog boxes on the user's screen, informing them that the use of the applet and its function is high-risk. This occurs both when transferring files and when accessing local storage.

**INTELLIGENT AGENT STORAGE MANAGEMENT** Assuming Java can be used for Internet storage integration, the next question is what would it be used for? Java's portability enables it to be downloaded or uploaded and to provide distributed consistent storage or

data management functions. Java technology could be deployed as intelligent agents that travel with the data they manage. The concept of intelligent agents for network management is certainly nothing new, but its use for Internet storage management represents a completely new type of technology development. Intelligent agents for storage could provide a variety of distributed and portable management functions, including the following:

- ▼ **Data moving** An agent transferred to a machine/device performs transfers to or from the Internet. This could be something like a Java FTP client (which already exists, as well as an FTP Java bean) or a Java application that complies with WebDAV to provide transfer functions. This function is somewhat controversial as a facility for storing data.
- **Delivery notification and audit** Agents that travel with transferred data and notify the sender of successful delivery to the receiver. The same or other agents could produce logs used for audit purposes.
- **Data synchronization** Agents that synchronize data across disparate systems and devices. This could include a synchronization mechanism that notifies and updates users working simultaneously on the same data file with change information.
- **Data compression and encryption**
- **Document routing** Agents that ensure that certain types of documents are routed to the right people in the proper sequence.
- ▲ **Policy enforcement** Agents that enforce data management policies such as retention, redundancy, and backup policies.

## WebDAV (Reprise)

A fair amount has already been said about WebDAV in this chapter. In short, WebDAV has incredible potential to integrate storage technology on the Internet. No other technology development effort has the potential impact WebDAV does.

## Have Meta data Will Travel: XML Encoding

One of the most significant developments in Internet standards over the last several years has been XML (Extensible Markup Language). While XML is generally viewed as one of the foundations for e-commerce applications, it is an extremely versatile technology with broad applicability. The strength of XML is its ability to communicate descriptive information, or Meta data, about the data being transferred. XML is a logical choice for transmitting information that could be used by Internet storage services. That's why it has been included as a key part of WebDAV.

## File Access Through Content Sealing

A potential problem with Internet storage integration is locating data. Many readers are probably familiar with the frustration of not being able to find the particular file, or ver-

sion of the file, that they need. This happens easily enough working within the confines of a single server, to say nothing of the size and dynamic of the World Wide Web. Locating the right version of a file is critical to the efficiency of a distributed workgroup collaborating on a project electronically. There is sometimes considerable confusion about which document version is the right one.

A little-known technology with excellent potential to solve this problem is *content sealing*. Content sealing enables data to be stored and accessed based on unique identifiers generated from the actual contents of the data. Files are retrieved by exchanging and acting on a file's *seal*, as opposed to scanning servers, directories, and file systems.

Wave Research (<http://www.waveresearch.com>) has developed the fundamental technology of an Internet storage system based on content sealing. Built on top of Internet standards such as HTTP, XML, and IP multicast, their e-CLIP (file seals), Java-based client agents and FilePool storage form a system they call *content addressable networking*.

E-CLIP file seals are formed using 128-bit MD5 cryptography techniques, virtually guaranteeing uniqueness for all files. Because they are based on a file's content, e-CLIP file seals also provide integrated integrity checking. Client agents provide access to FilePool servers and perform the sealing work on behalf of users. FilePool servers provide storage and storage management services on sealed files.

Client agents submit files and their seals to a primary or local FilePool server. From there, files are replicated or transferred according to system management policies. Access to files in the system starts with opening the file's seal. The client agent sends the seal to its primary server, which determines whether or not the file is stored within its pool. If it is, the file is loaded from the server. If not, the FilePool server communicates with other known FilePool servers to locate the file. This is roughly equivalent to having server-side retrieval agents in server systems.

Users exchange data by exchanging e-CLIP file seals. This is similar to exchanging an URL, but has the added benefit of integrated data authenticity and integrity. As users become comfortable working with Internet storage, storage systems that provide both ease of access and data authentication/integrity are likely to become very important.

## SUMMARY

This chapter was a departure from the previous chapters in this book. Instead of looking at traditional network storage techniques and technologies, this chapter examined how storage technologies can be used to support Internet computing and how storage applications are being developed that are accessed over the Web.

Both NAS and SAN products have strong features to help IT organizations build high-availability 24/7 web sites. A key feature of storage networking is the ability to segregate storage management from server management, allowing applications to be moved from server to server with a corresponding small impact on the storage and storage management at the site. The pace of change of the Internet is application driven, not server or storage driven. Server and storage infrastructure strategies that are modular and independent of each other, and are organized along application boundaries, will provide the growth potential needed to maintain a thriving and competitive e-business endeavor.

This chapter also looked at the types of services and technologies being offered to business and individual customers over the Internet. This is a fast-changing environment that will no doubt see a great deal of creativity and innovation in coming years. The ability to store, access, share, and exchange data over the Internet using a number of methods will provide new ways of working in the near future. One of the more interesting aspects of this will be the way in which the Internet is used, as distinguished from how the World Wide Web is used. It is hoped that the reader has gained an expanded understanding of these possibilities from this chapter and will be able to draw his or her own conclusions as Internet storage evolves over the next decade.

# Index



@backup web site, 559  
1000 MHz 64-bit processors, 171  
100BaseT switched topology, 50  
10BaseT switched topology, 50  
1244 Mass Storage Reference Model, 363  
1394 (Firewire), 27  
19mm tape devices, 234  
256MB/sec. technology, 65  
404 file not found errors, 535  
50-pin devices for SCSI buses, 63  
68-pin devices for SCSI buses, 63  
802.2, 387  
802.3 specification for collision, 92-93  
802.3X flow control, 415-416  
8mm tape devices, 232

---

## ▼ A

AAL5 (Asynchronous Transfer Mode  
Adaption Layer) protocol, role in fibre  
channel networks, 387  
Access arrays, reduced mode operations  
for, 204-205  
Access characteristics of FTP, RFC 1867,  
and WebDAV, 554-556

Access gate, role in integrated SAN  
backup systems, 363  
Access methods for IBM S/390 MVS  
I/O systems, 484-485  
ACLs (access control lists), 263, 446  
Active X, using for Internet storage  
applications, 563  
Add-in cards, configuring for  
servers, 172  
Address masking techniques for  
managing I/O paths in SANs, 340  
Addresses, consolidating with virtual  
drives, 197  
Addresses in fibre channel networks,  
claiming in loop initialization  
process, 411  
Administering open systems  
servers, 16  
Agent software components for  
network backup, 243  
Aging indicator/discard processor for  
caches, 155  
AIT (Advanced Intelligent Tape)  
technology, 237-238  
AIX systems, 65  
Algorithms for disk caching, 146-154