SpinFS® Architecture
Basic Concepts Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: http://www.netapp.com

Part number 215-00839
April 2004

Copyright and trademark information

Copyright information

Copyright © 2004 Network Appliance, Inc.; copyright © 2002-2004 Spinnaker Networks, LLC. All rights reserved. Specifications subject to change without notice.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980-1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ITS KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED OT, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED, IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Linux Kernel software that was developed and modified under the terms of the GNU General Public License, version 2. For a complete machine-readable copy of the Linux software and Linux Kernel software distributed with this product, please send an email request to:

This product includes the Java FTP Client Library that was developed under the terms of the GNU Lesser General Public License, version 2.1.

This product includes CIMOM software that was developed and modified under the terms of the Storage Networking Industry Association Public License, version 1.1. These software modifications are available via anonymous FTP from the Network Appliance web site.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT).

This product includes Kerberos software that was developed by M.I.T. and that was modified under the terms of the M.I.T. license.

This product includes Net-SNMP software that is covered by the following copyrights:

- Copyright © 1989, 1991, 1992 Carnegie Mellon University. All rights reserved.

- Derivative work—Copyright © 1996, 1998-2000 The Regents of the University of California. All rights reserved.
- Copyright © 2001-2003, Network Associates Technology, Inc. All rights reserved.
- Copyright © 2001-2003, Cambridge Broadband Ltd. All rights reserved.
- Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

This product includes software developed by the University of California, Berkeley and its contributors.

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the United States, Canada, the European Union, and Japan. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. DataFabric, FilerView, NearStore, SecureShare, SnapManager, SnapMirror, and SnapRestore are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp—the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, gFiler, MultiStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapVault, vFiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States. Spinnaker Networks, the Spinnaker Networks logo, SpinAccess, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, and SpinStor are registered trademarks of Spinnaker Networks, LLC in the United States, other countries, or both. SpinAV, SpinManager, SpinMirror, and SpinShot are trademarks of Spinnaker Networks, LLC in the United States, other countries, or both.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Basic Concepts of the SpinFS Architecture

This document provides an overview of the SpinFS® architecture and the SpinServer® system. It gives a summary of current standards in storage networking technology, then discusses the SpinServer system at a high level.

For more information about the concepts discussed in this document, see the *SpinFS 2.5 Administration Guide* and the *SpinServer 4100 Installation Guide*.

Current Standards in Storage Networking Technology

This section discusses current standards in storage networking technology, including network-attached storage and storage area networks, traditional file systems, and file-access protocols.

Storage Networking Technology: NAS and SANs

Currently, storage networking technology is divided into two major categories: network-attached storage (NAS) and storage area networks (SANs). These technologies are discussed in the following sections.

Network-Attached Storage (NAS)

Network-attached storage (NAS) systems include a server that is dedicated to file sharing and a collection of storage disks. Users can directly access the files that are stored on the disks and controlled by the server. NAS servers typically support one or more file-access protocols such as NFS and CIFS (see “File-Access Protocols” on page 2 for more information). A NAS server does not provide any of the applications that a typical server offers, including email or authentication. NAS enables additional disk storage space to be added without decommissioning the network while maintenance and upgrade tasks are being performed.

Storage Area Networks (SANs)

Whereas NAS systems provide file access to end users, storage area networks (SANs) provide storage-device access (for instance, access to tape drives and raw disks) to servers. Users do not directly access the devices; instead, users access their servers, which in turn access the devices.

Traditional File Systems

The basic function of any file system, whether simple or complex, is to store and retrieve data. A file system uses a structure of files and directories to maintain and locate data.

The term *file system* can describe any of the following:

- The entire file tree from the root directory downward
- A particular type of file system: local disk or network based
- The data structure of disk storage or another type of media-storage device
- The portion of a file-tree structure that is attached to a mount point on the main file system to make that portion accessible to users

A traditional network-based file system is a grouping of files and directories that can be accessed by clients across the network, regardless of the server on which the data is actually located. A network-based file system enables central administration of the various devices that store and serve data. A distributed file system presents the appearance of a single, unified file system, when the files are actually distributed among many servers across the network. This distribution of data over the network in a distributed file system is invisible to the end user, and is irrelevant to the completion of user tasks.

File-Access Protocols

To access files on a traditional network-based file system, UNIX clients use the Network File System (NFS) and Windows clients use the Common Internet File System (CIFS).

These file-access protocols enable clients to make direct connections to a file server device that resides on the network and to use the network-based file system to access data as if all files resided on the client's local hard drive.

The SpinServer Model

This section introduces the SpinServer system, lists its components, gives an overview of the system from logical and physical viewpoints, and discusses environments into which the SpinServer system is typically deployed.

The SpinServer System

The SpinServer system consists of two primary components:

- The SpinServer, a high-performance file server
- SpinFS, a distributed, virtualized file system

Running together, these components minimize the disadvantages of a traditional NAS system and provide the advantages of distributed file systems and the benefits of virtualization to NAS storage, including the following:

- High-performance delivery of content and data for Network File System (NFS) and Common Internet File System (CIFS) clients
- High capacity for storage
- A global file-system architecture that enables end users to view a cluster of SpinServers as a single file system
- Storage capacity that can be expanded without system downtime or disruptions
- Simple management and maintenance

The components are discussed in greater detail in the following sections.

Hardware Components

The center of the SpinServer system is the SpinServer, an Intel-based file server with the following features:

- Two Gigabit Ethernet ports for client connections.
- Two Gigabit Ethernet ports for cluster connections. SpinServers can be networked together into a cluster of multiple servers called a SpinCluster®; additional SpinServers can be added to the SpinCluster to increase the cluster's storage capacity and overall performance.
- One Ethernet port for management connections.
- Two dual-port Fibre Channel Host Bus Adapters (HBAs) for storage connections. Disk storage is on Redundant Array of Independent Disks (RAID) sets and Just a Bunch of Disks (JBOD) disk enclosures.
- Two redundant internal SCSI drives for the SpinFS software.
- Two redundant power supplies.

For more information on the hardware components of the SpinServer system, see the *Installation Guide*.

Software Components

SpinFS, the file-system software, runs on the SpinServer and provides the following components:

- SpinCluster for server clustering.
- SpinShot™ for creating snapshots of data.
- SpinRestore for restoring data from SpinShots.
- SpinMove® for nondisruptive file-system movement within a SpinCluster.
- SpinMirror™ for mirroring data within a SpinCluster.
- SpinManager™ for system management. SpinManager includes both a command line interface (CLI) and graphical user interface (GUI); both interfaces provide full access

to and control over the SpinServer system.

- SpinHA® for a high-availability system that has no single point of failure.
- SpinAV™ for virus scanning.

For more information on the software components of the SpinServer system, see the *Administration Guide*.

The Logical View

This section describes the logical view of the SpinServer system, including the virtualization of servers, file systems, interfaces, and storage pools. This section also discusses clustering and high availability.

Overview

The basic unit of storage on a SpinServer is the *virtual file system* (VFS), which is a conceptual container for a set of logically related directories and files. Virtual file systems create a single logical storage area that can be seamlessly managed across diverse platform types and geographically dispersed locations.

Virtualization of file systems simplifies storage administration responsibilities by creating a virtual view of all storage disks, connecting those disks to a server, and presenting this system as a single resource to the storage administrator and users.

A *virtual server* provides access to a set of virtual file systems through a set of *virtual interfaces*.

Virtual Server

A virtual server is a scalable server built on a single physical SpinServer or a cluster of SpinServers. A virtual server represents a separate file-system namespace with separate network access and provides the same flexibility and control as a dedicated server.

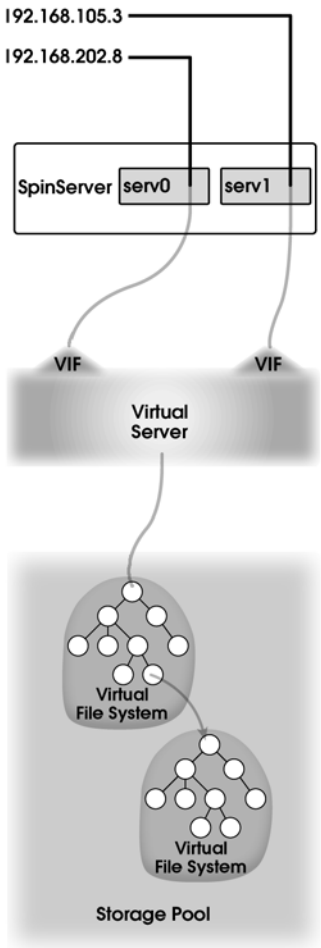

Figure 1. A virtual server built on a single SpinServer with a single storage pool

Each virtual server operates independently from other virtual servers that share the same host. The architecture of the virtual server is transparent to end users; the users see only a single virtual server with its virtual file systems. Each virtual server can point to numerous virtual file systems and can have an unlimited number of virtual interfaces.

Virtual File System

Each SpinServer has a collection of physical storage disks attached to it. The physical storage is divided into logical *storage pools* (SPs). A VFS lies at the level immediately above the storage pool and is the primary unit of storage management for a SpinServer or a cluster of SpinServers.

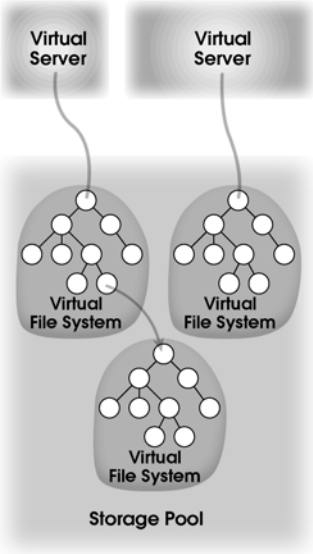

Figure 2. Multiple VFSs on a single storage pool can be connected to different virtual servers.

Any number of VFSs can be created on a given storage pool. Each VFS can be mounted within another VFS to create a file-system tree, as shown in Figure 3. VFS creation, deletion, and mounting are considered administration tasks. The VFS itself is not seen by users; only the file-system tree within a VFS is visible.

VFS Mount Points

For UNIX systems, the NFS concept of mount points means that you create a directory on an NFS client and then mount a remote file system on top of it. When a client accesses that directory or its contents, it is actually accessing the remote system. (For Windows NT systems, this functionality does not exist; it is available starting with the Windows 2000 concept of reparse points.) SpinServer VFS mount points are similar to UNIX mount points. In the SpinServer model, a mount point creates a junction that points from a directory in one VFS to the root directory of another VFS. VFS mount points are transparent to NFS and CIFS clients.

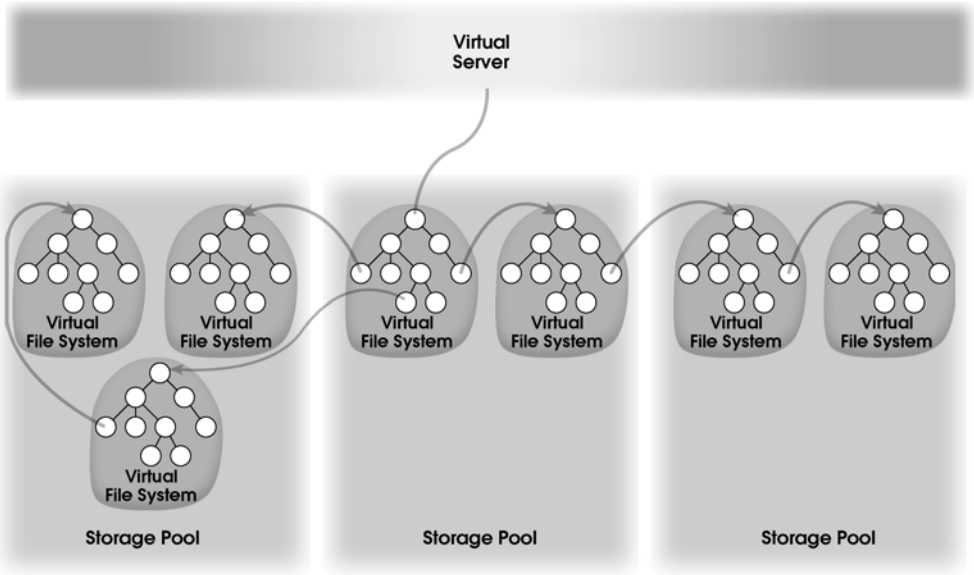

Figure 3. Multiple VFSs can be linked together by using mount points, creating a uniform namespace.

Uniform Namespace

The exact location of the VFS is transparent to end users of files and directories in the file system. Every user at every location sees the same collection of data files, and in this way, is presented with the same namespace.

Virtual Interfaces

Traditional systems closely align the Ethernet hardware with their associated IP addresses. In addition, traditional systems have only one IP routing table, which prevents accidental creation of duplicate IP addresses.

The SpinServer provides a virtual network-interface layer that defines the IP addresses and the IP routing table. Multiple virtual interfaces can use the same physical port, as shown in Figure 4. Each virtual network interface can reference one or more physical network interfaces for failover. If one physical interface fails, a second one takes over.

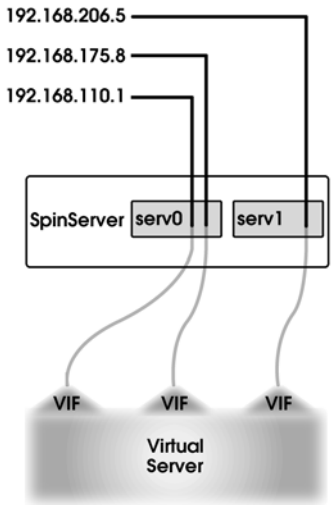

Figure 4. A virtual server can have multiple virtual network interfaces.

Data Management

In the SpinServer model, all data is contained within virtual file systems. This provides the following benefits for data management:

- Each VFS is assigned a quota of allocated storage (for instance, 1 GB) and a minimum reservation of storage space (for instance, 200 MB). The minimum quota guarantees that the specified amount of storage is available. This is done by reserving storage from the underlying storage pool. The maximum quota limits the amount of data that can be written to a VFS, but no reservation is made. Oversubscription is possible. The quotas can be modified at any time. Uses for quotas include ensuring that a minimum amount of space is available for large data sets and databases and enforcing limits on users' consumption of storage space.
- VFSs can be moved from one storage pool or server to another. A moved VFS maintains the same mountpoint it had before it was moved.
- Snapshots of VFSs can be made. SpinFS's snapshot implementation is called SpinShot. A SpinShot of a VFS is a new VFS, located on the same storage pool as the original VFS, that represents the contents of the original VFS at a particular point in time. A SpinShot shares disk blocks with the original VFS and thus is very lightweight. For instance, a SpinShot of a 2-GB VFS might use only a few kilobytes of disk space, depending on the extent of changes to the original VFS since the creation time of the SpinShot. If you create SpinShots by using the VFS Scheduler, the SpinShots are automatically mounted; if you create a SpinShot manually, you specify whether or not the SpinShot is automatically mounted.
- VFSs can be copied. A copy is a complete version of the original VFS with the same access (read-write or read-only) as the original VFS. It can be on the same server or a different server, and in the same storage pool or a different storage pool. It contains a complete copy of all data from the original VFS; it does not share blocks with the original VFS. For example, a copy of a 2-GB VFS uses 2 GB of disk space, whereas a SpinShot of the same VFS might use only a few kilobytes of disk space, depending on

the extent of changes that have been made to the original since the SpinShot was created. Copies are not automatically mounted.

- VFSs can be mirrored. A VFS mirror is a read-only copy of a VFS on a different storage pool; it can be synchronized with the original VFS whenever the administrator desires. Mirrors are automatically mounted.

For all VFS types, VFS operations such as reading and writing can continue while management operations are being performed, regardless of the size of the original VFS. For instance, if a 100-GB VFS is being copied to a different storage pool, the original VFS is fully available for reading and writing while the copy is being made. Additionally, if a parent read-write VFS fails, one of its SpinShots or mirrors can be promoted to serve as the parent read-write VFS.

Typical uses for the various types of VFSs include load balancing and disaster recovery. As an example of load balancing, assume that you have two storage pools of equal capacity named sp0 and sp1; the storage pools are served by SpinServers named Server0 and Server1, respectively. If sp0 contains 100 VFSs and sp1 contains 20 VFSs of similar size, you can move VFSs from sp0 to sp1 until each storage pool holds approximately the same amount of data, thus balancing the load between the SpinServers and the RAID sets that constitute the storage pools. As an example of disaster recovery, assume that the VFSs on sp0 are mirrored to sp1. In the event that the RAID sets that comprise sp0 fail beyond the point of recovery, the data that was contained in sp0's VFSs can be retrieved from the mirrors on sp1.

Other uses for the various types of VFSs include testing and development environments and data-warehousing applications.

The following figures illustrate the VFS types and operations.

Figure 5 shows a simple VFS configuration before any SpinShot operations are performed. In the figure, a single storage pool contains VFSs named FOO and BAR.

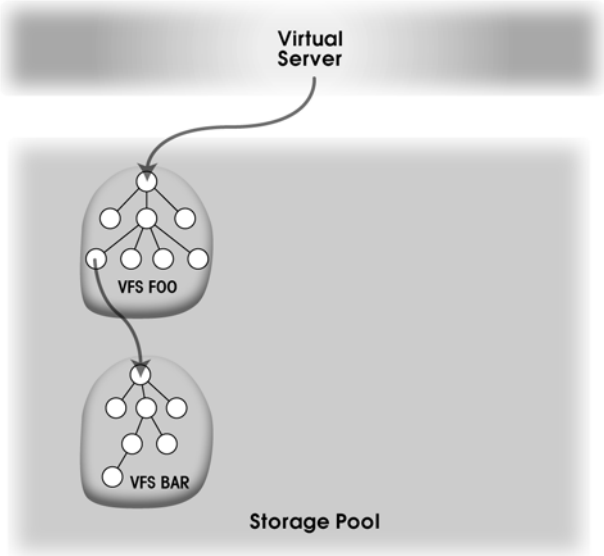

Figure 5. Starting position for SpinShot operations

In Figure 6, several SpinShots are made of the VFS named BAR. These VFSs reflect the changes made to the VFS at different points in time. The SpinShots can be used to retrieve data that was accidentally deleted or modified.

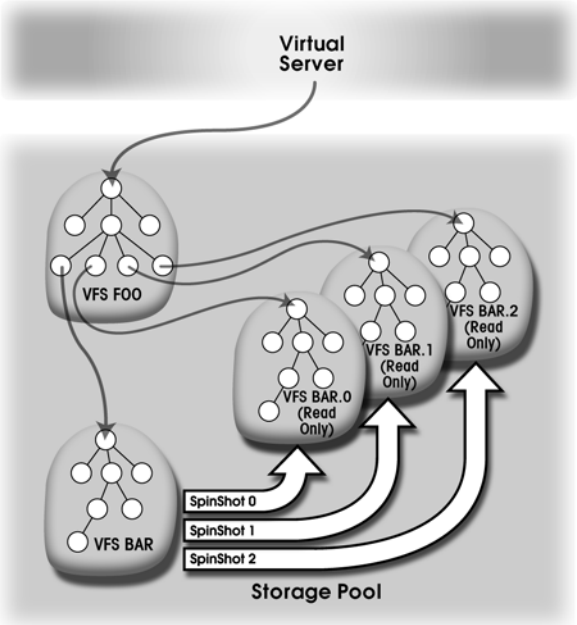

Figure 6. VFSs can use SpinShots to back up data.

In Figure 7, one-time, unmounted SpinShots are made of the VFSs named FOO and BAR. The SpinShots can be mounted and used as reference points as the data in the source VFS is modified.

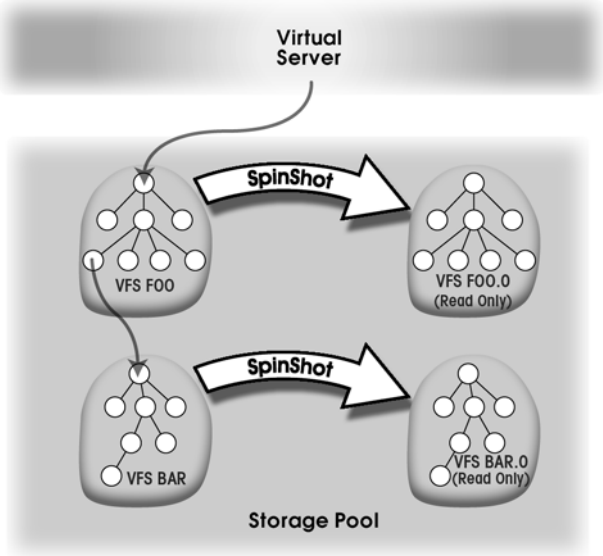

Figure 7. You can make one-time SpinShots of VFSs.

Figure 8 shows a simple VFS configuration before any mirror, move, or copy operations are performed. There are two storage pools. The first storage pool contains VFSs named FOO, BAR, and BAS. The second storage pool contains a VFS named BAT.

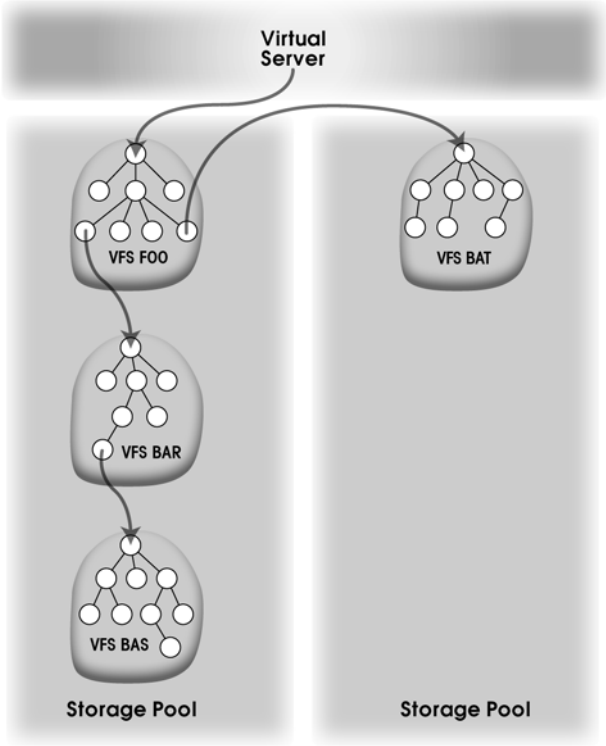

Figure 8. Starting position for mirror, move, and copy operations

In Figure 9, the VFSs named BAR and BAS are mirrored to the second storage pool. The mirrors can be used for data recovery in the event of a disk failure, or for fast read-only access to data.

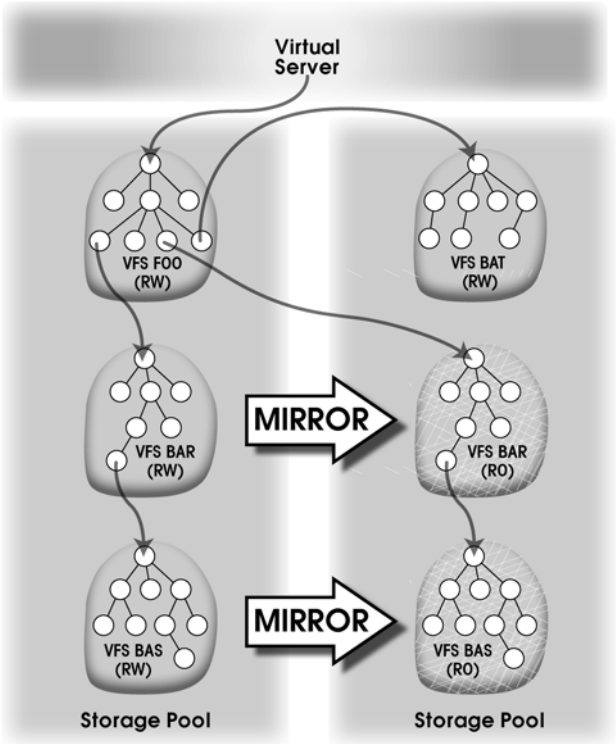

Figure 9. VFSs can be mirrored to a different storage pool.

In Figure 10, the VFS named BAR is moved to the second storage pool. After BAR is moved, there is an equal number of VFSs on each storage pool.

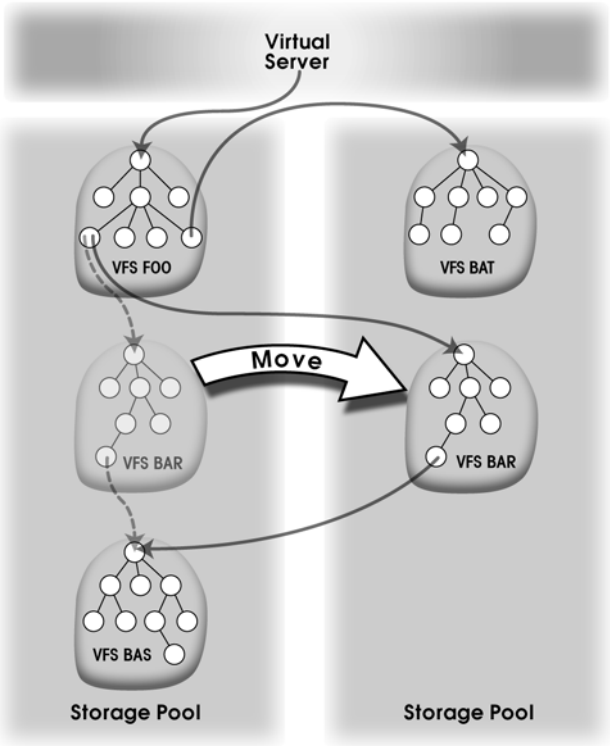

Figure 10. VFSs can be moved to a different storage pool.

In Figure 11, the VFS named BAR is copied to the second storage pool. After its creation, the copy can be mounted and used for testing and archiving.

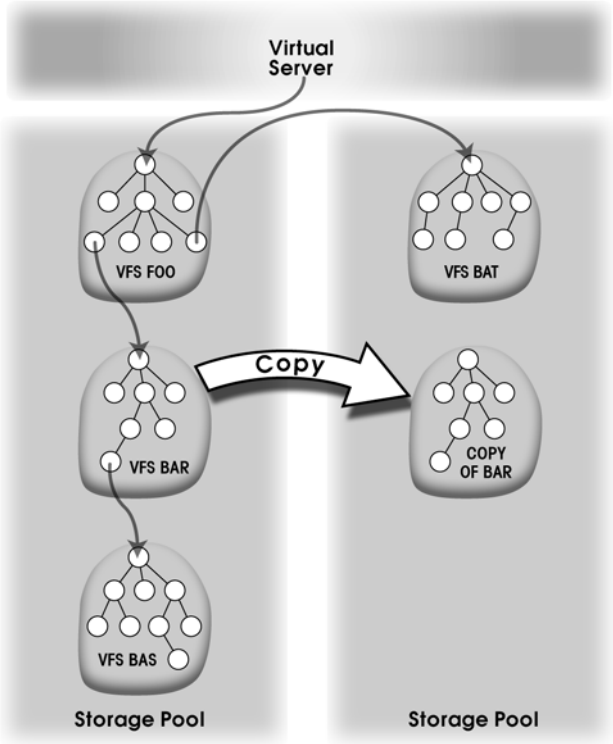

Figure 11. VFSs can be copied.

Clusters

You can create a grouping of SpinServers to form a scalable cluster, the SpinCluster, as shown in Figure 12. Each SpinServer in the SpinCluster can see the same virtual file systems as any other SpinServer in the SpinCluster. The total file-system namespace, which comprises all of the VFSs and their resultant paths, is global across the cluster.

The cluster enables multiple SpinServers to pool their resources into a large virtual server and to distribute work across the cluster. Clustering also enables continuous service to end users if individual SpinServers go offline or are taken offline. When a new SpinServer is added to a SpinCluster, there is no need to update clients to point to the new server. The existence of the new server is transparent to the clients.

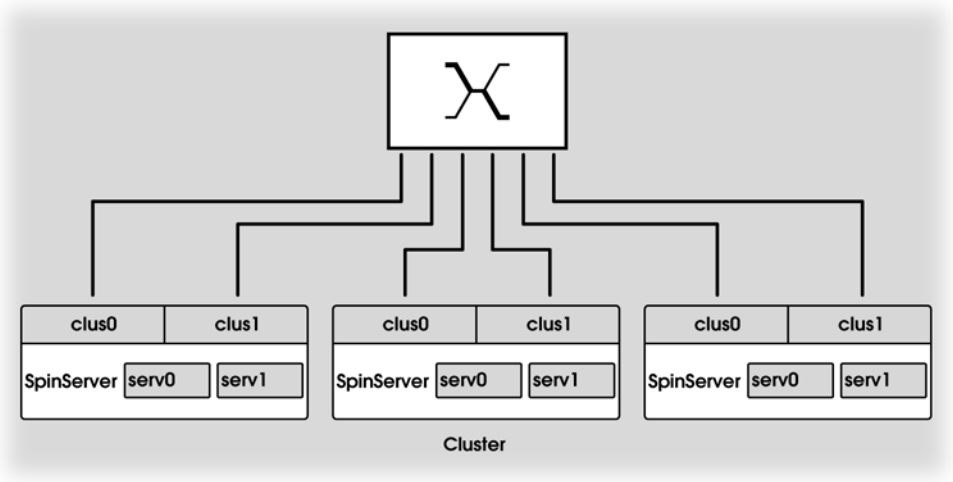

Figure 12. Multiple SpinServers can be grouped together into a cluster by using an Ethernet switch.

The key concept in the cluster is *quorum*, which is a simple majority of SpinServers that participate in the cluster. Each SpinServer in the cluster is known as a *node*. Each node in the cluster participates in a voting protocol that elects one node *master*; each remaining node is a *secondary*. The master is responsible for synchronizing information across the cluster. When quorum is formed, it is maintained by continual voting; if the master goes offline, a new master is elected by the nodes that remain online. Quorum must be maintained for the cluster to function fully. Because there is the possibility of a tie in a cluster that has an even number of nodes, one node can be configured to hold *epsilon*, which is an extra fractional voting weight. For instance, in a four-node cluster, quorum can be maintained if two nodes cast the same vote and one of those nodes holds epsilon. The system administrator sets epsilon on one node at most. Epsilon can be unassigned and reassigned as needed.

High Availability

A high-availability (HA) system builds on the features that are already provided by the base SpinServer system, including scaling, clustering, and RAID redundancy. An HA system ensures that data remains available even in the event of a SpinServer failure. Advantages of a high-availability system include the following:

- No single point of failure in the storage solution. If a component of the RAID set fails, the RAID architecture ensures that another component takes over for it. Building on that capability, if a SpinServer fails in an HA configuration, another takes over for it. Data and all management functions remain available regardless of a failure at any point in the system.
- Less need for system-administration resources. If a failure occurs, the system automatically compensates and continues to serve data; no human intervention is required.

High availability is a requirement in any environment that requires uninterrupted access to data, including global operations that require daily 24-hour availability and high-volume, data-intensive applications such as databases.

A high-availability system consists of two or more SpinServers and the optional SpinHA software. In an HA system, two or more SpinServers share responsibility for a particular storage pool. If one SpinServer fails, the other automatically takes control of its disks and continues without losing any committed data. Additionally, virtual interfaces can be configured so that if a server port fails, the virtual interfaces are migrated to another port.

The HA configuration guards against a SpinServer failure but not against a catastrophic failure of multiple disks (a failure that is beyond the ability of the RAID subsystem to contain). Because RAID sets are highly reliable, the HA configuration is an optimal solution for many storage administrators.

The physical setup of an HA system consists of two or more clustered SpinServers and one or more RAID and JBOD disk enclosures. The SpinServers communicate with each other across an Ethernet network connection. Each SpinServer communicates with the disk enclosures across Fibre Channel connections. If a server in the cluster fails, other servers in the cluster detect the failure across both the Ethernet and Fibre Channel connections. Failover is executed according to cluster topology and user-defined failover configurations. The goal of SpinServer failover mechanisms is to maximize the availability of data to users and to ensure the continuing functioning of the cluster. In a properly configured cluster, all user-visible resources on one server can be served by other cluster members.

The logical view of an HA configuration consists of two or more SpinServers that share responsibility for a storage pool. The SpinServer that actively serves the storage pool is called the *primary*; the other server is called the *active secondary*. Additional servers can be added to the configuration as *passive secondaries*. If the primary fails, the active secondary takes over as primary; if any passive secondaries are available, one is promoted to active secondary. A passive secondary is promoted to active secondary on the basis of its *priority*. For instance, if there are three passive secondaries with priorities of 1, 2, and 3, the passive secondary with priority 1 is the first to be promoted to active secondary, followed by the passive secondaries with priorities 2 and 3.

To illustrate a standard HA configuration, assume that you have two SpinServers named Server1 and Server2, and one storage pool named sp0. Server1 is the primary; Server2 is the active secondary. There is no passive secondary. The following scenarios can occur:

- If Server1 fails, Server2 takes over as primary and continues to serve sp0. Server1's virtual interface is failed over to Server2.
- If Server2 fails, Server1 continues as primary and continues to serve sp0. Server2's virtual interface is failed over to Server1.

If additional nodes are available in the cluster and are configured as passive secondaries, one or more of those passive-secondary nodes is promoted to active secondary if either the primary or the active secondary fails.

Use SpinManager to designate the SpinServers for a given storage pool as primaries, active secondaries, and passive secondaries and to configure virtual interfaces for failover. You can also manually fail over and promote servers, and manually fail over virtual interfaces. This is useful if you are testing an HA configuration or if you are balancing the load in a cluster.

The Physical View

The virtual components of the SpinServer are backed by the physical disks of the storage system.

Storage Pools and Disks

The VFSs are contained in storage pools, which are represented by the storage system as one or more RAID sets. Each RAID set comprises 3 to 16 disks, which can be located on RAID and JBOD enclosures.

A storage pool is a logical grouping of physical storage, as shown in Figure 13. Physical storage is defined in terms of storage units. A storage unit can be a single JBOD disk or a RAID set. A storage pool contains a defined amount of physical storage that can be expanded dynamically at any time. For example, if a storage pool is made up of three disks, an additional disk can be easily added to the storage pool, and the logical size of the storage pool can be increased accordingly, without disrupting any user or process currently using that storage pool. Storage pools can include hot spare disks that hold data if one of the other disks fails.

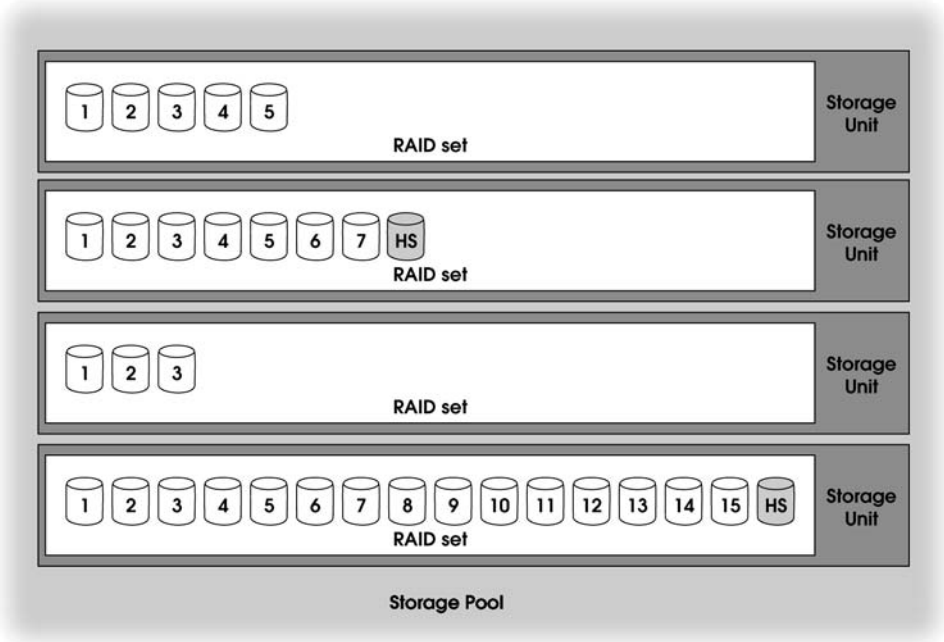

Figure 13. A storage pool contains one or more storage units. Each storage unit is a single RAID set.

Ports and Network Interfaces

Network ports represent physical Ethernet connections. Network interfaces are associated with network ports and have assigned IP addresses.

The network interface for the management port in the SpinServer system must be configured with a single IP address. The file-server ports can be configured with one or more interfaces and IP addresses.

IP Addresses

Each virtual interface attached to a virtual server is assigned an IP address, as shown in Figure 14.

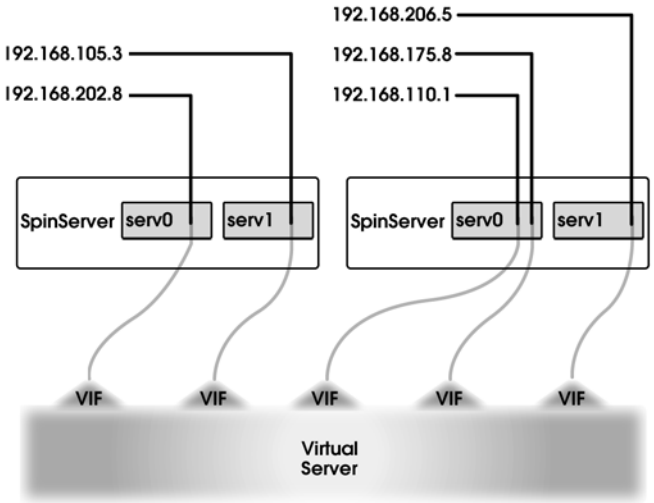

Figure 14. An IP address is bound to a specific virtual network interface through a specific physical port.

Advantages of the SpinServer System

This section discusses the advantages of the SpinServer system, including virtualization and scaling, performance, and management optimizations.

Virtualization of NAS

Historically, NAS has been viewed as lower-end storage. NAS servers have traditionally had relatively low capacities and performance. Most storage-management vendors have not considered virtualizing NAS because these servers have not been able to support the large application input/output (I/O) demands that benefit most from virtualization.

Furthermore, while NAS servers are relatively easy to deploy, the growing number of NAS servers causes rising management and administrative concerns and costs.

SpinServer Optimizations

For NAS virtualization to be a successful improvement, it must address the scaling, performance, and management drawbacks of traditional NAS system designs.

SpinServer Scaling Solutions

The SpinServer eliminates the one-server, one-file system design. When more storage space is needed, additional SpinServers can be added to the cluster, each with RAID and JBOD disk enclosures added. These upgrades are invisible to users and applications alike.

Multiple file systems can be linked across multiple servers to create a single logical file system, as shown in Figure 15. By using a distributed file system that enables multiple servers and their associated storage disks to be addressed as if they are part of the same seamless storage space, the SpinServer provides scalability without system downtime.

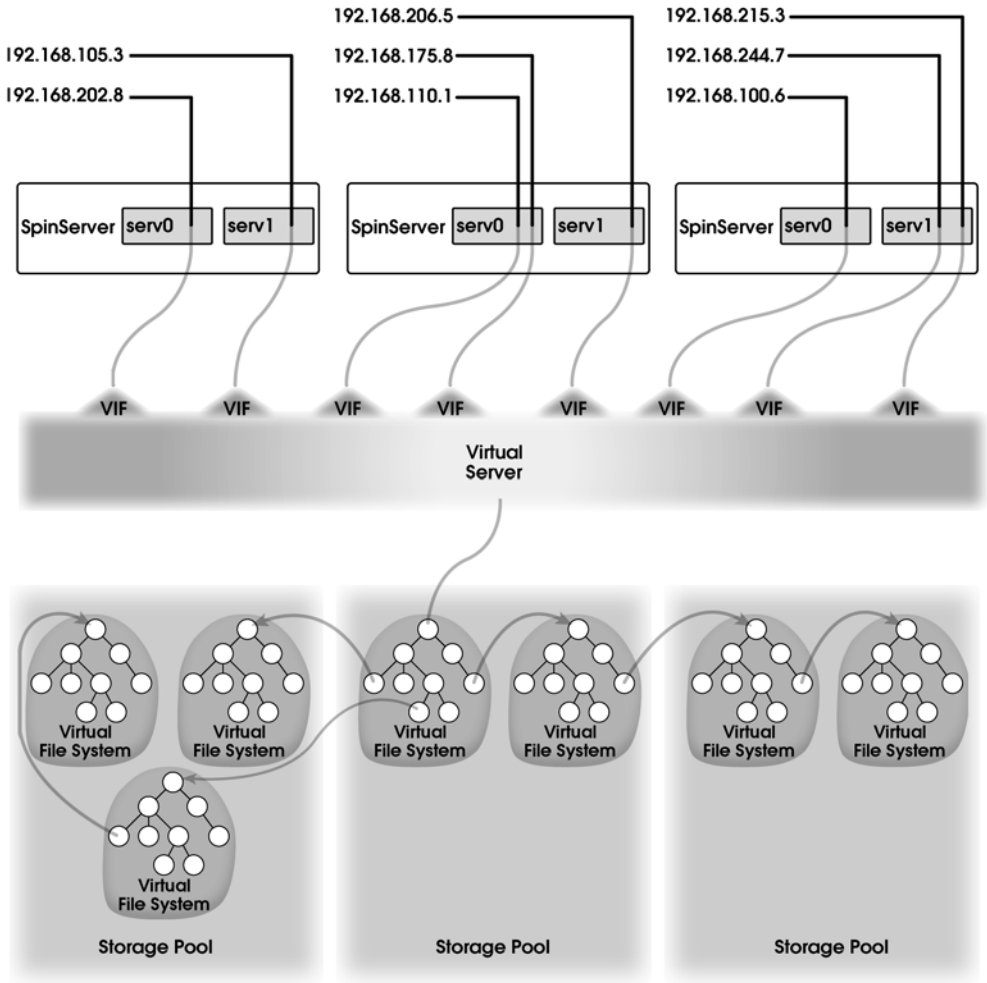

Figure 15. A single file system can be distributed across multiple servers, which are associated with a single virtual server.

SpinServer Performance Solutions

Using a distributed file system enables the SpinServer to improve application I/O performance and scaling capabilities. With all SpinServers managed as a single storage space, each new SpinServer adds new I/O channels for users.

SpinServer Management Solutions

By virtualizing data servers, storage space can be presented and managed as a single storage pool. The SpinServer serves the same data to all clients, regardless of the file-access protocols they are using.

The virtualized storage pool is managed with SpinManager, SpinFS's management software. The command line interface (CLI) and the browser-based graphical user interface (GUI) present two views of the SpinServer system and can be used interchangeably.

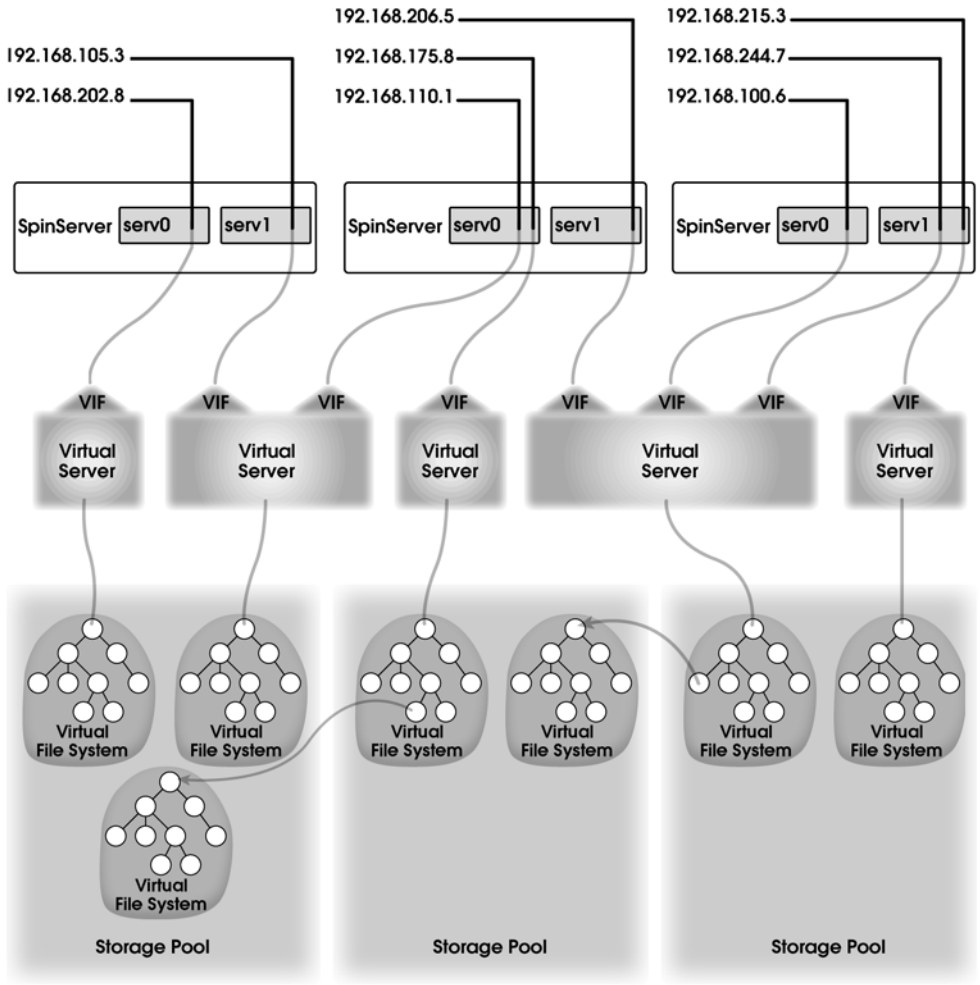

Figure 16. Multiple virtual servers can access the same storage pool.

As shown in Figure 15, multiple virtual servers can store files on the same storage pool, but each virtual server has its own VFS, and its files are inaccessible to other virtual servers.

Typical SpinServer Applications

This section discusses environments and applications that are well suited to the SpinServer solution.

Large User Community

In this environment, a large number of users require access to file and print services. Each user has a home directory.

With the SpinServer system, each user's home directory is implemented as a separate VFS with a quota of allocated storage (for instance, 1 GB) and a minimum reservation of storage space (for instance, 200 MB). You can add users to any SpinServer in the cluster based on criteria such as free space on storage connected to the SpinServer, not on the location of servers or data.

As the number of users and amount of data grows, you can add new SpinServers and capacity without any changes visible to users. You can move users' VFSs to different SpinServers and perform maintenance tasks on the SpinServer cluster without disrupting users' access to their data.

Branch Office with WAN Cluster Link

In this environment, a cluster of SpinServers is maintained at a central location, and a smaller cluster (even a single SpinServer) is kept at a branch office. The two clusters maintain communication over a wide area network (WAN) connection. The SpinServers provide a single namespace for users at both locations; in this case, each user has her or his own VFS, and all of the VFSs are mounted on a top-level directory named `/user`. VFSs for users at each location are stored on the SpinServer cluster at that location (central office or branch office). Both types of users have fast access to their data over the local area network (LAN) at their location and can also access data at the other site over the WAN. This provides users with the fastest possible access to their data at both locations.

The top-level `/user` directory is mirrored to ensure speed and availability of data. In the event of a WAN failure, the SpinServer system continues to run autonomously at both locations. Users at both the central location and the branch office have uninterrupted access to local data, and requests for data from the other location are queued to be processed when the WAN link becomes available again.

Parallel Data Processing

In this environment, numerous data-processing servers require high-bandwidth access to a large data set. With the SpinServer solution, the total bandwidth available to the data-processing servers increases near-linearly with the number of SpinServers in the cluster. You can redistribute the data online (for instance, move a growing data set from one SpinServer to another) without updating the configuration of the data-processing servers and without interrupting their access to the data.

Multiple Types of Storage

In this environment, different types of storage hold different types of data. For example, data for active projects can be located on high-performance storage pools, but data for inactive projects can be moved to lower-cost storage pools. Data can be moved between the different types of storage without changes to the namespace and without disrupting access to the data. You can match the type of storage to the importance of the data that it holds, thus using storage resources more efficiently and saving money.

Databases

In this environment, a database is stored on one SpinServer in a cluster. For disaster recovery, the database is mirrored to another SpinServer at a remote site across a WAN or metropolitan area network (MAN); the mirror is replicated every five minutes. The mirror replicas (SpinShots) provide a low-cost backup of the database at different points in time; only data that has changed is included in each SpinShot. Because the SpinShots are online, it is easy to recover different versions of the database as needed.

This configuration provides a solution for high-performance database applications with multiple clients, such as a database that serves a web server farm. Updates (writes) to the database are made only on the primary database; these updates are then propagated to the mirrors via replication. Read-only clients can then access the mirrors; local mirrors enable scaling to many thousands of database users, and WAN link failures do not interfere with read-only clients.

NAS and File Server Consolidation

In this environment, a SpinServer cluster is implemented to replace multiple standalone NAS servers, file servers, or both. Each legacy NAS server or file server can be replaced with a virtual server on the SpinServer system. Each virtual server can be given the same name as the legacy server it replaces, so end users do not have to migrate their NFS mounts and CIFS shares to new host names. A SpinServer cluster can host up to 8 virtual servers that serve CIFS clients and up to 256 virtual servers that serve only NFS clients. The ability to migrate legacy host names to virtual-server names simplifies the transition from the legacy system to the SpinServer cluster.

There is no sharing of data among the SpinServer's virtual servers, each of which has its own VFSs and IP subnet to provide security similar to that of a firewall. Multiple departments and users can thus share the same servers efficiently, with no danger of security breaches.

High-Availability Systems

In this environment, two SpinServers share responsibility for a particular storage pool. Multiple SpinServers and storage pools can be configured into a high-availability (HA) system. If one SpinServer fails, the other automatically takes control of its disks and continues to provide full access to data without losing any committed data. Additionally, virtual interfaces can be configured so that if a server port fails, the virtual interfaces are migrated to another port. A SpinServer can be the primary server for one or more storage pools while being assigned as the secondary (backup) server for other storage pools; thus, all resources in the SpinServer system are used efficiently, with no servers waiting unused until another server fails.

Each storage pool is configured to fail over independently, enabling you to balance the workload in the event of a SpinServer failure. For instance, assume that you have two SpinServers in an HA system, and each SpinServer is the primary server for one storage pool, as shown in the following table:

Table 1. SpinServers and Storage Pools in an Example HA Configuration

Server	Pool
Server1	PoolA
Server2	PoolB

In this example, each storage pool can be configured to fail over to a different SpinServer. For instance, PoolA can be configured to fail over to Server2, and PoolB can be configured to fail over to Server1. If a server fails, its storage pool fails over to a different server, thus distributing the workload across the cluster. There is a limit of four storage pools that can be configured to fail over per cluster node; when designing the failover configuration, you must assure that no node in the cluster can ever be responsible for more than four storage pools.