realtimepublishers.com™

The Definitive Guide™ To

Building Highly Scalable Enterprise File Serving Solutions

PolyServe™

Chris Wolf

Copyright Statement

Chapter 5: Building High-Performance, Scalable, and Resilient Linux File-Serving Solutions

The last chapter took a close look at the world of Windows file serving. This chapter will take a similar approach with Linux file serving. Although many of the challenges facing file serving today remain consistent between both Windows and Linux operating systems (OSs), the approach to solving the challenges presented by enterprise file-serving for these OSs certainly differ.

In this chapter, you will see the world of file serving from a Linux perspective. Along the way, you'll get a close look at the several challenges facing file serving on Linux platforms. We will also examine the current alternatives (both commercial and open source) for solving the performance, scalability, and availability file-serving issues.

Another major aspect of Linux file serving is the ability to integrate Linux file servers onto Windows-based networks. With a Windows Active Directory (AD)-dominated client base, building Linux file-serving solutions that can seamlessly integrate into an AD infrastructure is deemed critical by many organizations. To that end, this chapter will also introduce several of the technologies that promote file serving across the heterogeneous enterprise.

> 📖 Chapter 6 will provide detailed procedures and examples of Linux and Windows integration concepts, such as simplifying authentication using winbind single sign-on and mapping user home folders between both Windows and Linux desktops.

Before we turn to examining the technologies that are being used to solve today's Linux file-serving problems, let's first take a look at the current file-serving landscape.

Challenges Facing the Linux File-Serving Landscape

Today's Linux-based file servers face similar challenges to their Windows-based counterparts. Among these challenges are:

- Performance
- Scalability
- Availability
- Integration

Let's start with a look at performance.

Performance

As an organization grows, so does its demands on file serving. To accommodate growth, several elements of the file server may need to be evaluated:

- CPU utilization
- Memory usage
- Disk performance
- Network bottlenecks

Any of these issues can seriously degrade system performance. Problems such as CPU or memory usage may be overcome with a simple upgrade. The same may hold true for disk performance. Upgrading to a U320 SCSI or SATA 2.0 hard disks could be a relatively inexpensive solution, depending on the server capacity.

Network bottlenecks are often the result of having a single network access point for a file server. This setup is often the case with traditional standalone file servers as well as Network Attached Storage (NAS) appliances. In these instances, often one of the easiest ways to streamline performance management is to consolidate to a shared data cluster. Shared data clustering not only gives you the ability to balance client traffic across several servers but also can provide an alternative to decommissioning servers that have reached their maximum CPU or memory limit and thus cannot be upgraded further. Later in this chapter, additional time is spent analyzing the benefits of shared data clustering as the baseline for Linux file serving in comparison with the traditional approaches.

> 🕮 For more information about performance tuning and data path optimization, turn back to Chapter 3.

Scalability

In time, scalability issues often result in many of the performance problems that were noted in the last section. As your organization’s data requirements increase, how does your file server respond? In some organizations, scalability problems are not easy to predict. In some instances, server and storage resources are over-allocated due to anticipating too much growth for one division within an organization. On the flip side, if other resources grow beyond your existing forecasts, some servers may quickly reach capacity. Running at capacity could result in several problems, such as hitting a performance bottleneck or running out of available disk space.

To be fully prepared for the pains of scalability, it is important for your file-serving infrastructure to be just as dynamic as the flow of the business processes within your organization.

Availability

With data access being critical to countless business processes, availability of data is also a significant consideration among today's Linux file servers. If a server crashes due to a hardware or software failure, or even from human error, how does the network respond? If the answer is that the administrators are running around scrambling for parts or are troubleshooting software, it means that a particular IT shop is not taking advantage of the many high-availability technologies that are currently available. If a file server is crucial to your organization's day-to-day operations, its data should be resilient to any server-based hardware or software failure.

Integration

Integration is another significant concern among those managing Linux file servers. If your organization is running a Windows domain, ensuring that your Linux file servers and domain controllers can seamlessly work together is also very important to the success of your file-serving infrastructure. Managing permissions and authentication between to two OS platforms in many cases presents challenges for administrators. However, with knowledge of the right tools and integration techniques, the two OSs can play together.

> 📖 Linux-Windows integration is not a topic to be taken lightly; therefore, most of Chapter 6 deals with how to effectively mesh the two environments together.

Another major integration concern with Linux file-server management is that of configuring multiple file servers to coexist in a SAN. Although all major Linux distributions offer fibre channel support, most have limited support in terms of distributed file locking across shared data in a SAN. Another weakness that exists in some of today's Linux file-serving solutions is a lack of reliable multipath support in the SAN. To take advantage of a redundant SAN, predictable multipath support on fibre channel HBAs attached to Linux servers is crucial. When faced with these problems, many Linux shops have turned toward tested and certified solutions offered by third-party hardware and software product vendors.

The previous sections have hit on the major problems that exist in the Linux file-serving landscape; let's look at the methods many organizations are currently using to provide file services to their networks.

Existing Linux File-Serving Solutions

Today, there are predominantly four architectures for offering Linux-based file serving:

- Standalone
- NAS
- DFS
- Clustered

This section will take a look at each of these four approaches.

Standalone

The standalone approach to file serving has stood the test of time and is still well suited for many small businesses. With this approach, a single server provides shared data access to users and applications. This approach is suitable for small organizations that do not live and die by the availability of their file services. If availability is critical, one of the next three architectures would be a better bet.

NAS

NAS has been a very popular architecture for Linux file serving in recent years. As many NAS appliances are easy to deploy, include built-in redundant components, and can offer several terabytes of storage, they have been viewed as an easy choice for many organizations.

As the last chapter mentioned, one of the problems faced by NAS appliances, however, is growth. If an organization outgrows one NAS, they will need to buy another one. NAS appliances run on proprietary hardware, so a NAS cannot be redeployed for other uses if it no longer serves a file-serving need. Another drawback to NAS is sprawl. If an organization deals with file data growth by continuing to add multiple NAS appliances to the LAN, management costs for a network that could grow to host several NAS devices would inevitably go up. One other problem with NAS appliances relates to performance. It is difficult for NAS appliances to be as resilient to high network traffic as other architectures such as shared data clusters.

One final drawback to NAS-based file serving as seen by many organizations is the high cost of a NAS appliance. As nearly all NAS vendors sell products that run on proprietary hardware, cost is another factor that sways organizations toward other Linux-based file serving technologies.

DFS

Like the Windows DFS options discussed in Chapter 4, Linux file servers can also participate in a DFS hierarchy. Linux file servers running DFS via Samba 3.0 can accept connections from any DFS-aware Windows clients, such as Windows 98, Windows 2000 (Win2K), or Windows XP.

With DFS support on Samba, there are two ways to integrate Linux file serving into a DFS hierarchy:

- Create links on a Microsoft DFS root that map to Samba Common Internet File System (CIFS) file shares on a Linux file server
- Configure the Linux file server as the DFS root

Most AD shops that run DFS configure Windows DFS controllers as DFS roots and simply create DFS links to any CIFS file shares on Linux Samba servers. This approach allows organizations to take advantage of some of the Windows DFS features that have not yet made it into Samba, such as DFS root replicas and AD site-awareness.

If your preference is to run your entire file-serving infrastructure on Linux, you may opt to configure the DFS root on a Linux box, then point each DFS link to other Linux file servers. DFS is unique in file-serving architectures in that it does not have to represent an absolute choice. Instead, DFS can complement other file-serving approaches such as standalone, NAS, or clustered. The ability to deliver transparent access to file shares could free up administrators to migrate file shares to other servers without having to impact users. Instead, all that would need to be updated would be the DFS link that exists at the DFS root so that it references the new shared folder location.

> 🕮 Basic Samba setup is covered later in this chapter.

Clustered

Another major approach to Linux file serving is to implement a clustered file server. For Linux file serving, two open source cluster solutions currently exist:

- Linux-HA failover clustering
- LVS load-balanced clustering

These solutions are described in the next two sections.

Failover Clustering

Open source failover clustering on Linux is provided by the High-Availability Linux Project (http://www.linux-ha.org). Linux-HA clusters can be configured on nearly any Linux distribution. The key to the operation of Linux-HA clusters is Heartbeat. Heartbeat is the monitoring service that will allow one node to monitor the state of another and assume control of the cluster's virtual IP address if the primary node fails. Heartbeat also provides the ability to automate the startup of services on the standby node.

In a traditional heartbeat scenario, two virtually identical servers are configured with one acting as the primary server and the second as the standby server. Both servers are kept in sync by replication from the primary server to the standby server, and the standby server will routinely send a "heartbeat" signal to the primary server, which, if it is up and running will respond. If the primary server fails and the heartbeat signal goes unanswered, the standby server will assume the role of the primary server.

Many Linux vendors have jumped on the Heartbeat bandwagon. One such vendor is SUSE, which includes the Heartbeat setup packages on its SUSE Linux Enterprise Server setup CD. For distributions such as Red Hat Enterprise Linux, you can download Heartbeat and all necessary dependant packages from http://www.ultramonkey.org. Ultra Monkey provides the software and documentation of Linux-HA on Red Hat distributions. Figure 5.1 shows a simple Linux-HA failover cluster.

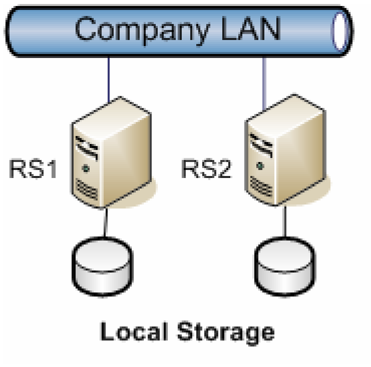


Figure 5.1: A 2-node Linux-HA failover cluster.

Note that in Figure 5.1, each node is maintaining its own copy of local storage. For file serving, this setup can prove to be very challenging. In order for each cluster node to present a consistent view of file system data, the local storage on each node will need to be continually synchronized. To maintain consistency across the local storage in the cluster, many organizations turn to `rsync`. With `rsync`, you can configure incremental block-level replication to run between each node in the failover cluster. Doing so will ensure that the second node in the cluster (RS2) will have up-to-date data in the event of a failover of the first node (RS1).

Of course, this functionality comes with a few significant drawbacks. For the sake of supporting failover, you would need to double your storage investment. For clusters consisting of more than two nodes, this investment would be proportionally higher. As you can imagine, this presents significant problems when facing storage growth. If the servers are configured to replicate every 15 minutes, for example, then the standby server may come online at a disadvantage. To achieve true high-availability failover, it's best to implement a shared storage environment so that when the standby server is called into action, it has access to the file system at the point where the primary server left off—without any replication delay.

> For more information about configuring incremental file replication using rsync, visit http://rsync.samba.org.

Load-Balanced Clustering

Most Linux load-balanced clusters are based on the Linux Virtual Server (LVS) Project. Compared with the Microsoft network load-balanced cluster architecture, you will see that LVS uses a fundamentally different approach. With LVS, one or two servers outside of the cluster are used to distribute client traffic among cluster members. Thus, to build a 3-node LVS cluster, you'll need at least four servers. Figure 5.2 illustrates this configuration.

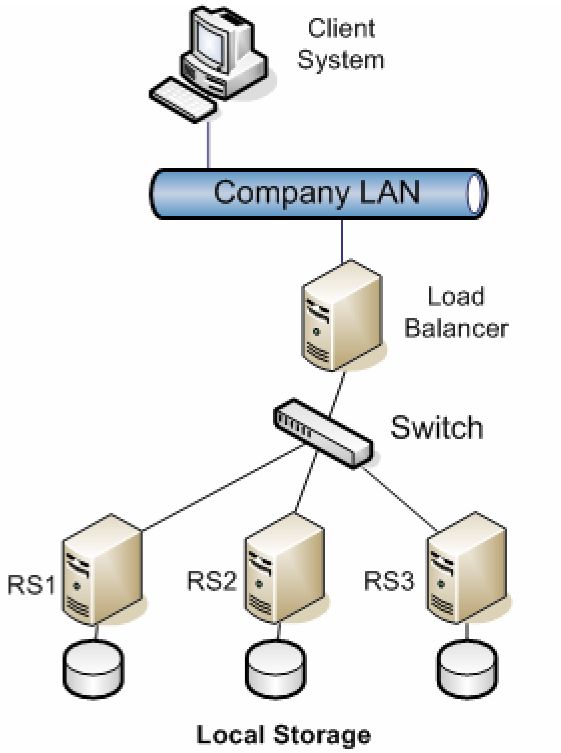


Figure 5.2: A 3-node load-balanced cluster.

In Figure 5.2, the server labeled as Load Balancer accepts incoming client requests and directs them to an internal Real Server (RS). Each RS is a cluster node. With the load balancer directing client traffic, the RS nodes in the cluster can be located anywhere that has TCP/IP connectivity to the load balancer. Thus, each RS does not have to be on the same LAN segment. As the load balancer is the director for all client requests, having one server as the load balancer does have one fundamental flaw—fault tolerance. If the load balancer fails, the entire cluster is brought down. To avoid this problem, most LVS cluster implementations use two systems as load balancers. One system serves as the active load balancer, and the second system is passive, only coming online in the event that the active system fails. Figure 5.3 shows a fault-tolerant LVS cluster.

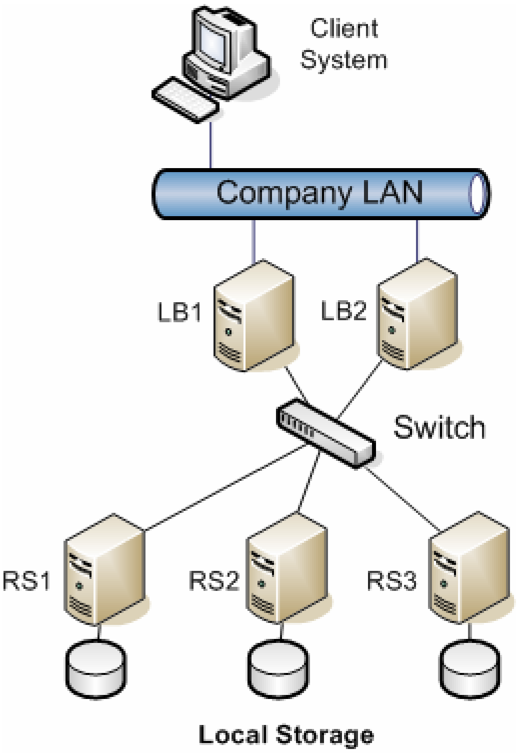


Figure 5.3: A 3-node fault-tolerant load-balanced cluster.

As with the failover cluster, the LVS load-balanced cluster by default allows for each real server to maintain independent local storage. This setup again means that to maintain consistency across the cluster, a replication tool such `rsync` will need to be employed.

Now that you have seen the basic operation of an LVS cluster, you may be wondering whether the load balancer acts as a bottleneck for client access. The answer lies completely in LVS cluster architecture that is applied to the cluster.

LVS Architecture

LVS is generally configured in one of three ways:

- LVS via Network Address Translation (NAT)
- LVS via IP tunneling
- LVS via direct routing

In the next three sections, we'll look at each of these unique configurations.

LVS via NAT

With the LVS via NAT architecture, the load balancer server is dual-homed and NATs all traffic to the real servers on an internal LAN. Figure 5.2 and 5.3 show this configuration. With NAT, each load balancer server directs client traffic into the internal LAN and to a real server. When the real server replies, the reply goes back through the load balancer system before returning to the requesting client.

This approach can present both a performance bottleneck as well as scalability limits. Most LVS cluster implementations cannot scale beyond 10 to 20 nodes and still see any gains in performance.

LVS via IP Tunneling

Several advantages exist with the LVS via IP tunneling, most notably scalability. Unlike configuring LVS via NAT, the IP tunneling approach causes the load balancer server to direct client requests to the real servers via a Virtual Private Network (VPN) tunnel. Replies from the real servers will use a different network. This approach does not have the scalability limitations of LVS via NAT.

With use of VPN tunneling, this cluster can easily be distributed among multiple sites and connected via the Internet. However, this approach is usually best suited for load balancing between FTP servers and is rarely applied as a high-performance file-serving solution.

LVS via Direct Routing

The LVS via direct routing approach is similar to LVS via NAT, except that reply traffic will not flow back through the load balancer; instead, replies will be sent directly from the real servers to the requesting client. As with LVS via NAT, real servers connect to the load balancer via the LAN. Replies from the real servers would return to the client over a different LAN segment that is routable to the requesting client.

Unlike the LVS via IP tunneling approach, this method is more sensible for LAN-based file serving. However, it is still far from the best solution for enterprise file serving. The currently available commercial solutions are far superior to their open source counterparts.

Although open source clustering technologies have emerged as methods for increasing the availability and performance of file servers, many organizations are wary of open source technologies due to a lack of support. If a failure occurs, help may be days (instead of minutes) away.

Commercial File-Serving Solutions

There are several commercial file-serving solutions in the Linux space, including:

- PolyServe NAS Cluster
- VERITAS (now part of Symantec) Cluster Server
- Red Hat Linux Cluster Suite and GFS

In the next three sections, each of these enterprise file-serving solutions will be looked at in closer detail.

PolyServe NAS Cluster

PolyServe NAS Cluster provides all the benefits of NAS (consolidation, ease of management, high availability) as well as all the advantages of both Linux-HA and LVS clustering. PolyServe NAS Clusters offer failover support for file-serving applications and offer true shared data clustering. In a PolyServe Matrix Server cluster, each node in the cluster shares a common storage pool in a SAN. Thus, with all cluster shares being in a common location, there is no need to replicate file server data between nodes. In comparison with the 3-node Linux-HA cluster shown earlier in Figure 5.1, migrating to a PolyServe NAS Cluster platform will allow you to immediately triple the amount of storage available for the cluster. Assuming that a Linux-HA failover cluster had 500GB of local storage attached to each node, the cluster would have 1500GB of total storage, with only 500GB that is truly writable. The reason is that the shared cluster storage in each node must mirror the storage of the other nodes in the cluster. If the same storage resources were applied to a PolyServe NAS Cluster, all 1500GB of storage would be writable. Figure 5.4 provides a comparison between a PolyServe Matrix Server cluster and a Linux-HA cluster.

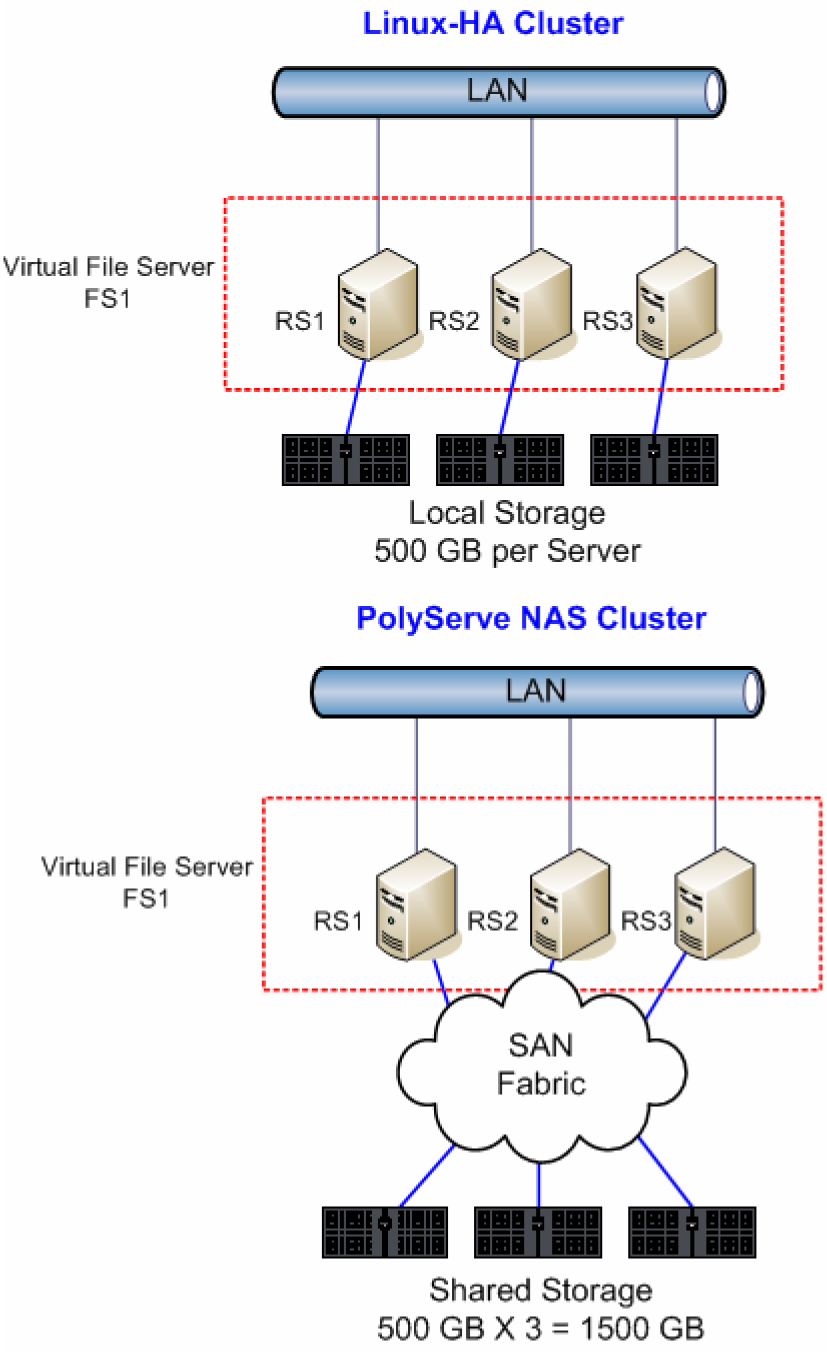


Figure 5.4: PolyServe NAS Cluster vs. Linux-HA cluster.

The fact that multiple nodes in a PolyServe NAS Cluster can simultaneously access shared files provides for high-performance load balancing as well as failover support. Thus, with this architecture, you can get the benefits of open source clustering products as well as a maximum return on your storage investment.

Aside from PolyServe's better approach to clustering, it also has advantages over traditional NAS vendors such as Network Appliance and EMC. Unlike NetApp and EMC, PolyServe's NAS Cluster can be deployed on industry-standard Intel or AMD platforms running Linux. Unlike NAS, the answer to performance bottlenecks is not through a separate NAS; instead, you can simply add another node to the cluster.

> 🕮 For more information about the PolyServe NAS Cluster solution, download the white paper *UNIX to Linux Migration* at http://www.polyserve.com.

VERITAS Cluster

Similar to the Windows clustering solution described in Chapter 4, VERITAS offers a comparable clustering solution for Linux. Although this product offers failover support, it does not provide the load balancing support or shared data capability that is found in PolyServe NAS Clusters.

VERITAS does make up for its lack of load-balancing support by offering other features such as an intelligent agent that can dynamically move a virtual server in the cluster to an underutilized node, which is similar to a solution from PolyServe, which also supports the movement of virtual IP addresses. VERITAS clusters can scale to 32 nodes, giving you plenty of room for growth potential. If performance and availability are primary concerns, the VERITAS cluster solution has trouble delivering in performance-demanding environments. This shortcoming is essentially due to the fact that VERITAS Linux clusters can only provide failover support and does not allow multiple nodes in the cluster simultaneous access to the same file.

Red Hat Cluster Suite and Global File System

Red Hat offers its own commercial clustering product, which is the company's adaptation of the Linux-HA project. Unlike Linux-HA, which is available for free via download and with SUSE Linux, Red Hat's Cluster Suite must be purchased as a separate add-on to the Red Hat Enterprise Advanced Server OS. The Red Hat Cluster Suite provides support for as many as 8-node failover clusters.

The Red Hat Cluster Suite supports shared storage via SCSI or fibre channel, a management UI to simplify configuration, and a shared cluster quorum. In a significant diversion from many traditional Linux server-management practices, Red Hat only supports management of its Cluster Suite using its Cluster Manager GUI tool. If you want to change cluster configuration files via a text editor, you're on your own! The Red Hat Cluster Suite also supports Global File System (GFS), which provides for better integration with storage networks. GFS supports simultaneously reads and writes to a single shared file system in a SAN. This feature allows clusters configured in the Red Hat Cluster Suite to offer both failover and load-balancing support, similar to the PolyServe NAS Cluster.

Deploying Performance-Based Scalable Linux File-Serving Solutions

Now that you are aware of the available alternatives, let's take a look at some considerations for deploying Linux file-serving solutions.

Pre-Deployment Considerations

The tendency of IT administrators is often to deploy first and customize later. For those that practice this approach, planned customizations take months or even years to complete. After all, with the file server deployed and operational, justifying spending additional time on the file server may be difficult, especially if you're like many IT folks and have countless other tasks on your list.

To deploy a file server right the first time, planning has to be an important part of the process. One major part of the planning process is deciding which technologies should be used to complement the file server. Table 5.1 lists the most common file-serving problems as well as the available technologies that can alleviate or manage the potential problems.

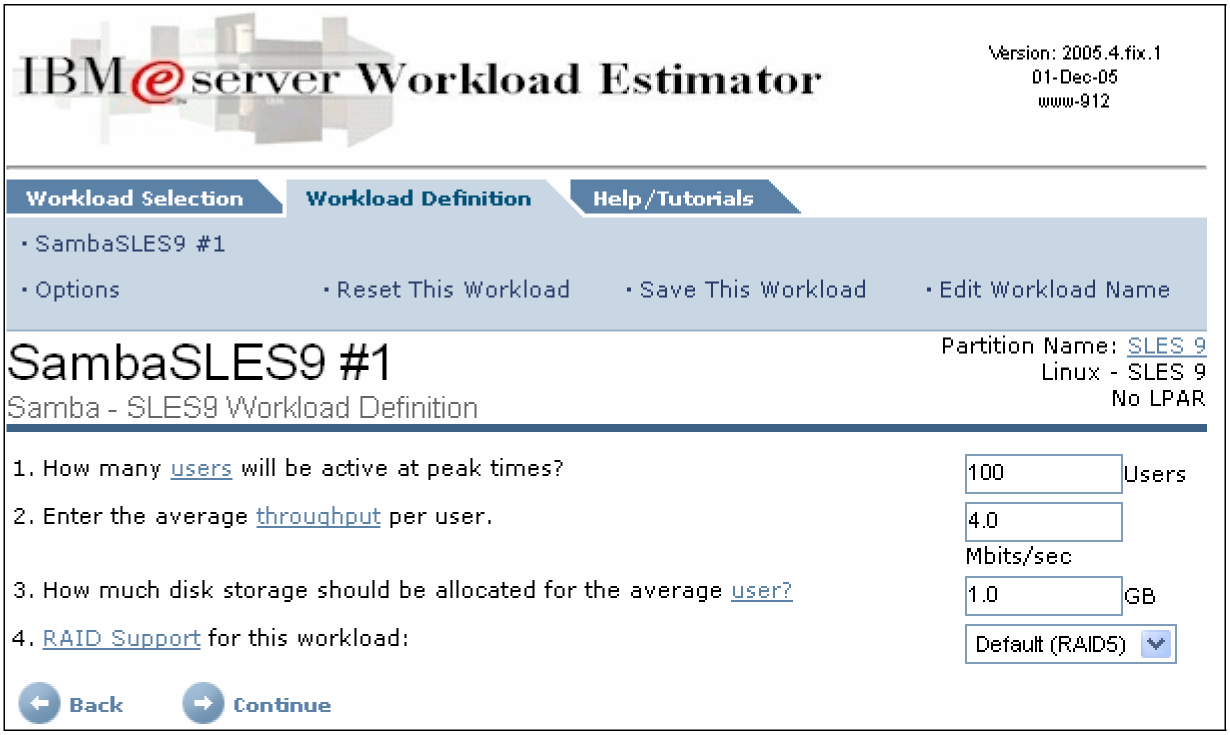
File-Serving Problem	Solution
Limit user usage of file-server resources	Deploy and configure disk quotas
Provide failover support	Deploy and configure a Linux-HA cluster
Provide failover and load-balancing support	Deploy a third-party product
Provide antivirus protection	Deploy an antivirus solution that is compatible with any installed file-serving applications as well as your backup product
Prevent unauthorized access	Determine the necessary permissions for each user or group that has access to the server

Table 5.1: Solutions for the most common file-serving deployment problems.

With some of the general requirements under your belt, let's look at the process of sizing up both server and storage requirements.

Server Sizing

One of the most difficult aspects of deploying any server is the process of determining the server's hardware requirements. This task can be difficult and the result is often an educated guess based on past experience. To help administrators in their quest to build servers that are perfect for their needs, many hardware vendors offer online sizing tools. To aid in the reliability of the tools, sizing tools are typically organized by server purpose, such as file server, and OS. One such tool is the IBM eServer Workload Estimator, which is available at http://www-912.ibm.com/wle/EstimatorServlet. Figure 5.5 shows this tool.



SambaSLES9 #1

Samba - SLES9 Workload Definition

Partition Name: SLES 9
Linux - SLES 9
No LPAR

1. How many users will be active at peak times? 100 Users
2. Enter the average throughput per user. 4.0 Mbits/sec
3. How much disk storage should be allocated for the average user? 1.0 GB
4. RAID Support for this workload: Default (RAID5)

Figure 5.5: The IBM eServer Workload Estimator tool.

In the example in Figure 5.5, the Workload Estimator is being used to size a Samba server running on SUSE Linux Enterprise Server 9. Note that the tool allows to you size server hardware requirements based on factors such as concurrent user sessions, average user throughput, and average storage allocation per user.

Once you provide the estimator with the necessary information (or accept the default settings), the tool will recommend server hardware that will meet your performance requirements. For SUSE Linux Enterprise Server 9 Samba servers, IBM offered the general guidelines that Table 5.2 shows.

Environment	Recommended CPU	Recommended RAM	Server Platform
Large (400 concurrent users)	1.9GHZ 4-core	4GB	P5 550 Express
Medium (200 concurrent users)	1.9GHZ 2-core	2GB	P5 520 Express
Small (85 concurrent users)	1.65GHZ 1-core	2GB	P5 505 Express

Table 5.2: IBM Linux file-server sizing recommendations.

☞ If your preferred server vendor does not offer an online tool to assist in Linux file-server sizing, you can probably pass along your requirements to your local vendor representative. The local rep should be able to use an internal tool or consult an engineer to arrive at the proper server sizing requirements for your environment. As each server application and server uses system resources slightly differently, there is no one-size-fits-all tool for server resource sizing.

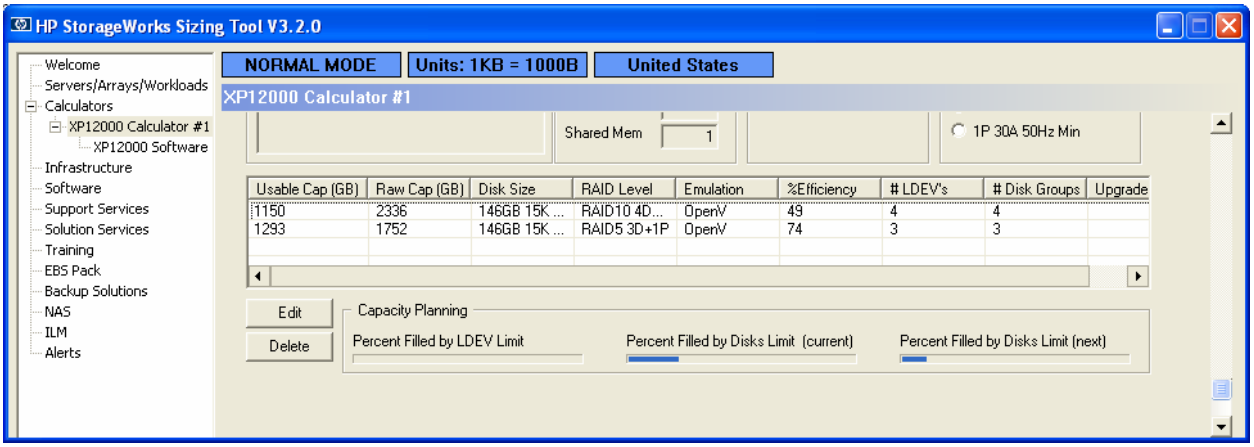
Storage Sizing

Storage sizing starts with allocating adequate internal disks for the OS, applications, log files, and the paging file. For file-server deployments, a best practice is to allocate 1.5 times the amount of physical RAM to the paging file. Thus, a file server with 4GB of RAM should have a 6GB paging file. For optimal performance, the paging file should be stored on a separate disk, which clears an I/O channel for just paging operations. For log file storage sizing, you should consult with the application vendors for each application running on the server.

Once you have determined the storage requirements for the OS, paging file, and applications, you can then move on to the storage requirements for the data itself. This value is often predictable because you should have on hand information about the current file server capacity as well as some historical data showing capacity over the past 12 to 18 months. For file server data sizing, a good practice is to requisition ample storage to meet the expected data growth for the next 18 to 24 months. When unsure about past storage growth, backup logs can usually provide the information you need. A simple method is to simply view the statistics for monthly full backups over the past year. This task should allow you to gauge the percentage of storage growth over the next 1.5 to 2 years.

Once you have a handle on how much storage you need, you can work with your preferred storage vendor to decide the type and size of disk drives that you'll need to purchase. As with server sizing, most storage vendors offer sizing tools that can assist you in determining the storage devices that will meet your disk storage requirements.

One such tool that can help in identifying the hardware components that could support your storage requirements is the HP StorageWorks Sizing tool, which is available at http://h30144.www3.hp.com/. With this tool, you can enter your planned capacity and RAID level and the tool will generate information about the hard disks to use to meet your requirements as well as the overall storage efficiency of your planned storage system. Being able to view efficiency is very helpful when comparing different RAID levels. Figure 5.6 shows a portion of the HP StorageWorks Sizing tool output.



Shared Mem 1

○ 1P 30A 50Hz Min

Usable Cap (GB)	Raw Cap (GB)	Disk Size	RAID Level	Emulation	%Efficiency	# LDEV's	# Disk Groups	Upgrade
1150	2336	146GB 15K ...	RAID10 4D...	OpenV	49	4	4	
1293	1752	146GB 15K ...	RAID5 3D+1P	OpenV	74	3	3	

Figure 5.6: Comparing RAID level efficiency using the HP StorageWorks Sizing Tool.

In the example that Figure 5.6 shows, a 1TB RAID 5 was compared with a 1TB RAID 10. The tool shows that the RAID 10 configuration would be 49 percent efficient, while the RAID 5 would be 74 percent efficient. The tool also allows you to see information about the disk size to be used as well as the total amount of storage to be purchased. For example, the 1TB RAID 5 would incorporate a total of twelve 146GB hard disks, total capacity of 1752GB. The usable capacity would be 1293GB. Once you have your Linux file-serving hardware sized, you are ready for deployment and management of the essential Linux file-serving services.

Managing Enterprise-Class Linux File Serving

Regardless of whether you have a standalone, NAS or clustered file server, the protocols that enable file sharing on Linux file servers are the same. This section will look at the roles of the following protocols and services as they pertain to Linux file serving:

- Network File System (NFS)
- Samba

Let's start with a look at NFS.

NFS

NFS has long been the de facto file-sharing protocol on UNIX and Linux servers. NFS has stood the test of time because it provides a simple and efficient means for sharing data between systems on a network. NFS has continued to evolve and get better with age as was proven by the recent improvements that were introduced in NFS v4.

What Is New in NFS v4?

NFS v4 is currently supported on both the SUSE 10 and Red Hat Enterprise Linux 4 distributions. NFS v4 offers several new features that both significantly improve the performance and security of NFS. The following list highlights some of the most significant improvements brought about by NFS v4:

- Improved security—Supports Kerberos v5 and Simple Public Key Mechanism 3 (SPKM3)
- Better ACL management—Supports named attributes; user and group information is stored in strings instead of numeric values
- Better firewall compatibility—The disparate NFS protocols (ACL, mount, NFS, NLM, and stat) are now combined into a single protocol specification
- File delegation—NFS clients can now modify files stored locally in their own cache without having to send requests back to the NFS server; this feature provides for a significant network performance improvement
- Lease-based file locking—NFS v4 clients lock files based on a share reservation; if an NFS v4 client loses contact with a server, once its lease on a locked file expires, that file is free to be accessed by other users
- Supports file migration and replication—File migration and replication are now supported via NFS

With a general overview of NFS under your belt, let's examine the steps for getting this service up and running.

NFS Setup Checklist

Setting up NFS is a relatively straightforward process. Let's start by looking at the general steps for configuring and enabling NFS on a Linux file server:

- Define the folders to publish as shares in the `/etc/exports` file.
- Set local permissions for each shared folder as necessary.
- Define the hosts and logical networks that are allowed access to the NFS service by editing the /etc/hosts.allow and /etc/hosts.deny files.
- Start the NFS service.
- Mount a shared folder from an NFS client.

Here's an example of /etc/exports configured to share a folder named /public:

```
/public/   *(ro,root_squash,sync)
```

Once the shares are defined in the /etc/exports file, you then need to ensure that you have the proper local permissions set for each exported folder. This step is necessary to prevent against unauthorized access, modification, or deletion of shared files.

Network access can be restricted on a host-by-host or network-by-network basis by editing /etc/hosts.allow and /etc/hosts.deny. When a connection is attempted to a Linux file server, the connecting host's IP address is first evaluated in the etc/hosts.allow file. If no match exists, the /etc/hosts.deny file is checked. If a match exists, the host is denied access. By default, if no match exists, the host is allowed access. If you want to deny all traffic from any hosts or services not explicitly listed in the /etc/hosts.allow file, you would add the following line to the /etc/hosts.deny file:

```
ALL:ALL.
```

Although denying all traffic not explicitly granted access is the most secure method of locking down a file server, you will need to remember this setting in the event that you are setting up additional network services or applications on the file server. If the new service or application is not allowed in /etc/hosts.allow, clients will not be able to connect to the service.

Once you have created the implicit deny rule in /etc/hosts.deny, you would then need to edit /etc/hosts.allow to grant access to the appropriate hosts or network segments. The following example shows how to configure /etc/hosts.allow to grant NFS access to hosts on the 172.16.1.0/24 subnet:

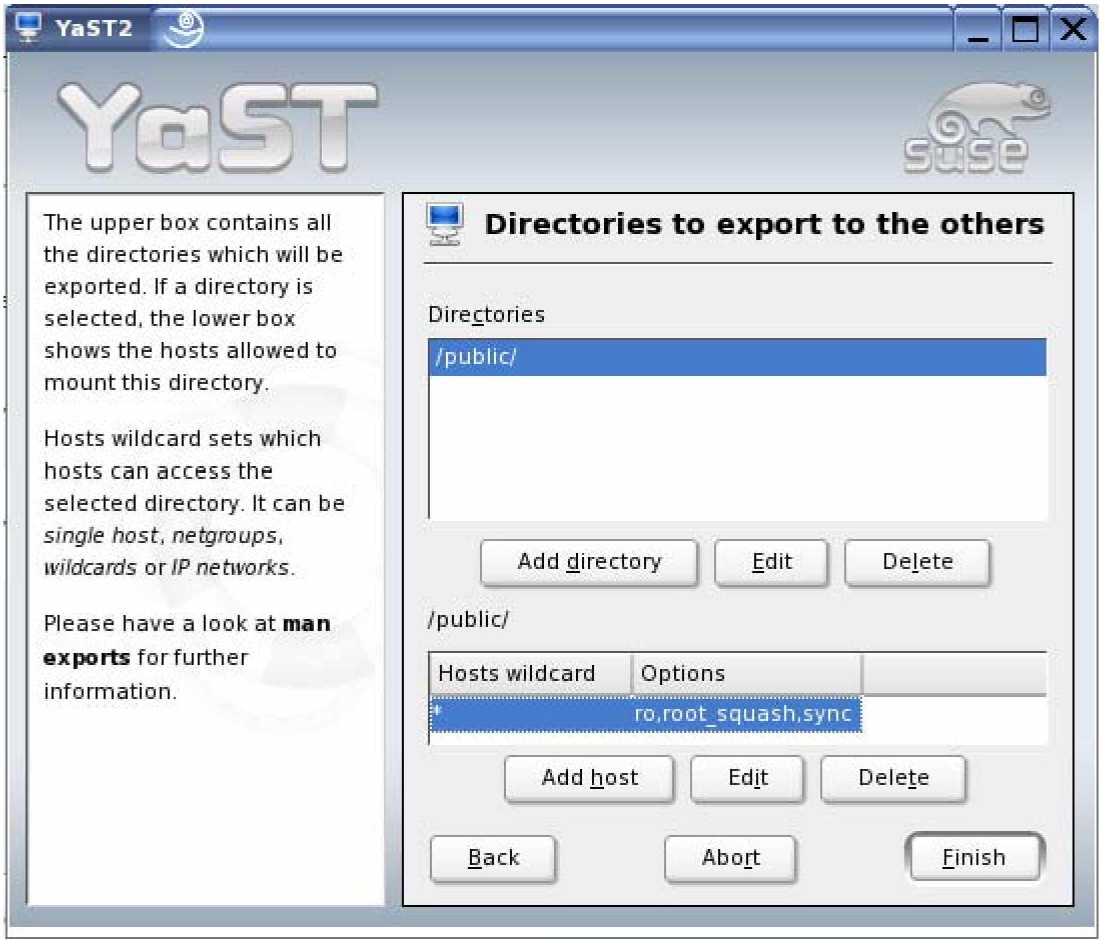
```
lockd: 172.16.1.

rquotad: 172.16.1.

mountd: 172.16.1.

statd: 172.16.1.
```

At this point, you can start the NFS service and you are on your way. Linux distributions are continually improving their GUI management tools, and such is particularly the case with SUSE Linux 9. NFS can be fully configured within minutes by using SUSE Linux's YaST, as Figure 5.7 shows.



The upper box contains all the directories which will be exported. If a directory is selected, the lower box shows the hosts allowed to mount this directory.

Hosts wildcard sets which hosts can access the selected directory. It can be *single host*, *netgroups*, *wildcards* or *IP networks*.

Please have a look at **man exports** for further information.

Directories to export to the others

Directories

/public/

/public/

Hosts wildcard	Options
*	ro,root_squash,sync

Figure 5.7: Configuring NFS using YaST.

Now that you have seen how to complete the initial setup of NFS, let's take a quick look at Samba.

📖 For more information about NFS, point your Web browser to http://nfs.sourceforge.net.

Samba

Samba provides the functionality for Linux file servers to host shared folders that are accessible via the CIFS protocol, which is the default file-sharing protocol for all Windows-based OSs. Both Red Hat Enterprise 4 and SUSE Linux 9 run Samba 3. With Samba 3, major improvements were made that allowed for reliable authentication between Windows AD domain controllers and Samba servers. Although the reliability improvements are significant, Samba's feature set is closer to that of a Windows NT Primary Domain Controller (PDC) than that of a Win2K or Windows Server 2003 (WS2K3) AD domain controller. This limitation of Samba is expected to change in Samba v4.

What Is Coming in Samba 4.0?

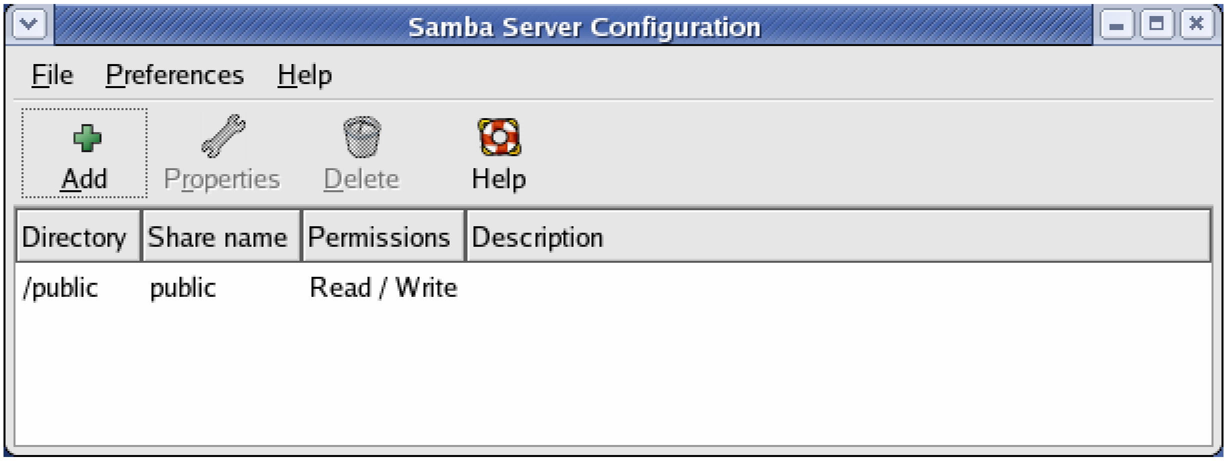
The upcoming release of Samba 4.0 is being hailed as Samba's first true challenge to AD. Among the planned features for Samba 4.0 are:

- Support for AD logon and administration protocols
- An internal Lightweight Directory Access Protocol (LDAP) server
- Internal Kerberos server
- Flexible (extensible) database architecture
- Full NTFS semantics
- Much better scalability

Many administrators have reveled in Samba 3 for its ability to provide highly available CIFS file serving. With so many planned enhancements in Samba 4, its pending arrival has garnered significant buzz in the industry.

Samba Deployment

Samba deployment is similar in approach to NFS deployment, with the exception that additional attention will need to be paid to Windows authentication—considering Samba file servers most often are used to provide access to Windows client systems. As with NFS, Samba can be configured using YaST on SUSE Linux or with the Samba Server Configuration tool (see Figure 5.8) on Red Hat Linux.



Directory	Share name	Permissions	Description
/public	public	Read / Write	

Figure 5.8: Red Hat Linux Samba server configuration.

The Samba server configuration is stored in the `/etc/samba/smb.conf` file. The following example shows the smb.conf file settings that match the /public share definition shown in the Samba Server Configuration tool earlier:

```
[public]
   comment = Company Docs
   path = /public
   writeable = yes
```

This code creates a writable share named "public." In addition to defining the shares and level of share access, you need to set permissions for the shared files and folders. In the next chapter, you will see how to set permissions on a Linux Samba server for Windows user accounts residing in an AD domain. As so many Samba issues in the enterprise are directly related to Windows, the bulk of the information on fully deploying Samba is provided in Chapter 6.

> 📖 For additional information about Samba setup and configuration, read the Official Samba HOWTO at http://www.samba.org.

Current Trends in Linux File Serving

Linux file servers have gained from three major trends in the IT industry:

- Migration from UNIX to Linux
- Server consolidation
- Storage consolidation

This section will look at the impact of these three trends on the Linux file-serving landscape.

Migration from UNIX to Linux

When Linux first burst onto the IT scene, many thought that it would be a serious challenger to Windows. Although the question of whether Linux will ever be able to overtake Windows still remains to be decided, UNIX OSs have suffered substantially at the hands of Linux.

To most, moving from UNIX to Linux is a no-brainer. Many of the enterprise applications that run on UNIX, such as Oracle, also run on Linux. As Linux OSs can run on industry-standard Intel-based hardware platforms, Linux servers are far less expensive than UNIX servers that run on proprietary hardware. Being proprietary can also mean that an organization will need to pay more to maintain a UNIX server. This cost is not only related to the proprietary hardware in the server but also the higher cost to pay an administrator that has the specialized skills to maintain the UNIX server. With Linux file-serving solutions being able to offer comparable performance to UNIX servers at a fraction of the price, migrating legacy UNIX boxes to Linux is a logical step.

Benefits of Consolidation

Another logical step that many have taken in file-server management is toward consolidation. Both proprietary UNIX servers and NAS appliances have been major contributors to server sprawl. For organizations that have anywhere from two to five NAS appliances, management overhead is becoming more difficult as the network expands. As with UNIX migration, the detriments to server sprawl are easy to spot and have led to a flood of organizations consolidating dozens of UNIX servers and NAS appliances to Linux clusters.

The bottom line with consolidation is that it can result in significant yearly savings. Take for example a consolidation project that reduces 60 servers to two 15-node PolyServe Matrix clusters. In this case, the TCO savings could easily reach several hundred thousand dollars a year. Fewer servers can also mean fewer software updates. With less to maintain, IT shops can stretch their budgets further.

Consolidation is not about getting smaller for the sake of getting smaller but is instead about getting the most out of your existing hardware investments. Having several servers with 30 percent CPU utilization, for example, means that you have several servers that have CPUs doing nothing 70 percent of the time. If your organization has paid for the hardware, it should very well get the most out of it.

Again, since consolidation is about reducing hardware costs and system management, it is important to keep in mind that file server consolidation is best suited for shared data clustering. Clustering provides the ability to configure failover support and load-balanced data access for critical file servers. Other approaches to consolidation, such as those that consolidate file servers to virtual machines, only reduce the amount of managed hardware on the network. They do not reduce the number of managed systems on the network and thus will not help with reducing software licensing costs. Thus, although there are several ways to go about file-server consolidation, consolidating to a shared data cluster that can offer the benefit of load balancing, failover, and streamlined management from a single console has been deemed the most logical methodology by many organizations in the IT community.

Storage Consolidation

Most organizations also consolidate their storage resources while in the process of consolidating file-server resources. Storage consolidation offers several benefits:

- More efficient utilization of purchased storage resources
- Simpler storage scalability
- Ability to back up and protect data using methods that are not available to traditional DAS storage
- Ability to share data between redundant servers instead of having each server maintain its own local copy of data files

When combined with server consolidation to a shared data cluster, sharing disk resources between servers in a SAN also allows for true load balancing of data access to storage. With consolidated storage, when a need arises for additional disk resources, the disks can simply be added to the SAN and then mapped to the server that needs them. This method is more efficient for managing storage than the traditional process of "marrying" a disk array to a single server.

If you allow it, the complexity of networks will only continue to grow over time. Warding off network complexity requires you to be proactive. The instinct to growth is always to buy more parts. More parts only further add to complexity and in turn more management costs. Streamlining your network with consolidated server and storage resources will ultimately lead to better TCO. When combined with shared data clustering, consolidation will also result in vastly improved reliability and performance.

Summary

In this chapter, you were presented with the state of the Linux file-serving world as well as best practices for optimizing Linux file serving in production. The final chapter will look at the management issues surrounding heterogeneous networks. In particular, you'll see how to configure winbind authentication on your Linux file servers for the sake of supporting user authentication to a Linux server via AD. For environments that are running both Windows and Linux desktops, you will also see how to set up user home folders to be shared across both Windows and Linux workstations.

After tackling the most challenging Windows-Linux integration issues, the chapter will then examine modern backup methodologies that are used to maintain data availability and disaster protection for both Windows and Linux file servers.

Download Additional eBooks from Realtime Nexus!