

1**1: FAN Basics**

This chapter covers some of the basics upon which the remainder of the book is built. This includes a discussion of what File Area Networks (FANs) are, why they are beneficial, and some of the protocols and products that can be used to create the infrastructure upon which to build a FAN. Since most readers are broadly familiar with these concepts, this chapter merely provides an overview.

File Area Networks

A large and growing percentage of corporate data takes the form of files. This includes unstructured data organized within a filesystem. File data is generally distinguished from “raw” block data which might be used for a back-end database on a business system, or enterprise email server.

All file data is ultimately stored as block data on the other side of a filesystem, whether it is located on a PC, a server, or a NAS appliance. All files are blocks, though not all blocks are files. The key differentiator is that file data is *accessed* as a file by the end user – such as word processor documents, or slide presentations. Block data is accessed as raw blocks, generally by an application such as a database, and not by an end user.

Given the growth in the number of files that IT departments need to manage, the increasing complexity of filesystems, and the large-scale applications that use

files, it is clear why file management has gained prominence. This is especially true when implementing application-level disaster recovery and Information Lifecycle Management (ILM). Such initiatives have created the need for a new type of file management: The File Area Network.

The FAN concept requires an increasingly sophisticated suite of file management technologies, including file-level descriptions and classification to attach policies to file data. To address this need, the industry is developing a wide range of products and services to help streamline the management of file-based data as part of an enterprise FAN. This book will focus on discussing the FAN products offered by Brocade.

The “network” portion of a FAN is the pre-existing corporate IP network. In addition, a FAN will make use of one or more upper layer network filesystem protocols such as the Network Filesystem (NFS) and the Common Internet Filesystem (CIFS). The FAN is, however, distinguished from the underlying network which transports it: “FAN” is a logical way to describe the hardware and software technologies used to organize, route, switch, and provide consistent access to massive amounts of file data. This is similar to other layered network models. For example, storage networking traffic may traverse Fibre Channel fabrics, DWDM MANs, and IP WANs, yet the SAN is still the SAN even when it sits on top of something else.

The goal of a FAN is to provide a more flexible and intelligent set of methods and tools to move and manage file data in the most cost-effective and controlled manner. To accomplish this, FANs provide several key functions:

- Enterprise-wide control of file information, including the management of file attributes

- The ability to establish file visibility and access rights regardless of physical device or location
- Non-disruptive, transparent movement of file data across platforms and/or geographical boundaries
- The consolidation of redundant file resources and management tasks
- The ability to support file data management in both datacenters and branch offices

At a high level, a FAN can be viewed as in Figure 1.

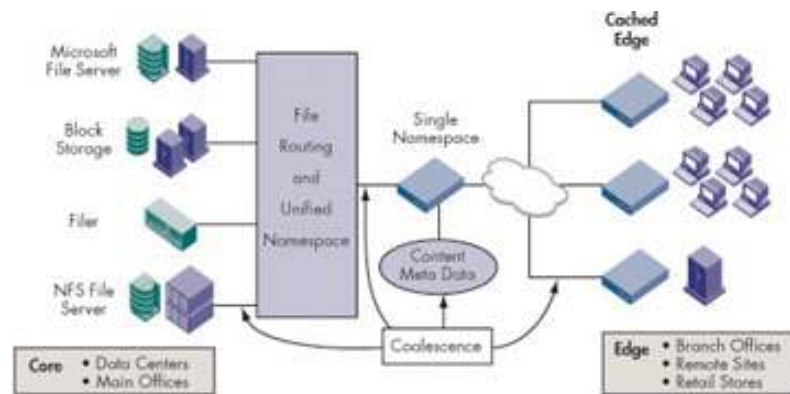


Figure 1 - High Level FAN Diagram

The key to this architecture is that the FAN provides “coalescence” between files stored in different locations and the file consumers (clients). In current datacenters, separate storage devices are truly separate, which is why administrators typically spend so much time mapping drive letters and handling complex change control processes during migrations. The “coalescence” principal means that a FAN will group separate components into one united file space. There are a number of objectives administrators have which are facilitated by this, such as:

- Make file location and movement transparent
- Centralize file storage and management for efficiency
- Reduce the cost of remote data backup
- Intelligently migrate files based on policies

- Consolidate branch office IT infrastructure
- Comply with regulations and corporate objectives

Many products and services operate in the FAN to allow these objectives to be met, such as a namespace unifier, file routing engine(s), meta data management, and remote office performance optimizers. Most of this book is dedicated to discussing what those products and services are, how they work, and how best to deploy them.

FAN Drivers

Before discussing how FAN technology works, it is useful to understand why FAN technology is needed. As the amount of file data has grown exponentially over the past few years, a number of factors have come together to create the need for file networking. These include:

Storage Management – Where does data reside? How does it get moved? How do people find it after it gets moved? With data spread across potentially hundreds of locations in an enterprise, how do IT departments manage drive letter mappings? These and other issues add to the complexity of managing a storage environment.

Administrators need to decide how to automate and use policies to manage storage infrastructure, and – ideally – find a way to manage DAS, NAS, SAN, *and* the files stored on those devices from a single location.

Storage Consolidation – It is desirable to consolidate many scattered storage resources into fewer centralized locations. Indeed, shared storage will have an increasingly important role in driving next-generation efficiencies across the enterprise. It simplifies management, and optimizes white space – the portion of any given disk which is not used for storing data. A very large portion of the data being consolidated consists of files, so an optimal consolidation solution must be intelligent at the file level.

Business Continuity / Disaster Recovery – More and more organizations are preparing for disasters. In many cases, this is driven by laws and regulations, in other cases by fiduciary duty to investors. One goal of solutions in this category is minimizing client downtime during an outage, but in all cases it is necessary to ensure the availability of mission critical data. Given that much if not most of the data being protected consists of files, this is a natural fit for FAN technology.

Storage Performance Optimization – When too many users access a single device, performance degradation inevitably results. To solve the problem, it is desirable to load balance across multiple storage devices. However, this can be tricky and time consuming to accomplish without automation, and doing it at all requires knowledge of file access patterns.

Data Lifecycle Management – DLM requires a method for creating storage tiers and aligning archival policies with regulatory compliance requirements. Additional issues include how to streamline the backup process and how to optimize the use of high end storage subsystems. Remember: most of the data being managed consists of files, so most of the knowledge about which bits of data need to “live” on what storage subsystems can only be obtained by looking at file-level properties. Merely looking at data blocks on a raw device will not help.

Remote Site Support – Managing remote site primary storage and backup can be a full time role. Traditional methods of centralizing management of highly distributed data can be equally problematic, for example in terms of performance and availability. File-level tools are needed to centralize data for ease of management, while maintaining performance and availability for remote users.

Data Classification and Reporting – Knowledge is power. Discovering storage capacity utilization and determining the business value of data is necessary in order to make good decisions about where to put data, how to protect it, and what kinds of new storage devices to purchase. Again, most of the knowledge needed to classify data, and most of the information needed in reports, is related to the file-level properties of the data.

FAN “vs.” SAN

The proceeding section illustrates some of the reasons why it is necessary for storage managers to move up the protocol stack all the way to the file level. It also illustrates the source of a common misconception: that File Area Networks are a technology *opposed* to Storage Area Networks.

In reality, the two technologies are more than complementary; they are actually symbiotic. SANs are a requirement for the most robust FAN solutions, and FAN solutions consist of tools that SANs simply cannot provide. As FANs make management easier at the file level, this will allow continued growth in data on the underlying storage subsystems... which are usually SAN attached. Remember: all file data is ultimately stored in block format, and block data is optimally stored on a SAN.

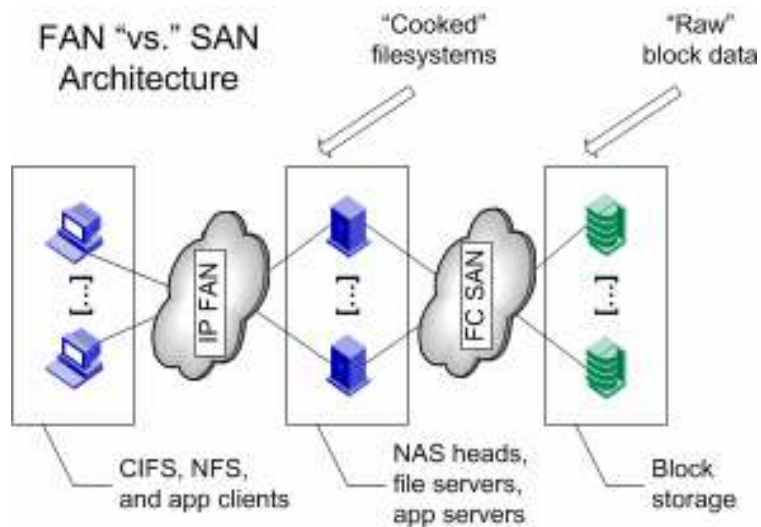


Figure 2 - FAN and SAN Architectures in Concert

FAN Support Products

FAN technology sits on top of other network infrastructure. Broadly speaking, there are seven major components to a FAN solution. Refer to Figure 2 while reading this list, and notice where each element would sit in the diagram:

- Clients which access files.
- Connectivity between clients and the file servers. This also allows clients to access namespace services.
- Policy driven file management and control, to align file locations and properties with business requirements.
- A namespace, with associated software and hardware to serve and maintain it.
- Devices to serve files, e.g. NAS heads and file servers
- Filesystems residing on these servers.
- Back-end storage devices, with an optional SAN.

Each of these elements requires underlying support technology. This section discusses some of those

technologies. Many comprehensive books have already been written about networking, and most readers will already be somewhat familiar with such products in any case. Therefore this section only provides a brief review and a discussion of how these products specifically relate to FAN architecture.

Underlying Network Components

A network hub or switch allows connectivity between its ports such that any port can “talk” to any or all of the other ports. Switches and hubs are both “Layer 2” devices (L2) in IP terminology. For example, in an IP/Ethernet network, switches operate at the Ethernet layer, which is the second layer in the Internet Protocol layered model.

Switches are distinguished from hubs in that switches do not have a “shared bandwidth” architecture. With a hub, if two ports are “talking” to each other, this precludes other ports from talking at the same time. There is only one port of bandwidth which is shared between all nodes. If a pair of devices should talk full speed to each other on a hub, this can preclude any other devices from talking at all until they are done. With a switch, on the other hand, connectivity is allowed regardless of activity between any unrelated pair of ports. It should not be possible for one IO pattern on a switch to “starve” another for bandwidth. This is one reason why Fibre Channel switches were successful in the SAN marketplace and FC-AL hubs quickly became obsolete: it is unacceptable for a host to be denied access to its storage for any length of time, and this happens more often than not with hubs.

A router is similar to a switch in that it provides potential data paths between all of its ports, but a router operates at a higher layer up the protocol stack. If switch operates at the Ethernet layer (L2), then a router operates at the IP layer (L3). This allows a router to connect

autonomous or semi-autonomous network segments together in a hierarchical structure, rather than a flat one.

Historically, routers were much slower than switches, and were not available for high performance applications. In fact, many IP routers were implemented in software rather than hardware. Modern IP networks generally use Layer 3 switches, which combine the hardware-accelerated speed of L2 switching with the intelligence of L3 routing in a single integrated platform.

In a SAN, the reliability and performance requirements for switches and routers are strict. The network is expected to deliver every frame without fail or delay except under rare and extreme circumstances, and to deliver all frames in order under virtually every condition. This is because the nodes and applications attached to a SAN were designed for direct attachment, where delay, out of order delivery, and frame loss simply do not occur. Any “working” SAN must make storage look as if it were directly attached from the point of view of each host, so that the SCSI protocol layer can work exactly the same way for SAN as it does for directly attached storage devices. As a result, hubs are almost never suitable for SAN use, and the vast majority of production SANs use Fibre Channel fabrics rather than IP/Ethernet. The breakdown at the time of this writing is more than 99% FC vs. less than 1% IP for production SANs.

In contrast, network filesystem protocols such as NFS and CIFS were designed with the assumption that the network could drop, say, 1% of packets on a regular basis – since IP networks often did that until very recently, and are still far less reliable than FC fabrics. In addition, the protocol designers assumed that performance on IP networks would be erratic and low compared to direct attached storage – again, because IP networks behaved that way when the protocols were designed. With upper level protocols in architected to compensate for unreliable

and slow underlying infrastructure, the requirements for transport are comparatively relaxed. It is therefore not surprising that the protocol of choice for FAN transport is IP, usually over Ethernet. Indeed, FANs are almost always built on top of the existing commodity Ethernet gear already in place in an enterprise, rather than using switches and routers built for block storage requirements. The breakdown of Ethernet vs. FC deployments for network filesystems is exactly the opposite of the breakdown for block-level storage networks.

The bottom line is that most FAN deployments will make use of IP/Ethernet gear. The network storage devices on the FAN will always have a block-level back end, which will often be connected via a SAN. In that case, the back end will almost always be Fibre Channel.

RAID Arrays

“RAID” stands for “Redundant Array of Independent Disks.” RAID subsystems have a set of physical disks which are “hidden” behind one or more RAID controller interfaces. The controllers present hosts with logical volumes that do not need to map directly to the physical disks. That is, the “picture” of the storage looks different to a host vs. the disks which are physically present in the RAID array. They group together physical disks to form logical volumes. This can be as simple as concatenating disks together so that many small disks appear to be few large volumes, or may involve complex layouts with redundancy and performance enhancements.

RAID arrays form the bulk of mass storage for the back-end of FAN solutions due to their high degree of configurability, enterprise class performance, and high availability options.

Remember that RAID arrays are block devices. In a FAN, a RAID array must have some sort of network storage front end processor: a device which takes the

“raw,” block-level data on the RAID volumes, configures it as a “cooked” filesystem, and presents that filesystem to a network interface using a protocol such as NFS or CIFS. This could be a general purpose server running a protocol stack, or special purpose appliance hardware. In some cases, a RAID array will be built into the same platform as the network storage processor. In larger scale solutions, RAID arrays and separate network storage processor nodes will be co-located on a Fibre Channel SAN, which will provide block-level any-to-any connectivity between the processors and arrays. This solution offers the best of both the block and file networking architectures, and is expected to be the norm for enterprise-class FAN deployments. See Figure 2 (p6) for an example.

Of course, any storage device could be used on the back end of a FAN. This includes S-ATA drives inside servers, JBODs, tapes, solid state media, and so forth. A detailed discussion of storage technology is beyond the scope of this work. See the book “Principals of SAN Design” for more information on this topic.

FAN Protocols

The products discussed in the previous section relies on a protocol, or rather, on several protocols in combination. FAN designers must be familiar with the characteristics of each protocol option when selecting equipment and planning for performance, availability, reliability, and future extensibility.

Protocols are behaviors that computers and network devices must follow in order to communicate. If devices on a network do *not* use the same protocols, they cannot communicate. Imagine a person who only speaks English trying to have a complex philosophical debate with another person who only speaks Swahili. Indeed, it is often hard enough to have a conversation if one person speaks American English and the other learned English in

England. (Or Parisian French vs. French spoken by Canadians, or Spanish spoken in Mexico vs. Spain, and so on.) Similarly, network devices must “speak” the same language (e.g. English), and use the same unofficial variations (e.g. American English). This means that industry-wide agreement is required on both “official” standards, and “de facto” standard implementation details.

Protocols apply at all levels of communication, from physical media and cabling all the way up to the application level. Many protocols at many levels are usually involved when two devices communicate. The entire group of protocols is collectively referred to as a *protocol stack*. Every piece of the stack must work properly for communication to occur.

This subsection discusses some of the protocols that are relevant to file networking today. The focus is on networking protocols such as IP, Ethernet, Fibre Channel, NFS, and CIFS. The discussion starts at the lowest level of the FAN stack and then works its way upwards.

SCSI

The Small Computer Systems Interconnect (SCSI) protocol is the basic building block of storage infrastructure today. It was originally designed as a Direct Attached Storage (DAS) protocol, with a short bus directly connecting a SCSI controller to one and only one host. SCSI could allow more than one storage node, but not *many* more than one, and the single initiator limit prevented the most meaningful solutions from being achievable. The overall architecture remained fundamentally the same until the advent of storage network optimized protocols.

Storage solutions today map the SCSI protocol onto some other transport, usually Fibre Channel. This mapping leverages the standards work already done on

SCSI – both the official SCSI standard and the de facto implementation standard practices – so that vendors building SAN-attached hosts and storage nodes have a head start at development and testing. This also allows SAN protocols such as Fibre Channel to be deployed with less risk: customers can be sure that most of the protocol “parts” of the network are extremely well-tested and stable, since the FC protocol inherits quite a bit of code from SCSI, which has already been validated in real-world environments for decades. Virtually all FANs use SCSI at some point in their back-end connection to block storage.

Fibre Channel

The leading protocol for SAN use is Fibre Channel. To date, Brocade alone has shipped many, many millions of ports of Fibre Channel infrastructure for production use, and other vendors have shipped similar amounts. Fibre Channel is the only protocol that was designed specifically for storage networking at every layer, and FC owes its popularity to technical superiority which resulted from that design.

Fibre Channel encompasses to a suite of protocols ranging from physical to application layers. It first appeared in 1994, and rapidly became the standard by which all other SAN protocols were judged. Fibre Channel has been in production use for years, and has a successful track record in mission-critical environments. Indeed, Fibre Channel has been so successful that it currently represents well over 99% of the overall SAN market installed base. All other SAN approaches combined – including all IP SAN technologies put together – amount to a fraction of one percent of production SANs. Current FC networks may now run at 1Gbit, 2Gbits, 4Gbits, or 10Gbits.

In addition to defining behaviors for Fibre Channel products and services, the FC standards define mappings

for higher-level protocols (e.g. SCSI or IP) to be carried over FC networks, and mappings for Fibre Channel to be carried over other protocols (e.g. ATM, SONET/SDH, or IP).

First, an application on the “host” generates a chunk of data which is to be written to the array. In the case of a FAN/SAN solution, the “application” is the filesystem, and the “host” could be a general purpose server or an appliance running NFS or CIFS. In either case, the application “tells” the operating system about the chunk of data – a file to be written – which tells the HBA driver. Typically, this means that the driver receives a pointer to a memory location which has the data, rather than receiving the data itself. (This improves efficiency a great deal.) The HBA driver tells the HBA hardware where to get the data, and where to put it. The actual data movement between memory and the network is accomplished by the HBA without using the main CPU at all. The HBA reads chunks of data out of main system RAM, maps the data through the FC layers, and deposits a stream of frames onto the network. At the other end, the RAID array performs a similar process to get the data onto the correct disk(s). In this case, the data never goes to an application *per se*; instead, it goes through a hardware-based volume management mapping process (e.g. RAID 5) to determine which physical disks need to receive or provide particular chunks from the data stream.

When the data is placed onto the SAN, it consists of a stream of *frames*. The frame has over 2k of room for payload. This section usually contains SCSI data. Of course, most filesystems today use block sizes greater than 2k. For this reason, Fibre Channel provides a mechanism for grouping over 65,000 frames into a *sequence*. The previous example illustrated how an HBA pulls a chunk of data straight out of RAM and converts it into frames. In point of fact, *sequences* of frames are the basic unit of data transfer from a hardware acceleration

standpoint, which means that well over a hundred Mega Bytes could be sent via a Fibre Channel HBA with the same CPU involvement that would be needed to send a single Ethernet packet. Fibre Channel goes even further than that, and allows many sequences to be grouped into a single *exchange*. Typically, each SCSI operation (read or write) is mapped to a single exchange ID.

IP and Ethernet

Internet Protocol (IP) is the standard for communication on the Internet, and the de facto standard for use within corporate LANs for applications such as email and desktop web servers. It is also the protocol of choice for the front-end component of file networks.

In most LANs, IP is carried over Ethernet. Upper level protocols such as NFS and CIFS are mapped on top of IP, usually with TCP in between for error detection. An IPv4 address consists of four bytes, usually represented in decimal format and separated by dots. For example, “192.168.1.1” is a standard format IP address.

There are advantages to IP when it is used in the way its designers intended. For example, IP was designed to support very large, widely distributed, loosely coupled solutions such as the Internet, and is therefore optimized to solve such design problems. The specifications for IP mandated a loose coupling between IP subnets as the most important design criteria. Any given connection was considered expendable as long as the overall network remained online. Fibre Channel, in contrast, was designed with support for high performance mission-critical storage subsystems as the most important factor. It did not need to be as scalable but it did need to be extremely fast and reliable compared to IP. Since upper layer FAN protocols were designed to use IP as a transport, the reliability and performance issues inherent

in IP do not pose the same challenges for FAN front ends as for SAN back ends.

Network Filesystems

There are a number of options available to map “raw” block data on a disk onto a “cooked” filesystem format. In Windows, NTFS and FAT are examples. In UNIX, there are many more options: XFS, UFS, VxFS, and so on.

Similarly, there are many options for mapping a cooked filesystem onto a network. However, two options make up the vast majority of the network filesystem marketplace, so this book will focus on them. They are NFS and CIFS.

The Network Filesystem (NFS) was developed by Sun Microsystems in the 1980s. It was the first widely deployed network filesystem. Today, it is the most typical choice for UNIX systems, though it can also be used for PC platforms with third party software.

In an NFS environment, one machine (the client) requires access to data stored on another machine (the server). The server may be a UNIX host, a Windows server running third party software, or an appliance. The server runs NFS processes, either as a software-only stack (e.g. a daemon in UNIX) or as a hardware-assisted stack (e.g. a chip in an appliance). The server configuration determines which directories to make available, and security administration ensures that it can recognize and approve clients. The client machine requests access to exported data. If the client is a UNIX machine, this is typically done by issuing a “mount” command. Once the remote filesystem is mounted, users can access the remote files as if they were located on the client machine itself.

When designing an FAN with NFS, it is important to consider a number of limitations and caveats.

For example, NFS version 2 only supported mapping over UDP, not TCP. UDP (User Datagram Protocol) over IP is a stateless and comparatively unreliable option. TCP (Transmission Control Protocol) over IP is still far less reliable than Fibre Channel, but it does deliver enough reliability to allow NFS to work in a more scalable and flexible manner. NFS version 3 or higher is required for TCP support.

It is also necessary to consider access control limitations. NFS does not provide a robust mechanism for granular definition of file access privileges. This can be a particularly challenging issue when implementing a FAN with both NFS and CIFS. Using NFS version 4 will address these limitations, but at the time of this writing, NFS version 4 is not widely deployed.

For those readers interested in the details of the NFS protocol implementation, look up RFC 1094, RFC 1813, and RFC 3530.

The Common Internet Filesystem (CIFS) was originally known as the Server Message Block (SMB) protocol. This is a proprietary protocol developed by Microsoft, and is used mainly for communications between Microsoft platforms and the network storage devices. It can also be used for sharing printers, serial ports, and other miscellaneous communications, but for the purposes of FAN deployments, only the file sharing aspects are relevant.

It is possible to access CIFS filesystems from UNIX platforms using third party software, and to present CIFS filesystems from UNIX servers for use by PC clients in a similar manner. Because CIFS is proprietary, both

approaches rely on reverse engineering and have significant limitations and caveats.

Like NFS, CIFS was not originally written for TCP/IP. In fact, CIFS was not originally written for IP at all: it was written for NetBIOS which would run on top of NetBEUI, IPX/SPX, or TCP/IP. In enterprise-class deployments today, CIFS is almost always mapped directly on top of TCP/IP, and for the vast majority of FAN solutions, this is the optimal approach.

Both NFS and CIFS have a common set of challenges which relate to the need for FAN technology.

For example, neither works well in a WAN environment. High latency connections between clients and servers reduce performance and reliability. This has resulted in a sub-optimal decentralization of resources in most large-scale environments. Some FAN components are designed to solve this problem.

Also, both protocols are designed to map the “mount point” for a remote filesystem (i.e. the location on the client at which users see the remote files) to the *physical* machine which serves the file data. In small environments this can work well. However, when there are hundreds of servers, each of which may have to be changed periodically, it can be extremely difficult to manage mount point mappings. Addressing this issue and its related problems is a cornerstone of the FAN model.

Finally, it is important to reiterate that both NFS and CIFS are “cooked” filesystems. All files ultimately reside in “raw” block format on storage. It is therefore necessary for designers to consider the architecture of the back-end block solution, as well as the FAN. For this reason, FAN designers should consult with SAN professionals. See the

books “Principals of SAN Design” and “Multiprotocol Routing for SANs” for more information on this topic.

Chapter Summary

FAN applications run on top of NFS and/or CIFS, which usually run on top of TCP, over IP, over Ethernet. Through this connection, the applications access files, and the underlying data within the file will reside on the “back end” of servers or appliances. That connection, in turn, is generally handled by some form of SCSI mapping, such as SCSI over Fibre Channel.

These applications are designed to help IT professionals eliminate or at least reduce the complexity of managing the ever growing amount of file data in their environments. FAN applications can make file location and movement automatic and transparent, centralize resources, consolidate management functions, reduce costs associated with backup and recovery, and automate regulatory compliance tasks. The remainder of this book discusses what those applications are, how they work, and how to deploy them effectively.

