Bonus Chapter

What Makes Servers Special

A server computer is different from a desktop computer. Though those differences can sometimes be a matter of degree, and may sometimes not be readily apparent upon a casual once-over, the primary thing that differentiates a server from a desktop is the word *more.* This usually expands into more CPU power (and often, multiple CPUs), more memory (usually twice as much or more than you'd put into a desktop machine), more storage (often, three or more times as much disk storage as you'd put in a desktop), and more networking capability. (Faster, more powerful network interfaces with at least two separate connections are pretty typical.)

But there's even more that goes on with servers that comes increasingly into play as the size of the audience they serve, or the importance of the applications and services they deliver, increases including such things as clustering, load balancing, mirroring, and much, much more. These are the topics you'll find covered in this chapter, with ample pointers to more information for those who want to dig deeper than this rapid survey of "server world" can deliver.

Servers Service Clients

In general, clients ask for services, and servers provide them, where both hardware and software are necessary to make networking work on any computer, either client or server. Here, we take a closer look at the various pieces and parts involved in a client/server relationship to help you understand what happens when a client requests a service from a server.

At the most basic level, a client must have a network connection available to transmit a request for services. Likewise, the client must have the correct software installed to formulate an intelligible request and pass it to the network, which is where a server can notice and respond to such a request.

Making the connection

To request network services, a client must have the following hardware:

- **Network interface:** A network adapter (often called a NIC, for network interface card, even when there's no separate card involved) allows a computer to interact with the network. Before any such adapter can transmit signals onto the network medium and receive signals from that network medium, you must configure it.
- **Physical connection:** The link between the computer and its network must also work properly. This means that clients can transmit outgoing signals and receive incoming signals through their network connections. Likewise, the network medium must be properly configured and interconnected for signals to travel from sender to receiver.

This hardware (adapter and cable, or some other network medium) takes care of the connections part in a three-part simple model for networking, which requires that connections, communications, and services all be available and working.

Software uses the connection

The software on the client computer handles the communications and services necessary for it to interact with the network. Here's a list of software that you normally find on a networked client computer, starting from the hardware level (or as close as software can get to hardware) and working up to the applications that request network services:

- **Network driver:** A special-purpose piece of software that enables a computer to send data from the computer's *central processing unit* (CPU) to the NIC when an outgoing message is ready to be sent. The network driver also forwards a request for immediate attention (called an *interrupt*) to the CPU when an incoming message arrives. You might say that the driver allows the PC to communicate with the NIC, which in turn communicates with the network.
- **Protocol stack:** A collection of communications software that provides the type of "shared language" necessary for successful networking. The protocol stack governs which formats network messages can assume, and it defines a set of rules for how to interpret their contents. Two computers must use the same protocol stack to communicate. We cover protocol stacks thoroughly in Chapter 4.
- **Redirector:** A redirector, or equivalent software, issues requests for remote resources or services to the protocol stack and receives the incoming replies from the protocol stack. With a redirector running in the background, applications don't need to be explicitly network aware because the redirector handles network connections.

- **Network-aware application:** Network-aware applications understand when service requests can be satisfied locally or must be satisfied remotely. In the latter case, a redirector may be present, but it may not necessarily handle certain types of network services (such as e-mail or Web-page access). However, the redirector can handle other types of network services, such as providing access to a file stored elsewhere on the network that's applied as an attachment to an e-mail message. In such a case, the redirector grabs a copy of that file across the network and attaches it to the outgoing e-mail message.

When a client makes a request for a resource or service that requires access to the network, either the application (if it's network aware) or a redirector (if the application isn't network aware) formulates a formal request for a remote service. Satisfying the request may involve the transfer of a small amount of data (as when requesting a listing of a directory on a machine elsewhere on the network). However, it may also involve transferring a large amount of data (as when sending a large file off to be printed or when copying a large file from the client machine to a server).

The request is ferried through the protocol stack that the client and server have in common. For short requests, a handful of short messages travel from the client and are reassembled and handled by the server. For large information transfers, the client breaks up the file into hundreds or thousands of small information packages, each of which is shipped across the network separately and then reassembled on the receiving end.

KEY CONCEPT

The protocol stack tells the network driver to send little packages of data (called *frames* or *packets*) from the computer, through the network adapter, and across the network to the intended recipient (the server). On the receiving end, the same process happens in reverse, with a few additional considerations that you find out about in the following section.

Servers Deliver Services

In the preceding section, you find out that clients ask for services and that servers provide them. What handling requests on the server side really means is that a special bit of software, called a *listener process,* runs continuously on the server and listens for requests for a particular service. When a request arrives, the listener process handles it as quickly as possible.

Servers thread through a maze of requests

What usually happens on most server operating systems — including Windows Server 2008 — is that the listener process simply recognizes that a request has arrived. The listener process checks the identity and the associated permissions of the client, and if the client is who he says he is and has the correct permissions for the service, the listener process grants the request for service. It does so by starting a temporary process (called an *execution thread* in Windows-speak; think of this as a very small program) that exists just long enough to handle whatever service the client requests — after which, the temporary process disappears. For example, a request for a particular file on a server would result in the creation of a temporary process that exists just long enough to copy the requested file across the network. As soon as the copy completes, the temporary process goes poof!

Using a listener process to create short-lived execution threads allows a server to handle large numbers of requests, because the listener process never stays busy for long handling individual requests. As soon as a listener process creates a thread to handle one request, it checks for other pending requests and handles them; otherwise, the listener process goes back to listening for new requests. Typically, a server has one or more listener processes for each service that the server supports. This is part of the reason why a server needs more horsepower than a desktop PC: Because it does more and generally manages many more execution threads at any given moment than a desktop, it generally needs more computing power (more powerful CPU or CPUs), workspace (more memory and disk space), and faster network communications (more and more powerful network adapters) to do its job.

Servers are demand-driven. That is, their job is to respond to requests for services from clients. A server rarely initiates activity. This reactive mode of server operation helps explain why the client/server model is also known as a *request/response* or a *request/reply* architecture, in which clients make requests and servers respond or reply to them.

Other than the necessary listener processes and a set of service applications that actually perform services, servers need the same hardware components that clients do. A server needs one or more network adapters, each with a working connection to the network, to allow data to enter and leave the server.

Software is similar on the server side

On the software side, servers also need the following elements so that their services can be available across the network:

- **Network drivers** enable the server to communicate with its NIC. This software lurks in the background and exists only to tie the computer to the NIC.
- **Protocol stacks** send and receive messages across the network. This software also lurks in the background and provides a common language shared with clients used to ferry information across the network.

- **Service applications** respond to requests for service and formulate replies to those requests. This software runs in the foreground and does the useful work. The service application includes the listener process, the temporary execution threads, and some type of configuration or management console so that it can be installed, configured, and altered as needed. Typical service applications include directory services (Active Directory), database engines (SQL Server or Oracle), and e-mail servers (Exchange).

REMEMBER

Most, if not all, software that resides on a server is network aware because delivering information across a network is a server's primary function.

Decoding a Client/Server Conversation

You may be wondering what the steps are in a conversation between a client and server. Examining the exact contents of such a message exchange wouldn't do you much good. However, the following sequence presents a typical request to print a file on a network printer (and, by necessity, through a print server) from a spreadsheet program:

1. A user requests print services in the spreadsheet program by clicking the printer icon or by choosing File➪Print.

   Assume that a network printer is set as the default printer for the designated print job.

2. The spreadsheet program formats the spreadsheet and then builds an appropriate print file.

   A print file includes the text and graphics that make up a file's content. It also includes instructions on how (bold, italic, and so forth) and where (top, bottom, left, right) to place the elements to be printed.

3. The spreadsheet program sends the print file to the printer.

4. The local networking software (assuming it's a Windows XP or Vista redirector) recognizes that the printer is on the network and sends a print request to print that file to the print server. The redirector accesses name and network address information through a Windows networking service (called the Browse Service, which talks to a browser server on the network) to figure out where to send the print file.

5. On the server side, the listener process recognizes and checks out the user's print request. We'll assume it's legal, so the listener process creates a temporary execution thread to handle delivery of the incoming print file packets from the client. This temporary thread tells the client to start sending the print file.

6. Having now obtained permission to start shipping the file, the protocol stack on the client chops the file up into small chunks (called *packets*) that are delivered to the temporary thread on the server.
7. The temporary thread on the server oversees delivery of the file and places it into a temporary holding area (called a *spool file*) where the print server stores all pending print jobs. The print server places the job in the *print queue,* which stores the print jobs in the order in which they are received.
8. When the print job reaches the head of the queue, the server creates another temporary thread to ship the job to the printer.

   In many cases, a different protocol carries data from the server to the printer than the one the client uses to ship data to the server in the first place.
9. In a final (and optional) step, the print server creates another temporary thread to send a message to the client computer stating that the print job is complete. Here, the same protocol used to transport the file from the client to the server is often used to send this message back to the client.

What's worth noting here is that a kind of conversation occurs between client and server. The client initiates this conversation when it asks for permission to print, and then it sends the print job to the print server. The server takes over from there, storing the incoming print file in its spool file, managing the queue, and then printing the file when its turn comes. The conversation ends when the server sends notification of job completion to the client.

Requests for other services, such as access to a database server, an e-mail server, or even a file server, are similar to the previous interchange. In such cases, the conversation usually ends when the server sends a data table, message, or file in reply to the client's initiating requests. This request/reply sequence is really what makes modern networks work. The reason servers demand more and more robust hardware isn't that what they do is radically different from what desktops do, it's that they do it on a much larger scale — in other words, MORE! This concept is subject to some interesting wiggles, as you see later in this chapter, and reflects what servers tend to do more of (networking and services delivery) and what they tend to do less of (presenting data for user inspection).

Examining Different Types of Servers

As we've mentioned, servers deliver many kinds of services, from file and print, to database, directory, e-mail, Web, and much, much more. Some servers take a generalist approach and, within their limits, try to do as much as they can for the users they serve. Other servers take a specialist approach, and concentrate on delivering specific types of services to certain classes of users.

Generally, as the workload increases and the number of users goes up, server roles tend to become increasingly specialized and interdependent. Lighter-duty situations allow servers to act as jacks-of-all-trades, but as demand for specific services skyrockets, specialization permits individual servers to be designed to handle specific jobs, as well as the underlying technology and available budgets permit.

This explains why servers also come in many forms, because form indeed follows function. And as that function gets more focused and specialized, so does the form.

Looking at server models from one manufacturer

We'll take a look at some Dell product offerings to help explain what happens to servers as their costs and capabilities increase and how multiplying servers tends to scale IT operations into increasingly specialized server architectures. We could just as easily pick other vendors that serve the same kinds of markets, such as IBM, HP, Sun Microsystems, or any of a host of smaller fry, but in the server world generally the way things work stays more or less the same across all these various players. To illuminate this concept, we compare and contrast the kinds of servers that Dell offers to small business customers.

TIP

When pricing servers, you must invariably start from published list prices from the vendor or an authorized reseller (VAR). Published prices for servers are interesting because you must add components to put a complete package together for purchase. Always treat published prices as base prices to which you must add more components to put something together you can actually buy and use on a production network.

On the small business side, Dell offers the following kinds of servers:

- **Tower servers:** A *tower server* comes in its own stand-alone enclosure that's designed to stand upright on the floor. Dell offers three different lines of tower servers called Essential (up to 10 employees), Enhanced (10–50 employees), and Elite (more than 50 employees). Prices vary from $400 to $1,700 for Essential, from $1,000 to $1,300 for Enhanced, and from $4,600 to nearly $14,000 for Elite models.

- **Rack servers:** A *rack server* is designed to be housed in a standard 19-inch equipment rack, and it's built in a carefully specified modular format measured in terms of storage units (abbreviation U; size 4.4 cm or 1.732 inches). Also called *rack-mounted servers,* these devices are typically 1U, 2U, or 3U in size, where racks typically accommodate from 20 to 30 U per vertical stack. Dell offers nine different server models with the following designations: entry-level, advanced, performance rack, business-critical, and performance. Prices range from $850 to $1,500, with high-end models that cost roughly $4,000 to $5,000 also available.
- **Blade servers:** *Blade servers* also mount in rack elements, but they don't include built-in power supplies and normally can't accommodate as much storage as rack-mounted servers. It isn't unreasonable to think of a blade server as a kind of "server on a circuit board" where the blade slides into a special enclosure from whence it draws power. A rack-mounted server can work without a rack; a blade server can't work without a blade enclosure (which usually comes in a 7U to 10U form factor). Blades start at $1,250 each, but an enclosure costs upward of $3,000.

Dell also offers the following, very appealing server solution bundles to small businesses as well:

- **Starter Network Server:** At prices from $1,200 to $1,500 and up, these servers come fully configured with the do-it-all Microsoft Windows Small Business Server 2008 installed, which offers basic file, print, fax, messaging, database, Web, and collaboration services for up to five simultaneous users without requiring the purchase of additional *client access licenses* (called *CALS* in MS-speak).
- **Exchange Server:** For $5,000 and up, these servers come configured with Microsoft Windows Exchange Server 2008 installed, with a built-in system drive and a 3-drive RAID 10 array, 4GB RAM, and 5 CALS. (Multipacks are available for about $34 per seat, $90 per seat for Small Business Server.)
- **Mobility Server:** Costing from $1,200 to as much as $8,000, this particular model is designed to support remote access via dial-in or high-speed Internet connections as well as local network access. It provides services to users who are on the go, as well as to those in the office. High-end versions of this server are similar to the aforementioned Exchange Server offering.

What you see here is an enormous range of capability and expense: from under $1,000 at the lowest cost to over $14,000 at the highest cost. There are also lots of choices you must make when putting your servers together, which of course is what makes servers more expensive and, in many ways, more interesting than desktop machines.

Rack-'em up!

We just talked about rack-mounted and blade servers. Now, let's talk about where those servers (and for blades, their enclosures) go: into equipment racks. Equipment racks are usually purchased in vertical bays, where a one bay unit will usually accommodate between 20 and 30 units of storage (20U to 30U). Racks are usually priced in increments in that range, so that it's understood a 22U unit is a single bay, a 44U unit is a double bay, and so forth.

Rack mounting is an attractive proposition for equipment buyers for many reasons:

- You can buy from multiple vendors, yet still be sure all the pieces and parts will fit together.
- You can maximize the use of the floor space in equipment rooms or closets, because many units can be stacked on top of each other before additional square footage is needed for more equipment.
- Racks permit you to estimate cooling and power requirements accurately, so they can be sure to deliver proper ventilation, cooling, and A/C power to the units to maximize equipment availability and longevity.

Typical servers vary from 1U to 3U in size, depending on their capabilities, storage capacity, and design. Other elements that go into equipment racks include uninterruptible power supplies (UPSs) that condition incoming A/C power, and provide enough battery backup to permit an orderly shutdown in the event of a power failure, or to keep gear running long enough for backup power sources such as a generator to be brought on line. Equipment racks also house various network appliances, switches, routers, and other network infrastructure elements required to bring servers and clients together. Figure BC-1 shows an example of one type of server rack.

Wiring management is also a key motivation for equipment racks. Rack-mounted devices are organized, tidy, and accessible from their front panels for installation and maintenance. Cables of all kinds are routed into and out of equipment racks through the back, so that power, networking, and other cables can be organized and managed with minimum muss and clutter. The more machines in a rack, and the more racks that cluster together, the more important this becomes.

Racks also provide additional physical security for the equipment they house, thanks to locking front and rear access doors. It's typical to place equipment racks inside access-controlled equipment rooms or closets, and to monitor who goes into those spaces, when, and for how long. Software monitoring lets organizations keep track of what those people are doing, an increasingly common activity intended to meet mandated auditing and tracking requirements (sometimes a matter of law or regulation, depending on the industry involved, but always a matter of good, sound IT policy and operations).

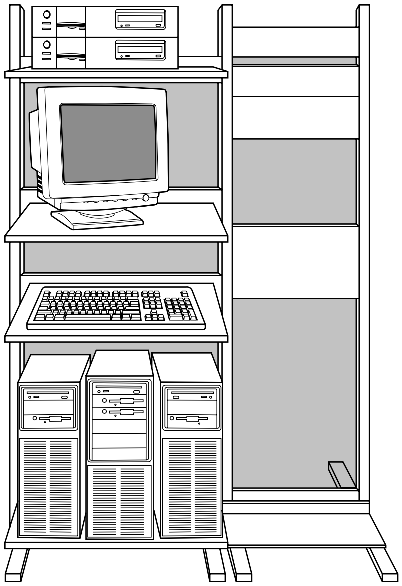

Figure BC-1: Choose the right type of server rack for your needs.

Light on the GUI, Heavy on Everything Else

One of the things that makes servers so different from desktops is that graphics performance is basically of little or no interest. Where those building desktop (or entertainment) PCs typically spend a lot of time, effort, and money in choosing just the right graphics card (or cards, for heavy-duty gaming machines, which can use two or even four graphics cards for extra graphics oomph), most servers use basic graphics circuitry built into the motherboard itself — or make do with extremely modest graphics cards when built-in graphics aren't available. (That's more the exception than the rule for server motherboards nowadays but by no means unheard of.)

Why is this? Simply put: Running the GUI on a server is needed only for setup and routine maintenance. Bizarre and occasional exceptions aside, nobody sits down in front of a server to get individual tasks done. Administrators are usually the only class of users allowed to do anything to a server, and they typically work from their desktops using remote management tools whenever possible. So they seldom sit down in front of the server either, except for occasional tasks that absolutely require them to do so. (And these are pretty few and far between, both in theory and in everyday practice.)

The bottom line is that servers have only skimpy graphics needs and can get by just fine on the graphics built into most motherboards — or with the capability you get from a $40 graphics card. All you really need is enough resolution to run the Windows GUI (which works very well at 1024 x 768 or at 1280 x 1024 resolutions that are well within the range of built-in graphics circuitry and low-end graphics cards).

Likewise, servers don't need big, fast, fancy monitors either. Individual servers work fine with 15-inch or 17-inch monitors, and most rack-mount or blade servers share a single monitor and keyboard setup for the entire rack, which is typically a flip-up LCD display, plus a modest mouse and keyboard, all tucked away in a 1U drawer rig in most rack environments.

Another area where servers need less capability is in the optical drive. Some desktop PCs include high-definition optical drives for HD-DVD or Blu-ray, especially on multimedia or entertainment models; most servers can get by with basic play-only DVD players, or with only modest recording capabilities.

Table BC-1 summarizes the differences between typical desktops and servers as Windows Server 2008 makes its way to market. Because these kinds of numbers tend to change rapidly, it's probably more important to pay attention to the relationship between the ranges than to the actual numbers themselves. Hopefully, they do make the point that more of everything is a valid server mantra, and far less applicable to even the most extreme desktops.

Table BC-1 Typical Desktops vs. Typical Servers

Type	*Desktop*		*Server*	
	Low	*High*	*Low*	*High*
CPU.n	1	2	1	4
CPU.c	1	4	1	4
RAM.gb	1	4	2	64
Disk.tb	0.5	2.0	1.0	8.0
RAID	No	Sometimes	Sometimes	Yes
Network.s	10/100	GbE	GbE	10 GbE
Network.n	1	2	2	4+

Key
.n stands for number of devices (CPUs and network adapters)
.c stands for total CPU cores (semi-separate devices on a single chip package)
.gb stands for gigabytes (of RAM)
.tb stands for terabytes (of disk space)
.s stands for speed (10/100 Mbps Ethernet, GbE = Gigabit Ethernet, 10 GbE = 10 Gigabit Ethernet)

Jokingly put, it's sometimes said that servers need expanded head and foot room, not just because they're often larger than desktop machines (1U and 2U rack-mount devices are smaller than you might think, however) but primarily because they benefit from more RAM (workspace for processing underway at any given moment) and disk storage (room for the results of processing, programs, and data to be stored). Because performance is so important on most servers, high-end, high-speed disk arrays (RAID) are far more common on servers than on desktops (though a few very high-end desktops do sometimes include them), and the total amount of disk space is often many times greater on a server than on a desktop PC.

Load-Balancing and Other Interesting Acts

As the sheer size and scale of workloads increase — and if you have trouble thinking about what this really means, think about the collections of machines called "server farms" that routinely labor away behind such well-known Web sites as Google, Yahoo!, MSN, and so forth — servers tend to gang up to share workloads. This usually requires some special-purpose hardware and software to manage incoming service requests and to distribute the load among however many servers may be available to pitch in and help out at any given moment.

One method for handling large amounts of processing employs what's called a *server cluster.* While a server cluster typically employs two or more servers to do the actual processing work, a server cluster looks just like a single server to an outside consumer of services. This approach uses a special hardware element called a *cluster controller* to handle the active workload. Sometimes clusters are designed to provide what's called high availability. In plain English, this means they must always be available for use (and to provide services). They include a special *failover capability* so that if any server in the cluster goes down, its workload automatically passes to another server with no affect on user sessions or service delivery.

Most of the time clusters are designed to provide improved performance, and do so by routing incoming service requests through one or more devices called *load balancers.* These devices (which may be special-purpose routers or specially-configured servers) keep track of all incoming service requests and all ongoing service delivery activities, and seek to smooth out the load across all available servers, so that none is either excessively idle or excessively busy.

Clustering is a realm of processing into which most of the people who read this book may never venture, though all of us will take advantage of such capabilities the next time we log into or use an Internet portal of any kind, or a high-traffic Web site. The servers that companies such as Google, Yahoo!, and Microsoft use in their clusters for such sites often cost six figures or more, and are backed up with equally expensive networking and storage infrastructures as well.

Server Appliances and Applications

Another emerging class of networking infrastructure that touches even the low end of the server and desktop computing spectrum is *network attached storage* (NAS) or *storage area networking* (SAN). This technology comes in the form of special-purpose devices that often use their own separate high-speed networks (gigabit speeds or higher are typical for storage applications), or that can share a single network with other applications. At the highest end of the computing spectrum, these network attached storage devices combine high-speed network interfaces on individual servers with a secondary, separate high-speed backbone network that lets storage servers whose specialized job is to provide and manage data storage and retrieval interact with other servers.

Because consolidation of storage has the same general benefits as consolidation of function in ordinary servers, storage servers can provide networked storage that's as fast or faster than local disk drives, with huge volumes of disk space and high levels of availability and reliability that even individual servers can be hard-pressed to match. Network attached storage also offers great economy and convenience, even for SOHO networks. As we write this chapter, you can buy pre-packaged NAS devices, also known as storage appliances, that deliver 1 TB or more of storage via Gigabit Ethernet at affordable prices (under $1,000 per TB as we write this chapter, with RAID support, plus backup and archival capabilities).

Other important server capabilities sometimes make their way into appliance form as well. It's not unusual to find stand-alone or add-on devices that work as print servers nowadays, to handle print spooling and output notification without requiring a Windows server to handle the job. In a manner of speaking, storage appliances do the same thing for file storage and access, and often come equipped to handle Macintosh and Linux clients, as well as Windows desktops.

We talk more about other network appliances in Appendix A, where we get into so-called Internet appliances and security appliances in more detail. In fact, you can buy specially designed appliances for under $100 that combine the functions of network storage, print server, security management, and a 4-port Gigabit switch, with a broadband connection to a cable or DSL modem that blurs these distinctions nearly beyond recognition!

Server Scalability, Reliability, and Availability

Earlier in the chapter in the "Examining Different Types of Servers" section, the rack servers we describe as examples include such designations as "performance" and "business-critical." To savvy server users (and buyers), this kind of language implies three inter-related characteristics: increased cost, increased capability, and increased complexity.

Server mavens are fond of specialized language and use the terms that appear in this section's title in specific ways, with equally specific meanings. In the following list, we define these terms to help you understand what's important when it comes to scalability, reliability, and availability; the costs, complexity, and capability that come into play; and why these characteristics may or may not matter for your specific needs and situation:

- **Availability:** The level of fault-tolerance in the server architecture, best understood as the server's ability to withstand failure of individual components, where high availability is a special characteristic that can be costly to assure. The most basic servers are not considered highly available. As buyers invest more in redundant parts — such as hot-swappable disk drives, power supplies, network interfaces, and so forth — availability levels and costs increase. The highest levels of availability occur when servers support failover capability (as described earlier in our discussion of clustering and load balancing) so that a hot standby server is ready to take over for an active server if and when the active server should fail, without affecting user service (or with only minimal delays or minor effects).

  TECHNICAL STUFF: In the most formal of terms, availability is measured as a percentage of uptime, which represents the total number of hours of actual operation divided by the total number of hours in the measurement period. High availability is often called "five nines" and refers to 0.99999 availability or 1 hour of inoperability for every 100,000 hours of operation.

- **Reliability:** The probability of failure for the server, most easily understood as the combination of all worst case figures for all component parts. As servers become more reliable, the cost of their components goes up accordingly. It's easy to confuse reliability (the probability of failure) with availability (the percentage of operation time over some time interval). But though the two concepts are clearly related (some of the downtime that availability measures may be caused by device failures that indicate reliability issues), they are not the same. Generally, server parts are more expensive than desktop parts at least in part because increased reliability adds to cost but also delivers increased availability as well.

- **Scalability:** The ability of a system to continue to perform well under increasing load or demand. A highly scalable system is one that handles peak demand with little or no measurable performance degradation; less scalable systems will register more performance degradation but will hopefully not fail. Scalability comes from proper planning and design of servers where workloads must be carefully estimated and some additional margin of safety (or error) be included to make sure the server can handle what it is asked to do.

  One good explanation for the ever-increasing popularity of rack and blade servers is that when combined with clustering technologies, they make it easy to add processing capacity in useful increments at affordable costs. Instead of migrating from older, less capable servers to newer, more capable ones, these technologies introduce the ability to scale at need or by deliberate plan to keep processing power in synch with demand and use.

High-end, more expensive servers generally possess more of these attributes than lower-end, less expensive ones. It's important to buy what you need when it comes to availability and scalability, but it's also a waste of money to buy much more of those things. As with so much else in life, it's a balancing act where you don't want to over-estimate your needs, or under-deliver necessary resources.

Super Server Resources

When it comes to understanding Windows Server 2008 hardware requirements, design, and components, you will find many useful and informative sources of information available online. Chief among these is Microsoft's Hardware Design Guide for Windows Server (you can find this document at `www.microsoft.com/whdc/system/platform/pcdesign/desguide/serverdg.mspx`). This document is regularly updated, but as we write this chapter, it has not yet been updated for Windows Server 2008. By the time you read this, it probably will be updated, however.

Another great resource that already includes numerous Windows Server 2008 updates is the Windows Server Hardware Platform page (`www.microsoft.com/whdc/system/platform/server/default.mspx`). Its coverage includes RAID adapters, hardware support issues and directions, plug and play device deployment, processor power management schemes, the extensible firmware interface, and hardware error handling. You will also find lots of links to related topics here, including PC, device, and driver fundamentals, numerous platform design guides, Windows Hardware Compatibility (WHQL) information, and lots more.

All the big server vendors offer lots of information and advice on building Windows servers, along with training, tutorials, and resources for current and potential customers. Here's a sampling of some of the many offerings you can find online:

- **Dell Small Business 360 (USA > Small Business > Dell Small Business 360):** Articles and materials include technology advice, business resources, industry guides and information, and success stories. Dell also provides access to various community forums, blogs, and RSS feeds through this Web page, with all kinds of product information and support links.

```
www.dell.com/content/topics/topic.aspx/global/shared/
        bizportal/en/resource_center?c=us&cs=04&l=en&s
        =bsd&~ck=sb3
```

- **HP Small & Medium Business Servers & Storage expertise center:** Product and solution information, plus pointers to the Small Business Connection, expert articles, Webinars, and more.

```
www.hp.com/sbso/serverstorage/index.html
```

- **IBM Small Business Resource Center:** Product, solution, and services information, with a special servers area, plus a large collection of server-related resources such as case studies, white papers, articles, and more.

```
www-304.ibm.com/jct03004c/businesscenter/small
        business/us/en
```

You can also turn up a wealth of resources on Windows Server hardware, planning, requirements, and administration at ServerFiles.com, a directory of resources for Windows network administrators and IT professionals.