Chapter 1

CLIMBING THE MOUNTAIN: 24/7 DATA ACCESS

Technology is, in many ways, more an art than a science. It is driven more by innovative minds than by textbook models. Much in the same way our drive to invent products for the future is often hindered by a lack of engineering ability, the development of products for the Internet generation has often times placed a distance between what the developer envisioned and the final product. After years of developing products under such constricting realities, it is easy to see how the integration of those products could be challenging. For example, the idea and development of tools to produce media-rich Web content was created many years before the telecommunications industry could provide the necessary bandwidth to make those products a reality. So, for many years, companies tried to integrate media-filled Web content with their own products and ideas, causing them to be challenged by the integration of products and the limitation of bandwidth.

Amidst the mass of connectivity now flowing through our homes and telephone lines, rarely a day goes by that we do not have an "imperfect" experience on the Internet. We all have left shopping cart Web sites due to badly written programs or slow response times. Even on our best day, we often cannot connect to our favorite site or online store. Most people don't give thought to why they cannot access a site; they just accept it as commonplace, for the Internet has always been known as an unreliable, public medium.

A chief executive from one of the largest software companies in the world once criticized the automobile industry, stating, "If the automobile industry had grown the way the technology industry has grown, we would all be driving hover crafts." In retort, a chief executive from a leading automobile maker said that although the automobile industry could have grown that fast, the result would have been unreliable automobiles that stopped for no reason, consistently failed to start, and required constant maintenance to keep on the road.

Although the automobile executive responded in jest, his point rings clear. People who purchase automobiles expect them to be reliable. They expect them to operate when the key is turned, to go when the accelerator is pressed, and to stop when the brake is applied. Unreliability and component failure is not acceptable.

In the technology arena, users accept intermittent failures as commonplace, viewing it as a price paid for innovation. The demand for bulletproof hardware and software has not been great because the personal computer, from the date of its inception, has always been riddled with a lack of *fault tolerance*—the ability to recover from a single point of failure (SPOF). As the computing environment in companies began to move from internal data processing to a model that was more accessible by their clients, it made sense to replace the large mainframe environment with a lower cost client/server architecture that is accessible by a World Wide Web browser—giving birth to the era of e-commerce.

Unfortunately, as the systems in use changed from large, reliable mainframe systems to low-cost client/server solutions, companies sacrificed fault tolerance for cost and the ease of integration with existing systems. Systems and applications became less reliable, even though they were more functional and easier to use. The sacrifice of fault tolerance was unavoidable due to a lack of client/server technologies that could provide the reliability and performance of a mainframe. But things are now changing.

Today, systems and services are being designed for high availability. Applications such as ERP (Enterprise Resource Planning) databases, business-to-business (B2B) e-commerce storefronts, and Web portals are in high demand. Along with that demand comes a need for servers that can provide data to users in a reliable fashion. Similarly to how mainframes were often used for data center environments that demanded 24/7 access, groups of servers, known as *clusters,* are now being deployed within data centers to rival, if not equal, the reliability and stability of the mainframe. A cluster, loosely defined, is a group of computers that work together to create a virtual server that can provide seamless services to a client or group of clients.

To acquire a better understanding of why a cluster is needed and how it provides the same reliability as a mainframe environment, several areas of concern must be addressed: fault tolerance, performance, scalability, and reliability.

1.1 Fault Tolerance

Fault tolerance is the ability of a system to respond to failure in a way that does not hinder the service offering provided by the server. This is often referred to as "graceful" response because it eliminates the failure gracefully, without interrupting service to users. Fault tolerance theory was applied to several other industries before it was ever used within a technology setting.

Consider the airline industry. If an airline did not have fault tolerance built into its airplanes, one small failure could bring a plane crashing to the ground. To avoid such catastrophe, airlines have eliminated every SPOF that they can. Most commercial airplanes have not one engine, but several, all of which are capable of keeping the plane in the air if one or more engines should stop working.

The average home computer user does not have fault tolerance built into his or her system. What happens if the power supply—the component that provides power to the machine—fails? The machine does not function anymore. What if the hard drive stops working? The user will likely lose not only the ability to run his or her computer, but also the data resident on the hard drive. Computers built for the consumer market have several SPOFs and typically do not provide fault tolerance. This is because of cost and the uptime factor. The machine you run at home does not provide services to other machines that may need to 24/7 access it, and therefore the cost of the fault tolerance is outweighed by the home user's need for inexpensive computing.

Many people run backup software to eliminate the risk of losing data, which is good practice, but backup software is a disaster recovery option, not a substitute for fault tolerance. Disaster recovery is a means of restoring your computer and all your data in the event of a failure, but it does not prevent the failure or assist the computer in functioning in spite of the failure.

In a corporate environment, fault tolerance is a bit more important, not necessarily for the workstations in use by the company's employees, but for the servers that contain the company's data. In most companies today, business cannot operate without the data stored on the server. How many times have you called a merchant to hear, "I would love to help you with that, but our systems are down right now." Companies have developed a dependence upon technology to the point that they cannot operate without the "system."

To ensure that downtime is minimized, network engineers apply fault tolerance theory to their systems and applications. They ask the question, If *blank* were to fail, would my system still run and provide service to users? That blank could be anything that assists in providing an application or service to the company's users. For instance, if the power were to fail, would the system still run and provide service to users? Unless your system is battery-powered or protected by an uninterruptible power supply (UPS) system, then the answer is no, and power becomes an SPOF for your system.

Fault tolerance theory should be applied not only to the environmental issues such as power and cabling, but also to the machine or server itself. Within the server, you have several points of failure: the power supply, the hard drive(s), the network card, the motherboard, the memory, and so on. A failure in any one of those areas can cause your system not to function. Hardware manufacturers have begun to provide fault tolerance within their architectures, often building redundancy in the power supply or network card, and systems such as *RAID* (Redundant Array of Inexpensive Disks) have been developed to provide fault tolerance to as many parts of the machine as possible. However, even in the most advanced systems built today, redundancy for all of the components is not available. If your server has redundant power supplies, redundant network cards, redundant hard disks, and redundant processors, and the machine experiences a motherboard failure, the system still goes down.

So, although every machine you run that provides business-critical functionality should have every form of redundancy available, even the most advanced hardware/software combinations don't offer complete redundancy and fault tolerance within a single machine.

1.2 Performance

Within any system, performance is always important. The performance of your personal computer in some ways dictates how fast you can get your work done. Servers are not any different, with the exception of user quantity. Personal computers typically have only one person to support, whereas servers have to support thousands, sometimes millions, of users. The ability of a machine to respond to a request within a certain period of time is greatly affected by the performance of the machine. Many different items make up the quality of performance.

1.2.1 CPU

The CPU—central processing unit—has often been referred to as the "brain" of the computer, and rightly so, for all information that is processed on a computer is controlled by the CPU. On a mainframe system, the CPU would often span several circuit boards, but in personal computers and servers, the CPU is housed within a small chip on the motherboard, called a *microprocessor*.

The CPU consists of two components: the arithmetic logical unit (ALU) and the control unit. The ALU is responsible for arithmetic and all logical operations performed on the machine, while the control unit is responsible for the extraction, decoding, and execution of instructions stored in memory as well as for calling on the ALU when necessary for computation.

Note

Remember, even though we use the decimal system and text to communicate, the computer uses only binary—a system of language that uses only 0's and 1's to communicate. The CPU plays a vital role in coding and decoding the 0's and 1's into a language that users can understand.

Within any computer, the CPU is what dictates how fast the machine can process information. The CPU must process all information that is sent to the screen (as well as a lot that is not). A slow processor will be evident to the user. Even more so, a computer that serves users on the Internet or within a large organization will suffer performance loss due to a slow processor.

Most of the servers that are manufactured today can now be equipped with dual processors: two processors within the same machine sharing the processing load. Some hardware manufacturers are producing machines that run between 4 and 64 processors within the same machine. As these types of machines, as they become more affordable, will assist systems administrators in eliminating the processor as a possible bottleneck within a system.

1.2.2 Memory

The memory within a machine, also called RAM (random access memory), is responsible for short-term storage for programs and applications. A computer uses RAM to cache working data. In the same way that a human brain has a short-term memory and long-term memory, a computer has RAM, which stores data temporarily, and a hard drive, which stores data for extensive periods. While a hard drive is nonvolatile memory, meaning that it retains data regardless of the power state; RAM is volatile memory, for it loses all data stored when a machine is powered off.

The importance of memory cannot be underestimated. When a PC or server loads a program or operating system, the files needed to run that application are loaded into memory. If there is not enough memory to load the applications and files being used, the system will be forced to store the excess data on the hard drive, where access will be significantly slower. To ensure optimal performance, assess the memory needs of your application prior to purchasing hardware and ensure that you have adequate memory not only for the application you want to run, but for the operating system and services as well.

1.2.2.1 Memory Types

There are two basic types of RAM, SRAM (static RAM), and DRAM (dynamic RAM). SRAM is memory that remembers its content, while DRAM needs to be refreshed every few milliseconds. While SRAM is more efficient, it is also more expensive and tends to be used for L2 cache rather than for system memory.

Because SRAM is much more difficult to produce, the industry has standardized with DRAM. DRAM comes in many flavors:

- FPM—fast page mode
- ECC—error-correcting code
- EDO—extended data output
- SDRAM—synchronous dynamic RAM

FPM RAM was the first type of RAM introduced to Pentium computing and came in SIMM modules that had to be loaded in pairs. The transfer type was 70ns and 60ns. Shortly after its introduction, FPM RAM was replaced with EDO RAM, a faster (60ns or 50ns), more reliable memory architecture. Because of the demand for fault tolerance, ECC RAM was created. ECC RAM provides error correcting within the RAM itself. ECC was much more reliable than its predecessors and became the standard for PC-based servers due to its parity-based error checking and recovery. Recently, SDRAM has become the most common memory in use due to the fast response times of 8ns to 12ns. SDRAM also integrates the error-checking technology of ECC RAM, making it the smartest choice for PCs and servers alike.

1.2.2.2 Virtual Memory

Although a PC or server may have a significant amount of RAM, there will always be times when the machine needs more memory than it has. To overcome this limitation in some of the first personal computers ever developed, engineers designed the operating system to use what is called virtual memory. By using virtual memory, an operating system sets aside a small portion of the hard drive to be used like RAM, so that if the machine needs to work with more information than can be stored in RAM, it will use the hard drive space as a "virtual" memory space. This type of process is often referred to as paging or accessing the page file, for the space used for virtual memory is actually a file stored on the computer's hard drive.

Virtual memory allows for computers that have a memory limitation to run programs they may otherwise not be able to run; however, there is a drawback. Reading and writing data to the hard drive is a much slower process than reading and writing data to RAM and is therefore a performance hindrance in many cases.

For most servers, a performance monitor program will detail the amount of paging taking place on a server. Paging should be minimal. If paging is consistently high, it is typically indicative of a lack of RAM within the machine.

1.2.3 Disks

The function of the hard disk should be central to any discussion of performance. If the CPU in a computer is the brain, and the RAM is the short-term memory, then the hard disk is the long-term memory. The ability to recall information that may not have been accessed for a long period of time is core to any computer system.

A hard disk—or hard drive—is a magnetic stack of flat metal platters that store information in contiguous sectors. A small arm, or spindle, moves back and forth across the disk to read and write information to the disk. As mentioned before, the hard disk, unlike RAM, does not lose its data when the computer is powered off. The data that is stored is permanent until it is deleted or overwritten.

The performance of the disk is related to the ability of the access arm to move back and forth across the disk and obtain information. There are two types of hard disks that are common in the PC market: SCSI disks and IDE disks. IDE stands for Integrated Device Electronics and is the standard mass-storage device interface for PCs. IDE allows for two devices to be connected to each interface, typically configured in a master/slave role. The transfer speeds used to be limited to 16 mbps (megabits per second), but recently have been enhanced to provide 33 mbps (ATA/33) in most machines and up to 100 mbps (ATA/100) in new models.

SCSI stands for Small Computer System Interface. SCSI, pronounced *skuzzy,* is a mass-storage device interface that is widely used within high-end computing environments. This architecture provides up to 160 mbps transfer rates. SCSI is unique because, unlike IDE, SCSI supports up to seven devices on one interface. Devices attached to a SCSI bus are daisy chained, using a special cable so they can all be controlled and maintained by the same controller. Because they are on the same controller, multiple disks can be written to and read from simultaneously. This makes SCSI a faster platform than IDE and more suitable to the server market. The difference, however, is in the cost. SCSI devices tend to be more expensive that IDE devices.

As you can see, the type of hard drive you use for a system can greatly impact the performance.

1.2.4 Network

Among all of the items that allow a computer to send and receive information, the network connectivity is the most important. The fastest computer in the world is of no value if it cannot serve its clients. It is vital that the speed of the network that a computer is connected to be fast enough to allow the processing within a machine to operate freely, without hindrance. Too often, individuals invest significant amounts of money into hardware and then use second-rate network standards to connect it to the network.

One of the very first LAN standards was called Token-Ring and ran at 4 mbps. This technology, pioneered by IBM, was on the forefront of networking and dictated that traffic flowing through a network should follow a ring pattern. This technology implemented a virtual ring that was passed from computer to computer on the network. When a computer had the token, it could transmit and receive all messages destined for other computers on the network. When the computer did not have the token, it remained silent. This created a highly fault-tolerant networking platform. Later, the Token-Ring specification was redesigned to operate at 16 mbps, which offered high-speed, fault-tolerant data transmission and a vast improvement over the 4 mbps standard. Unfortunately, strict patenting laws, driven by IBM, made the technology expensive to implement.

Shortly after the development of Token-Ring networking, another standard called Ethernet was being developed. Ethernet was a joint effort by Xerox, Digital Equipment Corporation, and Intel. Ethernet uses a bus or star topology and supports data transfer rates from 10 mbps to 1000 mbps. Unlike Token-Ring's token-passing technology, Ethernet provides a network where all machines can broadcast their information at any time. When more than one machine speaks at once, it causes a collision, which in turn causes all the machines on the network to "timeout" for a specific period of time before retransmitting. This technology, on the surface, may seem inferior to token-passing technologies because of the constant collisions, but Ethernet, due to its inexpensive cost and nonproprietary development, has far surpassed the installed base of Token-Ring networks.

From a performance standpoint, network speed should never be the bottleneck in the deployment of any system. Fast-changing technology horizons have ensured that high bandwidth remain inexpensive and easy to maintain. Technologies such as ATM, Gigabit Ethernet, and FDDI are all examples of high-speed networking options that can provide a system with high connection speeds and thus improved performance.

1.3 Scalability

Scalability is the ability of a system or service to adapt to increased demand. Scalability is essential to any system or service that may need to grow to meet the needs of a particular client base.

To understand scalability, a simple analogy is helpful. If you were to purchase a three–bedroom, two-bath home, and the following year you added one new family member, you would need a house that is scalable—meaning that you need a house that can be easily modified to meet the needs of a larger family. Adding a bedroom to the home would "scale" the house to meet the increased need. If you could not add a bedroom due to zoning restrictions or architectural issues, your home would be unable to scale. The only solution then would be to sell your home and buy another one that is larger.

In a networking environment, the same rings true. A network manager does not want to replace all of the networking equipment in the company to meet the needs of an additional fifty employees. It is more desirable to add the necessary equipment to the existing network so that the network will scale to meet the needs of the additional employees. Scalability is essential, not only for the cost savings, but for the ability to add additional capacity while minimizing downtime.

Consider the personal computer. When a person buys a personal computer from a computer store, they typically ask the question, Is it upgradeable? What they are actually asking is whether the computer will scale to meet their future needs. Most systems that you purchase will allow you to upgrade the memory, upgrade the video card, and in some cases, upgrade the processor. But some machines are built to reduce cost, not provide scalability. These machines usually have constraints that don't allow you to upgrade the memory or processor. These machines are cost effective, but can only be useful for about one to two years, whereas a machine that is scalable can be useful for about three to five years.

Systems that are designed for use as Internet servers or intranet servers have the same scalability issues, but also the added issue of what is often referred to as dynamic scalability. Dynamic scalability is the ability of a system or service to adapt to an increased demand within a short span of time. This span of time can be day-to-day, hour-to-hour, or minute-to-minute.

Many companies use Internet servers as a way to survey public opinion. Consider a company that decides to advertise a new Web site for the first time during a commercial break on Super Bowl Sunday. Once the commercial is aired, there will be literally millions of people hitting the site in response to the advertising. And seeing that the commercial most likely cost several million dollars, the owner of the site does not want to lose a single connection to the site. This scenario demands that the Internet server scale from a zero amount of traffic to one million plus hits worth of traffic within a matter of minutes. The ability for this system to handle the traffic is the system's dynamic scalability.

1.4 Reliability

Reliability is the ability of a system to provide a service without interruption. Although related to fault tolerance, performance, and scalability, reliability is the ability to provide continual service without regard to the amount of use or failure within the system. Reliability is related closely to availability, for the availability of a system proves its reliability.

Using the earlier example of the automobile, the reliability of a new automobile is expected. When the key is turned, the engine should turn over. A racecar driver knows the necessity of reliability. Mechanics go through the racecar in detail before the driver is allowed to race. The failure of one part within the vehicle could cause the driver to lose the race—or to lose his or her life.

Similarly, to ensure reliability, a systems designer must inspect every part of the system. A failure within any part of the system could cause the application or service to be unavailable to requests, proving its unreliability. To build reliability into a system, the designer must analyze all of the points previously discussed:

- Fault tolerance of the system
- Performance of the system
- Scalability of the system

Failure within a hard drive or network card could cause the server to quit servicing clients. Too many users could cause the system to respond slowly to requests, or to quit responding all together. If the application cannot scale to meet the needs of the clients as those needs change from day to day, the system could experience hours or even days of downtime caused by frequent hardware upgrades or replacements.

1.5 Designing Reliable Data Access

Throughout the corporate world, systems and services provide a communications and data storage channel that assists companies in generating revenue. The loss of productivity due to systems failures numbers in the billions of dollars per year. Client/server systems of the past have been powerless to reduce downtime, but things are changing. Now, more than ever, there are products and services available to keep systems up and running in spite of component failure, power failures, or user error, and although no software or hardware product can guarantee a zero percent failure rate, today's technologies can "virtually" eliminate hardware and software failures.

There are two scenarios that cause a system or service to be unavailable.

1. A component that is part of the system, or a dependency of the system, malfunctions, leaving the application or service (or path to the application or service) in an unusable or unavailable state.
2. The network load on an application or service is so great that the server cannot respond to client requests in a timely fashion.

To (virtually) eliminate these two events from taking place, a systems engineer or designer must seek to design systems that apply the following methodology:

1. Downtime due to failures within components of the system or components that the system is dependent upon does not affect the operation of the application or service being provided.
2. The application or service must respond in a timely manner to all requests, regardless of network load or requests being processed.

Note

It is important to note that no system can guarantee 100 percent uptime. There are many products and services on the market that tout figures such as 99.9995 percent uptime, but these are statistics typically taken from a controlled environment with minimal load on the server(s). Realistically, an engineer should never hope to achieve 100 percent uptime, but should strive to minimize unnecessary downtime due to component failures or software upgrades. The focus should always be the quality and reliability of service provided to the client, not the amount of uptime.

Let's take a brief look at how a designer might approach the construction of a reliable data access solution. To approach the design correctly, the designer must seek to provide the following:

- Fault tolerance
- High performance
- Scalability
- Reliability

The questions a designer must ask include, What are the needs of my application? Does the application need to operate 24 hours a day, 7 days a week? Where will the server physically be placed, and how will clients access it? Does the application need to be accessed by clients on the Internet? These questions help the designer discern what connectivity must be engineered to allow clients to access the application.

The designer continues the planning by envisioning the completed system and systematically tracing the connectivity from the client to the server, listing each component that is encountered as a client might access the system. Assuming this is an Internet application, let's start with the typical Internet client.

The client machine itself has many points of failure, but because they are beyond the control of the systems designer, they are ignored for the moment. The client machine is connected to the Internet through a dial-up line or a network connection of some sort. This is a critical point of failure, but again, beyond the control of the designer. The client makes a request for our application through an ISP (Internet service provider). The request travels across the Internet (through many routers and data lines) until it reaches the designer's Internet connection. At that point the request will most likely hit a Channel Service Unit/Data Service Unit (CSU/DSU), then a router, which will pass the request to a firewall. The firewall will pass the request through a cable to a switch or hub, which will pass the request on to the server, which is also connected with a network cable. The server will access the application by reading data from the hard drive or memory and send the desired information back to the client through the same route.

If you count all of the devices between the client and the server, you will easily see that there are many points of failure that could prevent the application from being accessible. Some of these items are not within the control of the designer, but many of them are. Take a look at the items that are within the control of the designer:

- The Internet circuit
- The DSU/CSU
- The router
- The firewall
- The switch or hub
- The server network card
- The server power supply
- The server hard drive
- The server memory
- All cables used to attach all network devices

All of the items listed can be controlled by the designer and must be evaluated to determine optimal fault tolerance. The failure of any one of these items will cause the application or service to be unavailable. Some of the items typically cannot be made fault-tolerant without great expense, but each item should be scrutinized to see what can be done to boost the reliability of the system. Table 1-1 lists each item and some of the fault tolerance options that are typically used.

Beyond fault tolerance, a designer must also consider the performance and scalability of the system. Hardware and software must be chosen that will provide high performance. The performance of a system is determined by usage. If you are in a controlled environment where the number of users is constant, it can be easy to determine performance requirements, but if you are building an Internet application, determining the performance requirements can be a bit tricky. A good plan is to always overestimate your need. If you believe the servers within your system need 256 MB of RAM to operate efficiently, buy 512 MB of RAM for each server. Always leave plenty of room. Don't just meet the requirements—exceed them. This will allow room for future growth.

Table 1-1 Fault Tolerance Options

Device/Item	Fault Tolerance Options
Internet Circuit	A secondary, low-bandwidth circuit could be installed to direct traffic through the event of a failure on the primary circuit.
DSU/CSU	DSU/CSU can be purchased with redundant power supplies. It is also a good idea to keep one DSU/CSU on the shelf in case you need to swap them.
Router	Routers can be purchased with redundant power supplies. Just as with the DSU/CSU, it is a good idea to have an extra router, configured exactly like the one you are using, just in case you need to swap it.
Firewall	There are different types of firewalls, so make sure your firewall is reliable and can handle the traffic. Many firewalls also have redundant power supplies and failover functionality.
Switch/Hub	Switches and hubs can be purchased to provide redundant power supplies and service.
Server Network Card	Within the server, you can use two network cards or you can build server clusters.
Power Supply	Most devices can be purchased with redundant power supplies. Also, make sure all power supplies use UPS.
Hard Drive	RAID 5 controllers can be purchased to make hard disks more fault-tolerant.
Memory	Clustering is the only answer to memory failure.
Cabling	Make sure all cabling in use is certified for the data that will travel on it.

1.6 Summary

As part of any enterprise application implementation, a systems designer must provide the applications and services in a way that is fault-tolerant and reliable. As computing has moved from mainframe-based systems to client/server systems, applications have suffered large blows to performance and reliability. The introduction of the cluster server seeks to change that. Already, cluster server software vendors are providing low-cost software options that allow systems engineers to rival mainframe reliability with client/server architectures.

As you will see, clustered and load balanced solutions meet the needs of fault tolerance by providing failover for systems and services that are critical to the enterprise. They meet the needs of performance and scalability by providing a system that can grow as needed, without having to replace hardware. They meet the needs of reliability and availability by providing a platform that can be upgraded, modified, and maintained without an interruption of service.