Chapter 3

Old Meets New: Distributed Computing

In This Chapter

- Taking a look at distributed computing through the years
- Exploring the elements of distributed computing
- Putting distributed computing together with hardware and software advancements

Distributed computing is not a new technology concept. It has been around for almost 50 years. Initially, the technology was used in computer science research as a way to scale computing tasks and attack complex problems without the expense of massive computing systems. One of the most successful early endeavors into distributed computing was a project funded by the U.S. Defense Advanced Research Project Agency (DARPA). The result of the organization's research was the development of the Internet, the first distributed computing network. You might say that it initiated a revolution that has led to a transformation of everything from commerce to healthcare, to transportation, and to human-to-human and machine-to-machine communications. In this chapter, we explain what distributed computing is and describe why it is the foundation for big data.

A Brief History of Distributed Computing

Behind all the most important trends over the past decade, including service orientation, cloud computing, virtualization, and big data, is a foundational technology called *distributed computing.* Simply put, without distributing computing, none of these advancements would be possible. Distributed computing is a technique that allows individual computers to be networked together across geographical areas as though they were a single environment, as shown in Figure 3-1. You find many different implementations of distributed

computing. In some topologies, individual computing entities simply pass messages to each other. In other situations, a distributed computing environment may share resources ranging from memory to networks and storage. All distributed computing models have a common attribute: They are a group of networked computers that work together to execute a workload or process.

Giving thanks to DARPA

The most well-known distributed computing model, the Internet, is the foundation for everything from e-commerce to cloud computing to service management and virtualization. The Internet was conceived as a research project funded by the U.S. DARPA. It was designed to create an interconnecting networking system that would support noncommercial, collaborate research among scientists. In the early days of the Internet, these computers were often connected by telephone lines (see Figure 3-1)! Unless you experienced that frustration, you can only imagine how slow and fragile those connections were.

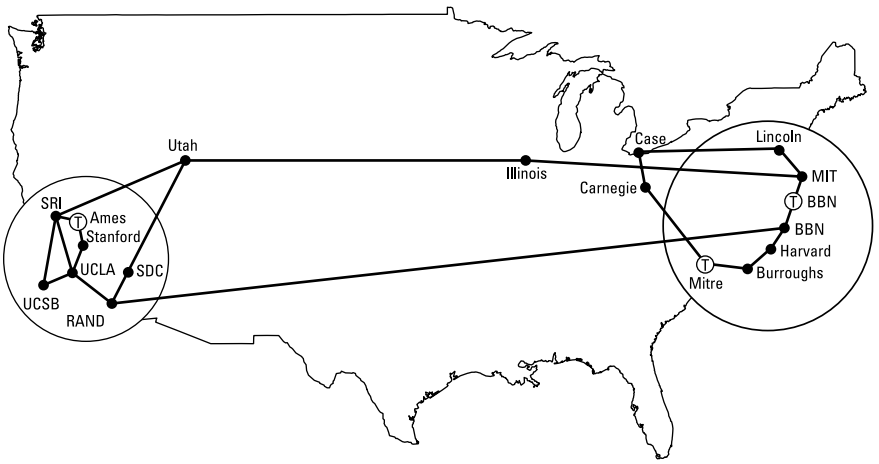

Figure 3-1: The early Internet connectivity model.

As the technology matured over the next decade, common protocols such as Transmission Control Protocol (TCP) helped to proliferate the technology and the network. When the Internet Protocol (IP) was added, the project moved from a closed network for a collection of scientists to a potentially commercial platform to transfer e-mail across the globe. Throughout the 1980s, new Internet-based services began to spring up in the market as a commercial alternative to the DARPA network. In 1992, the U.S. Congress

passed the Scientific and Advanced-Technology Act that for the first time, allowed commercial use of this powerful networking technology. With its continued explosive growth, the Internet is truly a global distributed network and remains the best example of the power of distributed computing.

The value of a consistent model

What difference did this DARPA-led effort make in the movement to distributed computing? Before the commercialization of the Internet, there were hundreds of companies and organizations creating a software infrastructure intended to provide a common platform to support a highly distributed computing environment. However, each vendor or standards organization came up with its own remote procedures calls (RPCs) that all customers, commercial software developers, and partners would have to adopt and support. RPC is a primitive mechanism used to send work to a remote computer and usually requires waiting for the remote work to complete before other work can continue.

With vendors implementing proprietary RPCs, it became impractical to imagine that any one company would be able to create a universal standard for distributed computing. By the mid-1990s, the Internet protocols replaced these primitive approaches and became the foundation for what is distributed computing today. After this was settled, the uses of this approach to networked computing began to flourish. Today, we take it for granted that we can create a network of loosely coupled computers that can exchange information and communicate at the right speed at the right time, as shown in Figure 3-2.

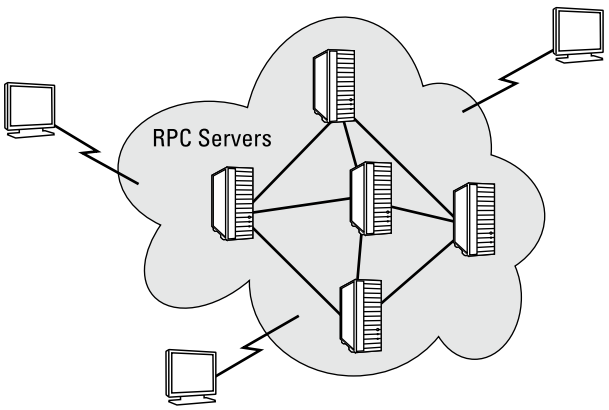

Figure 3-2: An example of how a Remote Procedure Call has been used in early distributed computing techniques.

Understanding the Basics of Distributed Computing

There isn't a single distributed computing model because computing resources can be distributed in many ways. For example, you can distribute a set of programs on the same physical server and use messaging services to enable them to communicate and pass information. It is also possible to have many different systems or servers, each with its own memory, that can work together to solve one problem.

Why we need distributed computing for big data

Not all problems require distributed computing. If a big time constraint doesn't exist, complex processing can done via a specialized service remotely. When companies needed to do complex data analysis, IT would move data to an external service or entity where lots of spare resources were available for processing. It wasn't that companies wanted to wait to get the results they needed; it just wasn't economically feasible to buy enough computing resources to handle these emerging requirements. In many situations, organizations would capture only selections of data rather than try to capture all the data because of costs. Analysts wanted all the data but had to settle for snapshots, hoping that they captured the right data at the right time.

Key hardware and software breakthroughs revolutionized the data management industry. First, innovation and demand increased the power and decreased the price of hardware. New software emerged that understood how to take advantage of this hardware by automating processes like load balancing and optimization across a huge cluster of nodes. The software included built-in rules that understood that certain workloads required a certain performance level. The software treated all the nodes as though they were simply one big pool of computing, storage, and networking assets, and moved processes to another node without interruption if a node failed, using the technology of virtualization. Chapter 5 covers virtualization and big data in more detail.

The changing economics of computing

Fast-forward and a lot has changed. Over the last several years, the cost to purchase computing and storage resources has decreased dramatically. Aided by virtualization, commodity servers that could be clustered and blades that

could be networked in a rack changed the economics of computing. This change coincided with innovation in software automation solutions that dramatically improved the manageability of these systems. The capability to leverage distributed computing and parallel processing techniques dramatically transformed the landscape and dramatically reduce latency. There are special cases, such as High Frequency Trading (HFT), in which low latency can only be achieved by physically locating servers in a single location.

The problem with latency

One of the perennial problems with managing data — especially large quantities of data — has been the impact of latency. *Latency* is the delay within a system based on delays in execution of a task. Latency is an issue in every aspect of computing, including communications, data management, system performance, and more. If you have ever used a wireless phone, you have experienced latency firsthand. It is the delay in the transmissions between you and your caller. At times, latency has little impact on customer satisfaction, such as if companies need to analyze results behind the scenes to plan for a new product release. This probably doesn't require instant response or access. However, the closer that response is to a customer at the time of decision, the more that latency matters.

Distributed computing and parallel processing techniques can make a significant difference in the latency experienced by customers, suppliers, and partners. Many big data applications are dependent on low latency because of the big data requirements for speed and the volume and variety of the data. It may not be possible to construct a big data application in a high latency environment if high performance is needed. The need to verify the data in near real time can also be impacted by latency. In Chapter 16, we address the issue of real-time data streaming and complex event processing, which are critical to applications of big data. When you are dealing with real-time data, a high level of latency means the difference between success and failure.

Demand meets solutions

The growth of the Internet as a platform for everything from commerce to medicine transformed the demand for a new generation of data management. In the late 1990s, engine and Internet companies like Google, Yahoo!, and Amazon.com were able to expand their business models, leveraging inexpensive hardware for computing and storage. Next, these companies needed a new generation of software technologies that would allow them to monetize the huge amounts of data they were capturing from customers. These companies could not wait for results of analytic processing. They needed the capability to process and analyze this data in near real time.

Getting Performance Right

Just having a faster computer isn't enough to ensure the right level of performance to handle big data. You need to be able to distribute components of your big data service across a series of nodes. See Figure 3-3. In distributed computing, a *node* is an element contained within a cluster of systems or within a rack. A node typically includes CPU, memory, and some kind of disk. However, a node can also be a blade CPU and memory that rely on nearby storage within a rack.

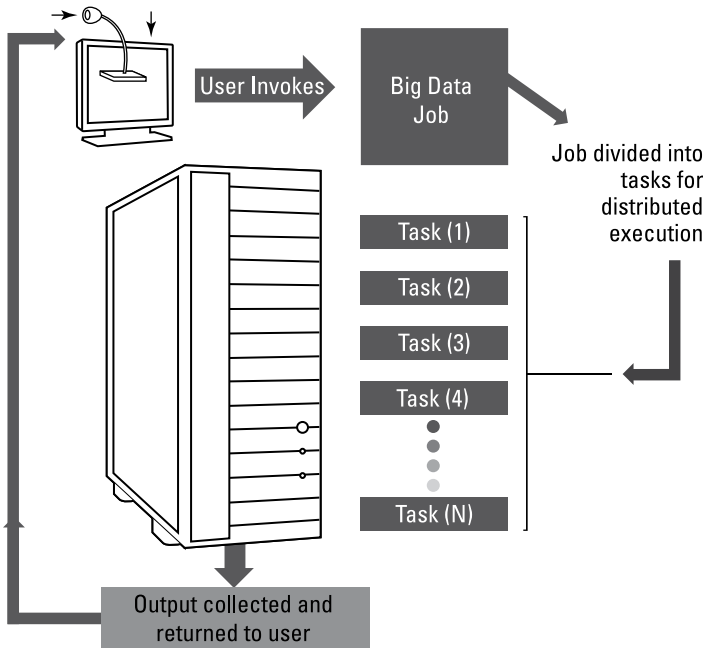

Figure 3-3: An evolution of distributed computing relies on jobs being divided for distributed execution of tasks.

Within a big data environment, these nodes are typically clustered together to provide scale. For example, you might start out with a big data analysis and continue to add more data sources. To accommodate the growth, an organization simply adds more nodes into a cluster so that it can scale out to accommodate growing requirements. However, it isn't enough to simply expand the number of nodes in the cluster. Rather, it is important to be able to send part of the big data analysis to different physical environments. Where you send these tasks and how you manage them makes the difference between success and failure.

In some complex situations, you may want to execute many different algorithms in parallel, even within the same cluster, to achieve the speed of analysis required. Why would you execute different big data algorithms in parallel within the same rack? The closer together the distributions of functions are, the faster they can execute. Although it is possible to distribute big data analysis across networks to take advantage of available capacity, you must do this type of distribution based on requirements for performance. In some situations, the speed of processing takes a back seat. However, in other situations, getting results fast is the requirement. In this situation, you want to make sure that the networking functions are in close proximity to each other. In general, the big data environment has to be optimized for the type of analytics task.

Therefore, scalability is the lynchpin of making big data operate successfully. Although it would be theoretically possible to operate a big data environment within a single large environment, it is not practical. To understand the needs for scalability in big data, one only has to look at cloud scalability and understand both the requirements and the approach. Like cloud computing, big data requires the inclusion of fast networks and inexpensive clusters of hardware that can be combined in racks to increase performance. These clusters are supported by software automation that enables dynamic scaling and load balancing.

The design and implementations of MapReduce are excellent examples of how distributed computing can make big data operationally visible and affordable. For more information on MapReduce, refer to Chapter 8. In essence, we are at one of the unique turning points in computing where technology concepts come together at the right time to solve the right problems. Combining distributed computing, improved hardware systems, and practical solutions such as MapReduce and Hadoop is changing data management in profound ways.