CHAPTER 1

Evolution of the Data Center

The need for consolidation in the data center didn't just occur overnight; we have been building up to it for a long time. In this chapter, we review the evolution of today's data center and explain how we have managed to create the complex information technology (IT) environments that we typically see today.

This chapter presents the following topics:

- "Consolidation Defined" on page 1
- "History of the Data Center" on page 2
- "Complexity in the Data Center" on page 6

Consolidation Defined

According to *Webster's College Dictionary*, consolidation is the act of bringing together separate parts into a single or unified whole. In the data center, consolidation can be thought of as a way to reduce or minimize complexity. If you can reduce the number of devices you have to manage, and if you can reduce the number of ways you manage them, your data center infrastructure will be simpler. With a simpler infrastructure, you should be able to manage your data center more effectively and more consistently, thereby reducing the cost of managing the data center and reducing your total cost of ownership (TCO).

When we first started working on consolidation methodologies in 1997, we focused on server and application consolidation; the goal was to run more than one application in a single instance of the operating system (OS). Since then, the scope has widened to the point that virtually everything in the corporate IT environment is now a candidate for consolidation, including servers, desktops, applications, storage, networks, and processes.

History of the Data Center

Over the last 40 years, the data center has gone through a tremendous evolution. It really wasn't that long ago that computers didn't exist. To better understand how we got to a point where consolidation has become necessary, it's worth taking a look at the evolution of today's computing environment.

The following sections address the role mainframes, minicomputers, and distributed computing systems have played in the evolution of the data center in a historical context. However, it is important to note that many of the qualities mentioned affect the choices IT architects make today. While mainframes are still the first choice of many large corporations for running very large, mission-critical applications, the flexibility and affordability of other options have undoubtedly altered the design and functionality of data centers of the future.

The Role of Mainframes

Mainframes were the first computers to gain wide acceptance in commercial areas. Unlike today, when IBM is the sole remaining mainframe vendor, there were several mainframe manufacturers. Because IBM has always been dominant in that arena, the major players were known as IBM and the BUNCH (Burroughs, Univac, NCR, Control Data, and Honeywell). These major players dominated the commercial-computing market for many years, and were the data processing mainstay for virtually all major U.S. companies.

The Strength of Mainframes

The strengths of mainframes make them valuable components to nearly every large-scale data center. These strengths include:

- **Power.** For many years, mainframes were the most powerful computers available, and each new generation got bigger and faster. While the power and performance of distributed computing systems have improved dramatically over the past several years, mainframes still play an important role in some data centers.
- **High utilization rates.** Because of the expense involved in purchasing mainframes and building data centers to house them, mainframe users tend to use every bit of available computing power. It's not uncommon to find mainframes with peak utilization rates of over 90 percent.
- **Running multiple applications through workload management.** Because of the large investment required to purchase mainframes, it is important for companies to able to run multiple applications on a single machine.

To support multiple applications on a single system, mainframe vendors, especially IBM, developed the concept of workload management. Through workload management, you can partition a mainframe and allocate its computing resources such that each application is guaranteed a specific set of resources. This ability allows corporate IT departments to provide their customers with very high application availability and very high service levels.

There is no doubt that mainframes are today's champions of workload management. This isn't surprising since this capability has been evolving over the last 30 years. For example, you can expect a fully implemented, highly evolved workload-management system to manage:

- Central processing unit (CPU) usage
- Dispatch priority
- Storage used
- Input/output (I/O) priority

Some workload managers have end-to-end management functions that monitor what is happening in the application and in the database, and that balance transaction workloads across multiple application and database regions.

- **Well-defined processes and procedures.** Because of their size, as well as their high cost, mainframes are run in data centers where specific processes and procedures can be used for their management. The IT environments that house mainframes are generally highly centralized, making it fairly easy to develop very focused policies and procedures. As a result, audits of mainframe environments usually show highly disciplined computing environments—a quality that further contributes to the mainframe's ability to deliver high service levels.

The Problem With Mainframes

While mainframes provide the power and speed customers need, there are some problems with using them. These problems include:

- **Financial expense.** The biggest drawback of using mainframes is the expense involved in purchasing, setting up, and maintaining them. When you exceed the capacity of a mainframe and have to buy another, your capital budget takes a big hit. For many years, mainframe manufacturers provided the only computing alternative available, so they priced their hardware, software, and services accordingly. The fact that there was competition helped somewhat, but because vendors had their own proprietary OSs and architectures, once you chose one and began implementing business-critical applications, you were locked in.
- **Limited creative license.** In addition to their high cost, the inflexible nature of the processes and procedures used to manage mainframe environments sometimes limits the methods developers use to develop and deploy applications.

- **Increased time-to-market.** Historically, the length of mainframe development queues was measured in years. In this environment, the ability of a business to change its applications or to deploy applications to meet new market needs may be severely limited.

As a result of the preceding characteristics, and as new alternatives have been made available, many businesses have moved towards faster and cheaper platforms to deliver new applications.

The Introduction of Minicomputers

During the 1970s and 1980s, minicomputers (minis) became an attractive alternative to mainframes. They were much smaller than mainframes, and were much less expensive. Designed as scientific and engineering computers, minis were adapted to run business applications. The major players in this market were DEC, HP, Data General, and Prime.

Initially, companies developed applications on minis because it gave them more freedom than they had in the mainframe environment. The rules and processes used in this environment were typically more flexible than those in the mainframe environment, giving developers freedom to be more creative when writing applications. In many ways, minis were the first step towards freedom from mainframe computing.

While this new found freedom was welcomed by many, minis had two significant deficiencies. First, because minis were small and inexpensive, and didn't need specialized environments, they often showed up in offices or engineering labs rather than in traditional data centers. Because of this informal dispersion of computing assets, the disciplines of mainframe data centers were usually absent. With each computer being managed the way its owner chose to manage it, a lack of accepted policies and procedures often led to a somewhat chaotic environment. Further, because each mini vendor had its own proprietary OS, programs written for one vendor's mini were difficult to port to another mini. In most cases, changing vendors meant rewriting applications for the new OS. This lack of application portability was a major factor in the demise of the mini.

The Rise of Distributed Computing

After minis, came the world of distributed systems. As early users of UNIX™ systems moved out of undergraduate and postgraduate labs and into the corporate world, they wanted to take the computing freedom of their labs into the commercial world, and as they did, the commercial environment that they moved into evolved into today's distributed computing environment.

One important characteristic of the distributed computing environment was that all of the major OSs were available on small, low-cost servers. This feature meant that it was easy for various corporate groups (departments, work groups, etc.) to purchase servers outside the control of the traditional, centralized IT environment. As a result, applications often just appeared without following any of the standard development processes. Engineers programmed applications on their desktop workstations and used them for what later proved to be mission-critical or revenue-sensitive purposes. As they shared applications with others in their departments, their workstations became servers that served many people.

While this distributed environment provided great freedom of computing, it was also a major cause of the complexity that has led to today's major trend towards consolidation.

UNIX Operating System

During the late 1960s, programmers at AT&T's Bell Laboratories released the first version of the UNIX OS. It was programmed in assembly language on a DEC PDP-7. As more people began using it, they wanted to be able to run their programs on other computers, so in 1973, they rewrote UNIX in C. That meant that programs written on one computer could be moved easily to another computer. Soon, many vendors offered computers with the UNIX OS. This was the start of the modern distributed computing architecture.

Although the concept of portable UNIX programs was an attractive one, each vendor enhanced their own versions of UNIX with varying and diverging features. As a result, UNIX quickly became Balkanized into multiple incompatible OSs. In the world of commercial computing, Sun became the first of today's major vendors to introduce a version of UNIX with the SunOS™ system in 1982. Hewlett-Packard followed soon thereafter with HP-UX. IBM didn't introduce their first release of AIX until 1986.

Although Linux and Windows NT are growing in popularity in the data center, UNIX remains the most common and most highly developed of these OSs. It is the only major OS to adequately support multiple applications in a single instance of the OS. Workload management is possible on UNIX systems. Although they are not yet in the mainframe class, the UNIX system’s current workload management features provide adequate support for consolidation.

Complexity in the Data Center

All of this freedom to design systems and develop applications any way you want has been beneficial in that it has allowed applications to be developed and released very quickly, keeping time-to-market very short. While this can be a tremendous competitive advantage in today's business environment, it comes at a substantial cost. As applications become more mission-critical, and as desktop servers move into formal data centers, the number of servers in a data center grows, making the job of managing this disparate environment increasingly complex. Lower service levels and higher service level costs usually result from increased complexity. Remember, as complexity grows, so does the cost of managing it.

The organizational structures that are typically imposed on those who make the business decisions that affect data centers and those who manage data centers further add to this complexity. In most of the IT environments we deal with, multiple vertical entities control the budgets for developing applications and for funding the purchase of the servers to run them, while a single centralized IT operations group manages and maintains the applications and servers used by all of the vertical entities. This organization is found in nearly every industry including, but not limited to:

- Commercial companies: Business units, product lines, departments
- Government: Departments, agencies
- Military: Service, division, military base
- Academic: Department, professor, grant funds

In this type of environment, vertical entities have seemingly limitless freedom in how they develop and deploy applications and servers. Further, operations groups often have little or no control over the systems they manage or over the methods they use to manage them. For these reasons, it is very common for each application-server combination to be implemented and managed differently, and for a data center to lack the operational discipline found in most mainframe environments.

In these environments, IT operations staff tend to manage systems reactively. If something breaks, it gets fixed. They spend their time managing what has already happened rather than managing to prevent problems. Because of this, the IT operations people are the ones who feel the pain caused by this complexity, and they are usually the primary drivers of a consolidation project.

The following section explains the causes and effects of server sprawl on your data center.

Causes and Effects of Server Sprawl

The most frequent complaint we hear from Sun customers is that they have too many servers to manage, and that the problem is getting worse. Each new server adds complexity to their environments, and there is no relief in sight.

In the distributed computing environment, it is common for applications to be developed following a one-application-to-one-server model. Because funding for application development comes from vertical business units, and they insist on having their applications on their own servers, each time an application is put into production, another server is added. The problem created by this approach is significant because the one-application-to-one-server model is really a misnomer. In reality, each new application generally requires the addition of at least three new servers, and often requires more as follows:

- **Development servers.** The cardinal rule that you should not develop applications on the server you use for production creates a need for a separate development server for each new application. This guideline increases the number of servers required per application to two.
- **Test servers.** Once your application is coded, you need to test it before it goes into production. At a minimum, this requires you to unit test the application. If the application will interact with other applications, you must also perform integration testing. This action results in at least one, and possibly two, additional servers for the testing process. Because many developers insist on testing in an environment that is as close to the production environment as possible, this condition often results in large, fully configured test servers with large attached storage and databases. The server population has now grown to three or four servers.
- **Training servers.** If a new application will be used by lots of people, you may need to conduct training classes. This condition usually results in another server, so now we're up to four or five servers.
- **Multitier servers.** Many applications are developed using an n-tier architecture. In an n-tier architecture, various components of the application are separated and run on specialized servers; therefore, we frequently see a separate presentation tier, business tier, and resource tier. This architecture exacerbates server sprawl and adds to the complexity of the IT environment.
- **Cluster and disaster recovery servers.** If an application is deemed to be mission-critical, it may require a clustered environment, requiring one more server. If an application is extremely mission-critical, for example, like many of those in the financial district of New York City, it will require a disaster recovery site that allows for failover to the backup site. These requirements have the potential to add one or two more servers.

Now you can see how a single new application adds at least seven new servers to a data center. This configuration is why we see customers with several thousand servers.

To fully understand how this type of server sprawl adds complexity to a data center, you must also recognize that each time you add another server to your environment, you are also adding:

- Additional data storage that has to be managed and backed up
- Additional networking requirements
- Additional security requirements

Probably the largest impact of server sprawl is the complexity that results from the methods used to manage the environment. In many distributed computing environments, we find that there are as many different ways to manage servers as there are system administrators. This is where the lack of discipline found in data centers really stands out. If you can somehow take charge of this complexity, you can eliminate much of it, and simplify your job.

The following chapters explain how you can sell and implement a consolidation project as a method for reducing complexity and its negative effects.

Summary

This chapter provided a definition of consolidation, and explained how data centers have evolved to a point where consolidation has become necessary. In addition, it explained the causes and effects of complexity in today's IT environment. In general, with complexity comes increased costs, decreased service levels, and decreased availability. Consolidation seeks to reverse this trend. It is a movement towards higher service levels and lower service level costs. This goal is the reason consolidation has been a hot topic for several years, and it is the reason today's economic environment has accelerated the move to consolidate not just servers, but everything in the IT environment.

As we dig deeper into consolidation in the following chapters, it's important to remember that the reason for consolidation is really very simple:

- If you consolidate such that you reduce the number of devices you have to manage, and if you reduce the number of ways you manage them, you can reduce the complexity of your environment.
- If you reduce the complexity of your environment, you can increase the efficiency of your infrastructure.
- If you increase the efficiency of your infrastructure, you increase service levels and availability, and you lower your TCO.