

## DATA COMMUNICATIONS MANAGEMENT

# IMPROVING DATA CENTER PERFORMANCE

Peter J. Manca

## INSIDE

Physical Complexity; Lengthy Deployment Cycles; Multiple Failure Points; Proprietary Architectures; The Closed Source Bottleneck; The Open Source Alternative; High Availability (HA); Cluster Types; Clustering Tools; Clustering Best Practices; Scalability; Vertical Scalability; Horizontal Scalability; Over-Provisioning; Solutions; Storage Area Networks; Processing Area Networks; PAN Benefits

The rapidly increasing use of technology to gain competitive advantage has led to unprecedented data center expansion. For both enterprises and service providers, the result has been tremendous pressure on the IT function — including a host of problems associated with deploying and managing the servers needed to support applications. Obstacles range from physical provisioning (power, networking, storage), to software infrastructure (enterprise management, maintenance), to advanced solutions (high availability, load balancing, clustering).

This sharp upturn in processing demand has been driven by several reinforcing factors, including:

- *Application complexity.* Almost as certain as death and taxes is the apparently immutable law that applications get more complex over time, with commensurately higher computing requirements.
- *More users and transactions.* The number of people and enterprises connected to applications has reached a level unimaginable just a decade ago. Likewise, the potential for problems has escalated. External networking means unprecedented volatility in load, with daily and weekly fluctuations of five or ten, to one now being commonplace. In response, many organizations have adopted the brute-force approach of over-provisioning for the highest predicted load as an inelegant solution to the problem.

**PAYOFF IDEA**

There have been profound changes in the operation of data centers. These changes are the result of a redefinition of these centers, from back-office operations to core production facilities for many knowledge-based industries. As a consequence, there is tremendous pressure on the IT function. This article proposes a rethinking of the server architecture as a solution to these challenges.

- *Increasingly layered and disaggregated architecture.* The standards underlying networked applications tend toward a complex, layered architecture that manifests itself as a disaggregated, distributed environment. Originally intended to minimize software and system intricacies, separating components onto different execution environments has increased data center complexity and exposed fundamental server limitations.
- *OS limitations.* Despite continued progress in operating system (OS) technology, experience has led users to isolate applications on separate servers for increased reliability and manageability.

The net effect on data centers has been profound. At the heart of the shift is a redefinition of the data center from a back-office operation to the core production facility for many knowledge-based industries. This transition has been most noticeable in financial services, with its long-standing use of electronic trading, but is also underway in other segments as rapid, barrier-free transfer and sharing of data has become a key competitive differentiator. While the sheer size of data centers has increased steadily, the number of servers has grown exponentially. Today, a large facility will house thousands of servers and tens of terabytes of storage, with the largest having more than 10,000 servers across multiple sites.

### **A REAL-LIFE SCENARIO**

To understand the difficulties associated with today's server architectures in a dynamic data center environment, one can examine the experience of an affinity portal company that provides Web hosting for clients. Not surprisingly, this company chose a proprietary UNIX platform to service a major customer. The company built a portal based on a three-tier model with multiple Web servers, application servers, and database servers residing on separate machines. In addition, it replicated the hardware for load balancing and high availability.

The complexity in building this portal stemmed from its physical provisioning — the racking of servers, routers, and firewalls as well as complex cabling for networking, keyboard, video, and mouse. In addition, scaling the system was nontrivial. When performance problems arose, the company added processors to each tier because it was unable to identify the exact source of the bottleneck. This required a series of forklift upgrades because, like most servers, the processing capacity on the UNIX machines could not be increased. The result was application downtime, which led to lost revenue for the client. Because of the difficulty in scaling and tuning its servers, the company ultimately lost its flagship account.

This article explores some of the major issues facing today's enterprise-class data centers as illustrated by the preceding example —

including physical complexity, proprietary architectures, high availability, scalability, and over-provisioning — and proposes a rethinking of the server architecture as a solution to these challenges.

### **CHALLENGE 1: PHYSICAL COMPLEXITY**

As applications and workloads require increasingly large, functionally partitioned, readily scalable infrastructures, the overall complexity of the environment rises rapidly. While the sheer number of servers deployed today is extraordinary, this does not begin to adequately capture the physical and logical demands of the environment — from the number of software images and physical disks to be managed, to the seemingly mundane but critical issues of cabling.

A single enterprise server requires power, connectivity to storage (increasingly via Fibre Channel to a SAN [storage area network]), and connectivity to the IP network (often via Gigabit Ethernet). In addition, it may require a separate management bus and KVM connections. For a truly reliable infrastructure, these must be in redundant pairs. If the server belongs to a cluster, it may also require a redundant heartbeat and cluster-management connection, and potentially redundant connections to dual-ported storage. The net effect is that a rack of 42 servers could theoretically require as many as 250 cables for an optimized installation. In reality, this is impossible to manage. Thus, real-world deployments are considerably less dense and less reliable than they should be.

As this example and [Exhibit 1](#) illustrate, the physical installation of systems, especially the cabling, is not just “an exercise left for the reader.” Cabling is a major stumbling block to rapid deployment and a common cause of failure, especially in connection with moves, adds, and changes (MAC). Sources in major enterprise data centers indicate that cabling can add days to an already long deployment cycle.

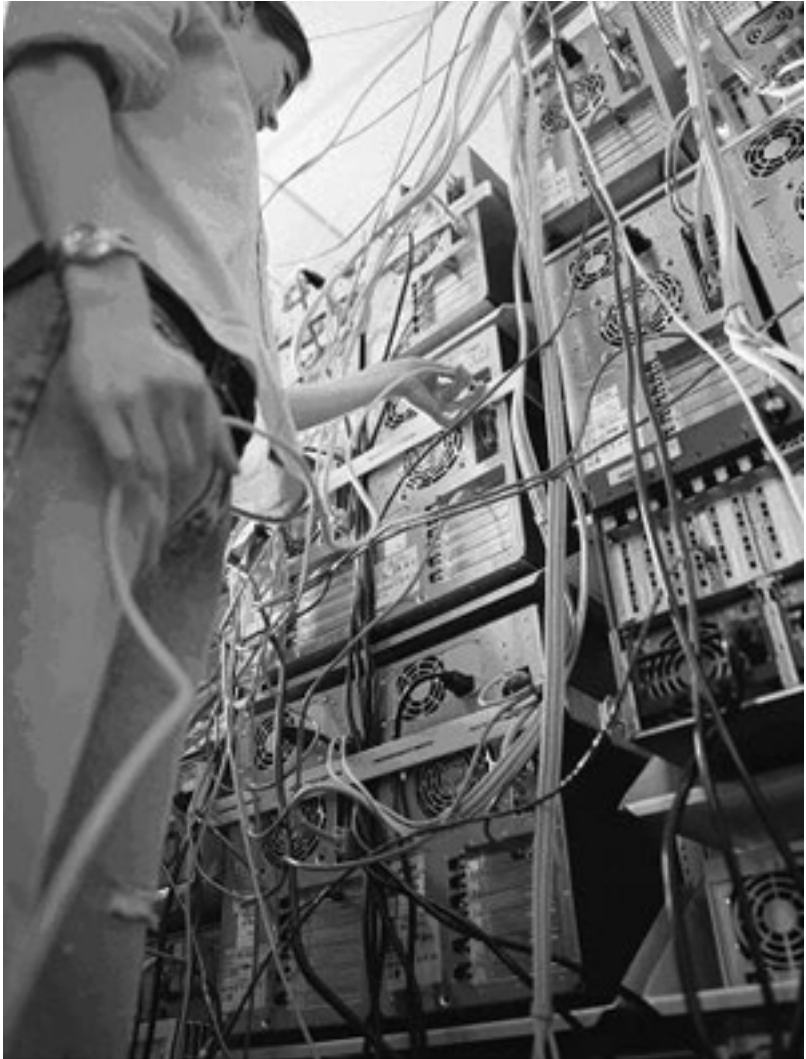
Vendors have responded on several fronts to the problems associated with cabling and physical complexity, including the recent move toward partitioning large, expensive SMPs into logical collections of smaller systems. While partitioning relieves some deployment complexity, it requires inherently more expensive systems and does not offer the same flexibility as a collection of separate and less tightly coupled servers.

Other piecemeal responses to cabling problems include hot-plug PCI chassis, shared networked KVM switches, daisy-chained management cables and rack-integrated Ethernet switches. However, none addresses the total needs of the modern data center.

### **Lengthy Deployment Cycles**

Because of this physical complexity, server deployment takes far more time than the amount of work would seem to warrant. Generally, the

---

**EXHIBIT 1** — Cabling Complexity

cause is organizational rather than technical. The deployment of a server typically involves multiple organizations, with separate groups responsible for provisioning storage management, network services, power, and, in some cases, security and systems management. Each organization has its own service level agreement, scheduling, and workload management. Deployment may also involve coordinating outside vendors such as storage service providers.

### **Multiple Failure Points**

Another unanticipated side effect of server proliferation has been excessive failure points. Storage, network, power, management, and KVM cabling are all physical points of failure. Paradoxically, duplicating them for high availability introduces additional failure points. In fact, failover clusters can be so complex that unless properly managed they actually *lower* the reliability of the overall environment. Change management is also a challenge because each physical or logical reconfiguration of the system can cause problems. Data center managers must constantly balance the fact that the most reliable system is the one seldom touched against requirements for rapid configuration changes and constant growth.

An ideal system would minimize the number of cables and connections required, and permit reconfiguration with virtually no physical intervention.

### **CHALLENGE 2: PROPRIETARY ARCHITECTURES**

The operating system chosen for the data center must provide ease-of-use, dependable operation, support for a wide range of commercial applications, a clear migration path (to protect the initial investment), and a non-proprietary approach (to avoid vendor lock-in).

### **The Closed Source Bottleneck**

Historically, computers have been developed for Microsoft® Windows® or a proprietary UNIX variant such as Sun Solaris or IBM AIX. The reason was simple: independent software vendor (ISV) support. In fact, the control these operating systems have had on application development and availability was virtually unbreakable, limiting the potential for useful evolution in any hardware. Furthermore, gaining access to source code and being able to implement changes typically required months of negotiating with the operating system supplier. Even then, developers were usually limited in the modifications they could make without breaking compatibility. For these and other reasons, closed source operating systems fall short in the delivery of new architectures. This simple limitation has possibly been the single most important reason for the lack of evolution in computer hardware.

### **The Open Source Alternative**

Fortunately, the arrival of Linux — an open source operating system — has changed the design equation to foster innovation. In fact, the impact of open source on enabling the development of new computing architectures is just now becoming apparent. The key is the controlled nature of creativity in the Linux community. A well-defined structure designed to

protect programming interfaces from frivolous changes enables rapid evolution of the computing stack while preserving the integrity of applications. Peer review by a cadre of the world's best engineers provides clarity, with well-designed, well-written, well-documented code winning out over other contenders. Thus, when Linux moves from technology release to commercial distribution, the code base is solid, stable, and reliable. This characteristic, shared by no other operating system, makes Linux adoption in the enterprise a winning proposition for IT professionals.

Likewise, the openness of Linux benefits users by creating leverage over vendors. Traditionally, once a company adopted a specific hardware platform, it was locked into the supplier's proprietary operating system. Ensuring compatibility and access to applications then meant buying ever-more costly OS upgrades. On the other hand, Linux, which can be purchased from a number of commercial suppliers or, for that matter, received gratis, eliminates vendor control. Customers now are free to choose products that best fit their needs and to choose vendors that provide the best products. This competitive "free-market" approach is also inspiring hardware providers to improve service responsiveness, enhance product quality, and lower costs to attract and retain customers.

In addition, Linux has enjoyed a rapidly increasing level of ISV support, including nearly universal availability on major Web server applications such as Apache, AOLserver, iServer, iPlanet, Roxen, Stronghold, and Zeus. Likewise, middleware applications such as ColdFusion, Total-e, WebLogic, and Websphere — as well as leading database products (including DB2, Informix, Ingres II, Oracle, and Sybase) — have been well received. Indeed, Linux can support the entire three-tier application infrastructure.

### **CHALLENGE 3: HIGH AVAILABILITY (HA)**

Traditionally, HA has been reserved for only the most mission-critical applications, such as those associated with revenue generation. However, as the networked business environment has evolved, so has the definition of mission-critical. Today, network elements such as firewalls and DNS servers, corporate mail systems, customer-facing CRM applications and other secondary systems are deemed mission-critical — with a corresponding demand for a more reliable underlying infrastructure. The industry has responded with various high-availability options, ranging from reliable storage to clusters to fault-tolerant systems.

In reality, most enterprise environments need a flexible repertoire of failover solutions, ranging from none (a perfectly reasonable alternative for some services), through simple failover, through complex nearest-neighbor and N+1 topologies. Unfortunately, clustering solutions are presently complicated and expensive, in part due to storage-management

dependencies, and seldom offer the complete range of failover modalities required.

### **Cluster Types**

Essentially, a cluster is a group of systems organized to share resources. While the shared resource in a cluster is typically processing capacity, cluster solutions dating back to the Digital VAXCluster were designed to share all resources equally and distribute loads as needed. Clusters should permit the easy addition or removal of a resource, provide a degree of high-speed communication among members, and understand some level of distributed resource ownership.

Clusters fall into three categories — high-performance computing (HPC), high availability, and load balancing — each with specific strengths to address a specific problem. With high availability clusters, the emphasis is on complete avoidance of planned and unplanned downtime. Thus, HA systems are designed around rigorous standards for redundancy with no single point of failure. The decision to implement HA clusters is generally business-driven; for example, if a brokerage firm is not processing stock transactions, it is not making money.

High availability clusters can be *stateful* or *stateless*. In a stateful model, a transaction in progress during a failure is not lost, although it may be rolled back and reissued. In a stateless environment, the transaction may be lost and the user required to reissue the request. For example, during an E-commerce transaction, a failed link involved in issuing an order may require user resubmission if the system fails. However, the exchange operation (e.g., credit card processing) cannot be lost or issued more than once. This combination most often compels enterprises to implement the more rigorous stateful approach, which is best achieved by combining a stateless hardware architecture with application software designed for statefulness.

### **Clustering Tools**

As clustering models and requirements have grown more complex, tools have been introduced to mitigate the complexity. Data centers should evaluate the various cluster-capable software and hardware systems against their specific needs.

For example, open source cluster products are built (or derived from proprietary solutions) to let users modify the source code. As the demand for Linux systems in the enterprise has continued to grow, commercial clustering packages have also appeared. In some cases, these commercial offerings are derived from portions of open source code; in others, the code is developed from scratch or is a part of proprietary code. The trade-offs between commercial closed codes and open source codes revolve around support, viability, responsiveness, and platform

availability. However, these lines have become blurred in that most open source programs have commercially available support while retaining their roots.

### **Clustering Best Practices**

In choosing a clustering solution, some basic questions should be asked. How large a cluster is required? What is the setup process? How will the cluster be monitored and tuned? What impact will clustering have on the business model? Recommendations from Giga Information Group and others on best practices for clustering reinforce the complexities and sensitivities of clustered environments:

- *Use only cluster-certified configurations.* It is important to cluster only those configurations of hardware and software that are certified by the vendor to be cluster-safe. Not all models and combinations, even from vendors that support clustering well, are safe to cluster. Attempting to cluster other configurations is an open invitation to failure and disavowal of vendor support. In the world of Intel®-based servers, this is especially important because the basic quality of implementation varies widely both across vendors and within a vendor's product line from low-end to high-end systems. Cluster-certified systems will have higher-quality engineering, and have undergone a certification process that includes qualifying the specific revision number and driver versions of all third-party supplied modules such as network interface cards and storage. This sensitivity to configurations can erase many of the hoped-for advantages of using industry-standard servers because the cluster-certified systems, in addition to being higher-quality, are also more expensive.
- *Use vendor clustering professional services for legacy servers.* Especially if you are new to clusters or do not have a mature internal staff, vendor-supplied professional services are likely to be required to install the systems, configure failover scripts, and test failover and fail-back. In addition to scheduling, the use of professional services has another drawback — it is expensive. With clusters taking tens of hours to set up and hundreds of hours per year to manage, the cost of provisioning has limited the penetration of HA to less than 10 percent of enterprise applications, while underlying demand is much higher.
- *Choose the right topology.* The cleanest and easiest form of failover clustering, active-passive failover, is unfortunately the most capital intensive, requiring an identical passive standby system for each active system. It is a telling commentary on the state of clustering technology that active-passive is still the *de facto* standard for database clusters. Active-active failover clusters, which allow one to run

simultaneous loads on two servers and fail one over to the other, are attractive in theory but very difficult to implement with conventional technology. Nearest-neighbor and other N+1 topologies require decisions about where to migrate the workload and what to do with the workload on neighbor systems. N+1 failover is, in general, the most complex to implement with legacy solutions, and in most cases the potential cost benefits versus simple active-passive failover are diminished by the higher cost of setup and management.

- *Plan for workload prioritization in the event of a failover.* Even with properly implemented active-active or N+1 topologies, one must address the issue of workload prioritization, because the total amount of processing capacity may be different in the failover configuration than in the base environment. This requires careful planning and constant monitoring, regardless of the selected failover topology.

An ideal system for the enterprise data center, either for traditional applications or for Web-resident E-business applications, would offer tightly integrated failover capabilities with flexible topologies, allowing the user to select the appropriate level of redundancy. In addition, it would simplify the process of configuring and managing clusters.

#### **CHALLENGE 4: SCALABILITY**

As experience in building E-business infrastructures has accumulated, so has awareness of the seemingly contradictory requirements for success: ensuring high availability in an increasingly complex environment while rapidly scaling the infrastructure to accommodate ever-changing load conditions.

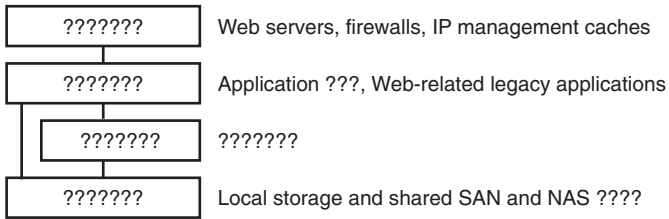
By considering the functional layers of the architecture in light of their own requirements, site architects can begin to define standard elements and practices for crafting resilient and scalable environments. The basic technical architecture for many data center applications is a layered set of services as shown in [Exhibit 2](#).

Defining a physical infrastructure based on a layered service architecture allows one to consider each layer as a separate entity, with its own rules for scaling and availability. The layers are connected via IP networking, and often will have security gateways (firewalls) between them. The resulting architecture provides separate arenas within which to apply the concepts of vertical and horizontal scalability.

#### **Vertical Scalability**

Vertical scaling of a service is the addition of resources into a single instance of the resource that provides the service. In the case of a Web server or transaction back end, it is making the server more powerful.

**EXHIBIT 2** — Basic Web Infrastructure Framework  
(Source: Giga Information Group.)



The usual ways to accomplish this are either to upgrade the server to a faster single CPU or add CPUs to an SMP.

Vertical scaling of a resource, where it can be done, is usually the most efficient way to scale, cost and reliability notwithstanding. It does not add complexity to the system and, with today's OS technology, is usually totally transparent to the application. This can be both a blessing and a curse; a blessing because it makes almost any thread-rich software run faster, and a curse because it may hide bad software architecture until the system is much larger and because it contributes nothing to enhancing availability.

Nonetheless, the pure scalability to be gained from simply throwing a larger system at the problem is impressive, albeit nonlinear. According to the SPECweb96 benchmark, the performance delta between a uniprocessor Intel system and an eight-way SMP is approximately one to four. Its successor, and much improved benchmark, SPECweb99, does not yet have enough data to reliably establish SMP scalability. On the commonly quoted TPC-C benchmarks, the performance of the largest single-image systems within a given vendor architecture is anywhere from ten to thirty times the performance of single-CPU systems.

These advantages have led data centers to deploy large SMPs and run them as partitioned systems, with multiple copies of the OS, to create a number of readily scalable virtual servers in a tightly coupled environment. While a step in the right direction, this approach suffers from several problems.

First, while very efficient architecturally, large SMPs are more expensive than the same resources spread across a number of smaller systems due to the increased complexity of the engineering needed to support multiple CPUs with cache coherency, larger memories, and the generally higher standard of reliability required for a large shared resource. Where they are being used appropriately, as large single-image engines for heavy back-end transaction processing, they are impressive resources, providing mainframe-level capabilities at substantially lower cost. However, when misapplied in an attempt to compensate for the physical complexity of a

multi-server deployment, they may work adequately but at an extreme price penalty. Moreover, large SMPs usually have limitations in configurability related to I/O and peripherals, and almost always come with some amount of local storage per partition, which vastly complicates clustering and storage management in general.

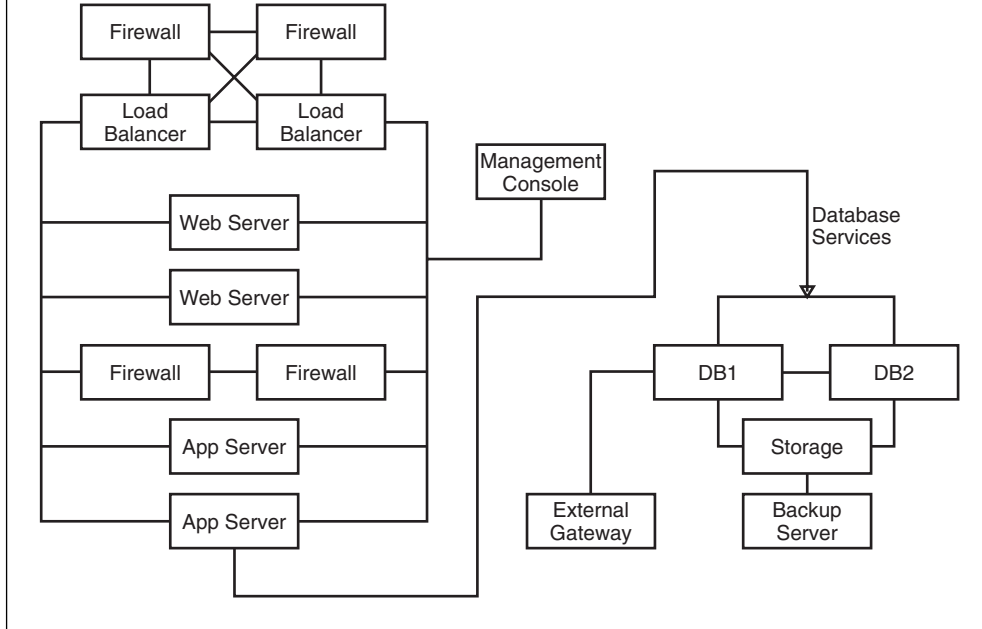
Thus, when we consider cost and flexibility, the decision to scale a service vertically becomes less straightforward and must reflect whether the service truly requires a large single image or can be spread horizontally across multiple, smaller independent platforms. While some functions, such as database images for online transaction processing and data-warehousing applications, still require a large single image, most data center requirements for SMP servers are for four-way and smaller systems.

### **Horizontal Scalability**

In particular, FTP (File Transfer Protocol), Web servers, and most firewalls and cache servers can be scaled horizontally using smaller systems operating in parallel. Through an external load-balancing device or locally resident software, this horizontal scaling creates a federated cluster of independent servers with no knowledge of one another at the OS level, making their administration and management simpler. They are, in effect, forming an application-specific cluster under control of a dedicated application — either the Web load balancer or the cache server's embedded failover and load-balancing facility — which does not intrude on the low-level management of the system as would an OS-level cluster such as that required for database failover.

This approach, however, presents its own challenges, primarily in the areas of storage management, network management, and physical cabling. To understand the roots of this complexity, consider the environment shown in [Exhibit 3](#). This diagram, despite its omission of directory services and all of the details of the internal routers and switches, gives some sense of the complexity of modern Web sites. Additional services and segmentation of the workload add to the complexity. It is not unusual for large Web sites to have more than 50 servers, and the largest sites have several hundred. This implementation, modeled after a mid-tier online catalog retailer architecture, is designed to be capable of supporting approximately 2000 simultaneous sessions with a pair of four-way RISC UNIX systems as application servers and similar systems as database servers from IBM, HP, Sun, or an Intel-based Linux server vendor. The application servers, with separate images executing on each machine, manage their own state information for failover. Despite the fact that it omits critical elements such as directory services, and assumes that the personalization services can run on the same servers as the primary application logic, this diagram begins to reveal the complexity inherent in even simple data center environments.

**EXHIBIT 3** — Infrastructure Complexity (Source: Giga Information Group.)



Thus, the issues surrounding physical provisioning are not resolved merely by establishing a cluster. While it may be easier to add a node to a cluster than to provision an entirely new, larger server to accommodate an increasing workload, the data center still has the daunting task of ordering the incremental machine; scheduling power, networking, and system-management personnel to set it up; and stabilizing the environment. While clustering allows data centers to more readily apply computing power where it is needed, today's clustering products neither migrate capacity to meet geographical concerns or application-specific needs, nor do they support secure virtual partitioning into dedicated machines. It is still a one-type-fits-all model.

These limitations manifest themselves in two ways. First, while clusters scale upward as servers are added, control over clustered resources may not scale as easily. Sharing disks, networks, and even printers is an exercise best left unconsidered. In fact, most clusters have an upper node limit, particularly high-availability systems wherein each machine must be aware of all others. Second, while clusters may scale as a whole, they lack the management flexibility to be partitioned into virtual servers with secure resources. This stumbling block is linked to a lack of trust, inasmuch as business units prefer not to share resources.

Finally, today's clusters do not offer adequate management flexibility to meet the needs of current or predicted usage. Consider a global E-commerce application. Each geographic region has unique needs related to data and will experience peak activities that require sizing to burst levels, generally while other regions are not utilizing their compute resources. A cluster solution should automatically reconfigure to meet the burst needs of each region using either a time schedule or a rules-based configuration.

### **CHALLENGE 5: OVER-PROVISIONING**

As data centers have moved toward the disaggregated infrastructure associated with distributed applications, they have suffered progressive declines in resource utilization. That is because ensuring responsiveness to business requirements has meant sizing systems to meet peak demand, thus leaving expensive processing cycles idle most of the time. Today, typical CPU utilization across a large organization is in the 19 to 23 percent range for non-mainframe servers, primarily because there has been no practical way to leverage unused resources during off-peak periods. Sizing to peak also means sizing to burst; because application access is bursty in nature, demand is not a smooth average yet is still somewhat predictable. Unfortunately, the limited flexibility of current server technology for allocating resources means that sizing to peak/burst means over-provisioning the hardware, increasing costs for equipment, space, and power.

As an example, consider some recent data, shown in [Exhibit 4](#), profiling five different E-business sites. With behavior like this, in many cases the best the IT staff can do is to over-provision, often by a multiple of four or more, against the load spikes. This does no harm in a technical sense but it is wasteful of capital budgets. A highly desirable attribute of a data center system would be the ability to rapidly add and subtract capacity from a group of resources while the system was running. To date, while vendors advertise the ability to do this with various load-balancing and clustering options, these solutions work only under some fairly limited conditions:

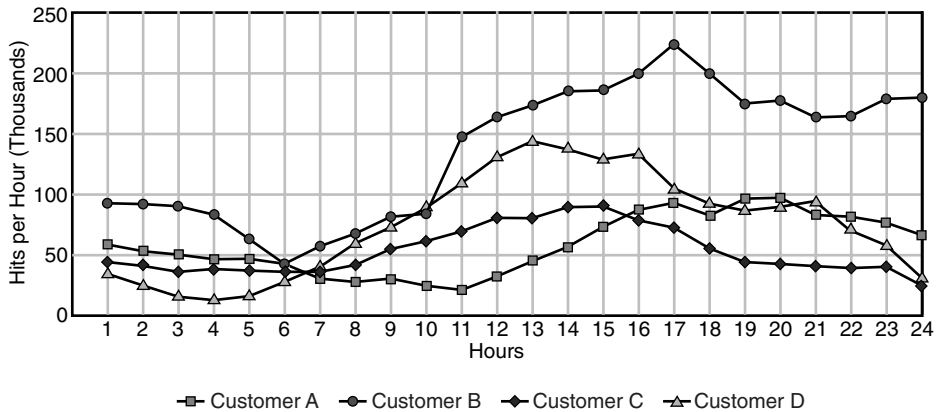
- The systems must be completely installed and cabled, with IP addresses assigned.
- Users must be responsible for keeping the systems in exact sync with respect to installed software, file system contents, etc.
- The clusters themselves, while improving, require significant upfront time to install and configure. A typical simple failover cluster requires 40 to 80 hours or more of time to set up and a significant investment in management time to maintain.

Vendors have also responded with COD solutions, which allow users to install capacity in anticipation of future needs, then activate and pay for it on demand. Unfortunately, these solutions come with significant drawbacks, most notably requiring the customer to buy a large UNIX SMP and use the COD capacity either to expand a large single image or add partitions to the machine. Although these large SMPs are always very expensive compared to a group of independent two- or four-way servers, they are used in this manner because, until recently, there was no other way to add small servers to an application on demand.

Another solution is server partitioning, wherein servers are designated to run services based on load rather than assigning one or multiple machines to a single service. An example is the traditional tiered software model comprised of client, middleware, and database services. It is not unusual to find the middleware service combined with one side or the other, an approach that reduces flexibility while increasing management complexity. Yet another approach to partitioning is sharing a large system among multiple business units or companies. However, the mere concept of sharing inspires IT professionals to reach for the aspirin bottle. In a word, the issue is *security*. Today, securing partitioned applications on a single server is not well understood and, without investing in a “secure OS,” fairly nonexistent.

A more common method is server consolidation. The server farm, with its racks of disparate systems, begins to resolve the sizing issue but significantly raises the level of pain inflicted on systems administrators. In fact, managing the server farm may be the single greatest challenge faced

**EXHIBIT 4** — Web Site Loads (Source: IBM.)



by data centers. Moreover, with costs for data center real estate increasing rapidly, cramming more capacity into less space is thought to be crucial. Yet density alone is not enough. Instead, server consolidation must result in “usable density,” which not only addresses the space issue but also considers management and application requirements.

## **SOLUTIONS**

The challenges of physical complexity, proprietary architectures, high availability, scalability, and over-provisioning faced by IT professionals will only increase as the transition to online business continues, with no letup forecast in the pace of infrastructure buildout over the next five years. To meet the expectations of both internal and external clients, data center managers must invest in technologies that enable them to add processing capacity without disrupting applications, and to easily allocate and reallocate resources as required. Fortunately, solutions are beginning to emerge that will provide data centers with the server, storage, and networking infrastructures needed to meet the demands of the new millennium.

### **Storage Area Networks**

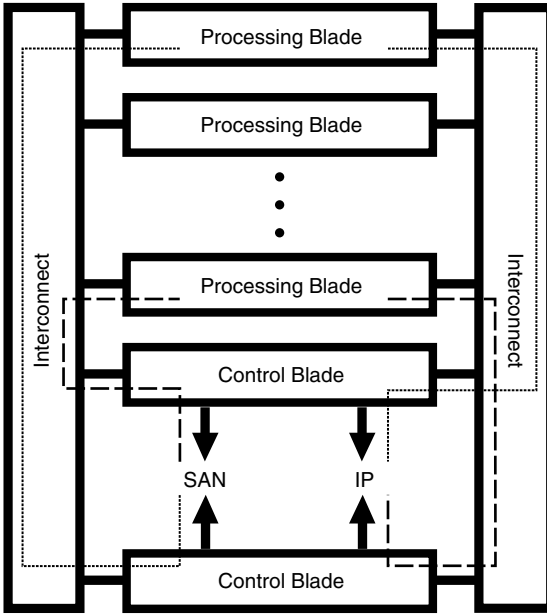
The past decade has seen the emergence of the storage area network (SAN) as a way to centralize, consolidate and virtualize the deployment and management of storage resources. This architecture was developed to address a set of complex problems that arose as storage moved from a central point (the mainframe-based data center) to a distributed approach based on the growth of departmental systems and applications.

Processing resources have the same inherent problems as storage resources when many discrete servers are deployed. As the number of servers and their local storage multiply and become organizationally and geographically dispersed, the cost and complexity of implementing, managing, and utilizing processing capacity increase dramatically. In a typical large data center, it can take eight to 12 weeks to deploy a new server with standard applications, while processor utilization may be as low as 15 to 20 percent.

### **Processing Area Networks**

Solving the major data center pain points requires rethinking the computing architecture. This architecture, called a processing area network (PAN), is intended to address the issues inherent in deploying and managing large numbers of servers and to leverage the increasing prevalence of SAN and network attached storage (NAS).

The PAN is synergistic with a physical bladed server architecture. Its fundamental premise — a network of diskless systems — has been tried before, but never with the integrated management needed for a commercial

**EXHIBIT 5** — Processing Area Network (PAN)

environment. Essentially, the PAN does for computing what the SAN did for storage — it transforms the provisioning of processing power from a complex physical process to a simple virtual process.

As shown in [Exhibit 5](#), a PAN pools together multiple processors in a redundant network via an integrated high-speed switch, which in turn is connected to external storage and IP network resources through consolidated connections. Significantly, the processing resources in a PAN are diskless, stateless, and anonymous — thus enabling flexible allocation and failover because one can easily take over the characteristics and configuration of another. At the same time, legacy server components such as Ethernet NICs (network interface cards), Fibre Channel HBAs, local disks, and external cables are consolidated or eliminated.

Instead, the PAN replaces error-prone physical components with virtualized switches and interfaces that allow users to network processors without the restrictions of conventional hardware. For example, an administrator creates the equivalent of a legacy server by associating an individual, stateless blade with disk and network resources. Users can also establish multiple logical PANs (LPANs) to allocate distinct, secure resources to enterprise divisions or individual customers. Moreover, the PAN architecture consolidates redundant storage and networking connections for the

equivalent of 24 servers into as few as four cables, while built-in clustering eliminates third-party software and duplicate hardware for provisioning high-availability services and applications.

All physical and logical resources are managed entirely through integrated management software, resulting in cost savings, unprecedented flexibility, and high system availability. Through a standard Web-browser interface, virtualized servers are provisioned, virtualized networks are created, high availability is configured, secure LPANs are established, and resources are reallocated on-the-fly as needs change.

### **PAN Benefits**

The implementation of a PAN increases the utilization of processing capacity, simplifies and speeds the installation and deployment of servers and applications, and simplifies and lowers the cost of high availability and server management. Combined with a physical bladed design, the PAN architecture offers users:

- *Efficient storage management.* PAN management software actively discovers connected disk resources and controls the mapping of disks to the PAN's stateless processing blades. The PAN can work with a full SAN fabric or provide many of the efficiency and manageability advantages of a SAN to simple JBOD storage.
- *Simplified physical installation.* Cable consolidation is inherent in the PAN. Because all I/O, both network and storage, is consolidated, cabling is based on the right balance of I/O resources and bandwidth for the computing complex, rather than on a fixed, mechanically defined number of cables per enclosure. A flexible, user-defined mix of Fibre Channel SAN, Gigabit Ethernet, and conventional Ethernet can be configured independently of processing resources. Additionally, blades are hot-pluggable and achieve power, network, and storage connections through a single blind-mate connector. Installing and provisioning a new server becomes a matter of minutes instead of hours or weeks.
- *Simplified HA provisioning.* With no persistent user data on its blades, the PAN can deliver reliable, integrated HA capabilities ranging from failover and reboot of a failed blade, to service failover to another running server, to load balancing and service clusters. Today, the PAN is the only commercial architecture capable of N+1 failover clustering in a completely heterogeneous environment.
- *Improved resource utilization.* Because the PAN treats server blades as a pool, resource utilization can be notably increased by assigning or removing blades from an application on-the-fly in response to changing business conditions. With most enterprises currently leveraging only 15 to 20 percent of installed processors, the PAN promises to improve capacity planning and overall resource utilization.

**SUMMARY**

The rapidly increasing use of technology to gain competitive advantage has led to unprecedented data center expansion. For both enterprises and service providers, the result has been tremendous pressure on the IT function — including a host of problems associated with deploying and managing the servers needed to support applications. While the sheer size of data centers has increased steadily, the number of servers has grown exponentially. Today, a large facility may house thousands of servers and tens of terabytes of storage.

Some of the major issues facing today's enterprise-class data centers include physical complexity, proprietary architectures, high availability, scalability, and over-provisioning.

In terms of complexity, a rack of 42 conventional servers can require as many as 250 cables for an optimized installation, including redundancy. In reality, this is impossible to manage and creates an unacceptable number of failure points. Thus, real-world deployments are considerably less dense and less reliable than they should be. In addition, sources indicate that cabling can add days to an already-long deployment cycle. Vendors have responded on several fronts, including the recent move toward partitioning large, expensive SMPs into logical collections of smaller systems. Other piecemeal responses include hot-plug PCI chassis, shared networked KVM switches, daisy-chained management cables and rack-integrated Ethernet switches. An ideal system would minimize the number of connections and cables required, and permit reconfiguration with virtually no physical intervention.

The operating system chosen for the data center must provide ease-of-use, dependable operation, support for a wide range of commercial applications, a clear migration path (to protect the initial investment), and a non-proprietary approach (to avoid vendor lock-in). While computers historically have been developed for Microsoft Windows or a proprietary UNIX variant, the emergence of Linux — an open source operating system — has changed the equation. Linux has enjoyed a rapidly increasing level of ISV support, including nearly universal availability on major Web servers, middleware applications, and database products. The openness of Linux also creates leverage over vendors. Moreover, peer review by the world's best engineers ensures that the code base is solid, stable, and reliable. These characteristics, shared by no other operating system, make Linux adoption in the enterprise a winning proposition for IT professionals.

Traditionally, high availability has been reserved for only the most mission-critical applications. However, as the networked business environment has evolved, so has the definition of mission-critical. Most enterprise environments need a flexible repertoire of failover solutions, ranging from none through simple failover through nearest-neighbor and N+1 topologies. As clustering models and requirements have grown

more complex, tools have been introduced to mitigate the complexity. Data centers should evaluate the various cluster-capable software and hardware systems against their specific needs. An ideal system would offer tightly integrated failover capabilities with flexible topologies, allowing the user to select the appropriate level of redundancy. In addition, it would simplify the process of configuring and managing clusters.

Another challenge is rapidly scaling the infrastructure to accommodate ever-changing load conditions. Vertical scaling of a service means adding capacity to a single instance of the resource which provides that service, that is, upgrading a server to a faster single CPU or adding CPUs to an SMP. To achieve this, data centers can deploy large SMPs and run them as partitioned systems to create readily scalable virtual servers in a tightly coupled environment. However, when misapplied in this manner to compensate for the physical complexity of a multi-server deployment, SMPs may work — but at an extreme price penalty. Thus, when one considers cost and flexibility, the decision to scale a service vertically becomes less straightforward and must reflect whether the service truly requires a large single image or can be spread horizontally across multiple, smaller, independent platforms.

As data centers have moved toward the disaggregated infrastructure associated with distributed applications, they have suffered progressive declines in resource utilization. That is because ensuring responsiveness to business requirements has meant sizing systems to meet peak demand, thus leaving expensive processing cycles idle most of the time. Today, typical CPU utilization across a large organization is in the 19 to 23 percent range for non-mainframe servers, primarily because there has been no practical way to leverage unused resources during off-peak periods. Vendor responses have included various load-balancing and clustering options, COD programs, server partitioning, and server consolidation. Yet none of these point solutions adequately overcomes the over-provisioning issue. Instead, a data center system should enable an administrator to flexibly add and subtract capacity from a group of resources on-the-fly.

Solving all of these challenges from a single platform requires nothing short of a new computing architecture, referred to as a processing area network, or PAN. The past decade has seen the emergence of the storage area network (SAN) as a way to centralize, consolidate, and virtualize the deployment and management of storage resources. Processing has the same inherent problems when many discrete servers are deployed.

A PAN pools together multiple processors in a redundant network via an integrated high-speed switch, which in turn is connected to external storage and IP network resources through consolidated connections. Significantly, the processing resources in a PAN are diskless, stateless, and anonymous, enabling flexible allocation and failover because one can easily take over the characteristics and configuration of another. At the same time, legacy server components such as Ethernet NICs, Fibre Channel

HBAs, local disks, and external cables are consolidated or eliminated. All physical and logical resources are managed entirely through integrated management software, resulting in cost savings and unprecedented flexibility. Other benefits include efficient storage management, simplified physical installation, cost-effective HA provisioning, and improved resource utilization. Essentially, the PAN does for computing what the SAN did for storage — it transforms the provisioning of processing power from a complex physical process to a simple virtual process.

---

Peter J. Manca is vice president of software engineering for Egenera, Inc.