I have to admit a great measure of sympathy for the IT populace at large, when it is confronted by the barrage of hype around grid technology, particularly within the enterprise. Individual vendors have attempted to plant their flags in the notionally virgin technological territory and proclaim it as their own, using terms such as *grid*, *autonomic*, *self-healing*, *self-managing*, *adaptive*, *utility*, and so forth. Analysts, well, analyze and try to make sense of it all, and in the process each independently creates his or her own map of this terra incognita, naming it *policy-based computing*, *organic computing*, and so on. Unfortunately, this serves only to further muddy the waters for most people. All of these terms capture some aspect of the big picture—they all

describe parts of solutions that seek to address essentially the same problems in similar ways—but they're never quite synonymous.

So what is this “grid” stuff really all about? How does it apply to the enterprise data center and why should you care?

GRID COMPUTING

At the heart of *grid computing* is the concept that applications and resources are connected in the form of a pervasive network fabric or *grid*. Additionally, the term *grid* implies

Enterprise GRID Computing

PAUL STRONG, SUN MICROSYSTEMS

Grid computing
holds great promise for the
enterprise data center, but many
technical and operational hurdles remain.

Enterprise GRID Computing

both ubiquity and predictability, with grids being viewed as very much analogous to electrical or power grids,¹ which are accessible everywhere and sharable by everyone.

Grid computing is an inevitable consequence of a set of long-lasting technology trends.

First, applications have evolved from once simple, often monolithic, computer- or server-centric binaries to services that are disaggregated and then distributed across the network. Web services and SOAs (service-oriented architectures) are at the forefront of this trend today.

Second, the network itself has evolved. The relentless increasing of network bandwidth has slowly eroded the boundary between the server and the network, turning a collection of interlinked, discrete resources into a fabric consisting of physical resources, such as servers, storage and network components, as well as logical resources—for example, operating systems, database servers, application servers, and other platform or infrastructure software. Although certain classes of applications still benefit from their components sharing memory or scaling on a large SMP (symmetric multiprocessor) computer, for many—and one could argue for most—applications, it is good enough to get one's data or to interact with other application components or resources across the network.

These trends have fueled each other for at least the past two or three decades, resulting in the application and infrastructure architectures we see today.

Grids, by their very nature, have a set of latent properties that extend those of a traditional computer and make grid computing compelling. These include, but are not necessarily limited to:

Massive scaling and throughput. Network connectivity enables groups of hardware and software components to be effectively combined to achieve greater performance and scaling. You can apply far more resources to a particu-

lar workload across a network than you ever could within a single, traditional computer. This may result in greater throughput for a transactional application or perhaps a shorter completion time for a computationally intensive workload.

Inherent resilience and availability. The use of multiple, replicated components within a grid enables greater resilience and availability.

Mutability and flexibility that result in greater efficiency and agility. Grids also offer the opportunity for true economies of scale through the sharing of a large pool of resources among many sets of workloads with differing profiles. There is sharing in the sense that a single resource may be shared by multiple applications at the same time—for example, two applications, or software components, consolidated onto a single physical server—and sharing in the sense that a resource may be repurposed over time, so that one application replaces another as needs change. This leads to potentially greater efficiency or utilization. In addition, the mechanisms that enable repurposing of resources give the grid a mutability that can translate into greater agility or responsiveness.

Service-oriented. The final distinguishing property of grid computing is a focus on managing applications and services rather than the individual resources within the fabric. The sheer number of resources that must be coordinated becomes so vast that more abstract management objects, such as a tier of a service or a complete business application, must be used so that management can scale.

Realizing these latent properties is made possible by a set of key mechanisms (described later), including virtualization, abstraction, and automation, as well as standards.

In many ways the notion of grid computing is evolutionary rather than revolutionary. It neatly captures the existing technology trend toward networked computing, the attributes that network-scale “systems” can deliver, the set of mechanisms that will enable them, and the desire for ubiquity and predictability in delivering business services through IT, all within a single, coherent context. What is revolutionary is the perspective that grid computing forces on architects and implementers. Grid computing mandates a systemic or holistic, rather than component-centric, approach to the management of networked applications and resources, especially within enterprise data centers.

GRIDS AND ENTERPRISE GRIDS

In the general sense, a grid may be defined as a bounded environment (i.e., a collection of networked applications—or services—and resources, which is treated as a

whole and within which grid computing is undertaken). The scope of a grid could range from a small departmental network to a vast collection of resources and services running in multiple locations, spread across the world, and owned by many organizational groups, government bodies, enterprises, or academic institutions.

When exploring the impact of grids within enterprise data centers, we use the term *enterprise grid* to capture the notion of a grid that is managed by a single entity or business. This is a very specific type of grid, in which there is a clear scope of control and responsibility for managing the grid to meet a specific set of business goals. The extent of an enterprise grid is defined in terms of organizational responsibility and not in terms of geography or asset ownership. Thus, an enterprise grid may span multiple locations or data centers. It may also consist of applications or services run on behalf of other organizations, such as in an outsourced environment. Enterprise grids must also support various types of workload (transactional, OLTP, batch, compute-intensive, and legacy) and a large, heterogeneous set of resources. This contrasts markedly with more traditional aggregation frameworks in the data center, such as high-availability clusters, load-balanced clusters (e.g., Web farms), or compute-intensive clusters, which are typically focused on a specific application, or type of application, and which are usually deployed on a relatively homogeneous set of resources.

The potential for greater efficiency and agility promised by enterprise grids is very compelling. Although the absolute scaling and efficiency benefits may not reach the magnitude of the generalized grid (think the whole Internet!), for large organizations with hundreds of applications and services, and hundreds of servers, disk arrays, and network devices, the potential benefits of applying grid architectures and technologies are huge. These benefits do not necessarily come easily, though. The adoption of grid technologies within the enterprise can be very challenging, especially in an operational sense. Nonetheless, grid computing captures a vision, architectural elements, and technologies that will almost certainly, over time, fundamentally change the way in which the data center is managed.

So, given all this, where is today's data center in terms of being an enterprise grid, and what are the challenges, in terms of the adoption of grid technologies?

THE ENTERPRISE DATA CENTER TODAY

Today's enterprise data center is a complex place. Each one usually hosts myriad applications or services running on a vast number of networked resources. Each

component in this fabric—whether an application or resource, whether physical or logical—is itself relatively simple, but once you put them all together, complexity increases exponentially. When you add a component, not only do you add to the total number of components, you may also add a new type of component and a set of relationships with existing components within the fabric. Visualize a typical enterprise application such as an electronic bookstore. The application may be broken down into tiers—for example, persistent storage or database, business logic, and presentation. Firewalls may exist between some of these tiers. Each tier may consist of a set of servers running application components and perhaps a clustering or load-balancing framework. Each server will be running at least one application component, which may depend on a certain version of an operating system, together with a certain set of patches, all running on a certain type of processor. Figure 1 illustrates the complexity in the average data center in the form of a simplified dependency graph for just one application. Add another 10 or 100 such applications and inter-relationships between them all and you get an idea of the complexity that has to be managed every day in a typical data center.

Complexity can lead to big problems. Managing such complexity is really hard. One thing that is certain today is that it is very, very hard to achieve economies of scale in the modern data center. Applications have evolved over the past 30 years, such that they are now created using more abstract and reusable building blocks, which are then distributed over a network. SOAs (service-oriented architectures) and Web services are the latest in a long line of ever-more-disaggregated and distributed application architectures. Likewise, data centers have evolved from monolithic mainframe environments, to networks of clusters and tiers of homogeneous servers and operating systems, to the fabric of heterogeneous resources we see today.

Management of the data center, however, which includes those applications, as well as the infrastructure that they run on, has not really evolved. We are still directly managing the same set of components (i.e., servers, storage devices, network devices, operating systems, database servers, application components), only now there are far more of them and they are connected in ever-more-complex combinations. How we manage infrastructure has to change; otherwise, we cannot realize the value of these complex applications and services.

Today, enterprises mitigate the effects of complexity by creating relatively static silos of infrastructure in a divide-and-conquer management approach. In a typical data

Enterprise GRID Computing

center, separate groups will manage the servers and their operating systems, the network components, the storage components, security, and sets of applications of services. This is illustrated in figure 2.

Complexity is addressed by effectively limiting the total number and types of components, and their relationships. This allows the performance, scalability, and

availability attributes inherent in network-distributed architectures to be exploited, but it is typically at the cost of efficiency and agility. Static silos result in spare or excess capacity on a per-silo basis, which is much less efficient than shared, dynamically assignable excess capacity. These static silos also result in a lack of agility, as new silos have to be created for new applications and services, rather than perhaps simply using existing excess capacity.

Today's data center could be viewed as a primordial enterprise grid. The resources form a networked fabric and the applications are disaggregated and distributed. Thus, the innate performance, scaling, resilience, and availability attributes of a grid are in some sense realized. The economies of scale, latent efficiency, and agility remain untapped, however, because of management silos. How can this be changed? And what is the difference between an enterprise grid and a traditional data center?

A SYSTEMIC VIEW OF THE DATA CENTER

At the core of the problem is the component-centric view of the data center. Operationally, everything revolves around explicitly managing discrete components and

Component Dependencies for an Application

Source: Enterprise Grid Alliance

FIG 1

their relationships, rather than the business applications themselves.

Enterprise grid computing requires a more systemic and holistic approach to managing the data center. The data center grid or network becomes the new system, so to speak—"The network is the computer" to quote one vendor. This philosophy is at the core of a number of vendor and standards initiatives, including Microsoft's DSI (Dynamic Systems Initiative),² Sun's N1 strategy,³ the GGF's (Global Grid Forum)⁴ OGSA (Open Grid Services Architecture),⁵ the EGA's (Enterprise Grid Alliance)⁶ Reference Model, and so forth.

It is this systemic approach and its application that fundamentally separates an enterprise grid from a traditional data center. The data center network becomes the design center, the context within which workloads (i.e., applications) and resources are managed. It encourages the development of an architecture that encompasses the data center, an architecture designed to enable management of services and applications through policy and automation, an architecture that recognizes that today's managed systems and stacks of software are merely components in a greater system that defines their functionality and relationships with one another.

Redefining the data center based on a holistic archi-

tecture then enables the development of consistent tools and products that can manage this new enterprise grid. A consistent architecture, coupled with industry standards, should ensure that the sets of tools that are delivered can be integrated in some way (yielding the horizontal integration layer described by Ian Foster⁷) to allow the data center to deliver value that is greater than the sum of the parts. This is the fundamental value of the holistic or systemic approach found in grid computing.

In many ways those tools, in aggregate, are analogous to an operating system. An operating system manages the life cycle of workloads, mapping and remapping them onto resources, in line with policy. In a traditional operating system, the workload consists of processes (binaries); the resources are processors and memory, together with storage, and network interfaces; and the policies are low-level rules, such as "nice" (priority) values for processes. In an enterprise grid meta-operating system (so to speak), the workload consists of network-distributed applications (ranging from traditional multitier applications to Web services and SOAs); the resources are servers, storage arrays, network devices, operating systems, databases, and other platform software; and the policies are SLOs (service-level objectives).

Just like a traditional operating system, an enterprise

Enterprise GRID Computing

grid management toolset depends heavily on a set of mechanisms—virtualization, abstraction, and automation—and an architecture that ensures their appropriate application. It is the development and adoption of these technologies within a consistent operational and architectural context that is key to realizing all of the benefits of true enterprise grids.

What are these mechanisms and how do they contribute to making the data center a more efficient and agile place—in fact, turning it into an enterprise grid?

VIRTUALIZATION

Virtualizing something, such as a server, disk, or operating system, means that its implementation (hardware or software) has been separated from the interface through which some other entity (a user or perhaps a piece of software) interacts with it. This is usually achieved by inserting a layer of software so that the underlying implementation can be changed without having to change anything that depends on or uses it. An example of this is LUNs (logical unit numbers) in storage. A LUN is a virtual disk. It behaves as a disk as far as an operating system that has a file system installed on it is concerned. Yet the LUN may, in fact, be realized as a single physical disk, as a partition on a physical disk, or as an aggregation of many physical disks, such as a RAID volume.

VMMs (virtual machine monitors), such as VMware⁸ or IBM's LPAR⁹ (logical partitioning), virtualize servers in much the same way. A hosted operating system behaves as if it is running on its own compute server, when in reality it is running on a software layer that presents it with all of the resources of a compute server. Similarly, Sun's Solaris Containers¹⁰ virtualize an operating system instance, allowing multiple applications to share it, each one within its own virtual operating system environ-

ment. In essence, virtualization ensures that the user or consumer of a resource or component is presented with a consistent interface that separates them from the underlying components.

Virtualization technologies can yield a number of benefits. They enable resources to be shared, improving asset efficiency. They enable resources to be aggregated, improving management efficiency. Finally, they improve agility by allowing components to be replaced or upgraded with minimum disruption.

We should note, however, that virtualization, in and of itself, does not necessarily solve the management scaling problem. If you virtualize a physical server so that you can host 20 operating system instances on it, then you may achieve better resource utilization with regard to the physical server and you may have only one physical server to manage. You still have 20 operating systems to manage and patch, however, as well as the virtualization software. If all 20 operating system instances are identical and tools allow them to be managed as one, only then will the management burden be significantly eased. True management efficiency is typically realized through the additional use of abstraction and automation.

ABSTRACTION AND AUTOMATION

Abstraction is the act of changing the exposed properties of one or more objects or entities, typically by creating a new type of object that encapsulates or hides the others. For example, a Web farm abstracts a set of servers, a firewall, and a load balancer. If you manage the Web farm at a high level of abstraction, you could manage properties of the farm, such as its access rights, its capacity, and the default operating system image for the servers and the types of servers to be used, without directly interacting with each individual component. A management tool automatically translates between the new management abstraction and the old one, based on policies. This is where the notion of policy-based computing intersects with grid computing. In this example, the tool could determine which unassigned servers in the enterprise grid support the desired operating system image and then provision those servers, load balancer, and firewall in line with those requirements. This could include the provisioning of policies for the load balancer and firewall.

This type of automation is essentially moving traditional IT management processes or practices, such as those captured in data center management standards—eTOM (Enhanced Telecom Operations Map)¹¹ or ITIL (Information Technology Infrastructure Library)¹², for example—from the human to the machine domain

and is often referred to as orchestration in the enterprise grid world.¹³ Abstraction and automation enable greater management efficiency by allowing many traditionally managed components to be more intuitively managed as a single entity. Complexity can be hidden. Of course, the complexity is still present, but the user or manager is no longer exposed to it, at least not where simple, common, repetitive tasks are concerned. Not only does automation improve management efficiency, it also improves and enables agility. It enables agility by reducing the risk of error associated with manual processes. This risk often prevents data center managers or systems administrators from taking advantage of the latent agility within their data center by, for example, repurposing servers to cater for load variations. Finally, automation improves agility through potentially reducing the time taken to execute a given process flow.

BRINGING THE THREADS TOGETHER

Virtualization, abstraction, and automation are mechanisms that are key to turning the modern data center into a genuine grid—an enterprise grid—and delivering greater efficiency and agility. These mechanisms are typically realized in combination in any given product—for example, in server and operating system provisioning tools, complex service and application life-cycle management tools, and service-level management tools.

The key to extracting the maximum value from these tools is that they share a common architectural and operational context.

A shared architectural context should ensure that the tools solve the right problems in the right way. This is the value of the various grid consortia—for example, the EGA and GGF—which are driving toward a standard set of requirements and a standard architectural model, respectively. Combining this with the use of standards for the various management protocols and mechanisms (many of which are nascent but nonetheless on the way) should allow data centers to choose the sets of interoperable tools appropriate to their needs, without fear of vendor lock-in.

REALITY CHECK

Ultimately, the grid computing paradigm really does offer the benefits enumerated in all of the hype. As with everything, however, there are trade-offs and areas where technology is immature or nearly nonexistent. This means that not all of the benefits can be realized yet. Setting appropriate expectations is key.

Today's data center is essentially a primordial enter-

prise grid where the latent properties of scaling, performance, resilience, and availability are being exploited. Because of the lack of a systemic architecture, consistent operational model, and mature virtualization, abstraction, and automation mechanisms, however, the potential economies of scale and agility of a true enterprise grid have yet to be fully realized.

The current buzz surrounding enterprise grids, policy-based computing, autonomic computing, and so forth is focused on solutions to these economies-of-scale problems.

The first thing that is required is a consistent operational model and a common set of requirements shared by the various vendors and standards bodies in the space. A sensible starting point is actually to use those standards (open or de facto) that have evolved to manage today's infrastructure, such as ITIL (in the commercial space) or eTOM (more typical in the telecommunications provider's space today). After all, if you are going to automate a process, you had better have one. Basing work on these has two benefits.

First, these capture today's best practices, and data center managers will want to minimize process change as much as possible when adopting new tools or technologies. The majority of the IT infrastructure management cost within a large company (a number of surveys put it at 70 percent) is taken up by the people cost associated with the management of infrastructure (network, storage, servers, operating systems, etc.) and applications. Minimizing process change lowers the cost of adopting new technologies and thus makes them more attractive.

Second, automation is about moving processes from the human or manual execution to the machine or automated execution domain. If well-established processes are already in place for the management of existing resources and services within an enterprise grid, then they would naturally form the basis for tools that fulfill a similar function.

So the foundations for a consistent operational model exist. What about an architecture?

As previously discussed, a number of proprietary vendor-driven architectures occupy this space, plus OGSA, which is quite a high-level standard architecture. The latter is not yet specific enough to the data center, nor detailed enough, to be actionable. The focus is on using Web services to manage the infrastructure and the applications, and thus it is not clear how this will support existing applications and infrastructure. Nonetheless, a great deal of effort is being expended in driving this forward.

Enterprise GRID Computing

Finally, there are the core enabling mechanisms. Figure 3 illustrates the various layers of virtualization and abstraction in the data center today. The diagram does not claim to be definitive, but merely serves to illustrate, for example, that physical components may be virtualized. These are then abstracted by operating systems and such. These may in turn be virtualized before being abstracted. Note that although there has been a great deal of media hype about the arrival of virtualization technologies, most of the technologies already exist. It is just that some of the newer ones are beginning to be adopted in production, and they are being combined with abstraction and automation products. These products are typically dealing with very specific areas of management, so there really isn't a single product that can act as the meta-operating environment for the enterprise grid yet. Nonetheless, solutions—albeit ad hoc and non-standards-based—may be built today.

GRIDS ARE INEVITABLE
Grids offer the opportunity for greater service performance, scaling, and availability, together with improved agility and efficiency.

Grids in general, and enterprise grids specifically, are inevitable. This is evident from the technology trends (i.e., from server-centric applications to network distributed services—as typified by Web services and SOAs—and from the server as the platform to the network as the platform).

Realizing all of the potential benefits, however, requires a great deal of effort focused on the management of the enterprise grid, in terms of both process refinement and technology—technology that virtualizes resources, making them easier to share and to replace with improved implementations, and technology that abstracts and automates, allowing management scalability to be achieved by slowly moving from a component-centric management model to a service- or application-centric model.

Many of the technologies exist, and real, quantifiable benefits may be achieved today through:

- The application of standard operational processes.
- The development of a rudimentary enterprise grid architecture in the data center or the adoption of a vendor-specific one (in lieu of a mature standard architecture).
- The use of various virtualization, abstraction, and automation tools focused on the problems most pressing to the business.

The Various Layers of Virtualization and Abstraction in the Data Center

biz process/ service	e-bookstore ERP service online bank		
virtualized platform	web server farm aggregations federation		clusters load balanced farms
platform instance	database	LDAP	web server application server
virtualized OE	network file systems— eg NFS, CIFS	virtualized OS— eg N1 grid containers, BSD jails, etc.	load balancers, global IP in clusters
OE	file systems	OS—eg AIX, HP/UX, Linux, Solaris, Windows, etc.	IP, TCP, UDP, etc.
virtualized physical	LUNs, volumes	VMMs & hypervisors hardware petitions	VLANs
physical	disks, array controllers, SAN switches	servers, blades, etc.	switches, routers, etc.
	storage	compute	network

Source: Enterprise Grid Alliance

FIG 3

In the longer term, enterprise grids will enable enterprise data centers to be truly agile, to evolve (perhaps organically), and to adapt rapidly to business needs. Through the development of appropriate management technologies they will also enable true economies of scale to be achieved within enterprise data centers.

This effort is well under way, and many benefits may already be accrued through the use of existing technologies. The journey is a long one, however, and expectations must be managed to avoid disappointment. □

REFERENCES

1. Foster, I. 2002. What is the grid? A three-point checklist; <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>.
2. Microsoft's Distributed Systems Initiative (DSI); <http://www.microsoft.com/windowsserversystem/dsi/default.msp>.
3. Sun's N1; <http://www.sun.com/software/n1gridsystem/>.
4. Global Grid Forum (GGF); <http://www.ggf.org>.
5. GGF Open Grid Services Architecture (OGSA); <http://www.ggf.org/documents/GWD-I-E/GFD-I.030.pdf>.
6. Enterprise Grid Alliance (EGA); <http://www.gridalliance.org>.
7. See Reference 1.
8. VMware; <http://www.vmware.com>.
9. IBM LPAR; <http://www-1.ibm.com/servers/eserver/series/lpar/>.
10. Sun Solaris Containers; <http://www.sun.com/software/solaris/ds/containers.jsp>.
11. ITIL; <http://www.itsmfusa.org/mc/page.do?sitePageId=2995>.
12. eTOM; <http://www.tmforum.com/browse.asp?catID=1647>.
13. Unlike eTOM, ITIL is not yet a prescriptive, formalized standard; rather, it is a set of informal guidelines.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

PAUL STRONG is a systems architect at Sun Microsystems where he focuses on grid standards and the N1 product set. He has worked at Sun for eight years, was part of the original N1 team more than four years ago, and has co-authored the book *Building N1 Grid Solutions* (Prentice Hall, 2004). Strong is chair of the Enterprise Grid Alliance technical steering committee and its reference model working group, where he edited/co-authored the EGA Reference Model. He holds a B.Sc. in physics from the University of Manchester, England.

© 2005 ACM 1542-7730/05/0700 \$5.00