Solaris™ Containers: Server Virtualization and Manageability

A Technical White Paper
September 2004

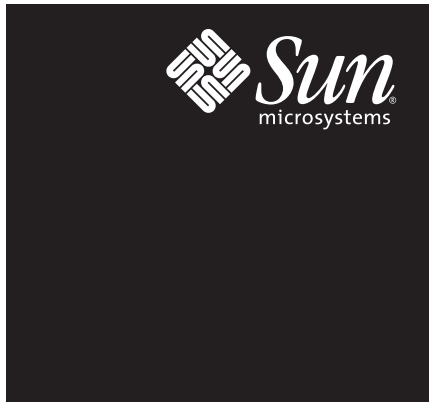


Table of Contents

Chapter 1

Executive Overview

After nearly a decade of experience with large-scale distributed computing, businesses have begun to understand how to build scalability and availability into their information technology (IT) infrastructure. Now, the escalating costs of managing vast networks of servers with software components installed on thousands of nodes have shifted the focus from increasing performance and availability to reducing the cost of IT infrastructure, increasing utilization of hardware, and better managing end-user service levels.

Sun's vision is that businesses can accomplish this through N1™ Grid computing, a new approach for managing data centers. N1 Grid views a data center as a single pool of resources delivering business services. Business services may be run on any system within the pool that provides the resources needed to meet performance and service levels. Creating and managing this single pool of resources requires *server virtualization*, a concept that allows servers and the network to be flexibly partitioned into independent execution environments within the same server and physical network.

Business services often consist of software applications that are distributed across multiple servers and operating system instances. To reduce costs, businesses are eager to consolidate these applications onto fewer servers, but are concerned about unexpected interactions between these applications. Virtualization enables the management of data centers to be treated as a fabric of interconnected computing resources rather than as a room filled with individual systems.

Solaris™ Containers (formerly N1 Grid Containers) technology is a key element in the implementation of an N1 Grid System, and provides flexible, software-defined boundaries for isolating software applications and services. These applications can then be managed independently of each other, even while running with the same instance of the Solaris Operating System (OS), without the extra overhead of managing multiple OS instances for each application.

Solaris Containers were first introduced with the Solaris 9 OS to provide full resource containment and control for more predictable service levels. With the release of the Solaris 10 OS, they extend this paradigm to create an execution environment for applications and services within a single instance of the Solaris OS and add security isolation to prevent unauthorized access and isolation from faults to minimize fault propagation and unplanned downtime. The Containers environment also includes enhanced resource usage accounting. This highly granular and extensive resource tracking capability can support advanced client billing models and enable capacity planning investigations.

Solaris Containers are planned to be the fundamental, ubiquitous management object in the Solaris OS, and will be used throughout Sun's SPARC®, AMD Opteron, and Intel x86 processor-based product range. This common approach will simplify service provisioning and make it easier to consolidate applications onto fewer servers, while also addressing concerns about resource constraints, fault propagation, and security. The ubiquitous management model also simplifies service-level management by permitting end-to-end services to be managed as a single unified object. The service management environment will be flexible and dynamic, and yet easy to understand and manage.

The primary benefits of Solaris Containers are:

- Reduced management costs through server consolidation, reduced numbers of OS instances, and end-to-end service-level management
- Increased resource utilization with dynamic resource reallocation between Containers
- Improved service availability by minimizing fault propagation and security violations between applications
- Increased flexibility through software Containers, which can be dynamically reconfigured
- Increased accuracy and flexibility in usage billing based on workloads rather than hardware or processes

Businesses can realize a significant reduction in total cost of ownership without sacrificing service levels through this intuitive, flexible management model.

Intended Audience

This paper describes an approach to resource management in a distributed computing environment. The intended audience is IT managers, architects, and executives in organizations that are implementing large-scale, distributed computing solutions. Chapter 3 discusses the concepts and key features of Containers; Chapter 4 illustrates how Containers may be used to implement various resource management scenarios; Chapter 5 takes a look at upcoming developments in Container technology; Chapter 6 provides a brief conclusion; and Chapter 7 provides references for further information on the software and technologies discussed in this paper.

Chapter 2

Introduction

As IT systems have evolved over the years, so have the expectations of both end users and IT staff. With each advance in technology, breakthroughs are quickly viewed as standard features, which are soon taken for granted. Early information systems were judged a success if they simply solved a business problem. It didn't matter if it required a team of programmers to maintain the system. As long as the business benefits could justify the cost, the systems were considered good investments.

As information systems matured, more users were given direct access to computerized business systems, and managers began to depend heavily on their computing infrastructure to conduct daily operations. Availability and scalability of their business systems became key concerns for IT managers, and the key benefit of building availability and scalability into their systems was measured by counting the lost productivity from either downtime or poor performance of their systems. Today, methods of implementing increased system availability and scalability are becoming well understood. By replicating key software processes across a multitiered architecture of servers, both system availability and scalability can be significantly improved.

Recently the focus has been on obtaining additional system availability or scalability in the most efficient way possible. However, when more components are replicated throughout an IT architecture to give it greater resiliency and throughput, the result is a sprawling, complex network of systems that are inefficient and difficult to manage.

Businesses can reduce costs by improving resource utilization and lowering system management costs. Capacity planning techniques generally include allocating extra capacity to handle occasional high peak loads. For example, brokerage firms often design their systems to handle trading volumes far above those that have ever been experienced in order to capture as much revenue as possible should a spike in demand occur. This means that normal trading activity levels may only require a fraction of the total available capacity of their IT infrastructure.

By allowing other applications to share resources with the trading systems when those resources are underutilized, a much more cost-effective implementation can be realized. In the event of a spike in trading volume, resources would be dynamically reallocated to the trading systems while the less-critical applications would receive less resources, ensuring that the primary trading systems function as needed. Sharing resources in this way reduces total capital investment costs and lowers system management expenses by reducing the total number of systems required. The struggle to make IT operations more efficient has turned the focus of IT managers towards increased utilization of hardware and software assets.

In order for server consolidation to be truly effective, businesses must still be able to manage each application independently. This requires the ability to control IT resource utilization, isolate applications from each other, and efficiently manage multiple applications on the same server. In other words, it requires establishing easily managed virtual server boundaries within a single system on which to provision applications.

Sun's Commitment

Increasing IT efficiency and improving hardware utilization is key to the success of enterprises. Through a long-term commitment to networking, client-server computing, UNIX® software, and open standards, Sun consistently delivers robust, flexible, and powerful software technologies, high-performance servers, and systems that meet the needs of a wide range of organizations. By providing technologies such as Solaris Containers to increase the utilization of existing resources, Sun enables safer, easier-to-manage IT infrastructures and extensive resource management capabilities, which help enterprises satisfy their diverse set of needs.

Chapter 3

Server Virtualization

The concept of consolidating applications onto a single system with a single operating system is not new. Mainframe systems have been doing this for years — just not at an affordable cost. It is simply more efficient to aggregate these applications together and share the available system resources. Through such consolidation, resources can be shared and reallocated on the fly instead of having dedicated, but inefficiently used, hardware to support each of these application services. Using this approach, resources can be logically moved when and where they are required. If done correctly, businesses can reduce the total expense of their computing infrastructure through reductions in the cost of capital equipment, maintenance, licensing fees, and system management.

Mainframes were the first to divide system resources into partitions to isolate applications from one another. This enabled customers to protect production environments while system updates or revised applications were tested in an isolated workspace that could not affect the production system. In 1996, Sun developed this idea one step further when it introduced Dynamic System Domains technology into the open systems market.

With Sun’s Dynamic System Domains, each domain runs its own copy of the Solaris Operating System and provides isolation so that system or application errors cannot impact applications running in other domains. A key difference between these domains and the logical partitions approach is the ability to dynamically adjust the resources within the partitions according to changes in demands. In the event of a resource shortage in one domain, the system can automatically borrow resources from another domain. This adds great flexibility to the way partitions are utilized while maintaining security and isolation from faults in other domains.

Dynamic System Domains virtualize individual servers; the N1 Grid System virtualizes an entire data center into a single pool of resources.

N1 Grid System: Sun’s Vision for the Data Center

The N1 Grid System is Sun's vision, architecture, and products for reducing the complexity of enterprise data centers by making entire data centers appear as a single pool of resources rather than a collection of separate systems, software, and storage. Today, each application in a data center typically has its own resources, with its own excess capacity to meet peak demand times. Applications have typically not been designed to share resources, resulting in poor overall utilization of hardware and reducing the total cost of ownership (TCO) achievable. With the N1 Grid System, data center resources may be managed as a single system delivering business services, putting idle resources to use, improving manageability, and decreasing TCO.

Note – In this paper, the terms *business service*, *workload*, or *application* may be used interchangeably to mean a collection of software components that provide a specific business function such as enterprise resource planning (ERP) or financial management.

A business service typically spans multiple computing tiers. The N1 Grid System will enable the administrator to describe the service levels, dependencies, and relationships between the different software components that comprise a business service. The N1 Grid System will then manage the end-to-end availability of the business service as a whole. By enabling IT staff to manage complete services rather than the individual components that make up those services, the N1 Grid System will reduce management complexity and increase system utilization, thereby reducing operating expenses and lowering TCO for the organization.

In an N1 Grid System, the process of mapping business services onto resources is known as provisioning. The N1 Grid System's services provisioning automates software and hardware installation and configuration, and allows for faster, more flexible deployment of services with lower risk of human errors. The N1 Grid System will dynamically provision resources for business services, obtaining resources as needed to meet peak demand. As demand drops off, resources are released for use by other services, an ability not provided by current capacity-on-demand solutions. By dynamically managing resource allocations based on predefined business policy, an N1 Grid System will maintain consistent service levels and enable IT staff to better meet service-level agreements and users' expectations — without overprovisioning servers to meet peak demands. Automation of error-prone manual tasks allows IT staff to focus on business needs and more rapidly deploy business services to meet those needs.

Pooling the available data center resources to maximize efficiency can make it more difficult to assign appropriate costs to individual business services. The billing and accounting features of the Solaris OS and the N1 Grid System provide a way to get accurate usage-based data on a per-service basis, not just on a per-process or system-wide basis. This greatly enhances the opportunity to measure the effectiveness of data centers equipped with the N1 Grid System. Line-of-business professionals have visibility into the actual costs per-business service and are able to make adjustments based on business objectives. Resources are logically provisioned and reconfigured as often as needed to meet rapidly changing business needs.

Solaris Containers: Advancing Server Virtualization

Sun has taken the server virtualization concept to a much higher level by allowing servers and their domains to be software partitioned to a far greater granularity by using Solaris Containers, a key element in the N1 Grid System approach to managing IT infrastructure. A Solaris Container defines a complete execution environment for a set of software components. It allows application components to be isolated from each other while sharing a single instance of the Solaris OS. It offers execution environments whose definitions are transportable from one server partition to another with minimal management overhead, enabling great flexibility in provisioning business services. The isolation, transportability, and resource sharing enable easier and more efficient consolidation of applications onto fewer servers and OS instances.

Solaris Containers (“Containers”) establish boundaries for consumption of resources such as memory, CPU time, and network bandwidth. As processing requirements change in line with business needs, one or more of the boundaries of a Container can be expanded to accommodate a spike in resource demand. Consider, for example, that an unexpected world event occurs and causes a surge in network traffic to a news-oriented Web site, which overloads the system and causes a reduction in service level. Wouldn’t it be helpful if the system on which the Web server is running automatically adjusted its resources to meet this business critical need? This is precisely what a Solaris Container is capable of doing. Fault and security boundaries are maintained when resource boundaries are updated, whether the update is done by an administrator or through predefined policies that result in automatic updates when certain conditions, like the one mentioned, are encountered. The next chapter provides some examples of how Containers can be used.

Chapter 4

Containers At Work

With the release of the Solaris 9 Operating System, many of the features necessary to create resource boundaries have been available through the Solaris Containers feature. These were expanded significantly in the Solaris 10 OS, and provide the ability to allocate resources such as CPU, physical memory, and network bandwidth within a single instance of the Solaris OS and allow enterprises to take advantage of the benefits of Containers illustrated in the following examples.

Example: Managing CPU Resources

Many enterprises depend on multiple instances of a database — for example, Oracle — which underlie critical applications such as enterprise resource planning (ERP) systems. Each database instance and front-end application typically resides on its own server, with its own dedicated CPU, memory, and storage, sized to meet peak demand times. In this example, a large, multinational enterprise has separate database instances, in a typical 3-tier architecture, for its European, Asian, and American operations. A simple view of this configuration is shown in Figure 1. Each of these systems is heavily used during the business hours for its operation's geography, but generally sees low usage outside of those hours, resulting in a significant investment in resources that are underutilized on each of the servers.

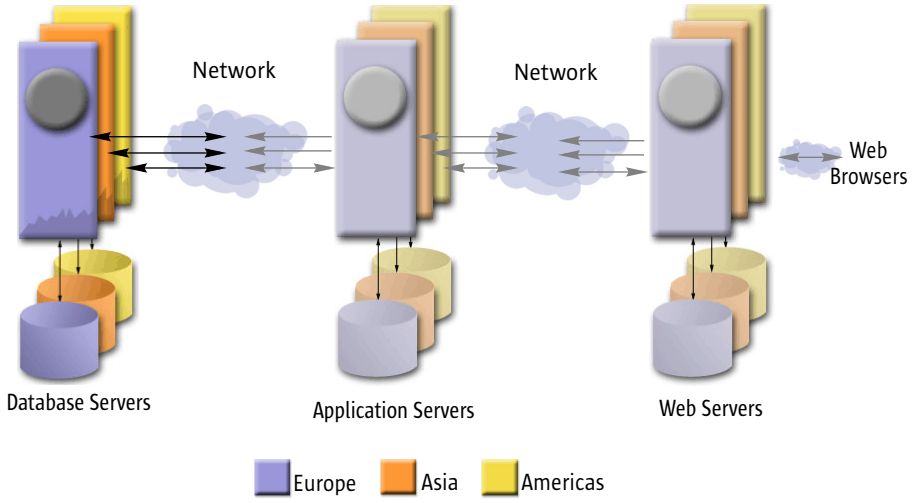


Figure 1. Resource management using separate systems

Database software tends to be more mature, and thus, more stable and reliable than newer types of software systems. Database systems generally request enough physical memory to ensure that their processes will remain in memory and avoid the impact on performance that results from data being swapped to disk. In this example, therefore, memory is assumed to be adequate and memory resource management is not a concern. The requirement is to manage CPU usage and ensure that each database instance receives the required resources during peak demand times. To manage CPU resources, the administrator can create a Container for each database instance and provision those Containers on a single database server, as shown in Figure 2. The CPU resources are managed using a specific piece of the Container technology: Solaris 9 Resource Manager.

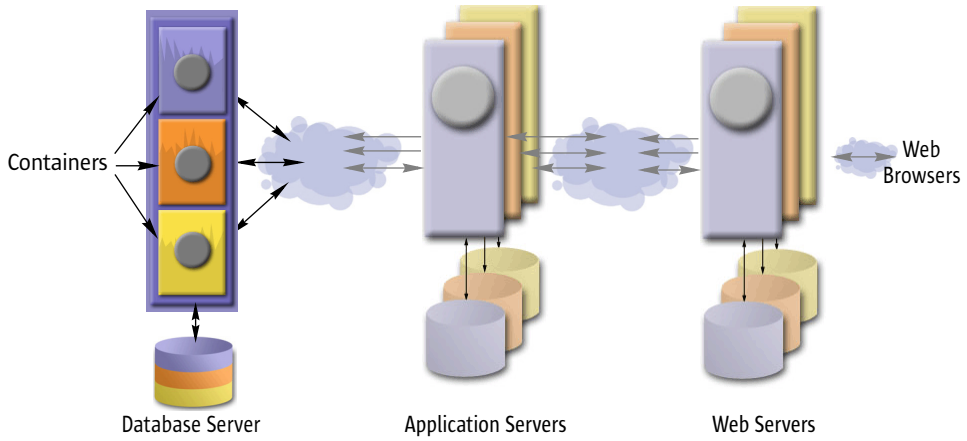


Figure 2. Database servers consolidated into separate Containers

Resource Management

Containers include a feature called Solaris 9 Resource Manager, which provides two significant capabilities for managing CPU resources: Resource pools and the Fair Share Scheduler (FSS).

Resource Pools

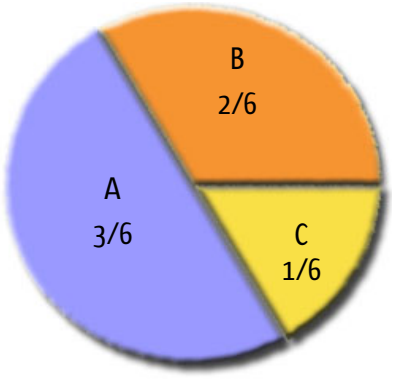
Solaris 9 Resource Manager uses *resource pools* to control system resources. Each resource pool may contain a collection of resources, known as *resource sets,* which may include CPUs, physical memory, or swap space. A *processor set* generally is assigned a subset of the total number of CPUs available on the system or in a Dynamic System Domain. For example, on a four-CPU system, an administrator could define two resource pools, each containing a processor set, with pool A's processor set containing one CPU and pool B's processor set containing the remaining three CPUs. Resources can be dynamically moved between resource pools as needed.

In order to assure the performance of a critical workload, it can be assigned to pool A by itself to avoid contention for resources with other workloads. All other workloads running on that system are assigned to pool B and share the three processors assigned to pool B's processor set. The use of processor sets in effect establishes the maximum CPU resources that may be consumed by a workload. The workload assigned to resource pool A, while it does not share the processor set, nevertheless has only one CPU available. The workloads running on resource pool B will contend for the CPU resources in pool B's processor set. This contention is managed through either the Solaris timesharing scheduler or the Fair Share Scheduler.

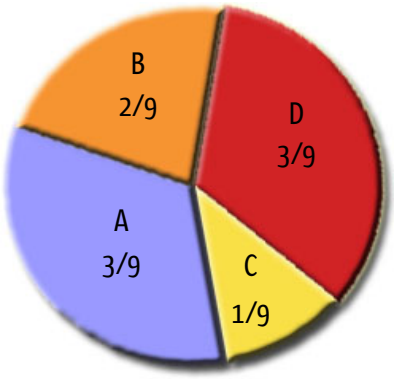
Fair Share Scheduler

Solaris 9 Resource Manager incorporates an enhanced Fair Share Scheduler (FSS), which may be used within a resource pool. When using FSS, an administrator assigns CPU *shares* to a workload that may comprise one or more processes. Shares enable the administrator to specify the relative importance of one workload to another, and FSS translates that into the ratio of CPU resources reserved for a workload. If the workload does not request CPU resources, those resources may be used by other workloads. The assignment of shares to a workload effectively establishes a minimum reservation of CPU resources.

For example, three workloads share a processor set using FSS. Workload A is assigned three shares, workload B is assigned two shares, and workload C is assigned one share, for a total of six shares. FSS allocates the CPU resources using the assigned shares. Workload A receives 3/6 of the available CPU resources, B receives 2/6, and C receives 1/6. When a fourth workload, D, is added and assigned three shares, the ratios change to 3/9, 2/9, 1/9, and 3/9 for each workload respectively, as illustrated in Figure 3. The relative importance of each workload remains the same.



Original CPU Share Assignments



CPU Share Assignments After Adding An Additional Workload

Figure 3. Fair Share Scheduler CPU share assignments

By setting minimum reservations (shares) and maximums (processor set resources) for each workload, the service-level needs for each operation can be flexibly managed while still providing a consistent level of service and meeting peak demand as needed.

In an environment where many workloads are sharing resources, enterprises will need to accurately allocate costs to the various business units using those resources and generate accounting information for resource utilization analysis and capacity planning. With Solaris 9 Resource Manager, extended accounting and reporting features are available that allow for tracking resource usage by business service workload, not just by process. Chapter 7 lists references for further information on Solaris 9 Resource Manager.

Using Containers and Dynamic System Domains

Prior to the availability of Containers, much of the consolidation described in Figure 4 could have been achieved with each database instance running in its own Dynamic System Domain, with its own copy of the Solaris OS, on a single server. Using Containers with Dynamic System Domains reduces the number of Solaris OS copies to be managed, resulting in a reduction in the system management workload and allowing for flexible reconfiguration of resources between Containers.

The enterprise in this example can go a step further, using Containers along with Dynamic System Domains. The production database instances can be consolidated into Containers within a single Dynamic System Domain running the Solaris OS release required for the production release of the database software. A second Domain provides an isolated environment to test a newer release of the database software that requires a different Solaris OS release. This scenario, illustrated in Figure 4, minimizes the number of operating system copies, isolates the production and test environments, and allows the hardware in the test domain to be reallocated to the production domain if needed.

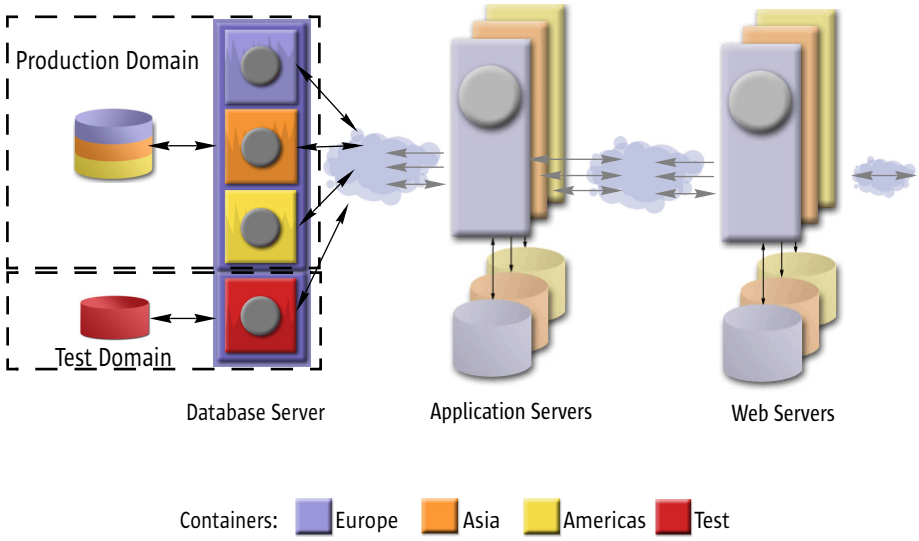


Figure 4. Database servers consolidated into separate Containers within Dynamic System Domains

Because the peak utilization times are different for each geography, the production domain can be configured as if for a single database instance. This configuration achieves a significant improvement in hardware utilization and a corresponding reduction in the number of OS instances that must be managed while maintaining performance levels and service availability.

Example: Managing Memory and Network Bandwidth

Many enterprises depend on multiple copies of application server software to support their critical business applications, such as financial management applications, customer management, and many others. A configuration typical of this environment is depicted in Figure 5, which shows each component hosted on its own server.

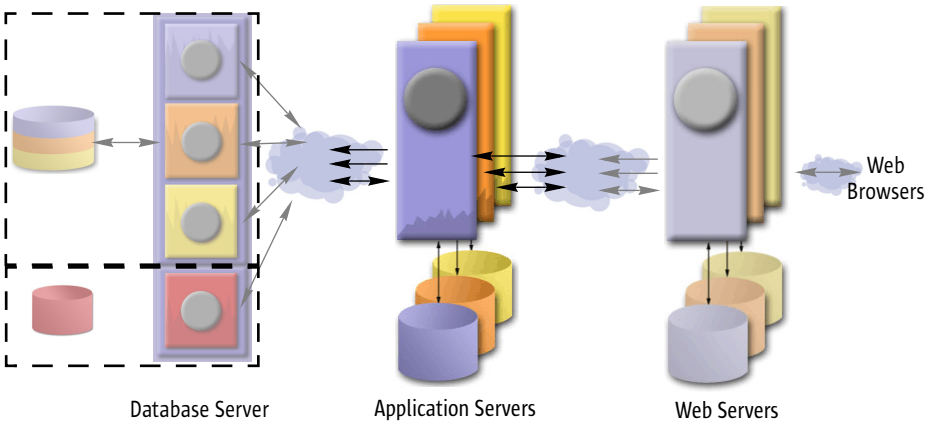


Figure 5. Application servers on separate systems

This example focuses only on the application server components and how they might be consolidated, and how memory and network resources can be managed as well as CPU resources.

Memory Resource Management

Because the business services provided by these application servers are critical to the running of the enterprise, it is extremely important that service levels are maintained and that performance meets expectations. Because most application servers are comparatively new, they may be somewhat less stable than other software such as database management systems which are much more mature. Less mature software, like application servers, may suffer from memory usage that grows over time or grows as the volume of users increases, or both. For these reasons, it may be prudent to place a cap on the amount of memory each application server can use to prevent excess memory usage by one application server workload from impacting the other workloads. Containers through Solaris 9 Resource Manager provide the capability to manage memory resources.

Network Bandwidth Management

The performance of application server software also frequently depends on available network bandwidth in addition to memory resources. Network resources can be managed through Solaris 9 Resource Manager's network bandwidth management facilities. Also known as Solaris 9 IP Quality of Service (IPQoS), this network management technology is fully integrated with the Solaris 9 OS. It provides the ability to classify and prioritize network traffic in accordance with the Internet Engineering Task Force (IETF) Differentiated Services (DiffServ) and IEEE 802.1d standards. These standards allow for the classification of network traffic on the basis of which application is generating the traffic, allowing the system administrator to more easily assign priorities. Network traffic limitations are handled either by the Solaris system or by a DiffServ- and 802.1d-compliant router or switch, and in a way that provides predictable performance levels for critical applications.

Multiple application servers can be consolidated onto a single system, as shown in Figure 6, with Solaris IPQoS used to manage network resources.

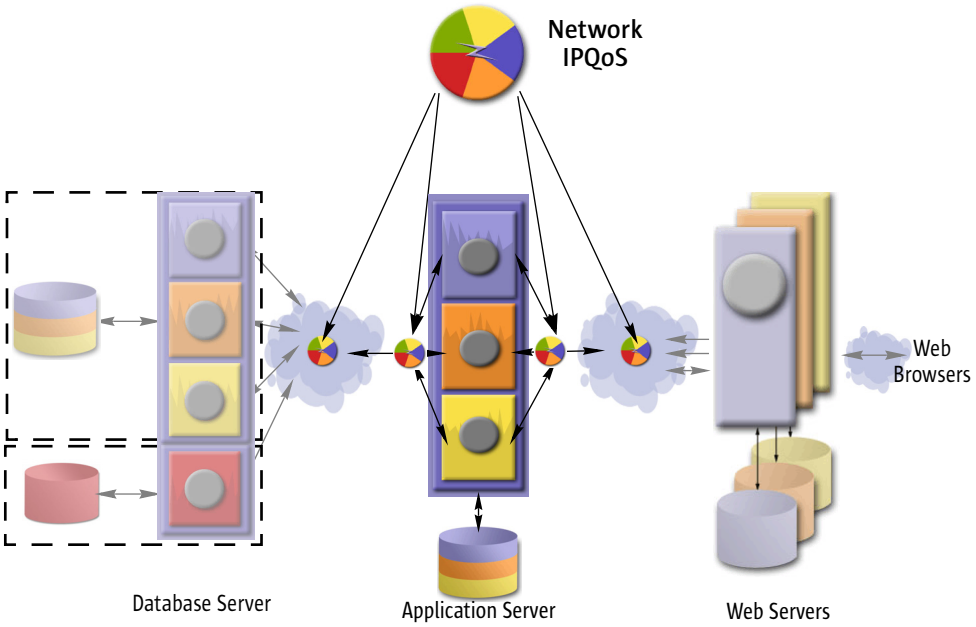


Figure 6. Application servers consolidated into separate Containers

In this example, Solaris 9 IPQoS on the server provides the ability to classify and prioritize network traffic, while a DiffServ- and 802.1d-compliant router or switch provides the same capabilities on the network.

Containers provide many of the features needed to create resource boundaries and flexibly manage CPU, memory, and network bandwidth. The ability to create multiple Containers within a single instance of the Solaris OS allows enterprises to easily reconfigure resources to meet rapidly changing business needs.

Chapter 5

The Evolution of Containers

Sun engineers are working to integrate the Container concept throughout Sun's software and system management product lines and to include enhanced security and fault isolation. The following example illustrates the usefulness of the security and fault isolation attributes of Solaris Containers.

Example: Security and Fault Isolation

A typical enterprise may provide a variety of services via external Web sites. These services could include an employee portal for access to internal networks, a Web store for selling products to customers, and a vendor portal for accessing supply chain applications. Each of these services has common elements, but must be isolated from one another to manage security and prevent errors caused by one service from affecting other services. Historically, isolation has been achieved by hosting these services in their own hardware and software environments. Figure 7 illustrates this environment. The database and application server pieces of the 3-tier architecture are already isolated on their servers, as shown in previous Figures. This example addresses the Web server segment of the architecture.

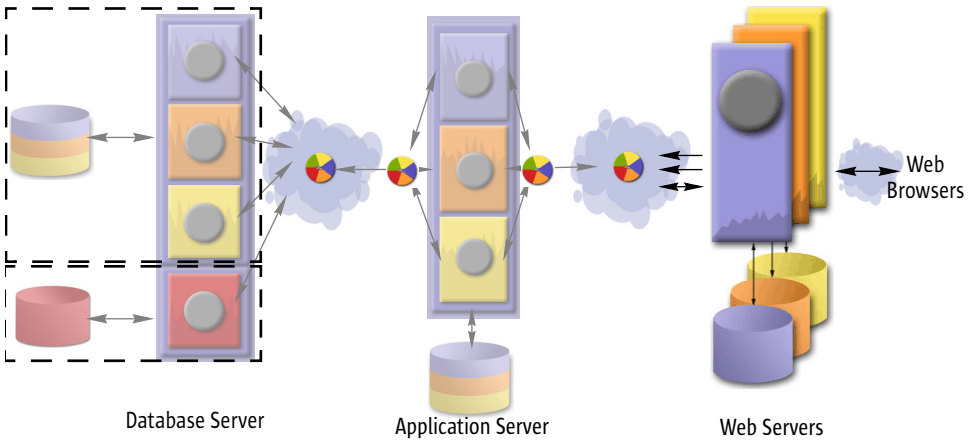


Figure 7. Web servers on separate systems

Solaris Zones Provides Isolation

In the Solaris 10 OS, Sun introduces new enhancements to Solaris Containers and provides a way to virtualize the OS. Solaris Containers provide the ability to run multiple virtualized OS instances on a single instance of the real OS and offer enhanced security and fault isolation in addition to the flexible resource boundaries offered by Containers in the Solaris 9 OS. The Container technology that provides this virtualization is called *Solaris Zones ("Zones")*. These Zones allow for safer consolidation of business services by providing separate name space, including its own IP addresses, file systems, unique root user password and password file, name server, and so on. Because Solaris Containers include Zones technology, this paper will use Containers as the name for the complete virtualized environment and Zones only when needed for clarity.

Communications between Containers use the same standard interfaces, such as the network stack and shared file systems, that are used in communications between separate systems. Should an error occur in a Container, it is designed as a blast shield that isolates and localizes the effects to just the services running in that Container and not in any other part of the system. In the unlikely event that an error is so severe that it would normally cause a server to reboot, the error would instead affect only the Container in which it occurred. The affected Container can restart, automatically or manually, without affecting any other Containers or services in the system and without the inconvenience and downtime normally associated with a full system restart. The ability to restart a single Container also addresses a significant concern in server consolidation projects: Applications that were designed to run by themselves on their own hardware can now be better equipped to share a system and OS instance, increasing overall hardware utilization and lowering TCO for the enterprise. Figure 8 depicts consolidated environments using Solaris Containers.

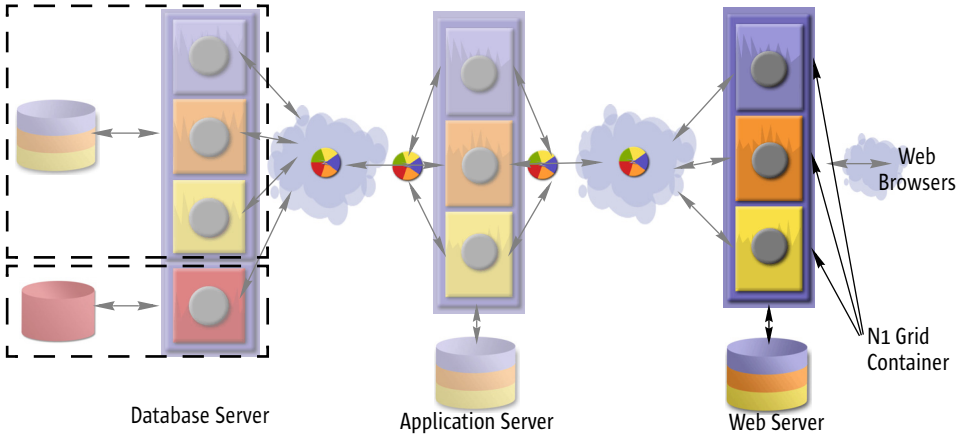


Figure 8. Web servers consolidated into Solaris Containers

Because Zones provide a complete virtual OS, Containers will appear to be completely separate systems from a user's perspective. In order for a user to access a Container, the user must have a valid user ID and use it to log in to the Container in the same way as with any separate system. Any administrative users must also have privileged access to their Container, privileged access limited to the Container in which it is provided for. Thus, workloads in a Container will be protected from common manual errors, such as unintentionally shutting down all databases instead of only a specific database instance. This separation also applies to common operating system tools. When used in a Container, tools such as `ps`, which normally lists all processes running on a system, will list only processes running in the Container, assuring the same level of security that would be expected between systems.

Other Applications and Features

In addition to an integrated command language interface, Sun offers a container management application from the Sun™ Management Center software portfolio called N1 Grid Console. This management tool provides a high-level service management facility that automates many of the container management tasks and simplifies the administrative model. The first release of this application, called N1 Grid Console - Container Manager 1.0 and shown in Figure 9, is available now and supports resource management in the Solaris 8 and 9 environments. Sun plans that the second release of the N1 Grid Console will provide management tools that simplify managing large numbers of Solaris Containers on multiple systems such that they are easily managed using a graphical tool.

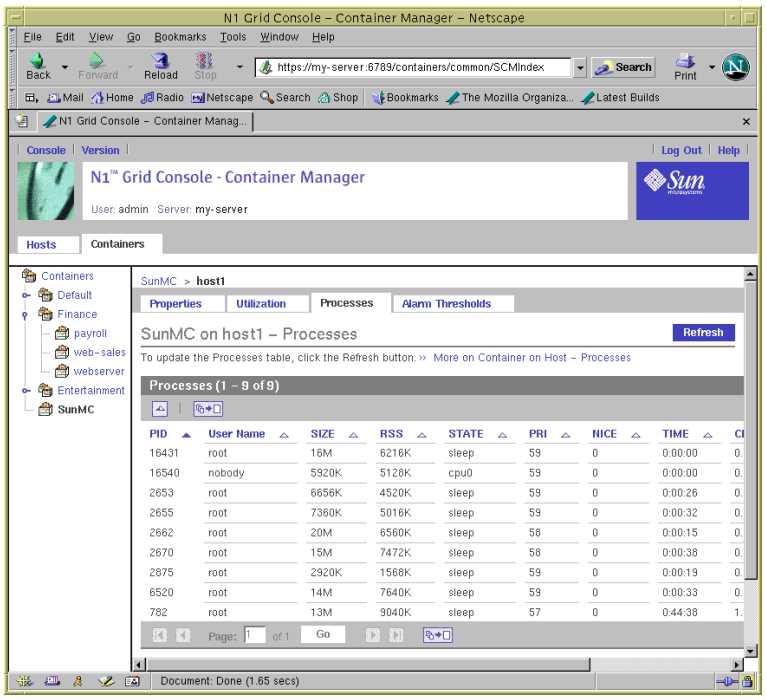


Figure 9. N1 Grid Console - Container Manager

Additional features are planned for Sun's resource management tools, such as the ability to establish *memory sets* and *swap sets* along the same lines as *processor sets* in Solaris 9 and 10 software. Also included in the Solaris 10 OS are *dynamic resource pools*, which provide the ability to specify an action to be taken when a particular event occurs. This could include such events as:

- In the case of a CPU failure, reallocate a CPU from another processor set
- When a CPU board is added to the system, where and how to allocate the new resources
- In the event of a cluster failover, how the resource allocations are changed
- Automatically move CPUs between processor sets depending on preset rules and the load within these sets

Sun plans to enhance all of its products to support this new management paradigm while continuing to provide application programming interfaces (APIs) that promote open standards.

Chapter 6
Conclusion

In order to prosper, businesses must maintain high service levels to satisfy customers and end users while also reducing their IT expenditures. These seemingly opposing objectives require being able to efficiently manage infrastructure and systems. The N1 Grid System and Solaris Containers provide a giant leap forward in manageability that enables businesses to reduce costs through improved efficiency and better resource utilization. This new approach will enable cost savings while continuing to meet the increasing demands of end users for predictable service levels.

Solaris Containers can have a significant impact on total cost of ownership for IT systems. They enable businesses to better manage large-scale systems through workload consolidation, without the traditional concerns about the impacts caused by resources, security, or faults. The primary benefits of Solaris Containers include:

- *Reduced Management Costs* — Applications can be more safely consolidated onto fewer servers, resulting in reduced management complexity and reduced management costs through reductions in OS instances and associated software licensing fees. Further, standardization on the Solaris Container model throughout the Sun product line simplifies service provisioning and reduces training and reconfiguration expenses.
- *Increased Resource Utilization for Lower Cost of Operations* — Dynamic resource reallocation improves resource utilization by allowing unused resources to be shifted to other Containers as needed. In addition, with fault and security isolation provided by Solaris Containers, applications with unpredictable workloads no longer require a dedicated and underutilized system, but may now be safely consolidated with other applications.
- *Improved Service Availability* — Fault isolation and security isolation protect applications from error propagation as well as intentional or unintentional intrusions that can affect performance or availability. The greater consistency and simplicity of this environment also reduces the chance of human errors that might impact availability.
- *Increased Flexibility* — Solaris Containers make it safer for businesses to implement server consolidation with less worry about applications being affected by resource constraints, faults, or security issues. In addition, Solaris Containers are easy to reconfigure and can greatly simplify the task of migrating applications from one set of hardware to another.

Because Solaris Containers isolate multiple execution environments in a single instance of the operating system, they extend the concept of server virtualization down to a sub-CPU level of granularity without unnecessary performance or management overhead. Some other vendors are approaching the need for containment by running separate copies of the operating system on top of a virtual machine. This means that each instruction executed from a contained application must pass through the virtual machine as well as through the operating system, which creates unnecessary system overhead and adds to management complexity by requiring administration of additional copies of the operating system.

In contrast, multiple Solaris Containers can run within a single copy of the Solaris OS to achieve isolation without the performance overhead of a virtual machine and without adding complexity to the management environment.

Solaris Containers change the management model to be more service-centric and allow more standardized automation of underlying resources and system management tasks. Using Solaris Containers allows enterprises to achieve greater systems management efficiency and improvements in utilization of their assets.

Chapter 7

References

Sun Microsystems posts product information in the form of datasheets, specifications, and white papers on its Web site at sun.com. Look for the these and other Sun technology white papers

Title	Location
Software Solutions — N1 Grid	• sun.com/n1
N1 Grid — Introducing Just in Time Computing	• sun.com/software/solutions/n1/wp-n1.pdf
Solaris (formerly N1 Grid) Containers — Feature Story	• sun.com/2004-0330/feature
N1 Grid Console	• sun.com/software/products/grid_console
Solaris 9 Operating System	• sun.com/solaris
Solaris 9 Resource Manager Datasheet	• sun.com/software/solaris/ds/ds-srm
Solaris 9 Resource Manager Software	• sun.com/software/whitepapers/solaris9/srm.pdf
Resource Management in the Solaris 9 Operating Environment	• sun.com/solutions/blueprints/0902/816-7753-10.pdf
Solaris 9 Resource Manager Technical FAQ	• sun.com/software/solaris/faqs/resource_manager.html
Solaris 9 Network Resource Management Using IPQoS	• sun.com/software/whitepapers/solaris9/ipqoswp.pdf
RFC 2475: An Architecture for Differentiated Services	• ietf.org/rfc/rfc2475.txt?number-2475
System Administration Guide: N1 Grid Containers, Resource Management, and Solaris Zones	• docs.sun.com/db/doc/817-1592
Sun/Oracle Best Practices	• sun.com/solutions/blueprints/0101/SunOracle.pdf

Please
Recycle

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN **Web** sun.com

The Network is the Computer™

Sun Worldwide Sales Offices: Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323 Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0 Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India–Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500 Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100 The Netherlands +00-31-33-45-15-000, New Zealand–Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China–Beijing +86-10-6803-5588; Chengdu +86-28-619-9333 Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Saudi Arabia +9661 273 4567, Singapore +65-6438-1888 Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-767-6000, Sweden +46-8-631-10-00, Switzerland–German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888 Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44 0 1252 420000, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800, or online at sun.com/store

09/04 R1.0