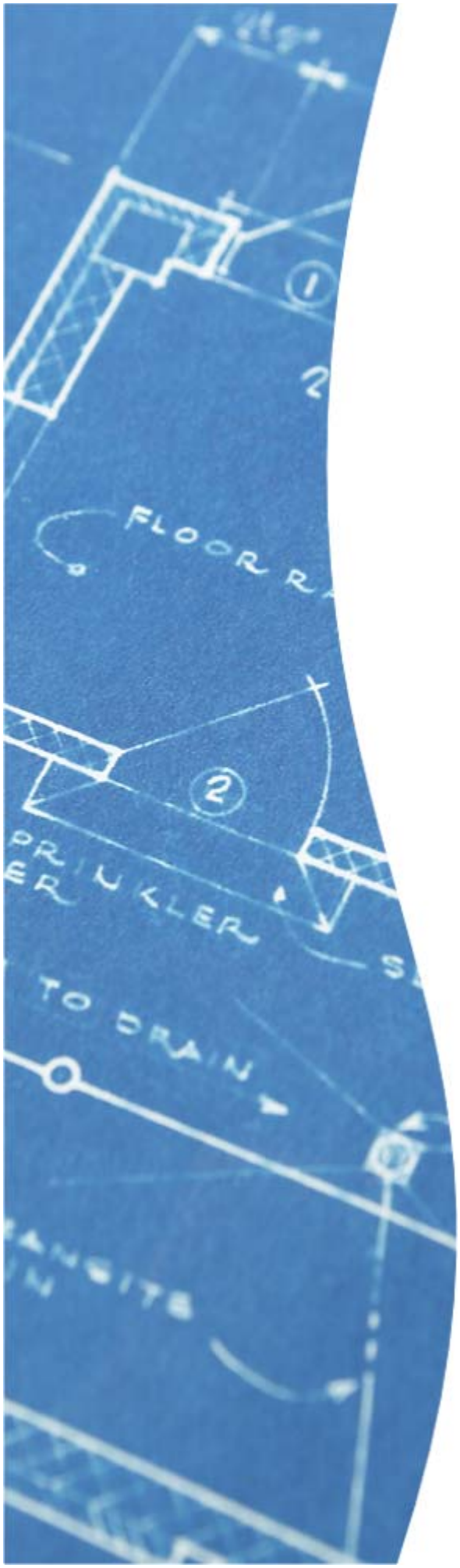

ENERGY EFFICIENCY STRATEGIES
SUN SERVER VIRTUALIZATION TECHNOLOGY

Jeff Savit, U.S. Client Solutions

Sun BluePrints™ On-line — August 2007

Part No 820-3023-10
Revision 1.0, 8/14/07
Edition: August 2007

Table of Contents

Sun Server Virtualization Technology

IT organizations everywhere are being asked to do more with less, and nowhere do they face more hard limits than in the space, power, and cooling capacities of their datacenters:

- Legacy, silo-architected applications have contributed to server sprawl because each server is sized for maximum workload requirements of a single application, and resources generally cannot be shared. The result is a large number of aging, energy-inefficient servers that run with low utilization levels most of the time.
- State-of-the-art servers, such as those from Sun, put so much computing power in the hands of IT organizations that they often can be tasked to support multiple applications at the same time. Consolidation efforts, however, are hampered by the need to align operating system levels, administrative domains, and security contexts on single operating system instances.

Virtualization is a key technology that addresses these issues. It helps consolidate legacy applications from multiple obsolete hardware platforms onto a smaller number of up-to-date, more powerful, and more energy-efficient servers. It supports moving today's applications from a large set of under-utilized servers to a smaller set of more powerful servers, helping to reduce the number of servers to house, power, cool, and maintain. Increasing utilization levels helps to reduce inefficiency, helping with the space, power, and cooling crunch.

Nearly every server virtualization technique provides a mechanism for putting an execution environment into a container that can be moved between servers, giving IT organizations the freedom to change the allocation of services to servers. This helps maintain service levels by allowing IT organizations to vacate a server for maintenance or upgrade without a large effort to re-host the services running on it. Rather than dedicating resources to an application, and wasting excess dedicated capacity, IT organizations can deploy them in virtual environments that can be scaled up or down, depending on their requirements.

The domain of virtualization includes server, storage, and network virtualization techniques. This Sun BluePrints™ series article describes two server virtualization technologies that are specific to Sun: Solaris™ Containers and Logical Domains. The article includes the following topics:

- "Virtualization Defined" on page 2 defines what server virtualization means in the context of this article,
- "Virtualization Technologies" on page 2 offers a brief overview of the history of virtualization technologies and the spectrum of possibilities they open up,
- "Solaris Containers" on page 4 provides an overview of Solaris Containers, and
- "Logical Domains" on page 7 provides an overview of Logical Domains, a virtualization technology for UltraSPARC® T1 and T2 processor-powered servers.

Virtualization Defined

Virtualization has been in commercial use since the 1960s, and is experiencing a resurgence because of its ability to help solve so many issues in today's datacenters. There are many kinds of virtualization. Storage virtualization presents mass storage capacity and service levels as a virtual resource without exposing the details of disk subsystems. Network virtualization provides virtual networks with security isolation and different service levels using the same physical infrastructure. Finally, server virtualization, the focus of this article, provides virtual servers on physical ones.

Server virtualization creates the illusion of multiple computing environments on a single computer. This allows IT organizations to run multiple, different application environments, even multiple operating systems, on the same server at the same time. Each behaves as if it had its own dedicated computer system.

Sun supports several different virtualization technologies, each of which provide different degrees of isolation, resource granularity, and flexibility. Sun supports virtualization technologies that allow multiple OS (and application) instances to run on the same server, while each instance has the illusion of owning its own hardware resources. Sun Dynamic System Domains provide hardware partitioning capabilities on selected Sun SPARC® Enterprise servers. Virtual machine technology including Xen, VMware Infrastructure 3, and Microsoft Virtual Server software provide software emulation of underlying hardware that supports multiple environments even on systems with only a single processor. Sun also has innovative virtualization technology that overcomes the limitations of both Dynamic System Domains and virtual machine environments: Solaris Containers provide security and resource isolation that allows multiple virtual Solaris OS environments to share the same OS instance. Logical Domains offers a hybrid of partitioning and virtual machine technology, allowing multiple OS instances to share the same hardware without the high overhead typical of virtual machine environments.

Virtualization Technologies

There are two different types of hardware virtualization technologies that have been in use for years, domains (or hard partitions) and virtual machines. (Figure 1):

- Domains are a hardware or firmware capability that divides a computer system's assets so they can host multiple OS instances. This was introduced to the mainframe world in the 1980s by Amdahl Corporation's Multiple Domain Facility, and introduced in open systems by Sun's Dynamic System Domains beginning in the mid-1990s.
- Virtual machines date to the 1960s with IBM's VM operating system. Operating system software, called a *hypervisor* or *Virtual Machine Monitor* (VMM), creates virtual machines, each of which has the illusion of owning its own hardware and is capable of running its own separate operating system instance.

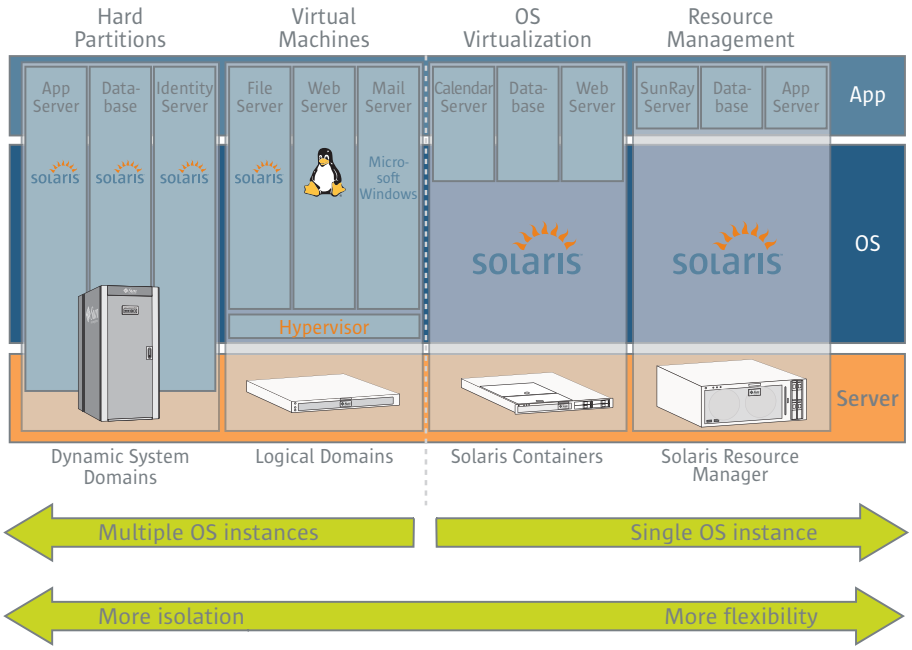

Figure 1. Sun offers a range of hardware and software-based virtualization technologies.

Today, in addition to Dynamic System Domains on supported UltraSPARC processor-powered Sun Fire™ servers and SPARC Enterprise servers, Sun supports flexible Logical Domains on its UltraSPARC T1 and T2 processor-powered Sun Fire servers. It supports Solaris Containers on all of its servers, and VMware Infrastructure 3 software on its x64 server product line.

Domains vs. Virtual Machines

Although there is some overlap between these categories, the simplest way of distinguishing between them is that domains are a hardware or firmware function that subdivides a computer's physical CPU and RAM. Domains, or partitions, typically have low overhead and high reliability with strong isolation between different domains. They usually have a fixed, architectural limit on the number of OS instances or amount of resource granularity or sharing.

Virtual machines are typically supported by a hypervisor, which is itself an operating system implemented in software. The hypervisor time slices between guest operating systems just as a conventional multiprogramming OS time slices between application processes. Hypervisors usually don't have fixed limits on the number of OS instances, and often permit fine-grained resource control. Many virtual machines can run on a single CPU, but have substantial overhead that can affect performance and limit the number of OS instances that can be run effectively.

Hypervisors give virtual machines the illusion that they can run with the processor in both privileged (OS level) and unprivileged (application program level) states. When operating systems execute privileged instructions, for tasks like changing memory mapping for virtual memory or performing I/O, these state-changing instructions and events in one virtual machine must not affect others. To enforce this, hypervisors run virtual machines without full machine access privileges.

This causes a program exception, or trap, which causes a context switch to the hypervisor every time the guest virtual machine's operating system executes a privileged instruction. The hypervisor then determines what privileged instruction the virtual machine was attempting to execute (start I/O, enable or disable interrupt masks, flush cache, switch address spaces), and then the hypervisor emulates that instruction on behalf of the guest operating system. Similarly, the hypervisor must reflect events such as timer or I/O interrupts to the guest OS.

Hypervisors give the guest OS the illusion that it has its own machine, but at a price in complexity and substantial overhead that sometimes exceeds 50 percent. The memory footprint of multiple complete operating systems - kernels, libraries and applications, and the housekeeping tasks each OS runs also adds to the overhead in running multiple virtual machines. Virtual memory operating systems, such as Solaris, Linux, and Microsoft Windows, running under a virtual machine hypervisor can suffer from double paging during memory shortages: a guest OS may need to swap out an old page to free up memory, but the hypervisor must read the page from disk so the guest OS can write it to its own swap file and then discard the contents. There are techniques to reduce these, and other pathological conditions, but this is a complex problem that can severely impact performance.

Solaris Containers

Solaris Containers is a virtualization technology from Sun that address the limitations of domains and virtual machines. Solaris containers combine Solaris Zones, a technology that provides isolation, with Solaris Resource Manager, which can be used to allocate resources to zones. Solaris Containers, introduced with the Solaris 10 Operating System, provides OS-level virtualization because it gives the appearance of multiple OS instances rather than multiple physical machines. Instead of virtualizing the hardware architecture of the underlying machine as described above for virtual machines, Solaris Containers provides multiple instances of the OS abstractions that Solaris provides to applications. This can be accomplished without the high overhead typical of virtual machines because there is no need to do processor instruction-level emulation. Another advantage of Solaris Containers is that because they run on the Solaris OS, they run on any SPARC, AMD Opteron™, or Intel® Xeon® processor-powered computer that runs the Solaris OS.

Solaris Container Implementation

The Solaris OS installs with a single global zone that has system-wide access to the entire Solaris instance. The global zone can be used to configure up to 8191 non-global zones. Each zone operates in a private, isolated context, with its own network address and identity, security context, and potentially different applications. Figure 2 illustrates how a single Solaris OS instance can support multiple customers on a single hardware platform. Customers can run their own applications and have access to their own resources.

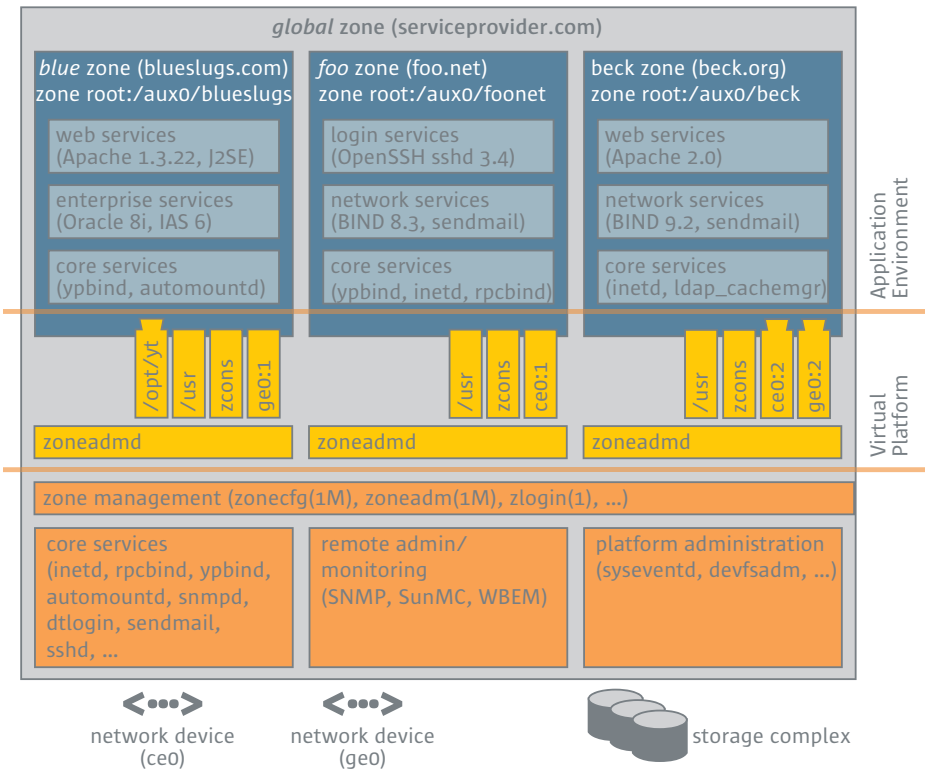

Figure 2. An example service provider server supports three different zones for three different customers, each of which run their own applications and have access to their own resources.

When a zone is created, it is provided with the following:

- A private startup and shutdown sequence and service management facility: each zone can be separately booted or rebooted without affecting other zones, and can run its own set of services.
- Private IP addresses and port ranges: these allow each container to be separately addressed on the network so that each container can independently host its own network applications. For example, an organization can create multiple zones, each of which hosts its own Web site using its own unique HTTP and HTTPS ports. This can be used to isolate different customers, or to enable developers to create and test

applications in private environments without being able to harm or conflict with one another. Applications in different zones communicate with one another via TCP/IP, just as if they were on separate physical machines. Inter-zone network traffic is handled by memory-to-memory transfer rather than the network interface. This allows extremely high bandwidth and message rates, and it allows network traffic to pass between zones securely, without the possibility of it being intercepted.

- Private authentication and name services: each zone can independently use its own password, DNS, LDAP or NIS naming services for users, passwords, services, and host names. A given userid can be defined in one zone and not exist in other zones. Having root access in one non-global zone has no value in the other zones.
- Private process lists: each zone has its own processes and cannot see processes in the global zone or other non-global zones. This provides privacy and safety for the different zoned environments, just as if they were on separate physical computers.
- Separate security, resource management, and failure scopes: each zone can have its own rules for authentication and authorization, allocation of computer resources, and containment of application or even system faults.
- Private file systems: these appear to the zone as the root of the file system. They can be set up in two ways: *whole root* and *sparse root*. Sparse-root zones optimize physical memory and disk space usage by sharing some directories, such as `/usr` and `/lib`, in read-only mode. Sparse-root zones have their own private file areas for read-write directories, including `/home`, `/etc`, and `/var`. Whole-root zones increase flexibility by providing complete read-write file structures for the zone. This lets the zone administrator install products or patches into what would be read-only directories in a sparse-root zone, but at the cost of increased disk space. Additionally, the global zone administrator can assign zones devices, UFS mount points, ZFS pools, loopback filesystems and disk assets. Filesystems that are private to zones typically are part of the global zone's filesystem hierarchy, and are mounted through loopback filesystem mounts (Figure 3).

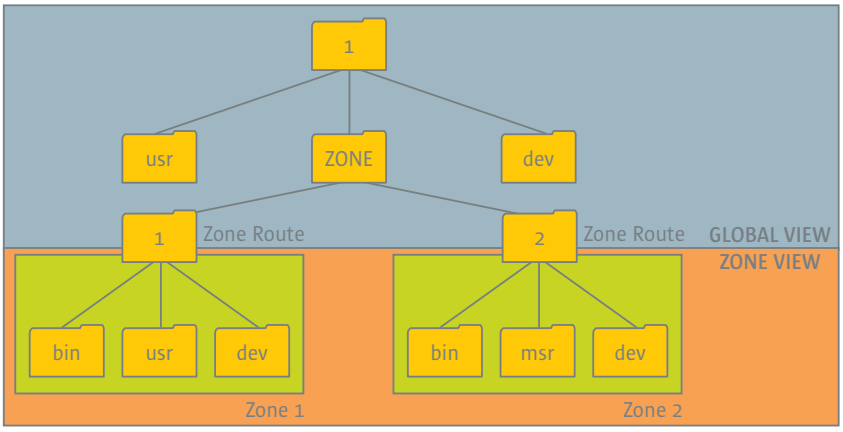

Figure 3. Filesystems that are local to a zone are typically part of the global zone's file hierarchy.

Low Overhead

Unlike virtual machines, Solaris Containers have minimal overhead. Isolation between zones is accomplished by restricting the scope of system calls to the zone from which they are issued, rather than the CPU-intensive task of emulating hardware architectures and instruction sets in software. This makes it possible to create hundreds, even thousands, of Solaris Containers on a single processor. Because of this negligible overhead, and unlike partitioning or virtual machines, Solaris Containers can be created in large numbers, for example, to give individual developers safe, isolated test environments.

Solaris Containers are available on any platform that runs the Solaris 10 OS, whether it is based on UltraSPARC, AMD Opteron, or Intel processors. Zones can be created by the global zone administrator in a few minutes. Solaris Containers can be cloned from a golden master, or even detached from one Solaris system and attached to another. Solaris Containers are included with the Solaris 10 OS, making it possible to have high-performance, low-overhead, low-cost virtualization on any Solaris OS platform.

All Solaris containers on the same platform run under the same OS kernel, so it is not possible to mix and match software that requires conflicting OS versions. The use of a single OS instance has the benefit that a patch or update applied to the global zone is applied to all of the non-global zones at once, instead of requiring individually maintained, separate OS instances within domains or virtual machines.

Logical Domains

Sun has recently introduced Logical Domains (LDoms), a virtualization technology for systems based on the UltraSPARC T1 and T2 processors, including the Sun Fire T1000 and T2000 servers. LDoms support separate OS instances without the overhead of virtual machines, and with more flexibility than hardware partitioning features such as Dynamic System Domains.

Minimal Context-Switch Overhead

Traditional virtual machines time slice physical CPUs among multiple virtual machines, intercepting and emulating privileged instructions. This is complex and CPU intensive — even simple context switching between virtual machines can require hundreds of clock cycles.

Time slicing is based on the historical design of computer systems where physical CPUs were relatively rare and expensive, and hence must be shared and time sliced. The UltraSPARC T1 and T2 processors turn the assumption of expensive and rare CPUs upside down. The UltraSPARC T1 processor's Chip Multi-Threading (CMT) design provides up to 32 logical CPUs (or *strands*) on a single processor, while the UltraSPARC T2 processor supports up to 64 strands. Using this multi-threading model, logical CPUs

become plentiful, and they can be allocated to logical domains and dedicated for their use. This eliminates the cost of context switching a single processor between virtual machines.

Although context switching still occurs between processes in the same logical domain, the UltraSPARC T1 and T2 processors significantly reduce context switch overhead. Context switching typically occurs when a domain references memory that is not currently in cache. Accessing main memory to fulfill the request requires many clock cycles on any system architecture. While memory is fetched, a logical CPU stalls while executing the single instruction causing the cache miss. By switching to another CPU strand on the same physical CPU core, UltraSPARC T1 and T2 processors lets another logical CPU continue instruction processing, during time that is unused memory wait time on most processors. On most CPUs, cache misses result in processor idle time. In contrast, the UltraSPARC T1 and T2 processors can use what is dead time on other processors to continue doing useful work.

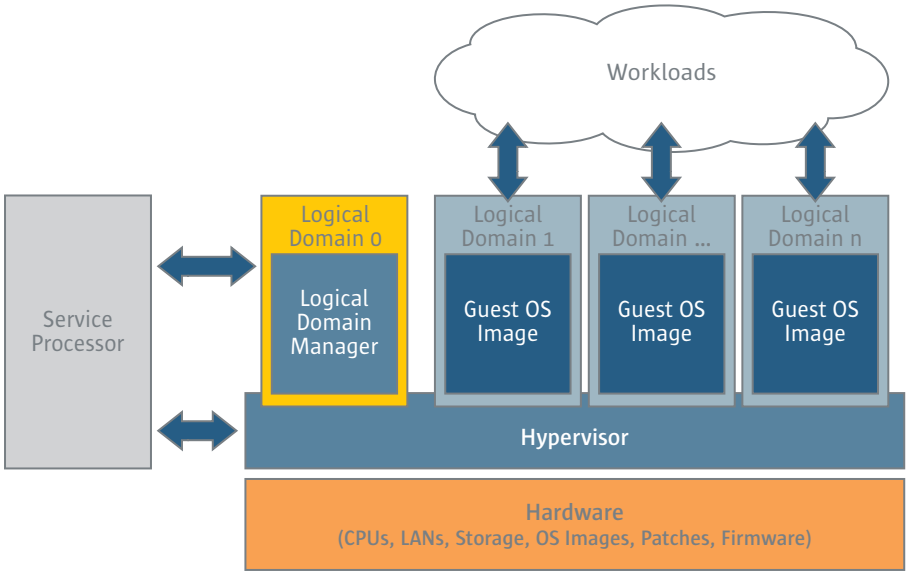

Figure 4. Logical domains allocate UltraSPARC T1 and T2 processor logical CPUs and memory to logical domains, allowing multiple guest operating system instances to be supported on the same CPU, but without the high cost of context switches between OS instances.

LDom Implementation

Logical domains are implemented using a very small hypervisor that keeps track of which logical CPUs, RAM locations, and I/O devices are assigned to which domain, and it provides communication channels to service domains (Figure 4).

Logical domains are designed for simplicity. Since each domain has its own dedicated logical CPUs, a domain can change its state (such as enable or disable interrupts) without having to cause a trap and emulation. That can save thousands of context switches per second, especially with workloads high in network or disk I/O activity. Since each logical CPU has its own private context in hardware, UltraSPARC T1 and T2 processors can switch between domains in a single clock cycle, not the several hundred needed for most virtual machines.

Using logical domains, logical CPUs can be added to or removed from a Solaris instance individually and dynamically. A CPU can be added to or removed from a logical domain without requiring a reboot. The Solaris OS can immediately make use of a dynamically added CPU for additional capacity. Each domain has its complement of disk, network, and cryptographic resources, which are assigned by commands issued by the control domain, which also runs the Solaris OS (Figure 5).

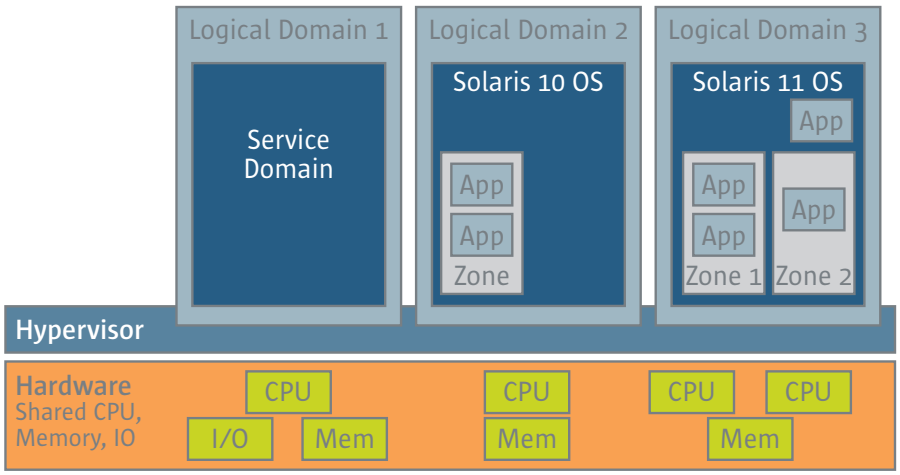

Figure 5. A service domain allocates CPU and memory resources to different logical domains, and can dynamically change the allocation without requiring a reboot.

Virtual network and disk I/O is provided to logical domains by service domains. A service domain is a separate instance of the Solaris OS that runs in a domain, but which has direct connections to network and disk devices. Regular domains, used for application processing, have network and device drivers that communicate with the service domains through communication channels provided by the UltraSPARC T1 and T2 processor firmware.

This allows service domains to export virtual LAN and virtual SAN services to other domains. Network resources can be set up for securely isolated VLANs, and disk resources can reside in virtual SANs, with actual disk images residing in physical disks,

disk slices, ZFS pools, or even standard UFS files. The Solaris OS instances can use these disk and network resources exactly as if they were dedicated physical resources.

Service domains are designed for high availability. They can be set up in pairs so that system and guest operation can continue if one is taken down for service, or due to a fault.

Logical domains make it possible to provide low-cost, high-efficiency domains on UltraSPARC T1 and T2 processors with highly granular and dynamic resource allocation. They can be used in parallel with Solaris Containers for the highest degree of flexible virtualization. Separate OS instances can be configured when different OS kernel levels are required, with each OS instance hosting many lightweight, virtualized Solaris Container.

Summary

Virtualization technology is a key to transforming an IT organization's server, storage, and network devices into a shared, global pool of resources. This can help reduce space, power, and cooling requirements while simultaneously helping make datacenters more flexible and agile. The two server virtualization technologies discussed in this Sun BluePrints article — Solaris Containers and Logical Domains — are of particular importance because of the low overhead they impose on the host server platform.

Solaris Containers abstracts a single Solaris OS instance into multiple containers, giving the appearance of a dedicated OS instance to each container. Solaris Containers is an excellent virtualization technology when a number of applications can run on the same OS instance. Logical Domains are used to partition the server CPU and memory, abstracting the hardware itself to multiple guest operating systems. Logical Domains can be used to support multiple OS instances on the same server, and each Solaris OS instance can also support multiple containers. With the ability to use each of these technologies independently, or in conjunction, Sun customers have a powerful set of tools for server virtualization in their datacenters.

About the Author

Jeff Savit is a Sun Microsystems Principal Engineer, Datacenter Ambassador, and OS Ambassador working in areas including the Solaris OS, Virtualization, Linux, Java™ technology, migration, and application and system architecture. Jeff is expert in virtualization and systems resource management, where he has years of production use and internals-level expertise which he now applies to Sun virtualization technologies.

Mr. Savit wrote or coauthored several books: "Enterprise Java", "VM & CMS: Performance and Fine Tuning," "VM/CMS Concepts and Facilities," and "IBM

Mainframes," and contributed to others: "VM Applications Handbook," "The REXX Handbook," and "The VM/ESA Handbook." He has also been published in SIGPLAN Notices, a journal of the Association of Computing Machinery. Mr. Savit has been a featured speaker at conferences including InternetWorld, SHARE, GUIDE, Australasian SHARE/GUIDE, and the Modula-2 Users Group. He is past president of the Metropolitan VM User Association. He also served on a computer science delegation to the People's Republic of China.

References

This Sun BluePrints article is part of Sun's Datacenter Reference Guide:

Black, Keith, *et. al.*, "Datacenter Reference Guide," *Sun white paper,* July 2007.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is

`http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at:

`http://www.sun.com/blueprints/online.html`

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN (9786) **Web** sun.com

Printed in USA 8/2007