OCTAVE ORGERON

an introduction to logical domains

PART 1

Octave Orgeron is a Solaris Systems Engineer and an OpenSolaris Community Leader. He currently offers consulting in the financial services industry but has experience in the e-commerce, Web hosting, marketing services, and IT technology markets. He specializes in virtualization, provisioning, grid computing, and high availability.

unixconsole@yahoo.com

IN RECENT TIMES, VIRTUALIZATION has become a requirement for many businesses looking to consolidate physical servers and increase utilization. This has led to many innovations at both the software and the hardware level to address the virtualization requirement. One such innovation is Sun Microsystems' new product called Logical Domains, or LDoms. This allows a single physical server to be virtualized into multiple discrete and independent operating system instances. LDoms present many opportunities for consolidation in modern data centers where physical space and power are at a premium.

This is the first of three articles that will introduce you to the Logical Domain technology. In this article, I will introduce the basic concepts and components of this new technology.

The Niagara Processor and the UltraSPARC Hypervisor

The Niagara processor is a chip multithreading design that leverages the power of multiple CPU cores running many hardware threads simultaneously. The first generation, known as the UltraSPARC-T1, was introduced on the Sun Fire T1000 and T2000 servers. The processor has up to eight CPU cores with four hardware threads each, for a total of 32 threads. The CMT design enables the processor to achieve significant increases in performance over the UltraSPARC IIIi processor for multithreaded applications. The processor has unique features, such as a cryptographic unit that traditionally would require an add-on accelerator card. In the future, the Niagara 2 platform will integrate more advanced features, such as 10-Gb Ethernet, enhanced cryptography, and enhanced floating-point performance. One of the more interesting features of the Niagara processor family is the support of a hypervisor for virtualization.

Hypervisors provide a virtualization platform for running multiple operating system instances. Hypervisors have been around since the 1960s, starting with IBM's CP/CMS, the ancestor of IBM's current z/VM solution. Until recently, such technology was only found on such proprietary platforms. However, with the advent of Xen and VMware ESX, hypervisors are becoming more commonplace. The hypervisor found in Sun's Niagara architecture, known as the UltraSPARC hypervisor, is a new addition to this growing virtualization methodology.

The UltraSPARC hypervisor is a thin layer of software stored within the ALOM CMT firmware. It creates a layer of abstraction between the operating system and the physical hardware. Traditionally, operating systems have the concept of nonprivileged and privileged access to the underlying hardware. The hypervisor introduces an additional layer of privileged access, known as hyperprivileged access. Hyperprivileged access enables the hypervisor to either expose or hide resources from an instance of an operating system. This allows resources to be grouped into logical partitions or domains. This is similar to Sun's Dynamic System Domains, with the main difference being that the resources are not electronically partitioned, but virtualized.

Resources such as CPU threads, cryptographic threads, and memory are partitioned into a logical domain. Other resources are virtualized and serviced through the use of Logical Domain Channels, or LDCs. LDCs provide secure communication and data pathways between LDoms and the hypervisor. This allows an operating system in one LDom to make an I/O request, which is serviced by another LDom that has privileged access to the underlying hardware. The abstraction reduces the I/O overhead in one LDom and passes it to another LDom that is capable of completing the request.

However, the hypervisor cannot accomplish this on its own. The processing of I/O requests requires CPU cycles, device drivers, etc. There is also the aspect of configuration and management of the platform as a whole. These different aspects of the platform lead to the division of responsibilities to unique logical domain types.

Logical Domain Types

There are several types of logical domains that can be configured. Each type plays a specific role in the logical domain architecture. Some of these roles overlap, but they can be separated for flexibility. The basic differences are shown in Table 1.

Logical Domain Type	Description
Guest	Domain that is a consumer of virtualized devices and services
I/O	Domain that has privileged access to a PCI-E controller but does not provide virtualized devices or services to guest domains
Service	I/O domain that has privileged access to one or more PCI-E controllers; provides virtualized devices and services to guest domains
Control	Service domain that runs management software to control the hypervisor configuration of the platform

TABLE 1: LOGICAL DOMAIN TYPES

GUEST DOMAINS

A guest domain is a virtualized environment that has no direct access to the underlying physical hardware beyond the CPU threads, cryptographic threads, and memory resources. It does not have direct ownership of any hardware devices. A guest domain does not provide virtual services or devices to other LDoms. It is a consumer of the virtual services and devices provided to it by the control and service domains. Guest domains consist of the following components:

- CPU threads
- Cryptographic MAU threads
- Memory
- Virtual console
- Virtual OpenBoot PROM
- Solaris 10 Update 3 or above
- Virtual networking
- Virtual storage

The guest domain is the target virtual environment for deploying applications and services. It functions as a normal Solaris instance with the exception that its underlying networking and storage are completely virtualized. This means that normal Solaris operations such as Jumpstart, package and patch management, running network services, account management, etc., all function without any changes. Also, advanced features such as boot disk mirroring or network multipathing function transparently. It is even possible to run Solaris Containers within a guest domain, adding another layer of virtualization.

I/O DOMAINS

An I/O domain is a virtualized environment that has privileged access to a portion of the underlying hardware platform. Specifically, an I/O domain has privileged access to a PCI-E controller and the devices that are connected to its ports. This allows it to have direct control over network ports and storage that are connected to that PCI-E device tree. However, an I/O domain does not virtualize access to its hardware for guest domains. As such, I/O domains differ from guest domains by having:

- Privileged access to a PCI-E controller and its devices
- Physical access to networking
- Physical access to storage

I/O domains may be useful for applications such as databases that require direct or raw access to storage devices. However, they do consume an entire PCI-E controller and the devices connected to it. This can reduce the flexibility of the hardware platform, but it may be of some use for specific applications.

SERVICE DOMAINS

Service domains are virtual environments that provide virtual resources to guest domains. The service domain takes ownership of one or more PCI-E controllers, similarly to an I/O domain. However, it virtualizes the devices connected to those controllers as a service for guest domains. This is accomplished by having the kernel device drivers, within the service domain, front-ended by virtual device services. When a guest domain interfaces with a virtual device, the request is handled by the corresponding service domain through LDCs. This happens transparently to the operating system in the guest domain. Service domains differ from I/O domains by having:

- Privileged access to one or more PCI-E controllers and their devices
- Virtualized devices and services for guest domains

THE CONTROL DOMAIN AND THE LOGICAL DOMAIN MANAGER

The control domain is a service domain with management software that is capable of configuring the platform. By default, the control domain is the first service domain for the platform and as such is referred to as the primary domain. This LDom can be accessed directly by the physical hardware console. This dual role allows the primary domain to configure, manage, and provide virtual services for the platform. The differences between the primary domain and a standalone service domain involve the former's physical hardware console, Logical Domain Manager software, and virtual console concentrator.

The Logical Domain Manager (LDM) software is the management layer that is aware of the mappings between the physical and virtual resources. The LDM software provides an easy command-line interface for the configuration and management of LDoms. Through the use of LDCs, the LDM software can control the hypervisor configuration. It also configures virtual services in service domains, controls dynamic reconfiguration, and provides virtual consoles for each LDom.

It is important to note that the control domain is the only domain that runs the LDM software and is responsible for configuring the server as a whole. For standalone service or I/O domains, no additional software is required beyond the standard Solaris installation.

Virtual Services and Devices

Logical domains are consumers in one way or another of virtualized services and devices. These virtualized services and devices form the building blocks for logical domains. They provide the processing, memory, and I/O components for logical domains. Tables 2 and 3 identify and describe virtual services and device types.

Virtual Services	Description
VLDC	Virtual Logical Domain Channels. These act as communication channels for logical domains and the hypervisor. Services such as dynamic reconfiguration, FMA events, Service Processor events, and communications between guest domains and services domains utilize VLDCs.
OBP	OpenBoot PROM. Each logical domain has its own OpenBoot PROM instance. The NVRAM variables are stored within the hypervisor.
VCC	Virtual Console Concentrator. The VCC provides a virtual console for each logical domain. This can only be provided by the control domain.
VSW	Virtual Switch Service. VSW provides virtual network access for guest domains to the physical network ports.
VDS	Virtual Disk Service. VDS provides virtual storage services for guest domains.

TABLE 2: VIRTUAL SERVICE TYPES

Virtual Devices	Description
VCPU	Virtual CPU. Each UltraSPARC-T1 CPU consists of 4, 6, or 8 cores with 4 threads. Each thread can be allocated as a virtual CPU.
MAU	Mathematical Arithmetic Unit. Each Niagara CPU core has a thread to a Cryptographic MAU, which provides accelerated RSA/DSA encryption.
Memory	Physical memory can be virtually mapped into a logical domain.
IO	PCI-E controller that is allocated to a service domain.
VCONS	Virtual Console. This port in a guest domain is connected to a VCC service in the control domain.
VNET	Virtual Network. This port in a guest domain is connected to a VSW service in a service domain.
VDSDEV	Virtual Disk Service Device. The VDSDEV is a physical storage medium that is virtualized by a VDS in a service domain.
VDISK	Virtual Disk. VDISK in a guest domain is connected to a VDS in a service domain.

TABLE 3: VIRTUAL DEVICE TYPES

In this article I have introduced the basic concepts and components of logical domains. By understanding the relationships among the different logical domain types and their virtual resources, it will be easier to explore this new technology. In my next article I will explain the installation and configuration of logical domains in detail.

RESOURCES

Home page for Logical Domains:
http://www.sun.com/servers/coolthreads/ldoms/index.xml.

Documentation for Logical Domains:
http://docs.sun.com/app/docs?q=ldoms.

Sun BluePrint document:
http://www.sun.com/blueprints/0207/820-0832.html.

Sun BigAdmin site for LDoms: http://www.sun.com/bigadmin/hubs/ldoms/.

Home page for OpenSPARC source code and specifications:
http://www.opensparc.net/.