OCTAVE ORGERON

# an introduction to logical domains

## PART 2: INSTALLATION AND CONFIGURATION

Octave Orgeron is a Solaris Systems Engineer and an OpenSolaris Community Leader. Currently working in the financial services industry, he also 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 THE AUGUST 2007 ISSUE OF *;LOGIN:*,** I explained the Logical Domains (LDoms) technology from Sun and what you can do with it. In this article, I will walk you through the installation process, explaining key requirements for proper installation, as well as suggesting choices you should make during the process.

## Prerequisites

For LDoms to function, you will need the correct platform, firmware, OS release, patches, and the Logical Domain Manager software.

Currently, LDoms are only supported on the UltraSPARC T1 (Niagara I) servers, as they are the only UltraSPARC platform with a hypervisor. You can find more information about those servers on Sun's site [1]. In the future, more platforms will be supported as the next-generation Niagara II servers are released.

Each of these servers requires firmware updates to fully support LDoms [2]. The firmware will update and enable the hypervisor software that is contained in the ALOM CMT service processor, which provides the platform lights-out management. In the installation section, you'll find an example of updating the firmware on a Sun Fire T2000.

It is important to have the correct Solaris version to support LDoms. Without the platform and driver support, LDoms will not function properly. The following versions of Solaris are supported:

- Solaris 10 11/06 Update 3 or higher [3]
- Solaris Express Build 57 or higher [4]

Solaris 10 is the commercial version of Solaris and Solaris Express is a preview of Solaris 11 based on the OpenSolaris source code. Solaris 10 should be installed if you require commercial support from both Sun and third-party vendors. However, Solaris Express can be utilized when such requirements are not a concern. Solaris Express provides a preview of developments and features that you will not find in Solaris 10. In this article, Solaris 10 will be utilized. When installing the operating system, it is important to keep in mind that it will become the control domain for the platform.

With Solaris 10, there are patches required to enable full LDoms support. These patches should be downloaded [2] and installed according to the installation instructions included with them.

The last component, the Logical Domain Manager (LDM) software bundle [2], includes the required software packages, installation script, and pointers to online resources.

## Installation

Once the operating system and any required patches have been installed, the installation of the firmware and LDM software can begin.

Upgrading the firmware is a multistep process that will require downtime for your server. The first step is to download the corresponding firmware patch for your server [2]. The patch will contain a firmware image file, an installation tool, and some documentation. The installation tool, sysfwdownload, will upload the image to the ALOM CMT service processor. The following example is based on a Sun Fire T2000 running Solaris 10:

```
# unzip 126399-01.zip
# cd 126399-01
# ./sysfwdownload ./Sun_System_Firmware-6_4_4-Sun_Fire_T2000.bin

.......... (10%).......... (20%).......... (30%).......... (40%).......... (51%)
.......... (61%).......... (71%).......... (81%).......... (92%).......... (100%)

Download completed successfully.
```

However, this does not upgrade the firmware. It merely uploads it to the ALOM CMT service processor. To perform the upgrade, you will have to first shut down the server:

```
# shutdown -y -g0 -i 5 now
```

Once the server has shut down, you will have to switch to the ALOM CMT console in order to upgrade the firmware. The console can be reached through the serial port or through the network management port [5]. It is important to ensure that the platform key switch is set to NORMAL to enable the firmware upgrade. Once that is accomplished, the firmware can be upgraded with the flashupdate command:

```
sc> setkeyswitch -y normal
Keyswitch is in the NORMAL position.
sc> flashupdate -s 127.0.0.1

SC Alert: System poweron is disabled.
................................................................................
................................................................................
......

Update complete. Reset device to use new software.

SC Alert: SC firmware was reloaded
sc> resetsc
Are you sure you want to reset the SC [y/n]?  y
```

Once the ALOM CMT reboots, the firmware upgrade is completed. You will notice a change in the versions of the hypervisor, OpenBoot PROM, and the POST diagnostics:

```
sc> showhost
Host flash versions:
  Hypervisor 1.4.1 2007/04/02 16:37
  OBP 4.26.1 2007/04/02 16:26
  POST 4.26.0 2007/03/26 16:45
```

At this point, the system can be powered on and the operating system booted.

Now that the firmware has been updated, it is time to install the LDM software. The software bundle includes the following:

- SUNWldm.v: LDM Software
- SUNWjass: Solaris Security Toolkit (a.k.a. JASS)

The LDM software is fairly small and contained within a single package. It contains the libraries, configuration daemon, command-line interface, SMF service, and man pages for the LDM software.

The Solaris Security Toolkit [6] or JASS is a security-hardening framework. This framework includes configurations that are called drivers. These drivers can disable services, change permissions, lock accounts, enable security features, etc., in a reproducible manner. The toolkit can easily be extended and customized for your environment. It is distributed with other Sun products, such as the management software for E25k, to provide recommended security settings. This is a purely optional component; the LDM software will function without JASS. However, its addition does provide a consistent and flexible security framework.

JASS is included with the LDM software bundle to help secure and harden the primary domain. This is accomplished through the ldm_control-secure driver, which is specifically designed for the primary domain and its services. It will disable all unnecessary services, enable many security features, and lock down access to only SSH.

The LDM software bundle can be installed manually, through Jumpstart, or through the use of the included install-ldm script. This script is included with the software bundle to automate the installation. It will present you with options for hardening the primary domain with JASS. The first option, “a,” will install the LDM and JASS software with the driver specifically for the primary domain applied; this is the recommended option. The second option, “b,” will only install the LDM and JASS software but will not apply any drivers. The last option, “c,” will install the LDM and JASS software but give you the option of selecting a driver to apply. Here is a sample installation session:

```
# Install/install-ldm
Welcome to the LDoms installer.

You are about to install the domain manager package that will enable you to
create, destroy and control other domains on your system. Given the capabilities
of the domain manager, you can now change the security configuration
of this Solaris instance using the Solaris Security Toolkit.
Select a security profile from this list:
a) Hardened Solaris configuration for LDoms (recommended)
b) Standard Solaris configuration
c) Your custom-defined Solaris security configuration profile
Enter a, b, or c [a]: a
The changes made by selecting this option can be undone through the
Solaris Security Toolkit’s undo feature. This can be done with the
‘/opt/SUNWjass/bin/jass-execute -u’ command.
```

At this point the LDM and JASS software is installed. It is now time to reboot the primary domain.

## Configuring the Primary Domain

The primary domain is the first service and the control domain for the platform. Now that all of the prerequisites are installed, it is time to configure the primary domain. The first step is to ensure that the required SMF services are running:

```
# svcs -a | grep ldom
online 18:34:15 svc:/ldoms/ldmd:default
online 18:34:15 svc:/ldoms/vntsd:default
```

The svc:/ldoms/ldmd:default service is responsible for managing the ldmd daemon, which communicates directly with the hypervisor for configuration and management tasks. The svc:/ldoms/vntsd:default service is responsible for providing the virtual network terminal services through the vntsd daemon. If these SMF services are not running, enable them with the svcadm command.

At this point it is good practice to add the following to your $PATH and $MANPATH shell configuration:

```
PATH=$PATH:/opt/SUNWldm/bin
MANPATH=$MANPATH:/opt/SUNWldm/man
```

After all of the prerequisites are installed, all of the resources in the platform are assigned to the primary domain. This can be verified with the ldm command:

```
# ldm list
Name      State    Flags    Cons    VCPU    Memory    Util    Uptime
primary   active   -t-cv    SP      32      32G       0.6%    1h 13m
```

As you can see, all 32 VCPUs and 32 GB of memory are assigned to the primary domain. To enable the creation of other logical domains, resources must be freed and basic services configured. The primary domain should be given at least one CPU core, or 4 VCPUs and 2 to 4 GB of memory:

```
# ldm set-mau 1 primary
# ldm set-vcpu 4 primary
# ldm set-mem 4G primary
```

In this example, a cryptographic thread of a MAU, 4 VCPUs, and 4 GB of memory are assigned to the primary domain. For these settings to take effect, the primary domain must be rebooted. However, before rebooting the primary domain it is good practice to configure the basic services that will support the creation of additional logical domains without causing further reboots.

Creating the virtual console concentrator or VCC service is essential to providing console access to any logical domains created in the future. Only the primary domain can be reached directly via the hardware console; all other logical domains must be reached through the VCC service. When you create the VCC service, a range of TCP ports must be specified. Each of these ports can be bound to one LDom and can be accessed through the telnet command.

```
# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

It is important to note that instances of services or devices can be freely named. In the example here, our instance of the VCC service is called “primary-vcc0.” The naming conventions used throughout this article take the form of <ldom>-<virtual service or device><instance>.

All virtual storage is serviced by the virtual disk service (VDS). Only one VDS can exist for each service or control domain. This service is created once in the primary domain:

```
# ldm add-vds primary-vds0 primary
```

The last services to be created are the virtual switches (VSWs); these enable logical domains to communicate with the physical network. A VSW should be created for each physical network port on the server. By default the Sun Fire T2000 is equipped with four embedded gigabit Ethernet ports:

```
# ldm add-vsw net-dev=e1000g0 primary-vsw0 primary
# ldm add-vsw net-dev=e1000g1 primary-vsw1 primary
# ldm add-vsw net-dev=e1000g2 primary-vsw2 primary
# ldm add-vsw net-dev=e1000g3 primary-vsw3 primary
```

Once the primary domain resources and services are configured, the configuration must be stored within the ALOM CMT service processor for the hypervisor to reference. This is accomplished by saving the configuration with the ldm add-config <name> command. In this example, I have called my current in-memory configuration "myconfig":

```
# ldm list-config
factory-default [current]

# ldm add-config myconfig

# ldm list-config
factory-default [current]
myconfig [next]
```

This will dump the configurations we have been entering into the ALOM CMT service processor and make it the configuration to use on the next reboot. Now that the current configuration has been saved, the primary domain must be rebooted.

When the primary domain reboots, the configuration will be updated:

```
# ldm list
Name      State     Flags    Cons    VCPU    Memory    Util     Uptime
primary   active    -t-cv    SP      4       4G        0.8%     7m

# psrinfo -vp
The physical processor has 4 virtual processors (0-3)
UltraSPARC-T1 (cpuid 0 clock 1000 Mhz)

# prtdiag -v | grep -i mem
Memory size: 4096 Megabytes
```

You can verify the configuration of the primary domain and the services we created with the ldm list-bindings command. Here is an example of the output reduced to show the key points:

```
# ldm list-bindings
Name:   primary
...
Vcpu:   4
...
Mau:    1
        mau cpuset (0, 1, 2, 3)
Memory: 4G
...
Vds:    primary-vds0
Vcc:    primary-vcc0
          port-range=5000-5100
```

```
Vsw:    primary-vsw0
...
            net-dev=e1000g0
...
Vsw:    primary-vsw1
...
            net-dev=e1000g1
...
Vsw:    primary-vsw2
...
            net-dev=e1000g2
...
Vsw:    primary-vsw3
...
            net-dev=e1000g3
...
```

As you can see, the VCPU, MAU, memory, VDS, VCC, and VSWs are configured. Now that resources and services are available, you can proceed to the configuration of your first guest domain.

## Configuring a Guest Domain

Guest domains are consumers of virtual devices and services. As such, these virtual elements must be configured and assigned. Typically, the following resources would be configured:

- VCPU
- MAU
- Memory
- OpenBoot PROM variables
- Storage
- Networking

To begin this process, the guest domain must be created:

```
# ldm add-domain ldom1
```

This will create a guest domain called “ldom1.” Resources can now be added to ldom1, starting with VCPU, MAU, and memory resources:

```
# ldm add-vcpu 4 ldom1
# ldm add-mau 1 ldom1
# ldm add-memory 4G ldom1
```

In this example, 4 VCPUs, a MAU, and 4 GB of RAM are allocated. Logical domains only require at minimum one VCPU, which is one of the 32 threads in the Niagara I processor. There is only one MAU thread for each CPU core, of which there are eight total on the Niagara I processor. This can be used to accelerate cryptographic software. Memory can be assigned in varying sizes, from 8K junks to gigabytes at a time.

Each logical domain has its own instance of the OpenBoot PROM (OBP). As such, standard variables can be configured as if it were a stand-alone server. These variables are stored in the hypervisor configuration. These variables can be defined from within the OBP or through the LDM software.

```
# ldm set-variable auto-boot\?=true ldom1
# ldm set-variable local-mac-address\?=true ldom1
# ldm set-variable boot-device=/virtual-devices@100/channel-de-
vices@200/disk@0 ldom1
```

This configures the OBP to auto-boot the logical domain, to configure unique MAC addresses for each network interface, and, finally, to boot off of the first virtual disk. The path defined for the boot disk uses the default device path for all logical domains. The disk target is defined by the last number in the device path.

It's now time to configure storage for your guest domain. There are several options for bootable storage with guest domains:

- Local storage
- SAN storage
- Virtual disk images

Local storage consists of physical disks that are not in use by any other logical domain, including the primary domain. The major limitation in this area is the limited number of physical disk slots on the current Niagara I product line. As such, an external storage array or SAN storage may make more sense.

SAN storage offers greater flexibility and redundancies. It also enables the ability to move logical domains between physical servers in the data center.

Virtual disk images are sparse files the are created with the mkfile command. These files can be virtualized to function as normal storage. This adds another layer of flexibility, since the virtual disk images can be stored locally, on SANs, or on NAS.

In this example, two virtual disk images will be created and added to the VDS service in the primary domain:

```
# mkfile 10g /ldoms/ldom1_vdsk0_10gb.img
# mkfile 10g /ldoms/ldom1_vdsk1_10gb.img
# ldm add-vdsdev /ldoms/ldom1_vdsk0_10gb.img ldom1-vdsk0@
      primary-vds0
# ldm add-vdsdev /ldoms/ldom1_vdsk1_10gb.img ldom1-vdsk1@
      primary-vds0
```

If we had wanted to add a SAN device, the command would look like this:

```
# ldm add-vdsdev /dev/dsk/c6t60060160B5681200944\
2F7677A81DB11d0s2 ldom1-vdsk2@primary-vds0
```

Now that the VDS devices are added, they must be assigned to the guest domain:

```
# ldm add-vdisk ldom1-vdsk0 ldom1-vdsk0@primary-vds0 ldom1
# ldm add-vdisk ldom1-vdsk1 ldom1-vdsk1@primary-vds0 ldom1
```

An important thing to keep in mind is that any virtual storage device bound to a guest domain will appear as if it were a locally attached disk. This removes any additional management layers and simplifies the storage stack for the kernel in the guest domain.

However, it is important to note that virtual storage devices are not presented with SCSI targets to Solaris in guest domains. As such, the device names will be missing the familiar target in the cXtXdXsX standard and appear as cXdXsX. For example, a storage device in the control domain may appear as c4t1d0s0, but in the guest domain it may appear as c0d1s0. The controller number will be determined by the order in which a VDS device was added to the guest domain. There can only be one VDS per service domain. By default the primary domain is the first control and service domain, so all of your disks will be under controller 0. The disk number is determined by the order in which you add the VDSDEV to the guest domain. In our example, ldom1-vdsk0 will appear as c0d0s0 and ldom1-vdsk1 will appear as c0d1s0 in the guest domain. This may affect JumpStart configurations, but it does not affect anything operationally.

Connecting the guest domain to the virtual switches will enable it to communicate with your networks. This involves configuring virtual network ports or VNETs to the VSWs that are connected to your networks. This will configure a unique MAC address automatically and allow access to the connected physical networks and to any other logical domains connected to the same VSW. If we name our VNET instances as ldom1-vnet0 and ldom1-vnet1, we get:

```
# ldm add-vnet ldom1-vnet0 primary-vsw0 ldom1
# ldm add-vnet ldom1-vnet1 primary-vsw2 ldom1
```

These VNET instances will appear, in the order of addition, as vnet0 and vnet1 to Solaris in the guest domain.

Finally, it is time to commit the configuration and start the guest domain:

```
# ldm bind-domain ldom1
# ldm start ldom1

 # ldm list
Name     State    Flags   Cons   VCPU   Memory   Util   Uptime
primary  active   -t-cv   SP     4      4G       0.6%   4h 8m
ldom1    active   -t—     5000   4      4G       0.2%   2m
```

Now you can connect to the virtual console of your guest domain by using telnet and the console number specified under the Cons column from above:

```
# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..

Sun Fire T200, No Keyboard
Copyright 2007 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.26.0.build_07, 4096 MB memory available, Serial #66831599.
Ethernet address 0:14:4f:fb:c4:ef, Host ID: 83fbc4ef.

{0} ok
```

The operating system can now be installed in the guest domain through the use of JumpStart. Once this is completed, you'll be able to log in and manage your guest domain:

```
$ ssh ldom1
Password:

ldom1:~ $ uname -a
SunOS ldom1 5.10 Generic_125100-04 sun4v sparc SUNW,Sun-Fire-T200
```

## Summary

In this article, you have been given a tutorial on the installation and configuration of logical domains. This tutorial should enable you to explore the use of LDoms and understand the technology. In the next article, I will discuss advanced topics and technology limitations. I will also compare the LDom technology with other virtualization solutions for the Solaris operating system.

## REFERENCES

[1] List of Sun servers that support LDoms: http://www.sun.com/servers/index.jsp?cat=CoolThreads%20Servers&tab=3&subcat=UltraSPARC%20T1.

[2] Download site for LDM software, platform firmware, and Solaris patches: http://www.sun.com/servers/coolthreads/ldoms/get.jsp.

[3] Download site for Solaris 10: http://www.sun.com/software/solaris/get.jsp.

[4] Download site for Solaris Express: http://opensolaris.org/os/downloads/.

[5] ALOM CMT service processor documentation: http://docs.sun.com/source/819-7981-11/index.html.

[6] Solaris Security Toolkit ( JASS) site: http://www.sun.com/software/security/jass/.