OCTAVE ORGERON

an introduction to logical domains

PART 4: ADVANCED TOPICS

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 FEBRUARY 2008 ISSUE OF *;login:*, I discussed advanced topics concerning networking and storage for logical domains (LDoms). In this final articleof the series, I will discuss other advanced topics that are key to the successful management of logical domains. These advanced topics will aid your design and implementation decisions concerning LDoms.

Hardware and LDoms

Since I started writing this series of articles, Sun has continued to release new hardware that supports LDoms. This includes UltraSPARC-T2 rackmount servers such as the T5120 and blade servers such as the T6320 [1]. LDom-capable equipment will continue to increase with the release of new sun4v platforms, such as the "Victoria Falls" and "ROCK" platforms [2]. This will help flesh out the medium- to high-end LDom-capable platforms and provide a full range of equipment to choose from.

Currently, the UltraSPARC-T1 and UltraSPARC-T2 platforms have significant differences that can impact your platform decisions with LDoms (see Table 1). The two differentiating factors are CPU features and I/O capabilities.

Feature	UltraSPARC-T1	UltraSPARC-T2
UltraSPARC-T2	8 maximum	8 maximum
Threads per core	4	8
Floating-point units	1 shared with each core	1 per core
Cryptographic units	1 (MAU) per core	1 (MAU/CWQ) per core
Memory controller	4 @ 23 GB/s	4 @ 50 GB/s
PCI-E controller	External ASIC connected to processor on the JBUS	Embedded into chip; 8 lanes @ 2.5 GHz with 2 GB/s bandwidth in each direction
PCI-E slots	Depends on model: ranges from 1 to 3	Depends on model: ranges from 3 to 6
PCI-X slots	T2000 has 2 PCI-X slots	None
Networking	Two external dual port 1 Gb Ethernet ASICs or through option cards	Embedded dual 10 Gb Ethernet on chip; two external dual-port 1 Gb Ethernet ASICs or through option cards
Storage	External SAS controller or through option cards	External SAS controller or through option cards

TABLE 1: COMPARISON OF ULTRASPARC-T1 AND ULTRASPARC-T2 PLATFORMS

These features play an integral role for designing solutions around LDoms. Table 2 outlines the key factors you should consider.

Feature	Consideration
Core	It is recommended that the primary domain have at least one core. So the number of cores available for guest domains is N – 1.
Threads per core	This determines the maximum number of VCPUs available for LDoms.
Floating-point units	The UltraSPARC-T1 platform floating-point performance suffered because only one FPU was available. This can negatively impact performance of heavy FP applications. Luckily, the UltraSPARC-T2 does not suffer from this limitation, as each core has a dedicated FPU.
Cryptographic (MAU) units	Only one LDom can own an MAU in any given core.
Memory controller	The memory bandwidth can impact the performance of applications that make heavy use of memory resources.
PCI-E controller	The UltraSPARC-T2 platform has a PCI-E controller embedded into the chip. This controller communicates with different PCI-E switches to which the PCI-E slots are connected. This enables higher I/O bandwidth to the PCI-E components.
PCI-E slots	The number of PCI-E slots determines the number of option cards that can be installed and utilized by LDoms.
Networking	Only the UltraSPARC-T2 has a dual 10 Gb Ethernet controller embedded into the chip. This increases throughput for high-performance network requirements, NAS, iSCSI, and streaming data.
Storage	The number of disks available internally can directly affect the number of guest domains that can be created when JBOD, SAN, NAS, or iSCSI storage is not available.

TABLE 2: FACTORS IN DESIGNING SOLUTIONS AROUND LDOMS

CPU Affinity

The UltraSPARC-T1 and UltraSPARC-T2 processors have a shared L2 cache that is utilized by all of the cores. Although you could technically divide a T5120 into 64 LDoms, that may not be practical for real workloads. For example, if you were to take a single core and allocate each VCPU out to separate guest domains with vastly different workloads, the probability of cache misses would increase. This causes the cache to work harder to feed each VCPU the required data. This can negatively impact the performance of your guest domains. To avoid this scenario, the following recommendations should be considered:

- Use whole cores for guest domains where possible for performance.
- Only allocate partial cores to guest domains that will host low-impact applications and services.
- Bind and start your larger guest domains first. This helps to ensure that your larger guest domains utilize a full core where possible. For

example, if you have a guest domain with eight VCPUs and another guest domain with two VCPUs, it makes sense to bind and start the larger guest domain first.

Split PCI-E Configurations

The PCI-E configuration on a platform can affect the ability to create a second I/O domain. As you recall, for an LDom to be an I/O domain, it must own part of the PCI-E bus. This is accomplished by assigning a PCI-E leaf to a guest domain as described in the vendor documentation [3]. You can then turn your I/O domain into a service domain by virtualizing networking and storage devices for your guest domains. However, not all platforms have multiple PCI-E buses available to be split among multiple service domains. The UltraSPARC-T1–based T2000 platform is one such machine that does have this support, but there are some limitations when using it:

- The T2000 only has a single SAS controller for the internal disk. By splitting the PCI-E configuration, only one of your domains can use the internal storage. As such, you'll need JBOD or SAN storage for your second I/O or service domain. Care must be taken to prevent the PCI-E leaf with the primary domain disks from being removed from the primary domain itself.
- One of your domains will have a single PCI-E slot and two 1 Gb Ethernet ports. The other domain will have two PCI-E slots, two PCI-X slots, and two 1 Gb Ethernet ports.

With the UltraSPARC-T2 platform, this capability is removed, as the PCI-E controller is embedded into the processor. All of the PCI-E components are connected via PCI-E switches. The only I/O component that can be split off to another domain is the NIU or 10 Gb Ethernet controller, which is also embedded into the processor. By doing so, you can have a guest domain capable of high-performance network throughput and communications.

In the future, the networking and storage controllers will be virtualized, with greater control for guest domains. The OpenSolaris NPIV project will enable a guest domain to have its own virtualized Fibre Channel HBA [4]. Also, the Hybrid I/O feature will enable a PCI-E leaf device to be assigned directly to a guest domain [5]. These features will become available in the future and provide greater flexibility for LDoms.

Dynamic and Delayed Reconfiguration

LDoms are capable of having virtual devices added and removed. Some of these virtual devices can be added or removed dynamically, which means that the LDom does not require a reboot for the change to take effect, whereas other virtual devices can only be added or removed when an LDom reboots. These differences are known as dynamic and delayed reconfiguration.

Currently, only VCPUs can be dynamically reconfigured with LDoms; all other virtual devices are relegated to delayed reconfiguration. As the technology evolves, more virtual devices will be capable of dynamic reconfiguration.

In this example, four VCPUs will be dynamically added to a guest domain:

```
ldom1:~ # psrinfo -vp
The physical processor has 4 virtual processors (0-3)
  UltraSPARC-T2 (clock 1417 MHz)
```

```
primary:~ # ldm list
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL      UPTIME
primary          active   -n-cv   SP      8     8G       0.3%      8h 46m
ldom1            active   -n---   5000    4     2G       48%       5h 52m

primary:~ # ldm add-vcpu 4 ldom1

primary:~ # ldm list
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL      UPTIME
primary          active   -n-cv   SP      8     8G       0.3%      8h 46m
ldom1            active   -n---   5000    8     2G       48%       5h 52m

ldom1:~ # psrinfo -vp
The physical processor has 8 virtual processors (0-7)
  UltraSPARC-T2 (clock 1417 MHz)
```

The VCPUs were added dynamically to the guest domain ldom1 without it having to be rebooted. This means that VCPU resources can be dynamically moved around depending on resource requirements. This can be useful for moving VCPU resources to where they are needed for application workloads.

Delayed reconfiguration requires the LDom to be rebooted. Multiple reconfiguration changes can be requested for the same LDom before it reboots, as they will be queued. Once a delayed reconfiguration operation for an LDom has been queued, reconfiguration requests for other LDoms are disabled until the queued requests are handled.

In this example, our guest domain will have memory and storage added:

```
ldom1:~ # prtdiag -v | grep Mem
Memory size: 2048 Megabytes

ldom1:~ # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0d0 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
          /virtual-devices@100/channel-devices@200/disk@0
       1. c0d1 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
          /virtual-devices@100/channel-devices@200/disk@1
Specify disk (enter its number): ^D

primary:~ # ldm add-mem 2g ldom1
Initiating delayed reconfigure operation on LDom ldom1.  All configuration
changes for other LDoms are disabled until the LDom reboots, at which time
the new configuration for LDom ldom1 will also take effect.

primary:~ # mkfile 5g /ldoms/local/ldom1/ldom1-vdsk2.img
primary:~ # ldm add-vdsdev /ldoms/local/ldom1/ldom1-vdsk2.img  ldom1-vdsk2@primary-vds0
primary:~ # ldm add-vdisk ldom1-vdsk2 ldom1-vdsk2@primary-vds0 ldom1
--------------------------------------------------------------------------------
Notice: LDom ldom1 is in the process of a delayed reconfiguration.
Any changes made to this LDom will only take effect after it reboots.
--------------------------------------------------------------------------------

        ldom1:~ # reboot

        ...
              ldom1:~ # prtdiag -v | grep Mem
              Memory size: 4096 Megabytes

              ldom1:~ # format
              Searching for disks...done
```

```
AVAILABLE DISK SELECTIONS:
       0. c0d0 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
          /virtual-devices@100/channel-devices@200/disk@0
       1. c0d1 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
          /virtual-devices@100/channel-devices@200/disk@1
       2. c0d2 <SUN-DiskImage-5GB cyl 17474 alt 2 hd 1 sec 600>
          /virtual-devices@100/channel-devices@200/disk@2
Specify disk (enter its number): ^D
```

Delayed reconfiguration requests can be canceled for an LDom. However, doing so will remove any queued items as well.

```
primary:~ # ldm rm-mem 2g ldom1
Initiating delayed reconfigure operation on LDom ldom1.  All configuration
changes for other LDoms are disabled until the LDom reboots, at which time
the new configuration for LDom ldom1 will also take effect.

Notice: this remove operation will prevent any future VIO device removal opera-
tion from being accepted for the duration of this delayed reconfiguration, i.e. until
the domain reboots or the delayed reconfig is cancelled.
primary:~ # ldm remove-reconf ldom1
```

Notice that this operation of removing memory prevented any further removal operations from queueing. The LDM software will alert you of such conditions.

There are a few caveats about dynamic and delayed reconfiguration that should be kept in mind:

- Be mindful of removing VCPUs from an LDom that has an MAU bound in the same core. The MAU may have to be removed first through delayed reconfiguration if the VCPUs being removed are the only ones assigned to the LDom from that core.
- For better cache coherency, an LDom should scale within a single core until additional VCPUs are required from another core.
- The ldm command will warn you if requests can be handled or if they must be held off until a reconfiguration operation has completed.

Configuration Management

The configuration for your LDoms should be backed up regularly. Each LDom configuration can be dumped into an XML configuration file. The configuration dump only contains the mapping of the resources and virtual devices that are configured for the LDom. However, this does not include the configuration of the underlying device services such as the VDSDEVs or the VSWs. This configuration file can be used to recreate or duplicate LDom configurations:

```
primary:~ # ldm list-constraints -x ldom1 > ldom1.xml
```

This configuration file can be used to recreate the LDom in a recovery scenario or when migrating a guest domain from one server to another. For example, if the above LDom were removed accidentally, the configuration could be restored:

```
primary:~ # ldm list ldom1
LDom "ldom1" was not found

primary:~ # ldm add-domain -i ldom1.xml ldom1

primary:~ # ldm list ldom1
```

```
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
ldom1            inactive -----                4        4G
```

```
primary:~ # ldm bind ldom1
primary:~ # ldm start ldom1
LDom ldom1 started
```

```
primary:~ # ldm list ldom1
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
ldom1            active   -t---   5000    4        4G       30%   0s
```

```
ldom1:~ # uname -a
SunOS ldom1 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120
```

This process cannot be used to restore the primary domain configuration. However, the XML dump can provide valuable information in the event that it must be recreated manually.

High Availability

Clustering today with LDoms is in its infancy. Many of the clustering products, such as Solaris Cluster and Veritas Cluster Server, are just beginning to support installation into control and I/O domains. However, they lack agents to properly support guest domains and the applications contained within them. This will change as the products mature to support LDoms. However, in the meantime you can create a standby environment for your guest domains in the event of a failure. For this you will need the following:

- Two or more servers that are configured similarly
- SAN or NAS storage for your guest domains

Here is an example of creating such a standby environment utilizing NAS storage:

```
primary:~ # mkfile 10g /ldoms/nas/ldom4-vdsk0.img
```

```
primary:~ # df -h /ldoms/nas
Filesystem             size   used  avail capacity  Mounted on
192.168.2.70:/export/ldoms
                       107G   10G    97G     1%    /ldoms/nas
```

```
primary2:~ # df -h /ldoms/nas
Filesystem             size   used  avail capacity  Mounted on
192.168.2.70:/export/ldoms
                       107G   10G    97G     1%    /ldoms/nas
```

At this point, we can create ldom4 on our first server:

```
primary:~ # ldm add-domain ldom4
primary:~ # ldm add-vcpu 4 ldom4
primary:~ # ldm add-mem 4G ldom4
primary:~ # ldm add-vnet vnet0 primary-vsw0 ldom4
primary:~ # ldm set-variable auto-boot\?=false
primary:~ # ldm add-vdsdev /ldoms/nas/ldom4-vdsk0.img ldom4-vdsk0@primary-vds0
primary:~ # ldm add-vdisk ldom4-vdsk0 ldom4-vdsk0@primary-vds0 ldom4
primary:~ # ldm bind ldom4
primary:~ # ldm start ldom4
LDom ldom4 started
```

```
primary:~ # ldm list ldom4
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL      UPTIME
ldom4            active   -n---   5004    4        4G       0.0%      34
```

Now we can dump the XML configuration of our guest domain ldom4:

```
primary:~ # ldm list-constraints -x  ldom4 > /ldoms/nas/ldom4.xml
```

The configuration can be imported on our second server, once the VDS device has been configured:

```
primary2:~ # ldm add-vdsdev /ldoms/nas/ldom4-vdsk0.img ldom4-vdsk0@primary2-vds0
primary2:~ # ldm add-domain -i /ldoms/nas/ldom4.xml

primary2:~ # ldm list ldom4
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
ldom4            inactive -----                4       4G
```

Once we have installed the OS into the guest domain on our first server, we can test our configuration:

```
ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120
ldom4:~ # shutdown -y -g0 -i 5
...

primary:~ # ldm stop ldom4
LDom ldom4 stopped
primary:~ # ldm unbind ldom4

primary:~ # ldm list ldom4
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
ldom4            inactive -----                4       4G

primary2:~ # ldm bind ldom4
primary2:~ # ldm start ldom4
LDom ldom4 started
primary2:~ # ldm list ldom4
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
ldom4            active   -n-cv   5004    4       4G       0.3%  5m

ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120
```

One could script this process to migrate guest domains between servers once the configuration is in place on each server. In the future, LDoms will also support the ability to migrate guest domains between servers without any downtime. This feature is called "Live Migration" and will be similar to the VMWare Vmotion feature [5].

Running Multiple Operating Systems

One of the strengths of LDoms is the ability to run multiple guest domains with different operating systems at the same time. You can install the following operating systems into a guest domain:

- Solaris 10 Update 3 (11/06) and above
- Solaris Express Community Edition, build 70 and above
- Solaris Express Developer Edition 09/07 and above
- OpenSolaris, build 70 and above
- Ubuntu Linux 7.10 and above

There are other operating systems that already have sun4v platform support or are developing support. The key to working with LDoms is to have support for the virtualized devices, such as the VNETs and VDISKs. Once the proper support is added to an OS, it can be used in a guest domain. Here

is a demonstration of different OSes running on a single physical server by using LDoms:

```
ldom1:~ $ uname -a
SunOS ldom1 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120

ldom2:~ $ uname -a
SunOS ldom2 5.11 snv_75 sun4v sparc SUNW,SPARC-Enterprise-T5120

root@ldom3:~ $ uname -a
Linux ldom3 2.6.22-14-sparc64-smp #1 SMP Tue Dec 18 05:40:10 UTC 2007 sparc64 GNU/Linux
root@ldom3:~# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=7.10
DISTRIB_CODENAME=gutsy
DISTRIB_DESCRIPTION="Ubuntu 7.10"

ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120

ldom5:~ $ uname -a
SunOS ldom5 5.10 Generic_120011-14 sun4v sparc SUNW,SPARC-Enterprise-T5120
```

As you can see, there are three guest domains running Solaris Express at different releases, one guest domain running Ubuntu Linux 7.10, and a final guest domain running Solaris 10 Update 4 (08/07).

This can be very beneficial for applications testing or development projects that require different OS versions, patch levels, or configurations. It can also be an efficient method for testing new products before migrating to them on the same hardware. The cost savings can be significant in both time and equipment.

Comparisons

There are many virtualization technologies today that can be utilized across a wide range of platforms and operating systems. As the demand for server utilization efficiencies increases, the requirement to leverage virtualization will become common practice in data centers. All of these technologies can be broken into three major categories, as outlined in Table 3.

Technology	Description
Hardware partitions	Hardware partitions are created from specialized ASICs and firmware that enable components in a platform to be grouped into smaller systems and electrically isolate them from failure. This provides the highest level of separation between multiple OS instances. This is seen on Sun equipment such as the E25k or the new SPARC Enterprise M9000.
Virtual machines	Virtual machines are created through software that is either in firmware or in a management OS instance. This software is able to virtualize or emulate the hardware into groupings capable of running isolated instances of an operating system. This provides many pros and cons depending on the requirements. Virtual machines are commonly seen in technologies such as VMware, Xen, Sun xVM, IBM LPARs, Parallels, and QEMU.

TABLE 3: VIRTUALIZATION CATEGORIES *(continued on p. 32)*

OS virtualization	OS virtualization occurs when a single OS instance is able to create an isolated run-time environment that closely emulates a standalone OS installation. When combined with resource management, this can effectively utilize hardware resources, because the overhead is very low. This is seen in technologies such as Solaris Containers (Zones), BSD Jails, IBM WPARs, OpenVZ, and Linux-VServer.

TABLE 3: VIRTUALIZATION CATEGORIES *(continued from p. 31)*

Logical domains are a hybrid of hardware partitioning and virtual machines. The control domain uses the hypervisor to partition CPU and memory resources into groupings for guest domains, whereas the service domain virtualizes the I/O components, such as networking and storage for guest domains. This interesting combination provides many benefits:

- High level of integration with the hardware via the sun4v hypervisor.-
- The ability to leverage built-in hardware features of both the UltraSPARC-T1 and the UltraSPARC-T2 processors, such as CMT, cryptographic engines, and 10 Gb Ethernet
- Reduced overhead for CPU and memory resources
- Flexibility in virtualizing I/O components
- The ability to leverage Solaris features such as ZFS and iSCSI for guest domains
- The ability to create Solaris Containers within LDoms, increasing the level of virtualization

Summary

This article has introduced you to advanced topics concerning the configuration and management of logical domains. With this knowledge, you should be able to explore this technology in greater detail and discover interesting ways in which it can be applied. This technology will continue to evolve and mature. As it becomes open sourced, you will be able to help with the development and advancement of this technology.

WHERE TO FIND MORE INFO

OpenSolaris LDoms Community: http://www.opensolaris.org/os/community/ldoms.

OpenSolaris LDoms Community discussion: http://www.opensolaris.org/jive/forum.jspa?forumID=203.

Sun LDoms home page: http://www.sun.com/servers/coolthreads/ldoms/index.xml.

Installing Ubuntu Linux on SPARC: https://help.ubuntu.com/community/Installation/Sparc.

My blog: http://unixconsole.blogspot.com/.

REFERENCES

[1] Sun LDoms home page: http://www.sun.com/servers/coolthreads/ldoms/index.xml.

[2] OpenSolaris LDoms discussion on future features: http://www.opensolaris.org/jive/thread.jspa?messageID=148688&r#148688.

[3] *LDM 1.0.1 Administrative Guide*: http://docs-pdf.sun.com/820-3268-10/820-3268-10.pdf.

[4] OpenSolaris NPIV Project: http://opensolaris.org/os/project/npiv.

[5] LDoms presentation by Liam Merwick: http://opensolaris.org/os/community/ldoms/files/LDoms-LOSUG-Oct-2007.pdf.