



# Logical Domain TOI

**“If you do what you've always done,  
you'll get what you've always gotten.”**  
- Anthony Robbins

**Maria Jose Avila**  
SSC Ft. Lauderdale



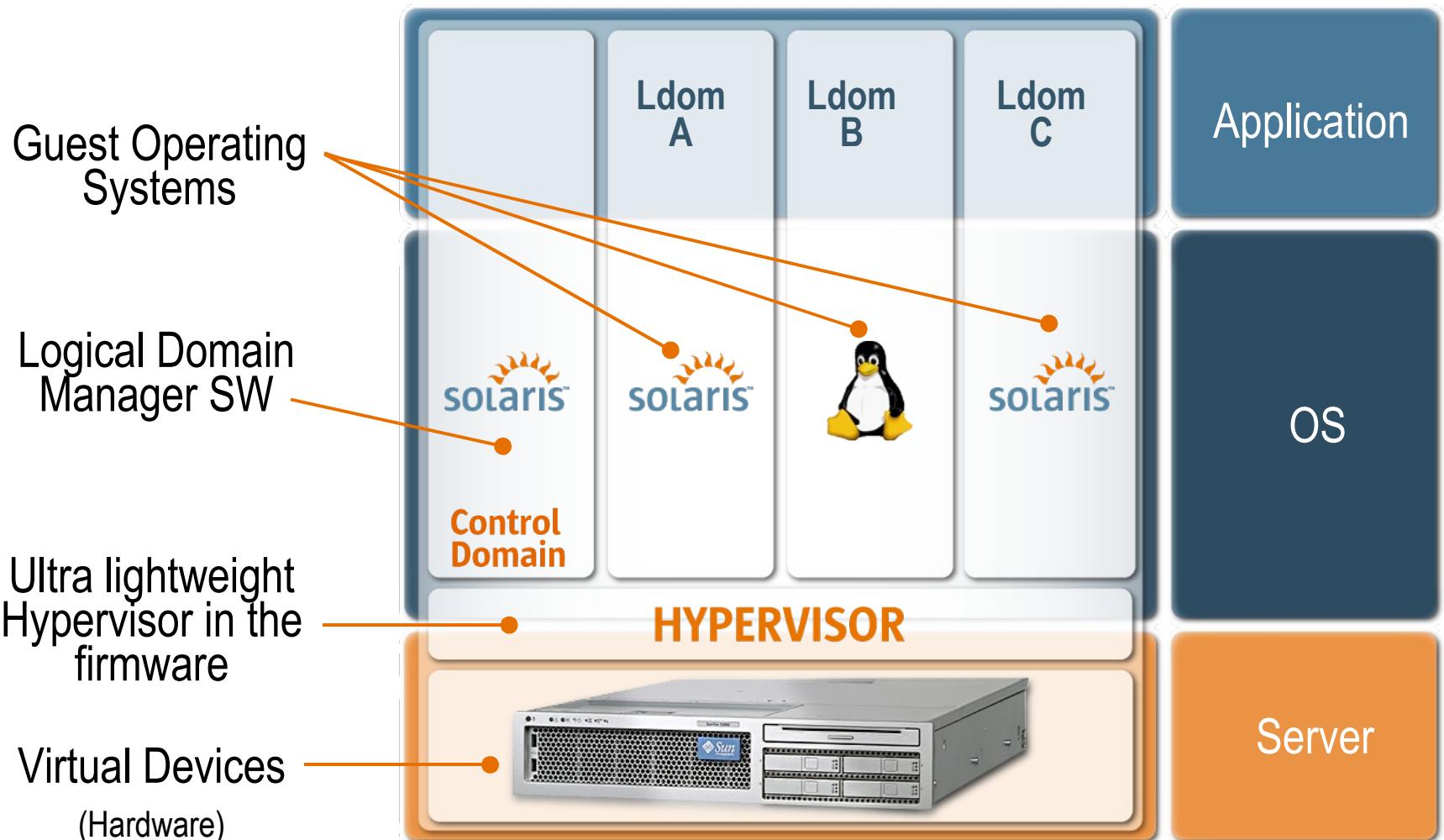


# Agenda

- Logical Domains
  - ◆ Architecture
  - ◆ Virtualized I/O (Disk, Network, Console)
  - ◆ Logical Domain Migration
  - ◆ What is new in LDOMs 1.2
  - ◆ Reconfiguration
  - ◆ Hardware & Split-bus
- Field LDOMs Architecture Request
- HA – LDOMs Demo
- What is Automatic Dynamic Reconfiguration (ADR)?, How to Demo it in a LDOMs environment?
- Q/A Session

# Logical Domains - Architecture

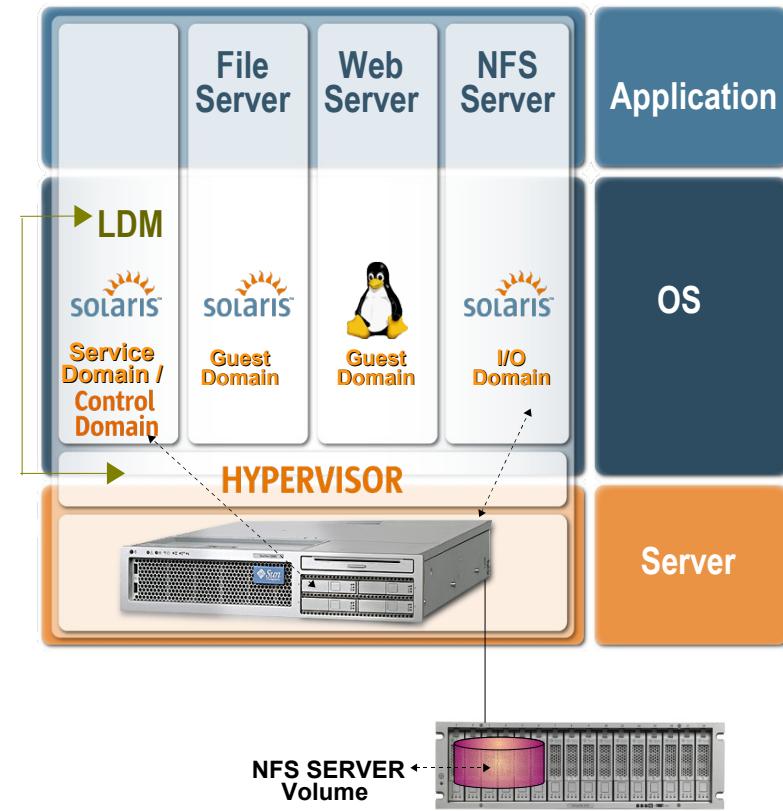
Supported Coolthread Platform



# Logical Domains - Architecture

A single logical domain may function:

- **Control Domain:** Executes Logical Domain Manager SW to govern logical domain creation and assignment of physical resources
  - ◆ Control Domain is usually acting as Service Domain and I/O Domain.
- **Service Domain:** Interfaces with the hypervisor in behalf of a guest domain to manage access to hardware resources (CPU, memory, disk, console, and cryptographic units)
- **I/O Domain:** Controls direct physical access to input/output devices, such as PCI Express cards, storage units, and network devices.
- **Guest Domain:** Utilizes virtual devices offered by service I/O domains and operates under the management of the control domain.



# Logical Domains – Command Line Interface

## Single Command “ Idm ” with multiples sub-commands

- Interfaces to the Hypervisor
- Logical Domain Manager Daemon “ Idmd ” should be running.
- Make sure to add the following paths in order to execute the command and access to its man page

```
root@t5240 # PATH=$PATH:/opt/SUNWldm/bin; export PATH (for Bourne or K shell)
root@t5240 # MANPATH=$MANPATH:/opt/SUNWldm/man; export MANPATH
%# set PATH=($PATH /opt/SUNWldm/bin) (for C shell)
%# set MANPATH=($MANPATH /opt/SUNWldm/man)
```

**TIP:**  
Add the definitions in root's .profile

- Idm sub-command for logical domain creation example:

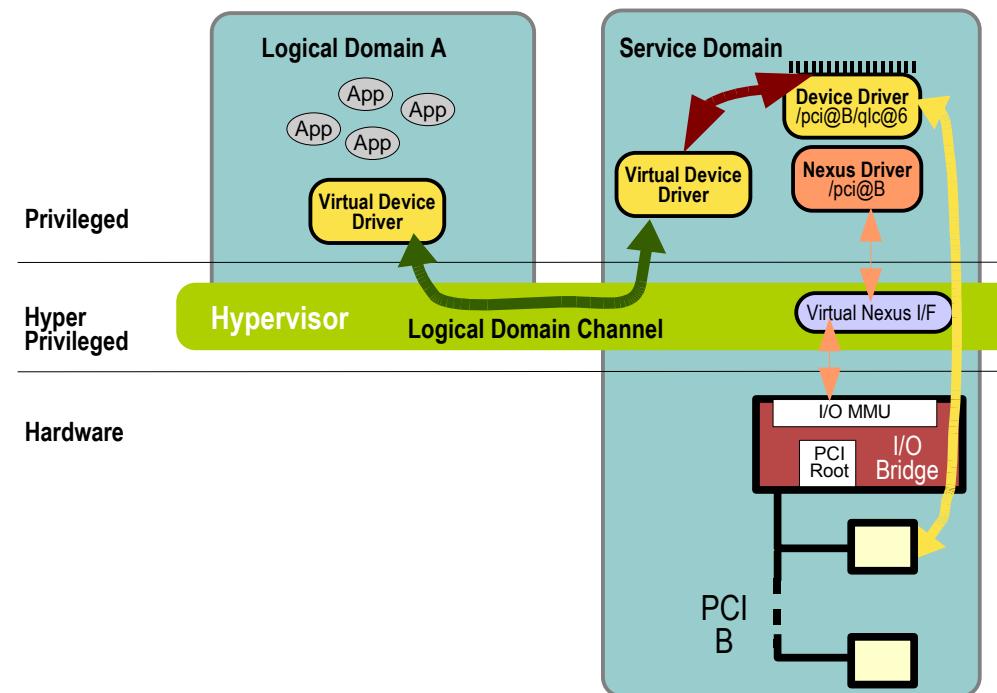
```
root@t5240 # Idm
...
domain ( dom )
add-domain (-i <file> | [mac-addr=<num>] [hostid=<num>] <ldom> | <ldom>...)
remove-domain (-a | <ldom>...)
list-domain [-e] [-l] [-o <format>] [-p] [<ldom>...]
      'format' is one or more of:
      console,cpu,crypto,disk,domain,memory,network,physio,serial,status
start-domain (-a | -i <file> | <ldom>...)
stop-domain [-f] (-a | <ldom>...)
bind-domain (-i <file> | <ldom>)
unbind-domain <ldom>...
migrate-domain [-n|--dry-run] <source_ldom> [<user>@]<target_host>[:<target_ldom>]
```

**EX:**  
root@t5240 # Idm add-domain ldom1

# Logical Domains – Virtualized I/O

## Virtual Disk Server (vds):

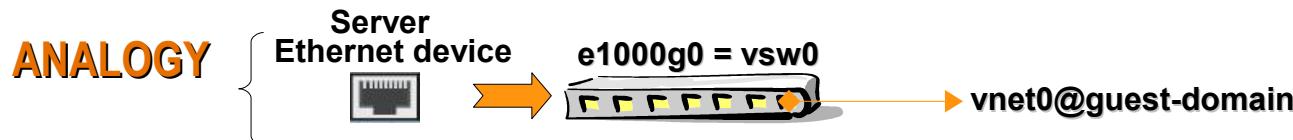
- Executes in the Service Domain
- Provides the guest domain domain with virtualized disk access, and export the physical block device to the logical domain in the form of virtual disk server device (vdsdev)
  - ◆ An entire physical disk
  - ◆ Single slice of a disk LUN
  - ◆ Disk volumes ZFS, SVM, VxVM
  - ◆ Disk image file
  - ◆ DVD-ROM device
- The virtual disk server device is associated to the guest domain by the virtual disk (vdisk)
- Virtual Disk Failover, adds support for disk multipathing which enables the virtual disk device in a guest domain to be serviced by multiple virtual disk servers



# Logical Domains – Virtualized I/O

## Virtual Network

- Virtual Network Device (vnet):
  - ◆ Emulates a ethernet device
  - ◆ Communicates directly with other vnet devices or the virtual switch device using LDC
  - ◆ MAC address created or assigned at the time of the logical domain creation, for jumpstart installation use this MAC address and not the one displayed in the banner for the guest domain
    - ◆ For manual MAC address allocation use the following range:  
00:14:4F:FC:00:00 – 00:14:4F:FF:FF:FF
- Virtual Switch Device (vsw)
  - ◆ Interfaces directly with a physical network adapter on a servie domain, and sends and receives packets on a virtual networks behalf
  - ◆ Functions as a simple layer-2 switch



- ◆ Internet Protocol Network Multipathing (IPMP) can be configure.
- ◆ 802.1Q VLAN-Tagging Supported - Solaris 10 10/08 OS and LDoms 1.1 softwareons as a simple layer-2 switch
- ◆ Network Interface Unit (NIU) Hybrid I/O

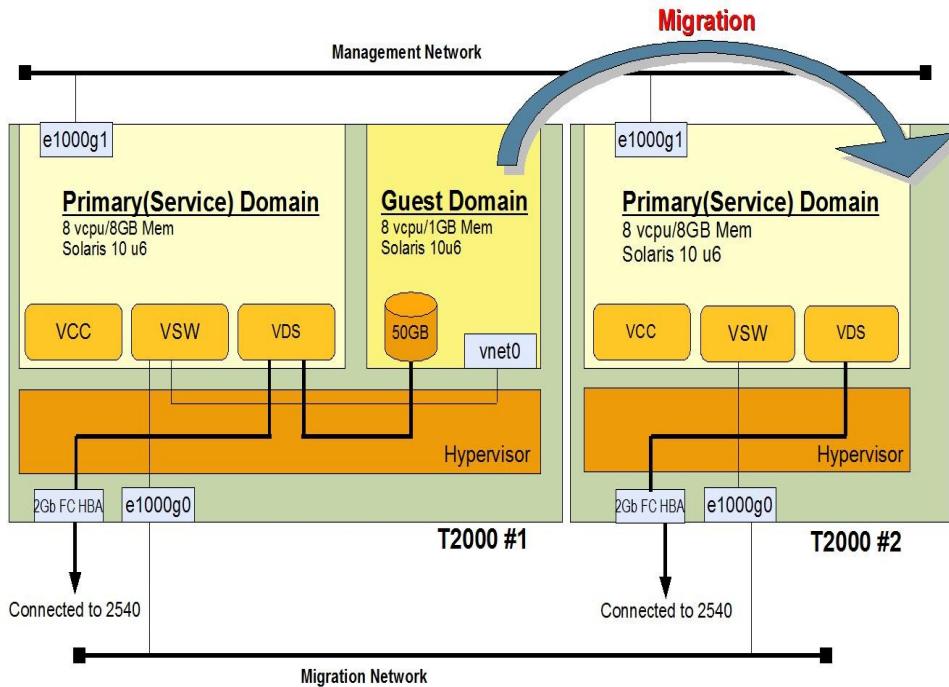
# Logical Domains – Virtualized I/O

## Virtual Console Concentrator (vcc)

- The vcc driver in Solaris communicates with all console drives in the guest Hypervisor LDC connections.
- Virtual console concentrator (vcc) functions as a concentrator for all domain's console traffic (except for the primary domain), and interfaces with virtual network terminal server daemon (vntsd) and provides access to each console through a UNIX socket.
- Virtual NTS daemon (vntsd) layered on top of vcc exports access via Unix sockets for connecting via telnet within control domain to logical domain consoles

```
root@t5240 # ldm list
NAME      STATE   FLAGS  CONS  VCPU MEMORY UTIL UPTIME
primary   active  -n-cv- SP    4    4G    0.2% 23m
Idom1     bound   ----v- 5000  4    4G
root@t5240 # ldm start Idom1
LDom alternate started
root@t5240 # telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^].
Connecting to console "Idom1" in group "dom1" ....
Press ~? for control options ..
{0} ok
```

# Logical Domain - Migration



**Save constraints information for each domain**

Save Constraints information for each domain into a XML file as a backup  
`# ldm ls-constraints -x Idom > Idom.xml`  
**Guest Domain Recreation**  
`# ldm add-domain -i Idom.xml`  
`# ldm bind-domain Idom`  
`# ldm start-domain Idom`

## Types of Migration

- ◆ Cold Migration
- ◆ Warm Migration
- ◆ In the targeted server define virtual disk device (path-to-disk)

# What is new in LDOMs 1.2 ?!?!?

## Features:

- Support for CPU power management
  - ◆ Enable power policy in your ILOM 3.0 firmware Platform, connect to the SP
    - -> set /SP/powermgmt policy=Performance | Elastic
    - ◆ Performance = the system is allowed to use all the power available
    - ◆ Elastic = The system usage is adapted to the current utilization level. For example power up or power down system components
    - Cli used: **ldm list -I primary | ldm list -I -p | ldm list-devices -a cpu**
- Support Jumbo Frames
  - ◆ Frames with payload sizes > standard ethernet MTU
    - ◆ standard ethernet MTU = 1500 bytes
    - ◆ jumbo frames now supported for vswitch and vnet
    - ◆ Supported MTU range: 1500 – 16000 bytes
    - Requirements:
      - ◆ need to configure physical switch/network for jumbo frames
        - mtu=<value> option for add-vsw, set-vsw, add-vnet, set-vnet
      - ◆ need to configure MTU of the physical interface (*S10 only*)
      - ◆ MTU of physical interface should be ≥ MTU of vswitch
- Restriction of delayed reconfiguration operations to the control domain
  - ◆ Restricts delayed reconfiguration operation to the control domain
  - ◆ For all other domains, you must stop the domain to modify the configuration unless the resource can be DR.

# What is new in LDOMs 1.2 ?!?!?

## Features:

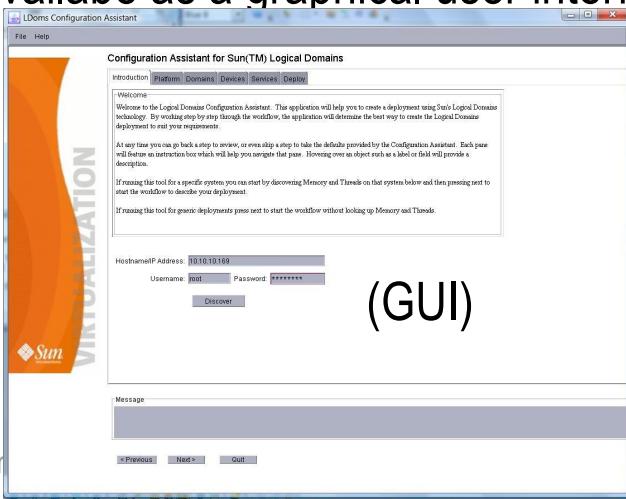
- Support for configuring domain dependencies
  - ◆ The LDM can be used to establish dependency relationship between domains, if the master domain fails a policy can be set to the slaves either to ignore, panic, reset, stop
    - EX: # ldm add-domain *failure-policy=panic* marlins
    - # ldm add-domain *master=margarita* florida
- Support autorecovery of configurations
  - ◆ The autorecovery policy specifies how to handle recovery of a configuration.
    - **autorecovery\_policy=1** (by default) → logs warning are saved in the ldmd SMF log file when an autosave configuration is newer than the corresponding running config. Manual recovery.
    - **autorecovery\_policy=2** Displays a notification message if an autosave configuration is newer than the corresponding running config, printed out of any ldom command the first time after each restart. Manual recovery
    - **autorecovery\_policy=3** Automatically updates the configuration if an autosave configuration is newer than the corresponding running configuration, This action overwrites the SP configuration that will be used during the next powercycle. Messages are logged in the ldmd SMF log file.
    - You may check the value

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
ldmd/autorecovery_policy integer 1
```
- API to support LDMD discovery
  - ◆ The Logical Domain Manager can be discovered on a subnet with multicast messages.

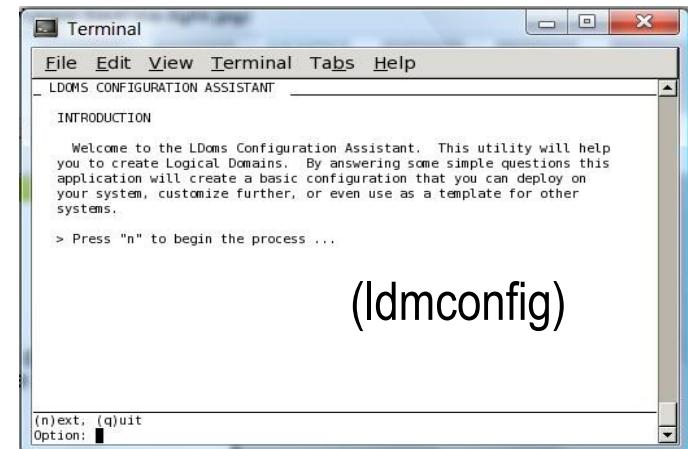
# What is new in LDOMs 1.2 ?!?!?

## Features:

- Support for export of same backend multiple times
  - ◆ A virtual disk backed can be exported multiple times either through the same or different virtual disk servers, and each exported instance of the virtual disk backend can then be assigned to either the same or different guest domain.
- Support for physical-to-virtual migration tool (P2V)
  - ◆ Automatically converts an existing physical system to a virtual system that runs in a ldom.
  - ◆ Source: Any sun4u SPARC with minimum Solaris 8 OS, sun4v system that run Solaris 10 OS.
- Support for configuration assistant tool
  - ◆ Runs on CMT servers
  - ◆ Leads you through the configuration of a logical domain by setting basic properties
  - ◆ Available as a graphical user interface (GUI) and terminal based tool (ldmconfig)



(GUI)



(ldmconfig)

# Logical Domains – Reconfiguration

## Dynamic Reconfiguration

- The ability to add or remove resources while the operating system is running.
- Is supported in the following resources
  - ◆ Virtual CPU – all solaris OS versions
  - ◆ Virtual I/O Devices – Supported in at least Solaris 10 10/08 OS
  - ◆ Memory - not supported
  - ◆ Physical I/O Devices – not supported
- To use the dynamic reconfiguration capability in the Logical Domain Manager CLI, you must have the Logical Domains dynamic reconfiguration daemon “drd” running in the domain you want to apply the change.

## Delayed Reconfiguration

- Operations take effect after the next reboot of the OS or stop and start of the logical domain if no OS is running.
- The LDM **ldm cancel-operation reconf** command cancels delayed reconfiguration on the control domain
- The delayed reconfiguration can be listed by using **ldm list-domain** command

# Logical Domains - Hardware

## Coolthread Servers

- Listing of some of the current systems available and their maximum Ldoms capabilities:

Processor Generation	System	Sockets	Cores per Sockets	Threads per cores	Max. Domains*	Max. I/O Domains	Min. Ldoms Version
UltraSparc-T1	T1000	1	8	4	32	2	1.0
	T2000	1	8	4	32	2	1.0
	T6300	1	8	4	32	2	1.0
UltraSparc-T2	T5120	1	8	8	64	1	1.0.1
	T5220	1	8	8	64	1	1.0.1
	T6320	1	8	8	64	1	1.0.1
UltraSparc-T2 Plus	T5140	2	8	8	128	2	1.0.3
	T5240	2	8	8	128	2	1.0.3
	T5440	4	8	8	128	4	1.0.3
	T6340	2	8	8	128	2	1.0.3

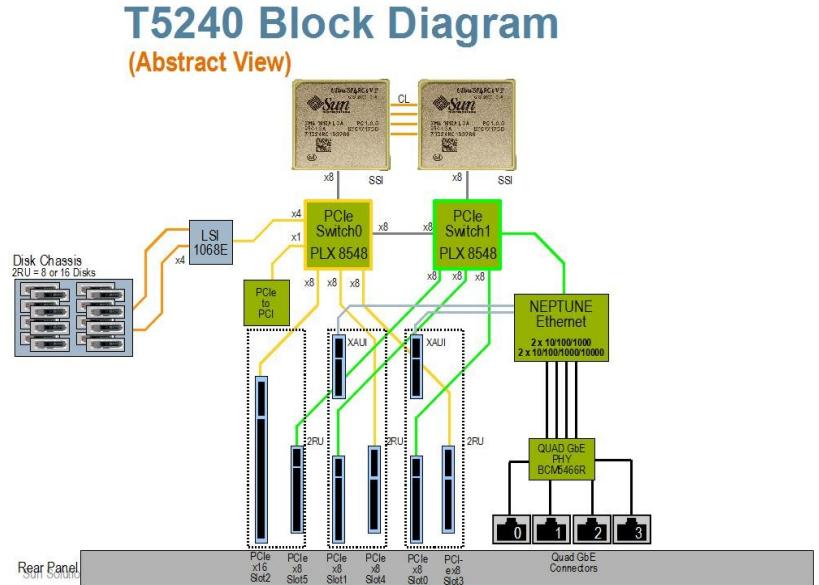
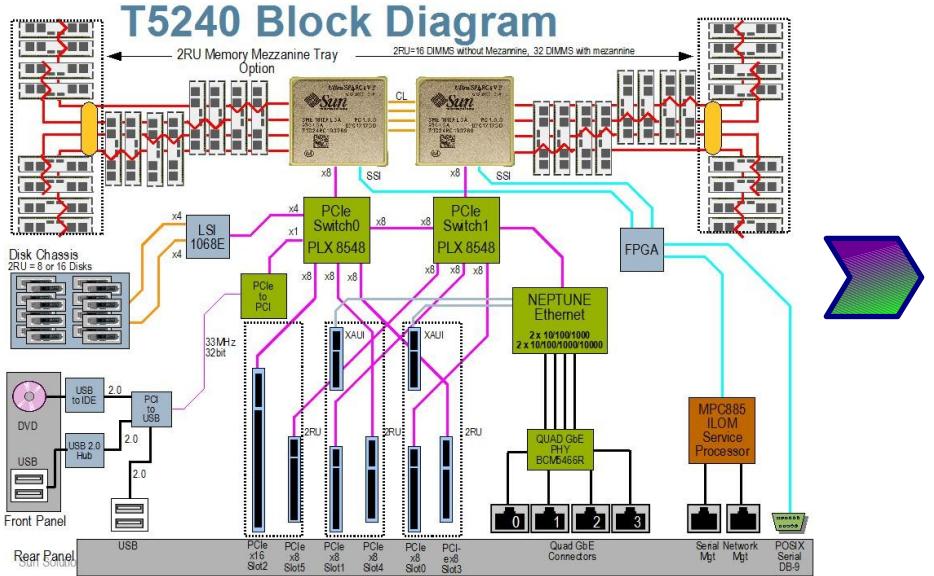
\* Depends on your virtual machine configuration based on your customer needs (apps, network, disk, etc)\*

Does Ldoms Software require a minimum firmware level? Yes!, It depends on the SW version you are installing

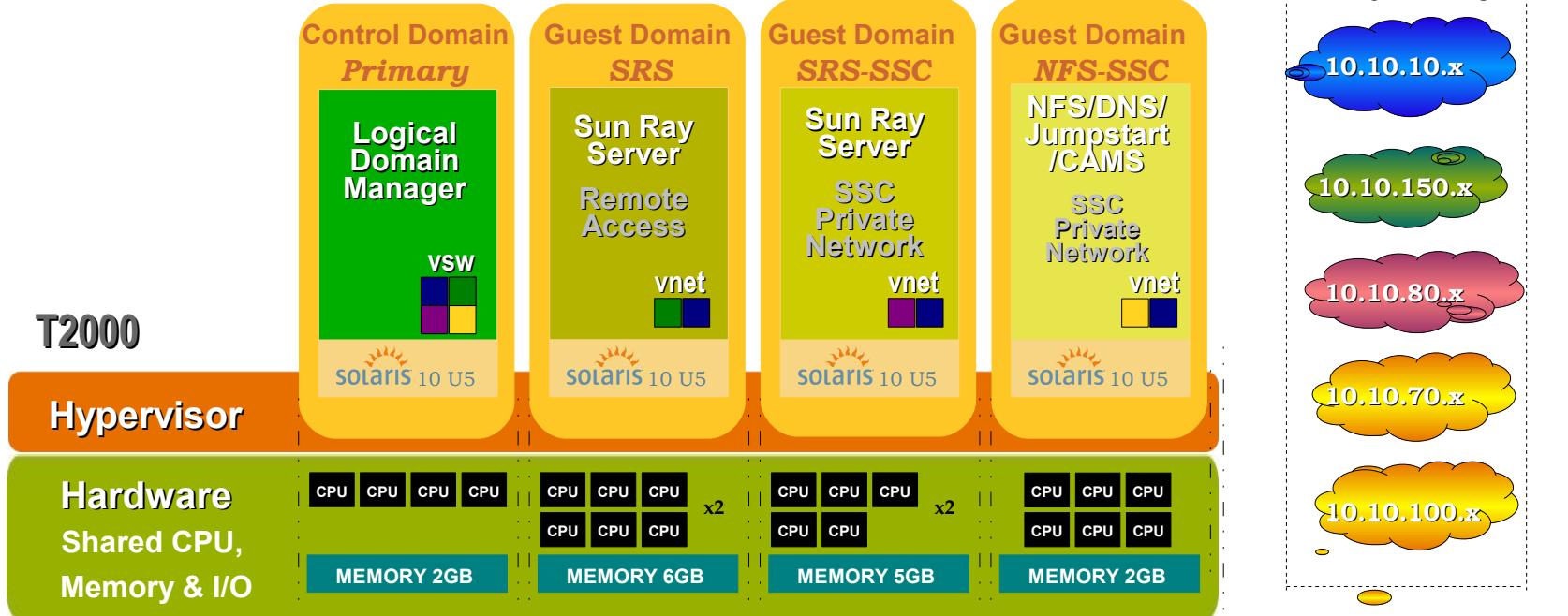
Please visit the following link: <http://cpubringup.sfbay.sun.com/twiki/bin/view/LDoms/LDomsMatrix>

# Logical Domains – Split Bus Configuration

- PCI-E buses on Coolthread platforms can be assigned to separate domains
  - ◆ Provides direct I/O access to two domains in the system
- I/O Domains can function as service domains and export virtual disks or provide network service to other domains
- Referred to “bus\_a”/“bus\_b” or “pci\_0”/“pci\_1” by the Ldoms Manager, depending of how many bus the server has.
  - ◆ Ensure following a split, each bus has appropriate disk and network available



# Case Study – SSC - LDOMs Based Management Infrastructure



**T2000 (front view)**



VDS

NAME	VOLUME	OPTIONS	DEVICE
primary-vds0	vol1	/dev/rdsk/c1t1d0s2	-> vdisk1 -> srs
primary-vds1	vol2	/dev/rdsk/c1t2d0s2	-> vdisk2 -> srs-ssc
primary-vds2	vol3	/dev/rdsk/c1t3d0s2	-> vdisk3 -> nfs-ssc

Note: Primary (Control Domain) -> /dev/rdsk/c1t0d0s2

Sun Microsystems Proprietary

**T2000 (rear view)**



- primary-vsw0
- primary-vsw1
- primary-vsw2
- primary-vsw3

# Field Request –LDOMs + S8Containers to host E10K legacy applications

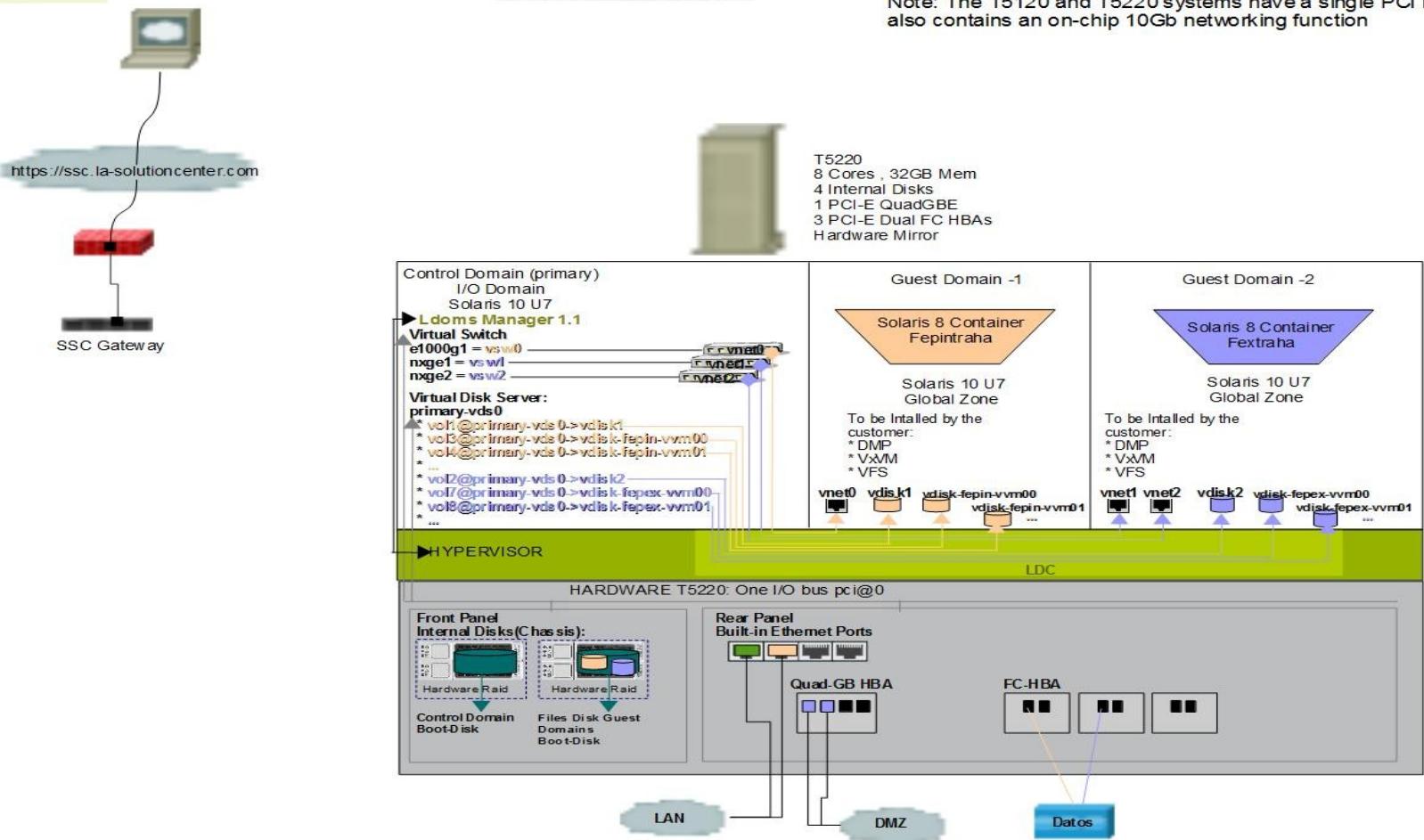


**Test for success.**  
We assembled the best team in the industry to assess unique business solutions.

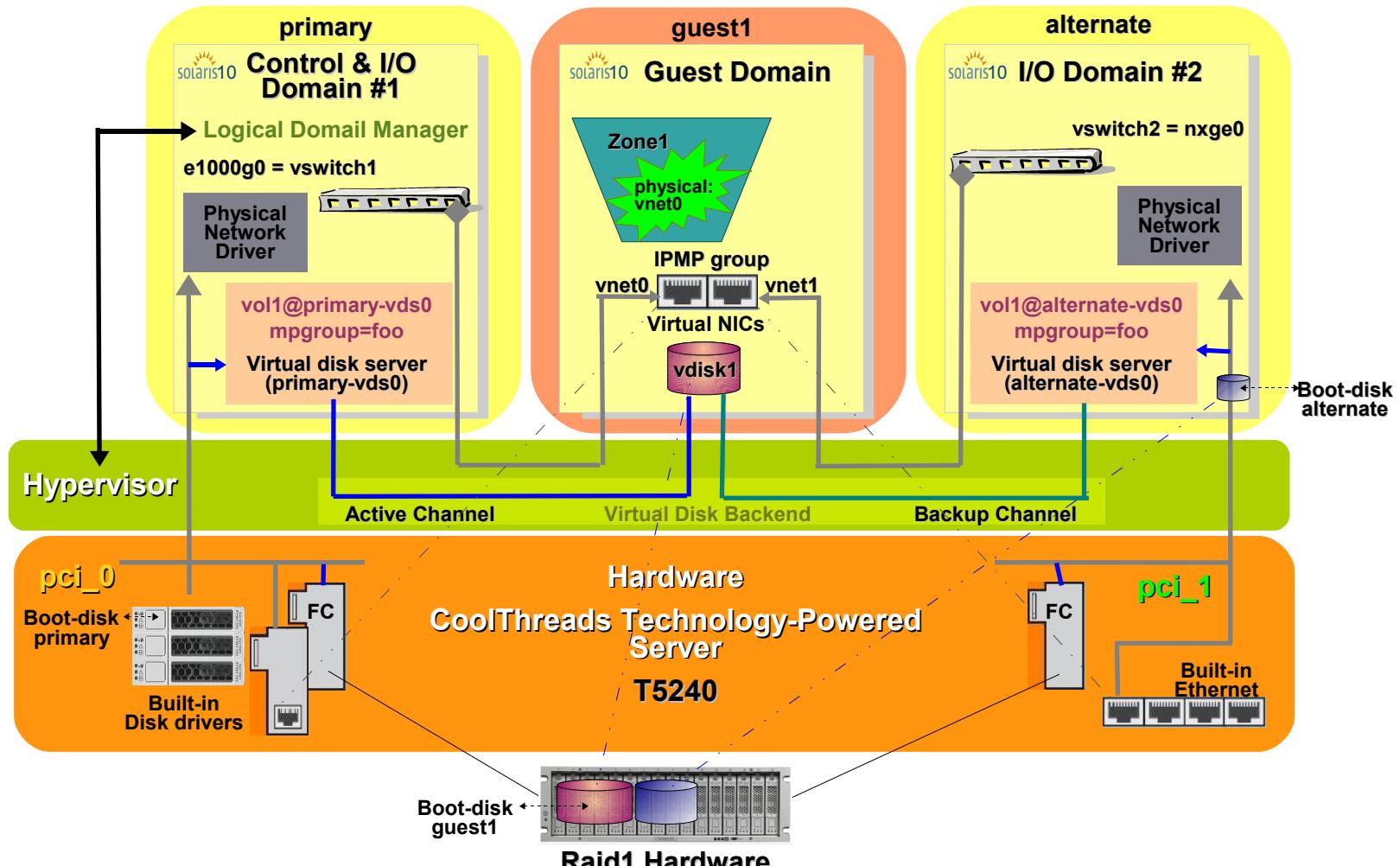


- Main Tests: Verify Feasibility of moving 2 Applications in Solaris 8 to Ldoms+ S8Containers on a T5220
- Validate the architecture for the applications on the ldoms/containers

Note: The T5120 and T5220 systems have a single PCI bus. The system also contains an on-chip 10Gb networking function



# Field Request – High Availability Configuration (Split-bus with Hardware Raid)



# Tips to identify I/O resources distribution within bus structure

- Check network I/O:
  - Built-in ethernet ports

```
root@t5240 # ls -l /dev/nxge*
```

Irwxrwxrwx	1	root	root	30 Feb 20 17:26	/dev/nxge -> ..../devices/pseudo/clone@0:nxge
Irwxrwxrwx	1	root	root	46 Feb 20 17:26	/dev/nxge0 -> ..../devices/pci@500/pci@0/pci@8/network@0:nxge0
Irwxrwxrwx	1	root	root	48 Feb 20 17:26	/dev/nxge1 -> ..../devices/pci@500/pci@0/pci@8/network@0,1:nxge1
Irwxrwxrwx	1	root	root	48 Feb 20 17:26	/dev/nxge2 -> ..../devices/pci@500/pci@0/pci@8/network@0,2:nxge2
Irwxrwxrwx	1	root	root	48 Feb 20 17:26	/dev/nxge3 -> ..../devices/pci@500/pci@0/pci@8/network@0,3:nxge3

- Pcie network card

```
root@t5240 # ls -l /dev/e*
```

Irwxrwxrwx	1	root	root	32 Feb 24 10:32	/dev/e1000g -> ..../devices/pseudo/clone@0:e1000g
Irwxrwxrwx	1	root	root	49 Feb 24 10:32	/dev/e1000g0 -> ..../devices/pci@400/pci@0/pci@9/ethernet@0:e1000g0
Irwxrwxrwx	1	root	other	29 Feb 20 16:52	/dev/eri -> ..../devices/pseudo/clone@0:eri

T5240 (Rear view)



# Control Domain Configuration

- Create default services for the control domain:
  - ◆ vdiskserver – virtual disk server
  - ◆ vswitch – virtual switch service
  - ◆ vconscon – virtual console concentrator service
- Default Services creation for the control domain

Create the virtual disk server (vds) that is going to allow importing virtual disks into Guest Domains

```
root@t5240# ldm add-vds primary-vds0 primary
```

Create the virtual console concentrator service (vcc) in order to use the virtual network terminal daemon (vntsd) and use it like a concentrator to other logical domains consoles

```
root@t5240# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

Create a SwitchService (vsw) in order to enable networking between the virtual network (vnet) emulated in each guest domain

```
root@t5240# ldm add-vsw net-dev=e1000g0 primary-vsw0 primary  
(corresponding to the pcie card installed)
```

*Note: this command had created a virtual switch service **primary-vws0** related to the network interface **e1000g0** and the control domain **primary**, in automatic way the MAC address is assigned, although you are also capable to add it manually, “make sure you do not duplicate it”, ex:*

```
root@t5240# ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=e1000g0 primary-vsw0 primary
```

# Control Domain Configuration

- Reconfigure cryptographic devices in the control domain

\* Reconfigure crypto units \*

```
root@t5240# ldm set-mau 0 primary
```

- Reconfigure the Control Domain in order to free resources. It will allow you to create the alternate domain and as many guests domains as you need.

```
root@t5240 # ldm set-vcpu 4 primary
```

```
root@t5240 # ldm set-memory 4g primary
```

```
root@t5240 # ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-ndc--	SP	4	4G	0.0%	6m

- Remove the appropriate bus that is going to be assigned to the alternate domain.

```
root@t5240 # ldm rm-io pci_1 primary
```

- Add a logical domain machine configuration to the system controller (SC). In this example, the following command would add or save a configuration using the name “**split-conf**”

```
root@t5240# ldm add-config split-conf
```

```
root@t5240# ldm list-config
```

factory-default (please do not erase this configuration it will allow you to recover the system to its manufacture configuration)

split-conf [current]

\* After reboot all the configuration will take effect \*

```
root@T5240# shutdown -y -go -i6
```

# Control Domain Configuration

- Verify the services and the bus bound to this domain

```
root@t5240 # ldm list-bindings
NAME      STATE   FLAGS  CONS  VCPU MEMORY UTIL UPTIME
primary   active  -n-cv- SP    4    4G    0.6% 3m
..
VCPU
VID PID UTIL STRAND
0 0 0.5% 100%
1 1 0.3% 100%
2 2 0.9% 100%
3 3 0.1% 100%
MEMORY
RA PA SIZE
0xe000000 0xe000000 4G
IO
DEVICE PSEUDONYM OPTIONS
pci@400 pci_0
VCC
NAME PORT-RANGE
primary-vcc0 5000-5100
VSW
NAME MAC NET-DEV DEVICE DEFAULT-VLAN-ID PVID VID MODE
primary-vsw0 00:14:4f:ff:ff:07 e1000g0 switch@0 1 1
VDS
NAME VOLUME OPTIONS MPGROUP DEVICE
primary-vds0
VCONS
NAME SERVICE PORT
SP
```

\* plumb the virtual switch \*

```
root@t5240# ifconfig vsw0 plumbr
root@t5240# ifconfig e1000g0 down unplumb
root@t5240# ifconfig vsw0 10.10.10.228 netmask 255.255.255.0 broadcast + up
root@t5240# mv /etc/hostname.e1000g0 /etc/hostname.vsw0
```

\* if you are using DHCP \*

```
root@t5240# mv /etc/dhcp.e1000g0 /etc/dhcp.vsw0
```

\* and edit /etc/hosts if necessary \*

# Alternate I/O Domain Creation

- Create the backup services for the control domain or the second I/O domain

```

root@t5240 # ldm create alternate
root@t5240 # ldm set-mau 0 alternate
root@t5240 # ldm set-vcpu 4 alternate
root@t5240 # ldm set-mem 4g alternate
root@t5240 # ldm add-io pci_1 alternate → bus available for this configuration
root@t5240 # ldm add-vds alternate-vds0 alternate
root@t5240 # ldm add-vsw net-dev=nxge0 alternate-vsw0 alternate
root@t5240 # ldm bind alternate
root@t5240 # ldm list
NAME      STATE    FLAGS  CONS   VCPU MEMORY UTIL UPTIME
primary   active   -n-cv- SP     4     4G    0.2% 23m
alternate  bound    ---v- 5000   4     4G
* Verify it's assigned services and bus *
root@t5240 # ldm list-bindings alternate
NAME      STATE    FLAGS  CONS   VCPU MEMORY UTIL UPTIME
alternate  bound    ---v- 5000   4     4G
...
IO
  DEVICE      PSEUDONYM      OPTIONS
  pci@500      pci_1
VSW
  NAME        MAC          NET-DEV DEVICE  DEFAULT-VLAN-ID PVID VID      MODE
  alternate-vsw0 00:14:4f:ff:ff:06 nxge0  switch@0 1
VDS
  NAME        VOLUME      OPTIONS      MPGROUP      DEVICE
  alternate-vds0
VCONS
  NAME        SERVICE      PORT
  alternate  primary-vcc0@primary  5000 → use this port to connect to it later on

```

# Alternate Boot-device Configuration

- Check alternate's raw configuration

```

root@t5240 # ldm start alternate
LDom alternate started
root@t5240 # telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^].
Connecting to console "alternate" in group "alternate" ....
Press ~? for control options ..
{0} ok devalias
alternate-vds0      /virtual-devices@100/channel-devices@200/virtual-disk-server@0
alternate-vsw0      /virtual-devices@100/channel-devices@200/virtual-network-switch@0
virtual-console     /virtual-devices/console@1
name                aliases
boot
* Let's check the path to disk and network available in the domain *
{0} ok show-disks
a) /pci@500/pci@0/pci@c/SUNW,qlc@0,1/fp@0,0/disk
b) /pci@500/pci@0/pci@c/SUNW,qlc@0/fp@0,0/disk → first pcie-fc port
q) NO SELECTION
Enter Selection, q to quit: q
{0} ok show-nets
a) /pci@500/pci@0/pci@8/network@0,3
b) /pci@500/pci@0/pci@8/network@0,2
c) /pci@500/pci@0/pci@8/network@0,1
d) /pci@500/pci@0/pci@8/network@0 → first built-in ethernet port (ngxe0)
q) NO SELECTION
Enter Selection, q to quit: q

```

# Alternate Boot-device Configuration

- Verify the pcie port's WWN where you have connected the array with the volume that is going to be used as alternate's boot-device

```

{0} ok probe-scsi-all
 pci@500:pci@0:pci@c/SUNW,qlc@0,1
 QLogic QLE2462 Host Adapter Driver(SPARC): 1.24 11/15/06
 Firmware version 4.00.26
 Fibre Channel Link down
 Possible causes: No cable, incorrect connection mode or data rate
 /pci@500/pci@0/pci@c/SUNW,qlc@0
 QLogic QLE2462 Host Adapter Driver(SPARC): 1.24 11/15/06 → this is the port connected to the array
 Firmware version 4.00.26
 Adapter portID - 13d00
 ***** Fabric Attached Devices *****
 Dev# 0 PortID 13f00 Port WWN 200500a0b8177b20
 {0} ok cd /pci@500/pci@0/pci@c/SUNW,qlc@0
 {0} ok .properties
 ...
 manufacturer      QLGC
 version           QLE2462 Host Adapter Driver(SPARC): 1.24 11/15/06
 model             QLE2462
 name              SUNW,qlc
 port-wwn          21 00 00 1b 32 05 7f a1
 ...
 More [<space>,<cr>,q,n,p,c] ?q

```

# Alternate Boot-device (Volume Mapping)

- Once you have performed the probe-scsi-all command, you are capable to select the WWN that you identify previously in the Common Array Manager SW at the time you create the initiator, and map the volume or boot-device to the domain.

https://arwen:6789 - Create New Initiator - Sun StorageTek™ Common Array Manager - Web Browser

Sun StorageTek™ Common Array Manager

Create New Initiator

Steps Help

**Step 1:Specify Initiator Properties**

1. Specify Initiator Properties

2. Review

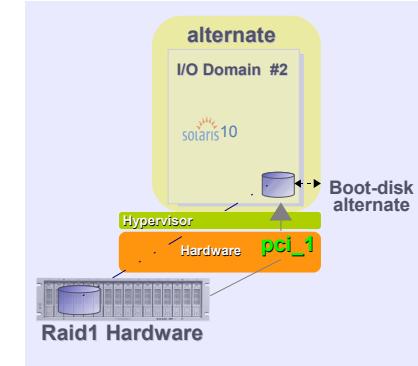
3. Results

\* Initiator Name: pcie5-port0 \* Unique Identifier:  \* Host Type: Solaris (with Traffic Manager) \* Host: T5240-alternate

Up to 30 characters, can contain 'A-Z', 'a-z', '0-9', '-' and '\_'. Spaces are not allowed.

\* Indicates required field

Previous Next Done



https://arwen:6789 - Create New Initiator - Sun StorageTek™ Common Array Manager - Web Browser

Sun StorageTek™ Common Array Manager

Create New Initiator

Steps Help

**Step 2:Review**

1. Specify Initiator Properties  
2. Review  
3. Results

Initiator Name: pcie5-port0  
Unique Identifier: **21:00:00:1B:32:05:7F:A1**  
Host Name: T5240-alternate  
Host Type: Solaris (with Traffic Manager)  
Host Group: N/A  
Host Group Members: N/A

Terminal

```
{0} ok cd /pci@500/pci@0/pci@c/SUNW,qlc0
{0} ok .properties
assigned-addresses 81060010 00000000 00000000 00000000 00001000
82060014 00000000 05000000 00000000 00040000
82060030 00000000 05040000 00000000 00040000
QLGC QLE2462 Host Adapter Driver(SPARC): 1.24 11/15/06
QLE2462
port-wwn 21:00:00:1B:32:05:7F:A1
node-wwn 28:00:00:31:03:03:7F:A1
reg 00060000 00000000 00000000 00000000 00000000
01060010 00000000 00000000 00000000 00001000
03060014 00000000 00000000 00000000 00001000
pclex1077.143.1077.143.3
pclex1077.143.1077.143.3
pclex1077.2432.3
pclex1077.2432
pciclass.c0400
1.24 11/15/06
short-version 00000000
devsel-speed
```

File Edit View Terminal Tabs Help

Previous Finish Cancel Done

- Provision the alternate domain

# Guest Domain Creation

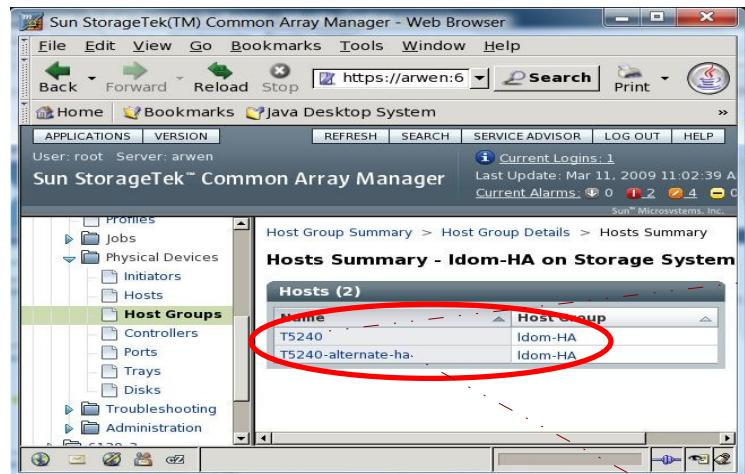
- Assign the resources for this new virtual machine for Sparc

```

root@t5240 # ldm create guest1
root@t5240 # ldm set-vcpu 4 guest1
root@t5240 # ldm set-mem 4g guest1
root@t5240 # ldm set-mau 0 guest1
* Virtual Network I/O Configuration *
root@t5240 # ldm add-vnet vnet0 primary-vsw0 guest1 } IPMP
root@t5240 # ldm add-vnet vnet1 alternate-vsw0 guest1

```

- Once you have the guest domain's volume created, proceed to map it to the I/O Domains



Sun StorageTek™ Common Array Manager - Web Browser

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop https://arwen:6789/se6130ui/login/PrimaryLogin Search Print

User: root Server: arwen

Sun StorageTek™ Common Array Manager

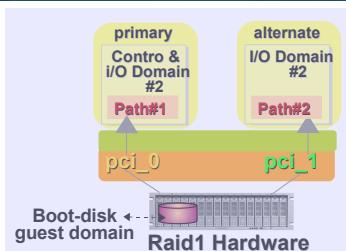
APPLICATIONS VERSION REFRESH SEARCH SERVICE ADVISOR LOG OUT HELP

Host Group Summary > Host Group Details > Hosts Summary

**Hosts Summary - Idom-HA on Storage System**

Hosts (2)

Name	Host Group
T5240	Idom-HA
T5240-alternate-ha	Idom-HA




Sun StorageTek™ Common Array Manager - Web Browser

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop https://arwen:6789/se6130ui/login/PrimaryLogin Search Print

User: root Server: arwen

Sun StorageTek™ Common Array Manager

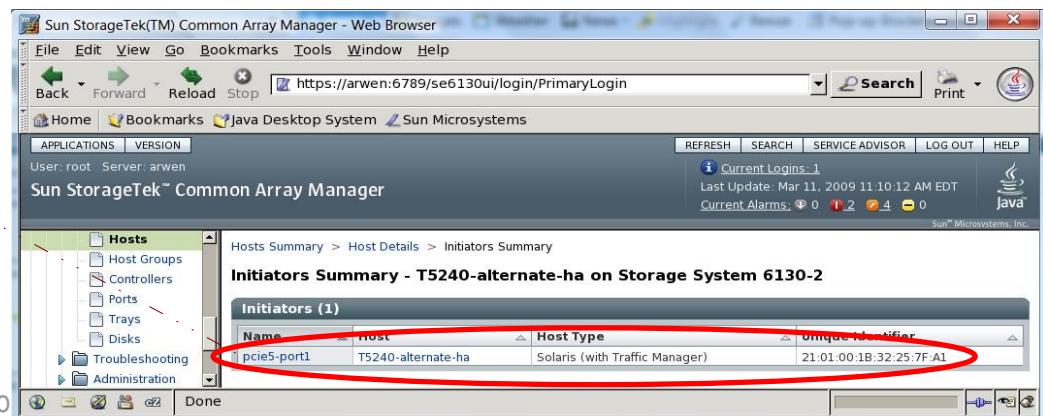
APPLICATIONS VERSION REFRESH SEARCH SERVICE ADVISOR LOG OUT HELP

Hosts Summary > Host Details > Initiators Summary

**Initiators Summary - T5240 on Storage System 6130-2**

Initiators (1)

Name	Host	Host Type	Unique Identifier
pcie3-port0	T5240	Solaris (with Traffic Manager)	10.00.00.00.C9.83.B4.8E



Sun StorageTek™ Common Array Manager - Web Browser

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop https://arwen:6789/se6130ui/login/PrimaryLogin Search Print

User: root Server: arwen

Sun StorageTek™ Common Array Manager

APPLICATIONS VERSION REFRESH SEARCH SERVICE ADVISOR LOG OUT HELP

Hosts Summary > Host Details > Initiators Summary

**Initiators Summary - T5240-alternate-ha on Storage System 6130-2**

Initiators (1)

Name	Host	Host Type	Unique Identifier
pcie1-port1	T5240-alternate-ha	Solaris (with Traffic Manager)	21:01:00:18:32:25:7F:A1

# Guest Domain boot-device

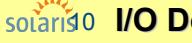
- Identify guest domain's virtual disk server device (shared LUN on a SAN) in each I/O domain

**primary**

 **Control & I/O Domain #1**

```
root@5240 # format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c1t0d0 <LSILOGIC-LogicalVolume-3000 cyl 65533 alt 2 hd 16 sec 273>
    /pci@400/pci@0/pci@8/scsi@0/sd@0,0
  1. c1t3d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
    /pci@400/pci@0/pci@8/scsi@0/sd@3,0
  2. c4t200500A0B8177B20d0 <SUN-CSM100_R_FC-0619 cyl 25598 alt 2 hd 64 sec 64>
    /pci@400/pci@0/pci@d/SUNW,emlx@0/fp@0,0/ssd@w200500a0b8177b20,0
Specify disk (enter its number):
```

**alternate**

 **I/O Domain #2**

```
root@alternate # format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t200500A0B8177B20d0 <SUN-CSM100_R_FC-0619 cyl 30718 alt 2 hd 64 sec 64>
    /pci@500/pci@0/pci@c/SUNW,qlc@0/fp@0,0/ssd@w200500a0b8177b20,0
  1. c1t200500A0B8177B20d0 <SUN-CSM100_R_FC-0619 cyl 25598 alt 2 hd 64 sec 64>
    /pci@500/pci@0/pci@c/SUNW,qlc@0,1/fp@0,0/ssd@w200500a0b8177b20,0
Specify disk (enter its number):
```

**Label the disk using either path**

# Guest Domain Creation

- Virtual disk multipathing is configured by putting the vdsdev representing the same virtual disk backend into the same multipathing group (mpgroup). Each service domain have access to the same virtual disk backend (for example a file on a NFS server, or a shared LUN on a SAN)

\* Configuring a shared Raid1 LUN on a SAN as guest1's boot-device\*

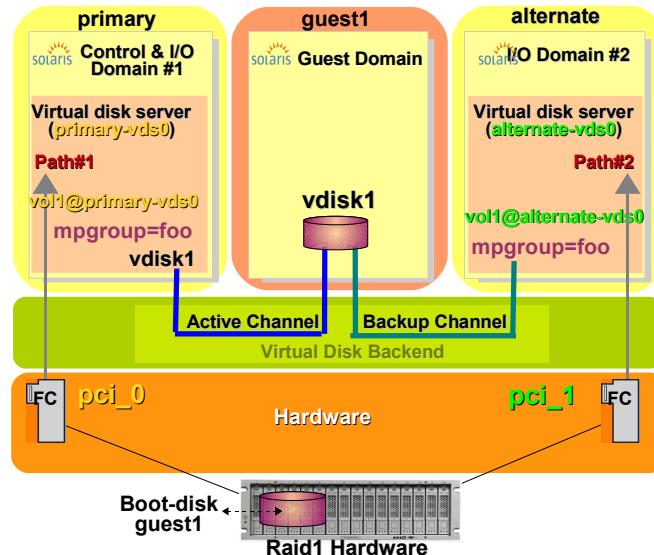
```
root@t5240 # ldm add-vdsdev mpgroup=foo /dev/dsk/c4t200500A0B8177B20d0s2 vol1@primary-vds0
```

```
root@t5240 # ldm add-vdsdev mpgroup=foo /dev/dsk/c1t200500A0B8177B20d0s2 vol1@alternate-vds0
```

\* The virtual disk will be accessible in the guest domain at the begining through the primary domain.

But if the primary domain goes down then the virtual disk will remain accessible through the alternate domain \*

```
root@t5240 # ldm add-vdisk vdisk1 vol1@primary-vds0 guest1
```



- Create the variables needed for the guest domain, bind it and provision the virtual machine

```
root@t5240 # ldm set-var auto-boot?=true guest1
```

```
root@t5240 # ldm set-var boot-device=vdisk1 guest1
```

```
root@t5240 # ldm bind guest1
```

\* Now fell free to provision your guest domain\*

```
root@t5240 # telnet localhost 5001 ..
```

Connecting to console "guest1" in group "guest1" ....

Press ~? for control options ..

{0} ok dealias

**vdisk1** /virtual-devices@100/channel-devices@200/disk@0

**vnet1** /virtual-devices@100/channel-devices@200/network@1

**vnet0** /virtual-devices@100/channel-devices@200/network@0 ...

{0} ok boot vnet1 - install

Boot device: /virtual-devices@100/channel-devices@200/network@0 File and args: - install

Requesting Internet Address for ...

# Guest Domain Creation

- Verify the services available for this domain

```
root@t5240 # ldm list -e guest1
NAME      STATE   FLAGS  CONS  VCPU MEMORY UTIL UPTIME
guest1    active  -t---- 5001  4    4G     25% 5m
...
VCPU
  VID  PID  UTIL STRAND
  0    8    100% 100%
  1    9    0.0% 100%
  2   10    0.0% 100%
  3   11    0.0% 100%
MEMORY
  RA      PA      SIZE
  0xe000000 0x20e000000 4G
...
NETWORK
  NAME      SERVICE          DEVICE  MAC        MODE  PVID VID
  vnet1    primary-vsw0@primary network@0 00:14:4f:fb:d0:33  1
  vnet2    alternate-vsw0@alternate network@1 00:14:4f:f8:22:3b  1
DISK
  NAME      VOLUME          TOUT DEVICE SERVER      MPGROUP
  vdisk1   vol1@primary-vds0      disk@0 primary    foo
VLDCC
  NAME      SERVICE          DESC
  ds       primary-vldc0@primary  domain-services
VCONS
  NAME      SERVICE          PORT
  guest1   primary-vcc0@primary  5001
```

# Guest Domain IPMP Configuration

- IP failover with IPMP configuration

\* Configure 2 fixed (or private) address and 1 floating (or public) address \*

```
root@guest1 # vi /etc/hosts
```

```
# Internet host table
```

```
#
```

```
::1 localhost
```

```
127.0.0.1 localhost
```

```
10.10.10.165 guest1 loghost → private ip
```

```
10.10.10.166 guest1-test1 } public ip
```

```
10.10.10.167 guest1-test2 }
```

```
root@guest1 # vi /etc/hostname.vnet0
```

```
guest1-test1 netmask + broadcast + group production deprecated -failover up \
```

```
addif guest1 netmask + broadcast + failover up
```

```
root@guest1 # vi /etc/hostname.vnet1
```

```
guest1-test2 netmask + broadcast + group production deprecated -failover up
```

```
root@guest1 # reboot
```

\* Verify guest1's network interfaces \*

```
root@guest1 # ifconfig -a
```

```
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
```

```
    inet 127.0.0.1 netmask ff000000
```

```
vnet0: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 2
```

```
    inet 10.10.10.166 netmask ffffff00 broadcast 10.10.10.255
```

```
    groupname production ...
```

```
vnet0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
```

```
    inet 10.10.10.165 netmask ffffff00 broadcast 10.10.10.255
```

```
vnet1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 3
```

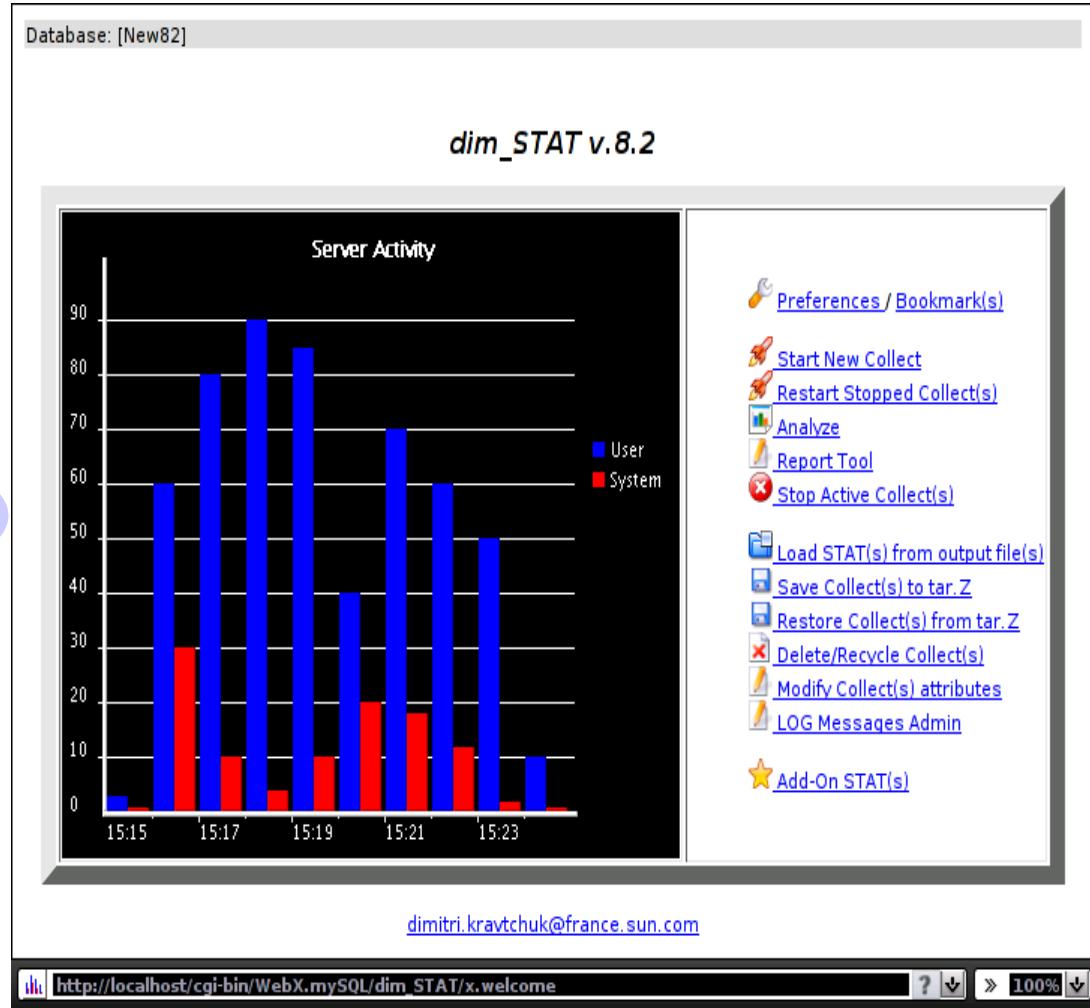
```
    inet 10.10.10.167 netmask ffffff00 broadcast 10.10.10.255
```

```
    groupname production ...
```

# What is Automatic Dynamic Reconfiguration?

## Special Guests:

- Dimitri Kravtchuk
- Matthieu Bordone





# Thank You!!

## Q/A

**Maria Jose Avila**  
[maria.avila@sun.com](mailto:maria.avila@sun.com)