

ORACLE®



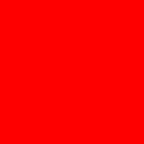
ORACLE®
VM

ORACLE®

LDoms Deep Dive – IO Best Practices for Oracle VM Server for SPARC

Stefan Hinker
EMEA Hardware Principal Sales Consultant





The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

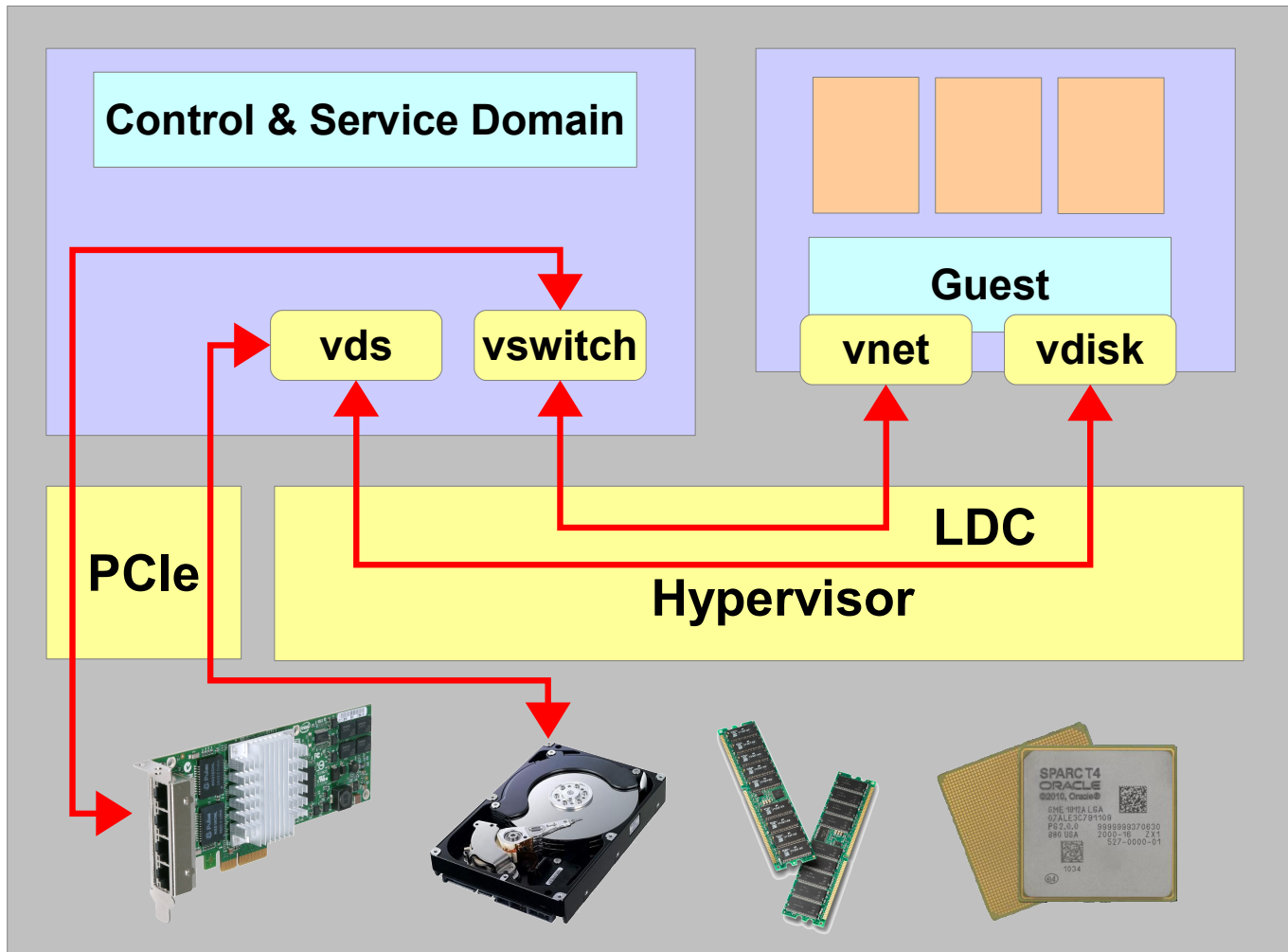
- **Introduction to LDom Virtual IO**
- Virtual Disk IO for High Performance
- Virtual Networking
 - Default Configuration
 - Reducing Latency
 - Saving on LDC Channels
- Not so Virtual IO
 - SDIO
 - SR-IOV
 - Root Domains
- Redundant IO



General Recommendations

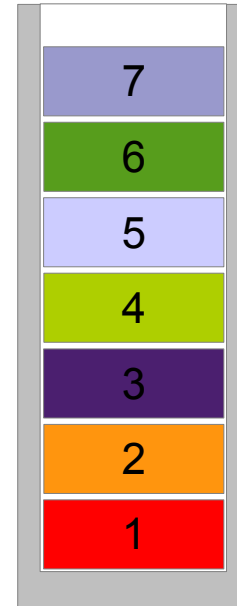
- Set the right expectations
 - Virtual IO isn't physical IO
- Use Solaris 11
 - at least in the Control- and IO-Domains
- Virtualization doesn't change physics
 - It does give us more options
- If “all virtual” isn't good enough, go physical

Domain Components



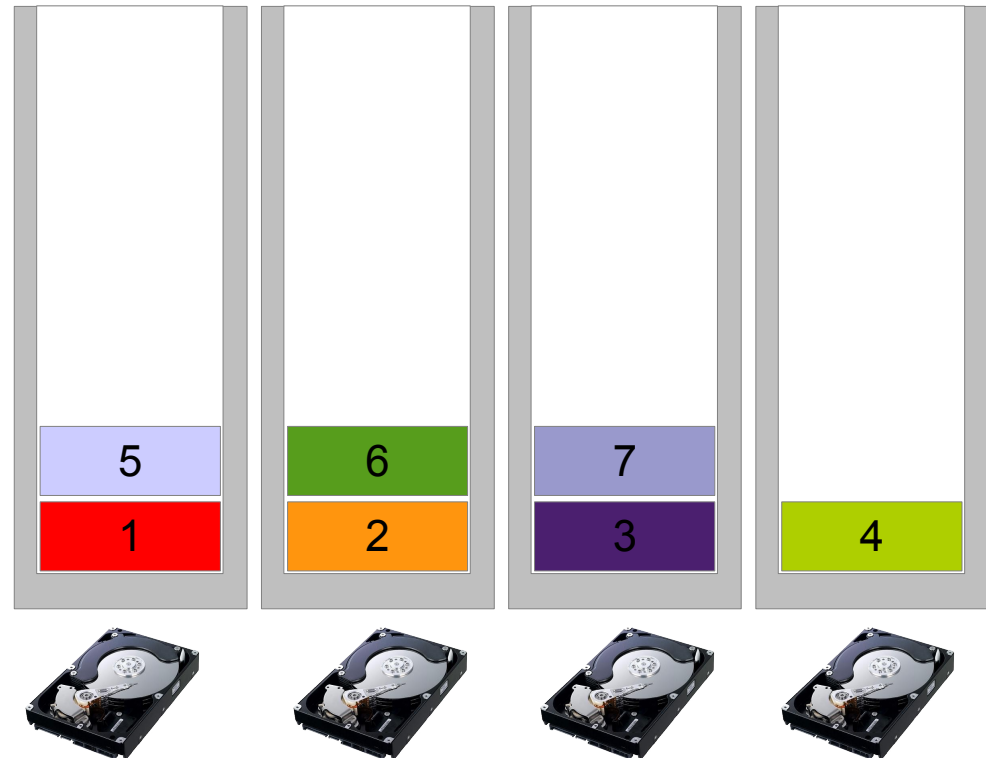
Virtual Disk IO – Considerations

- Each disk IO will see additional latency
 - impact depends on IO blocksize!
- One LDC per vDisk
 - read and write traffic sharing one channel
- One SCSI queue per vDisk



Virtual Disk IO – High Performance Recommendation

- Multiple vDisks
 - multiple SCSI queues
- Same service time per IO
- Less wait time per IO
- Better overall latency
- Higher throughput
- Redo Logs on dedicated vDisk
- Same approach as with traditional storage



Virtual Disk IO – Further Aspects

- NPIV
 - Multiple WWWns on a single SAN port
 - Supported by Solaris (10 and 11)
 - Usable for normal LDom disk backends
 - This is not a virtual HBA
- For highest performance requirements, use
 - SDIO
 - Root Domains

Agenda

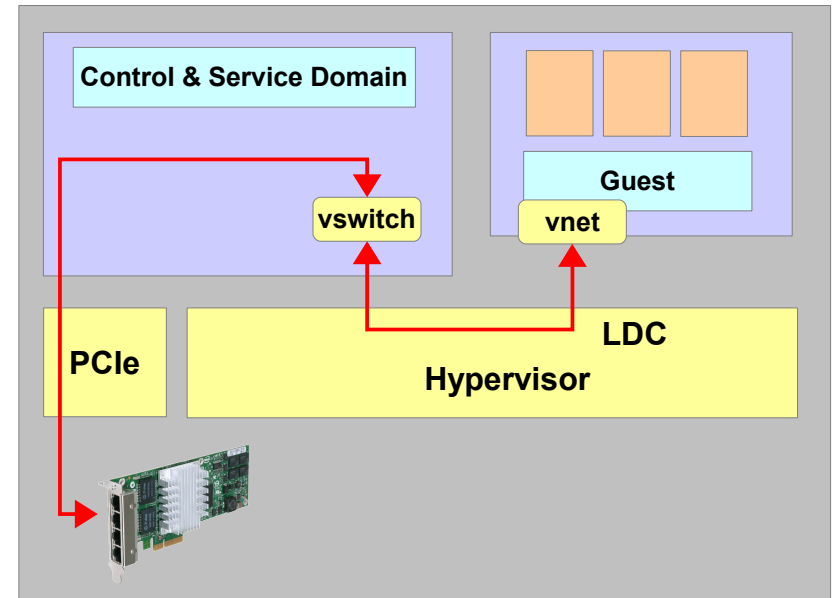
- Introduction to LDom Virtual IO
- Virtual Disk IO for High Performance
- **Virtual Networking**
 - Default Configuration
 - Reducing Latency
 - Saving on LDC Channels
- Not so Virtual IO
 - SDIO
 - SR-IOV
 - Root Domains
- Redundant IO



Virtual Networking

Default Configuration

- Network access using virtual switch connected through LDCs
- Very flexible
- Supports Live Migration
- Additional Latency



Virtual Networking

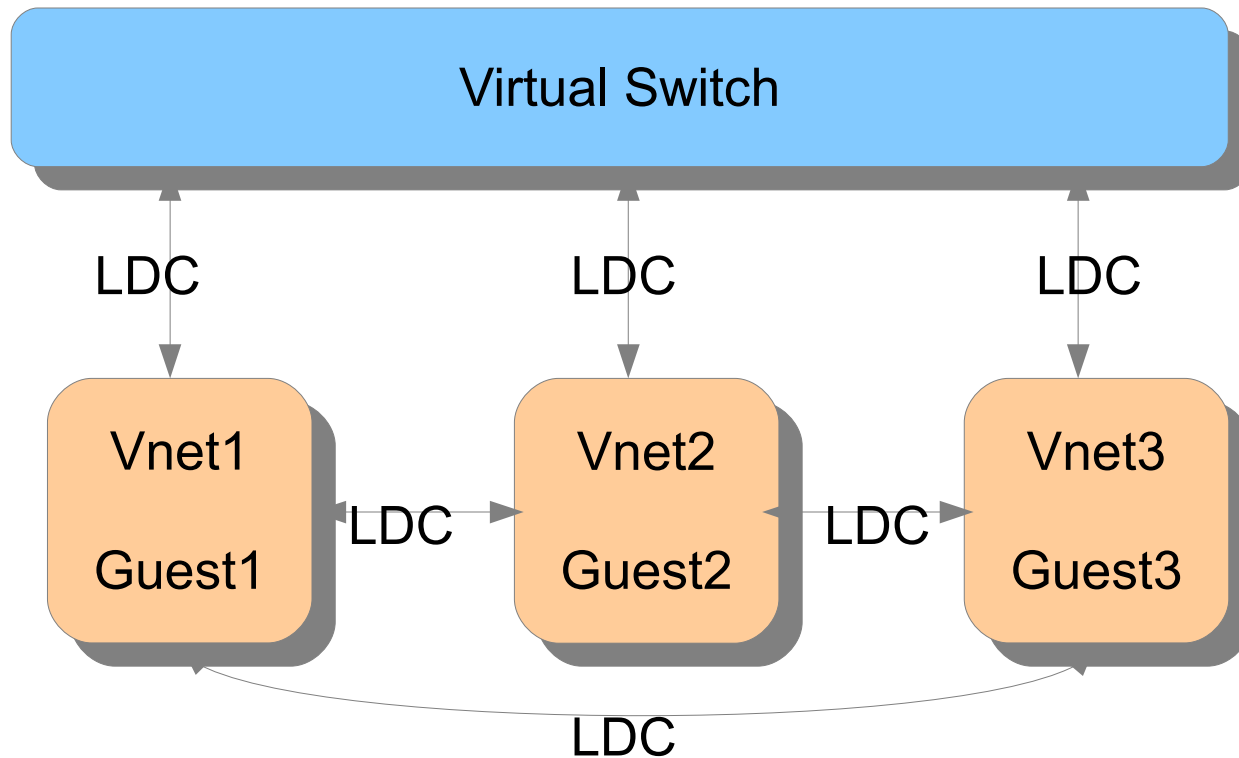
Reducing Latency

- Per-Domain feature: “extended-mapin-space”
- Requires Solaris 10u10 or Solaris 11
 - Guest & Control Domain
- Requires LDoms 2.2
- New virtual network implementation detail
 - Reduced CPU utilization
 - Reduced latency
- Requires a reboot of both guest and control domain
- Uses 4MB of free RAM per guest domain

Inter-Vnet LDC Channels

Reduce LDC usage for complex network setups

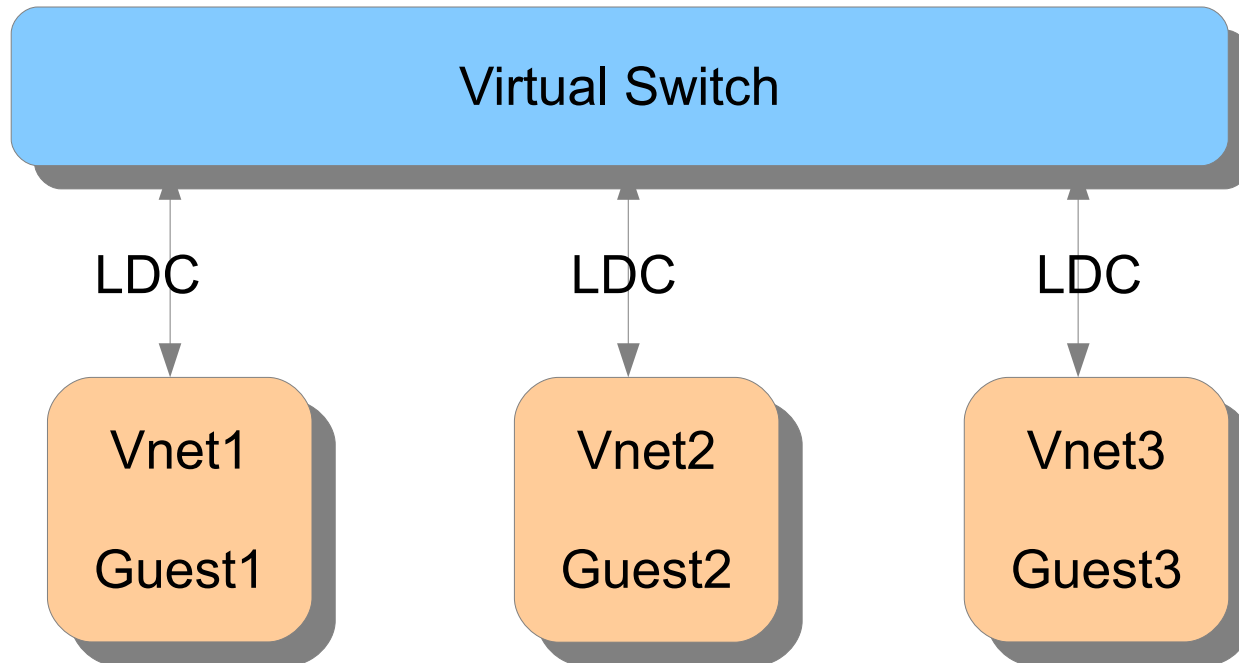
- Default behaviour: NxM LDCs



Inter-Vnet LDC Channels

Reduce LDC usage for complex network setups

- New (optional) behaviour: LDC tree



Inter-Vnet LDC Channels Details

- New CLI option 'inter-vnet-link'
 - Default: ON
 - Virtual Switch wide setting, affects all Vnets in a Virtual Switch.
 - Can be dynamically enabled/disabled without stopping the Guest domains
 - The Guest domains dynamically handle this change

Virtual Networking Recommendations

- Apply “extended-mapin-space” to domains with latency sensitive applications. That might be all domains...
- Jumbo Frames for high throughput
- If short on LDCs, use “inter-vnet-link=on”

Agenda

- Introduction to LDom Virtual IO
- Virtual Disk IO for High Performance
- Virtual Networking
 - Default Configuration
 - Reducing Latency
 - Saving on LDC Channels
- **Not so Virtual IO**
 - SDIO
 - SR-IOV
 - Root Domains
- Redundant IO



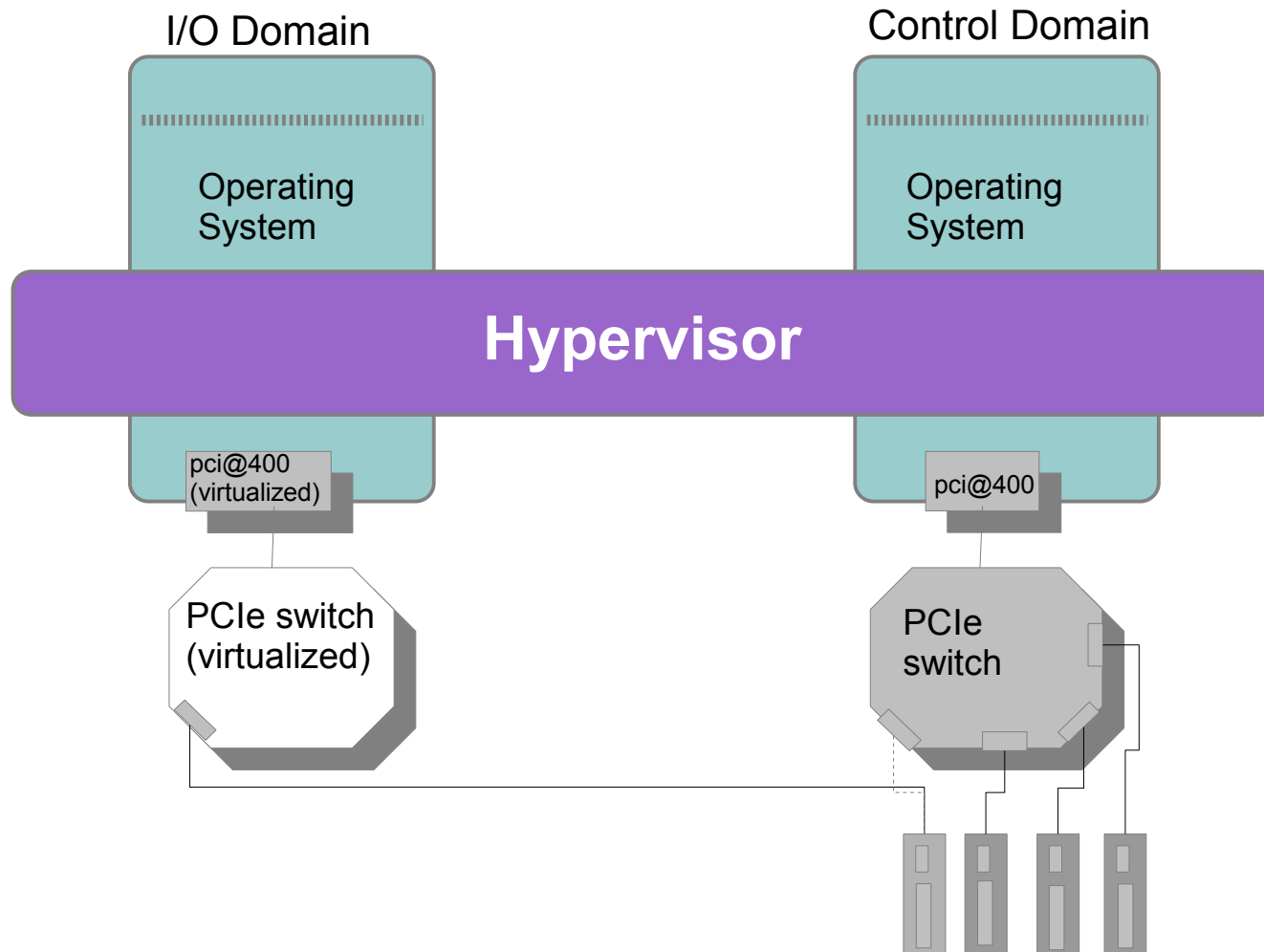
Not so Virtual IO

Hardware Access for Guest Domains

- Fully virtual IO
 - is very flexible
 - supports Live Migration
 - supports Dynamic Reconfiguration
 - is freely and abundantly available
- Virtualized Hardware
 - is as fast as “real” hardware
 - prohibits Live Migration
 - does not support Dynamic Reconfiguration
 - needs to be paid for
 - uses PCIe slots in your server

Not so Virtual IO

SDIO – Giving a PCIe Slot to a Domain



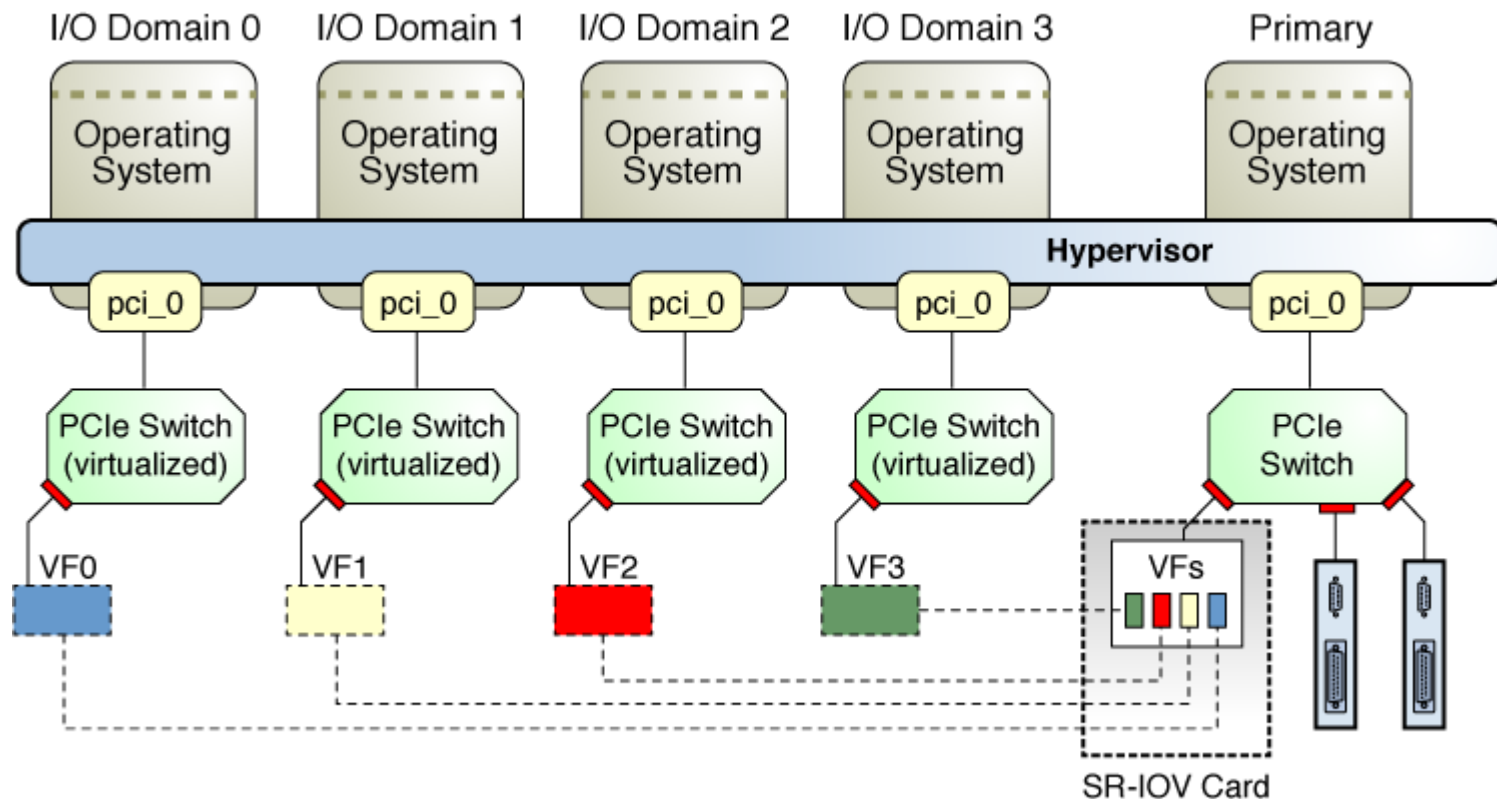
SDIO Considerations

- Has no performance penalty
- Supports devices other than disk and network
 - Tape being the most typical example
- Creates a dependency on the Control Domain
- Does not support Dynamic Reconfiguration
- Disables Live Migration of the guest
- Not supported by all PCIe adapters
 - See [MOS note 1325454.1](#) for details

Not so Virtual IO

SR-IOV – Network Hardware Virtualization

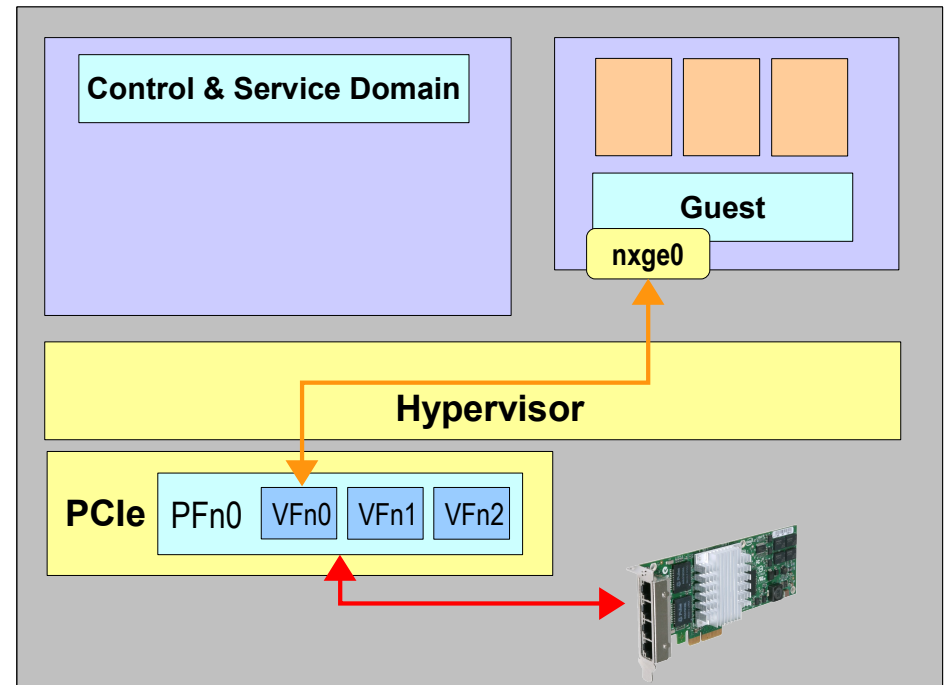
- PCIe Standard
- Requires hardware support on the card



SR-IOV in LDomS

Virtual Network Ports with Bare Metal Performance

- Create VFs in Control Domain
- Assign VFs to Guests
 - Don't forget MAC-Addresses
- Guest uses physical NIC driver (nxge, igb)
- Network-Only feature
 - no HBAs available that would support SR-IOV



SR-IOV Example

```
root@sun:~# ldm create-vf /SYS/MB/NET0/IOVNET.PF0 \  
    mac-addr=0:14:4f:fb:8a:20 alt-mac-addr=auto  
root@sun:~# reboot
```

```
root@sun:~# ldm ls-io
```

NAME	TYPE	DOMAIN	STATUS
----	----	-----	-----
/SYS/MB/NET0/IOVNET.PF0	PF	-	
/SYS/MB/NET0/IOVNET.PF0.VF0	VF	-	

```
root@sun:~# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 jupiter
```

```
root@sun:~# ldm ls-io
```

NAME	TYPE	DOMAIN	STATUS
----	----	-----	-----
/SYS/MB/NET0/IOVNET.PF0	PF	-	
/SYS/MB/NET0/IOVNET.PF0.VF0	VF	jupiter	

```
root@jupiter:~# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net2	Ethernet	up	1000	full	igbvf0

SR-IOV Considerations

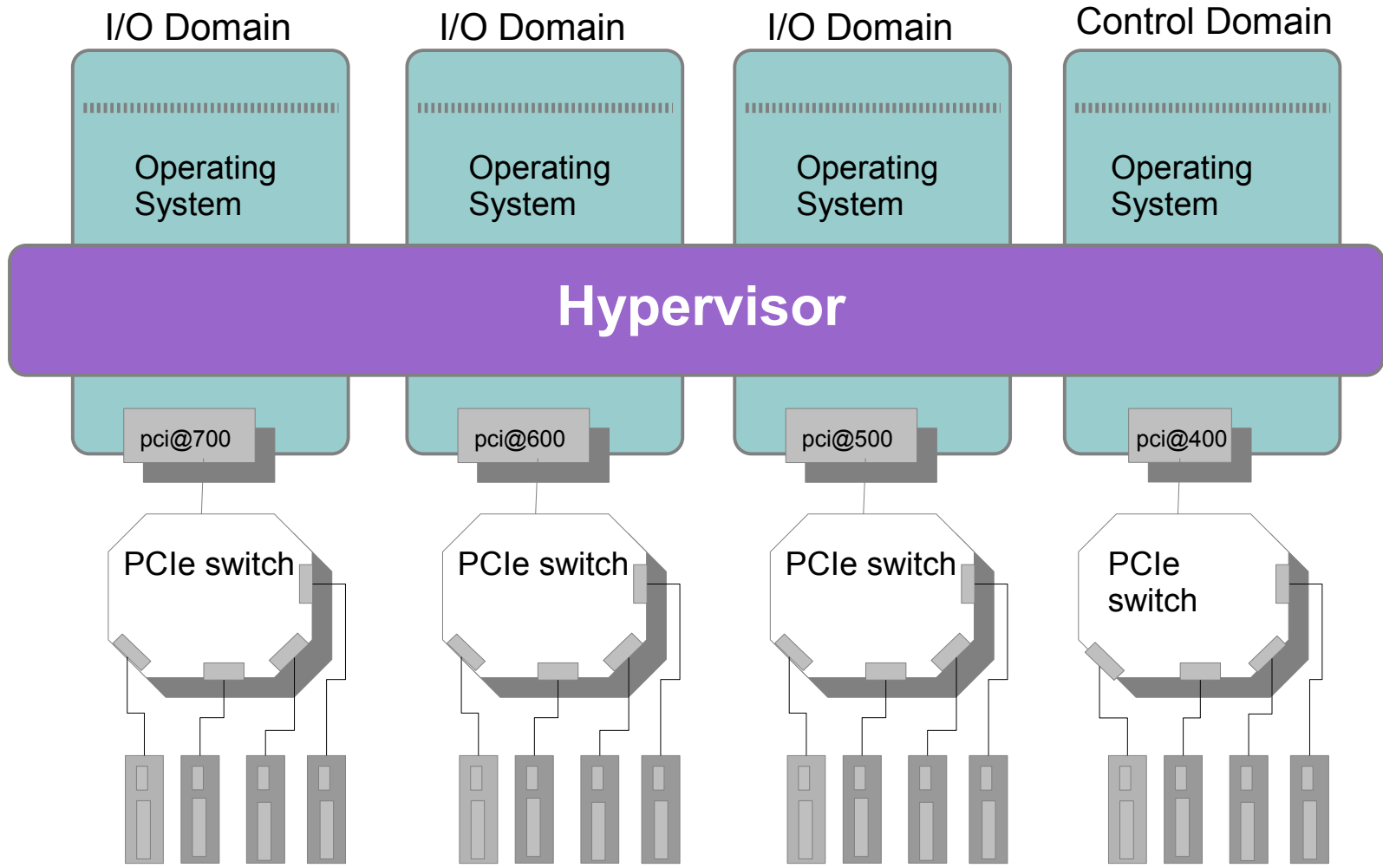
- Performance
 - near native latency & throughput
 - VFs somewhat limited compared to PFs depending on HW
- Better utilization of NIC hardware
- Cost reduction
 - fewer hardware adapters required
- Live Migration disabled for domains with Virtual Functions
- Dependancy on primary domain (similar to SDIO)
- Solaris11 VNICs on a VF are supported.
 - But limited to the number of alt-mac-addr assigned to it.
- Up to 15 IO-Domains per PCIe RC

SR-IOV Software Requirements

- Platform FW that supports SR-IOV.
 - The version of the FW is platform dependent.
- LDoms manager version 2.2 or later.
- Root domain OS that supports SR-IOV:
 - Solaris 11 or later
 - S10U11 – when it becomes available
- Guest domain OS:
 - Solaris 11 or later
 - S10U10 with a VF driver patches
 - S10U11 – when it becomes available

Root Domains

Is This Still Virtualization?

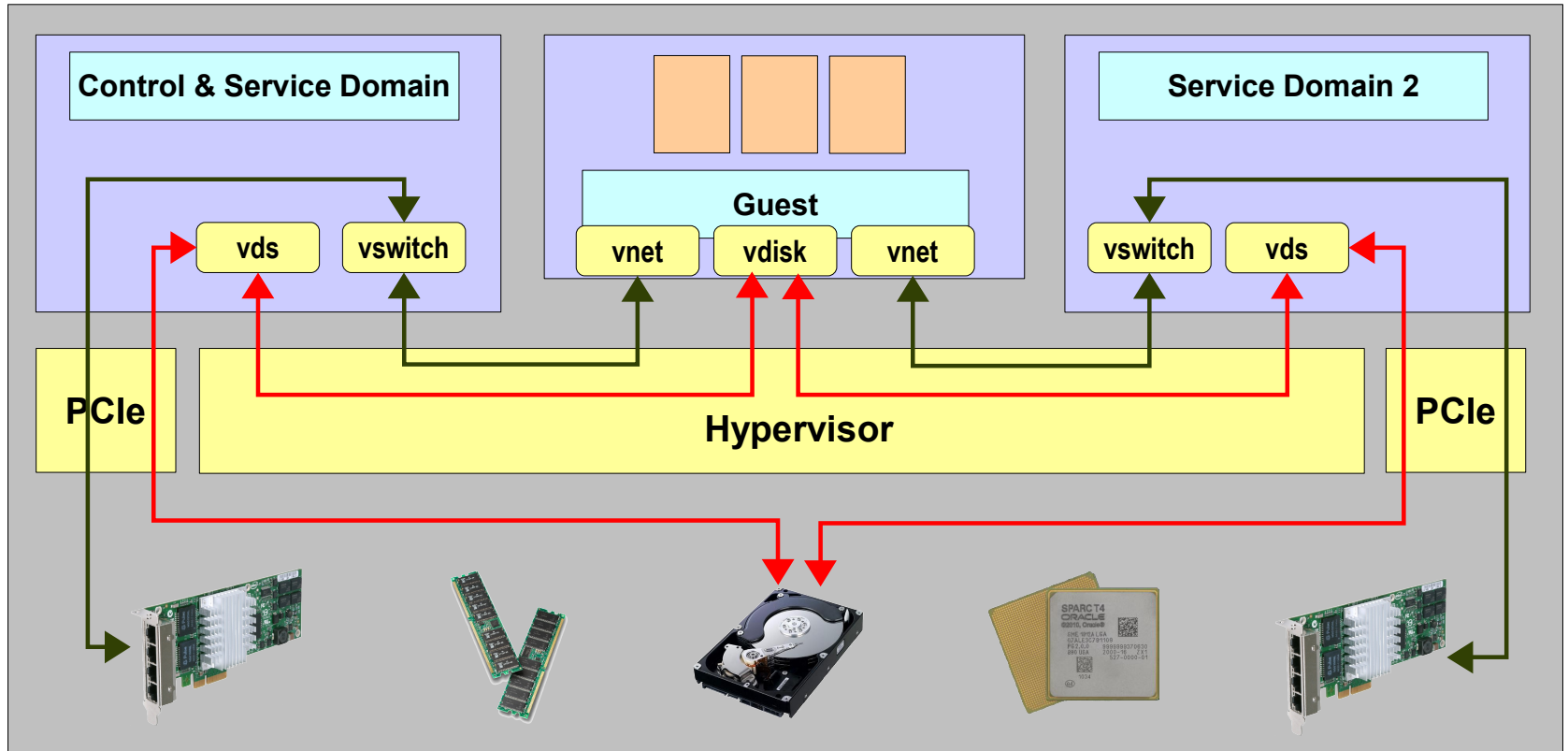


Agenda

- Introduction to LDom Virtual IO
- Virtual Disk IO for High Performance
- Virtual Networking
 - Default Configuration
 - Reducing Latency
 - Saving on LDC Channels
- Not so Virtual IO
 - SDIO
 - SR-IOV
 - Root Domains
- **Redundant IO**



Redundant IO Domains



Redundant IO domain (1): Create Domain

```
root@sun # ldm remove-io pci@400 primary

root@sun # ldm create io-domain
root@sun # ldm set-vcpu 16 io-domain
root@sun # ldm set-memory 8g io-domain

root@sun # ldm add-io pci@400 io-domain
root@sun # ldm add-vsw net-dev=igb0 switch-second io-domain
root@sun # ldm add-vds io-vds io-domain

root@sun # ldm set-variable auto-boot\?=false io-domain
root@sun # ldm bind io-domain ; ldm start io-domain

root@sun # telnet localhost 5001
```

Redundant IO domain (2): MP-IO for Guest

```
root@sun # ldm set-vdsdev mpgroup=mars \  
          mars.root@primary-vds  
root@sun # ldm add-vdsdev mpgroup=mars \  
          /guests/mars.root mars.root@io-vds  
  
root@sun # ldm add-vnet net1 switch-second mars  
  
root@sun # ldm set-vsw linkprop=phys-state switch-second  
root@sun # ldm set-vsw linkprop=phys-state switch-primary  
  
root@sun # ldm set-vnet linkprop=phys-state net0 mars  
root@sun # ldm set-vnet linkprop=phys-state net1 mars
```

Redundant IO domain (3): IPMP for Guest

```
root@mars:~# ipadm create-ip net0
root@mars:~# ipadm create-ip net1
root@mars:~# ipadm add-ipmp -i net0 -i net1 ipmp0
root@mars:~# ipadm create-addr -T static \
    -a 10.131.6.98/24 ipmp0/v4
```

```
shinker@mars:~$ ipadm show-if
```

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
ipmp0	ipmp	ok	yes	net0 net1
net0	ip	ok	yes	--
net1	ip	ok	yes	--



Stefan.Hinker@oracle.com
<https://blogs.oracle.com/cmt>

Hardware and Software Engineered to Work Together

ORACLE®