

Solaris Containers TOI

Jeff Victor

OS Ambassador

Sun Microsystems, Inc.

Advanced Containers: Objectives

- Technical Deep Dive: **How** Do I...?
- Architectural Best Practices: What **Should** I Do?
- Examples: What **Can** Be Done?
- Future: What Happens **Next**?

Agenda

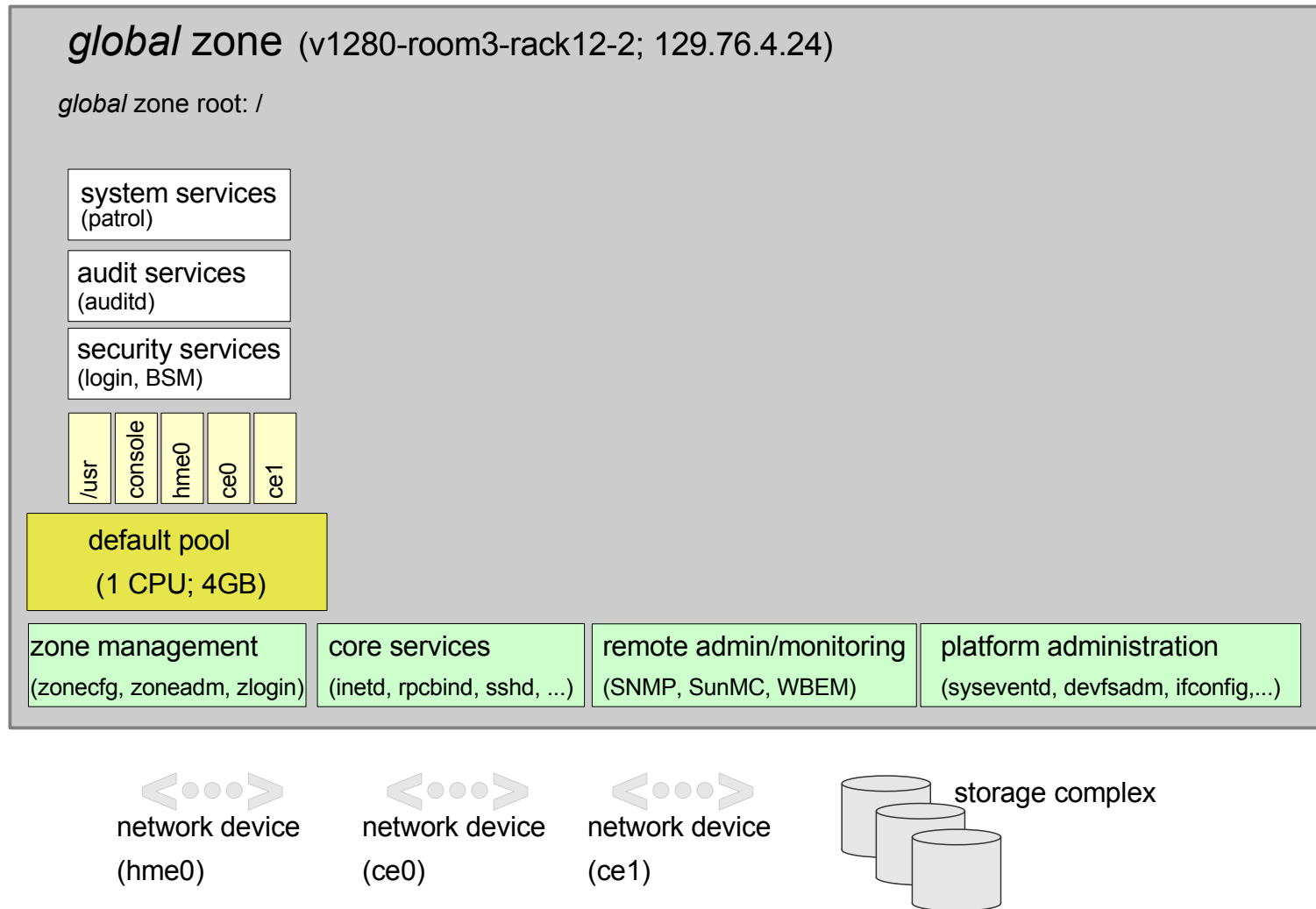
- Review of CLI
- Webmin GUI
- Consolidation Architecture
 - > Specific Topics
 - > Containers and Domains
 - > Sun Cluster 3.1 8/05
 - > Real-world Examples
- Solaris 10 1/06
- ZFS Integration

Solaris Containers

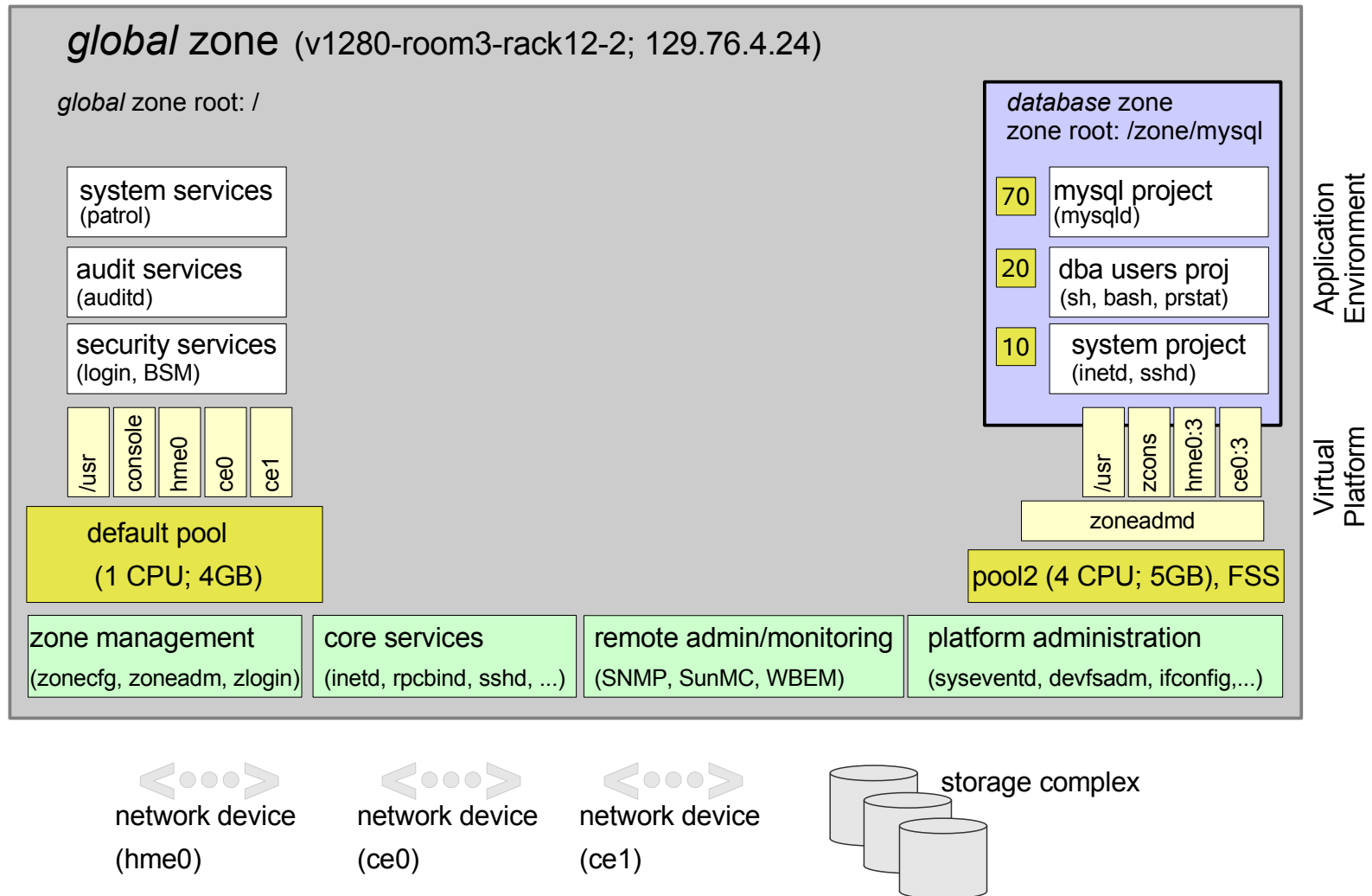
Review

- Solaris 10 added “zones” – isolated virtual application environments – to Solaris 9 Containers
- “An app in a zone thinks it's alone”
- Resource management tools provide extremely granular resource controls

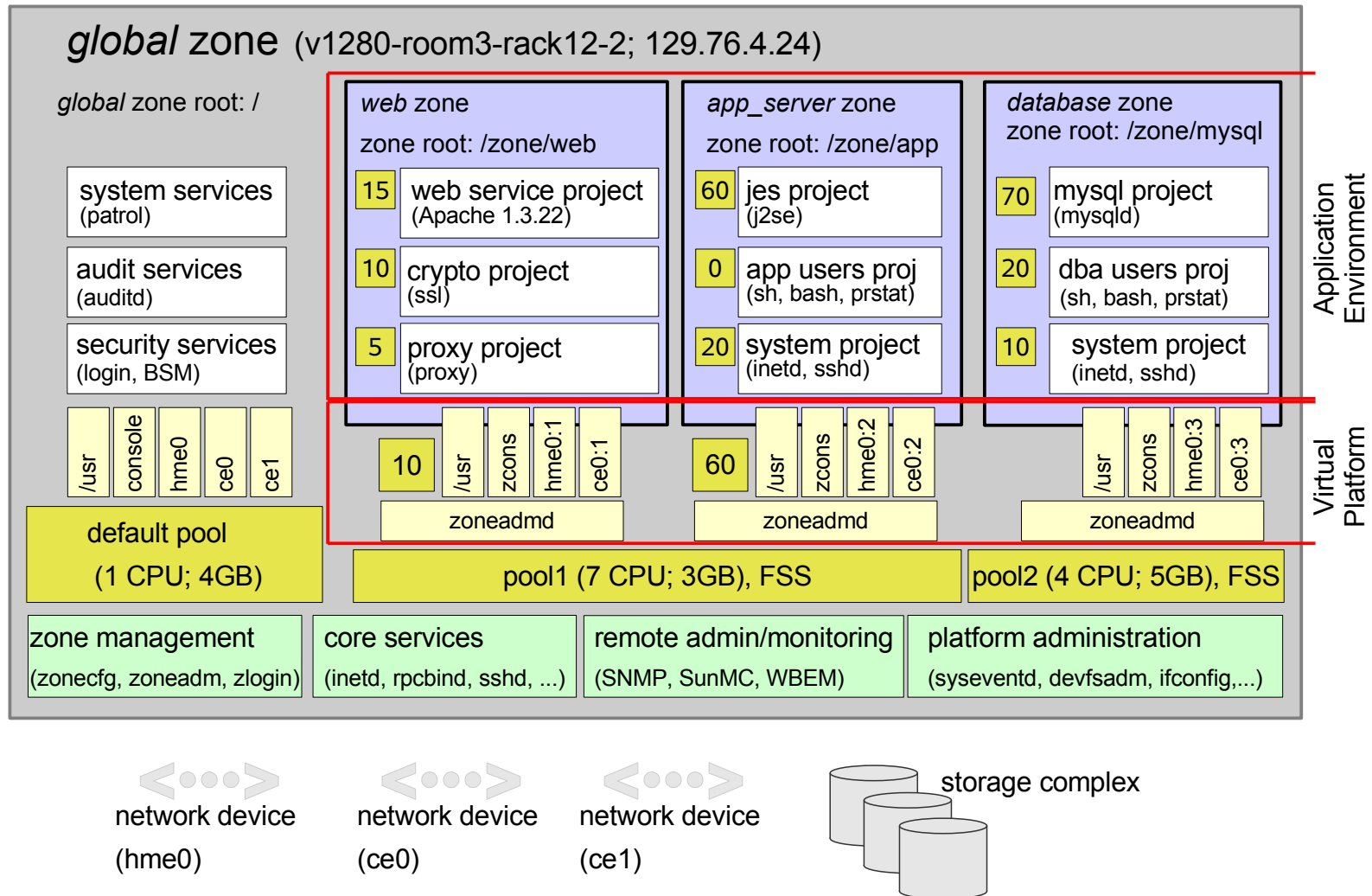
Solaris Containers



Solaris Containers



Solaris Containers



Sparse-Root or Whole-Root Zones?

- Sparse preferred when possible:
 - > Leverage ease of administration
 - > Fast creation
 - > Smaller disk footprint
- May need to use whole zones for:
 - > Specific application requirements
 - > Specific patch requirements

Solaris Containers

```
global# zfs create tank/drop
global# zfs set quota=16G tank/drop
global# zonecfg -z drop
zonecfg:drop>create
zonecfg:drop>set zonepath=/zones/zone1
zonecfg:drop>set autoboot=false
zonecfg:drop>add dataset
zonecfg:drop:dataset>set name=tank/drop
zonecfg:drop:dataset>end
zonecfg:drop>commit
zonecfg:drop>exit
global# zoneadm -z drop boot

drop# zfs create tank/drop/opt
drop# zfs set mountpoint=/opt tank/drop/opt
```

Review of Container CLI

zonecfg(1M): The Simplest Zone

```
global# zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/zones/zone1
zonecfg:zone1> exit
```

```
global# zoneadm -z zone1 install
global# zoneadm -z zone1 boot
global# zlogin -C zone1
```

zonecfg(1M): Network Interfaces

```
global# zonecfg -z zone1
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=bge0
zonecfg:zone1:net> set address=192.168.0.1
zonecfg:zone1:net> end
zonecfg:zone1> exit
```

This interface can be added to an IPMP group from the global zone.

Using a Filesystem with Packages

```
global# zonecfg -z zone1
zonecfg:zone1> add inherit-pkg-dir
zonecfg:zone1:inherit-pkgdir> set dir=/opt/sfw
zonecfg:zone1:inherit-pkgdir> end
zonecfg:zone1> exit
```

Adding a File System

UFS - Manually

- Method:

```
global# mount /dev/dsk/c1t0d0s6 /zones/zone1/root/opt/local
```

- RW or RO access in container and GZ
- Easily accessible from GZ (by root)
- Can be unmounted and remounted by GZ (if not used)
- Can use UFS on slices, SVM, lofi devices
- Simplest method to mount, complicates sys admin

Adding a File System

Automatic UFS Mount

- Cannot be mounted in GZ when container boots
- After container boots, GZ can unmount and re-mount
- Method:

```
global# newfs /dev/dsk/c1t0d0s6
global# zonecfg -z zone1
    add fs
        set dir=/opt/local
        set special=/dev/dsk/c1t0d0s6
        set raw=/dev/rdisk/c1t0d0s6
        set type=ufs
        add options [ro,nodevices]
    end
exit
global# zoneadm -z zone1 boot
```

Adding a File System

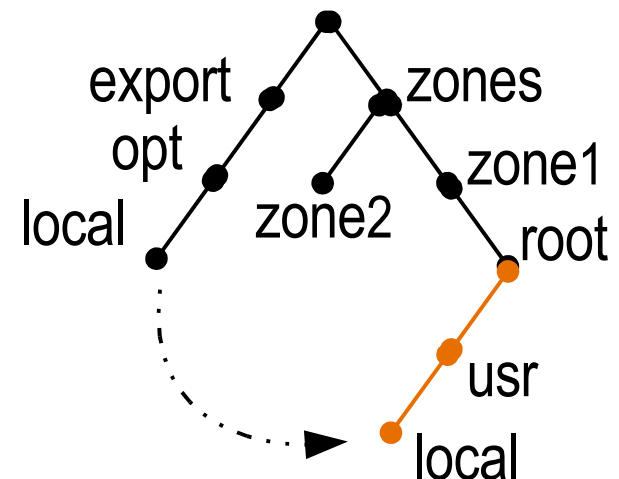
Automatic LOFS Mount

- Method:

```
global# newfs /dev/rdisk/c1t0d0s6
global# mount /dev/dsk/c1t0d0s6 /export/opt/local
```

```
global# zonecfg -z zone1
add fs
  set dir=/usr/local
  set special=/export/opt/local
  set type=lofs
end
exit
```

- Can mount in multiple zones



Adding a File System

LOFI Mount

```
global# mkfile 100m /zones/lofis/myfs
global# lofiadm -a /zones/lofis/myfs
global# newfs /dev/rlofi/1
global# zoneadm -z zone1 boot
global# mount /dev/lofi/1 /zones/zone1/root/opt/local
```

OR

```
global# zonecfg -z zone1
    add fs
        set dir=/opt/local
        set special=/dev/lofi/1
        set raw=/dev/rlofi/1
        set type=ufs
    end
exit
global# zoneadm -z zone1 halt; zoneadm -z zone1 boot
```

Adding a File System

File System Access - NFS

- Method:

```
zone1# mount -F nfs nfs-server:/export/zone1 /opt/myfiles
```

> or, in the container's /etc/vfstab:

```
nfs-server:/export/zone1 - /opt/myfiles nfs - yes -
```

- Completely managed by container
- Use /etc/vfstab to mount when container boots
- Global zone cannot “see” NFS-mounted files in a container
- Container cannot NFS-mount from its own GZ - use LOFS

File System via Direct Device Access

- Only method with management by container
- Be aware of security issues!
- Can use dsk, lofi, etc.

```
global# zonecfg -z zone1
    add device
        set match=/dev/rdisk/c1t0d0s6
    end
    add device
        set match=/dev/dsk/c1t0d0s6
    end
```

```
zone1# newfs /dev/rdisk/c1t0d0s6
zone1# mount /dev/dsk/c1t0d0s6 /opt/local
```

Choosing a File System Method

Use this information when choosing an FS method.

	LOFS	UFS	Device	LOFI	NFS
Use manually by GZ when zone down?	Yes				
Boot zone while manually mounted by GZ?	No				
Mount automatically when zone boots?	Yes	Yes	Yes ¹		Yes
Unmount by GZ after zone boots?	Yes, if not in use				No
Re-mount manually by GZ after zone boots?	Yes	Yes	Yes ²	Yes	No
Unmount by LZ (after booting)?	No	No	Yes	No	Yes
RW in GZ and RO in LZ?	Yes	No	No	No	Yes
Use SVM?	Yes	Yes	Yes	Yes	Yes
Use VxVM?	Yes?	Yes	No ⁴	?	Yes
Mount RW in multiple zones?	Yes	No	No	No	Yes

1. A bug affecting this has been fixed.

2. Use <zonepath>/root/dev/... to allow LZ to manage.

3. This will be fixed in 2006.

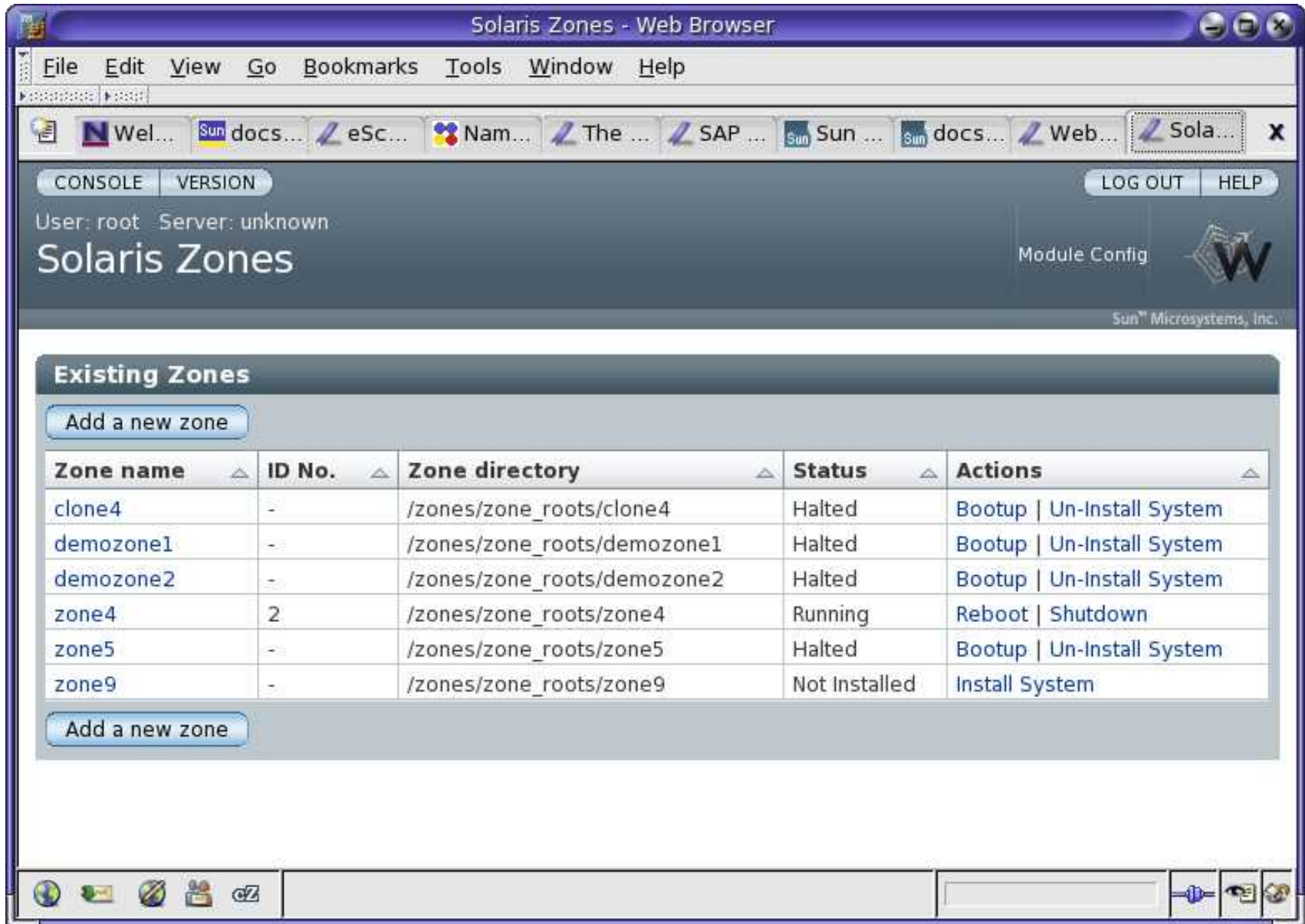
4. I don't know if Veritas plans to fix this.

Constraining Resource Usage

```
global# zonecfg -z zone1
set pool = mypool
add rctl
    set name = zone.cpu-shares
    add value (priv=privileged,limit=20,action=none)
end
add rctl
    set name = zone.max-lwps
    add value (priv=privileged,limit=100,action=none)
end
exit
```

Webmin GUI

Webmin GUI



The screenshot shows the Solaris Zones Webmin GUI in a web browser window titled "Solaris Zones - Web Browser". The browser's address bar shows the URL "http://localhost:10101/solaris-zones/". The page has a dark blue header with the "Solaris Zones" title and a "Module Config" link. Below the header, there are tabs for "CONSOLE" and "VERSION", and a "LOG OUT" button. The main content area is titled "Existing Zones" and contains a table of zones. The table has five columns: "Zone name", "ID No.", "Zone directory", "Status", and "Actions". The table lists six zones: clone4, demozone1, demozone2, zone4, zone5, and zone9. The "zone4" zone is the only one in a "Running" state. Below the table, there is a button to "Add a new zone".

Existing Zones

[Add a new zone](#)

Zone name	ID No.	Zone directory	Status	Actions
clone4	-	/zones/zone_roots/clone4	Halted	Bootup Un-Install System
demozone1	-	/zones/zone_roots/demozone1	Halted	Bootup Un-Install System
demozone2	-	/zones/zone_roots/demozone2	Halted	Bootup Un-Install System
zone4	2	/zones/zone_roots/zone4	Running	Reboot Shutdown
zone5	-	/zones/zone_roots/zone5	Halted	Bootup Un-Install System
zone9	-	/zones/zone_roots/zone9	Not Installed	Install System

[Add a new zone](#)

Webmin GUI

Create Zone - Web Browser

File Edit View Go Bookmarks Tools Window Help

CONSOLE VERSION LOG OUT HELP

User: root Server: unknown

Create Zone

Solaris Zones > Create Zone

New zone details

* Zone name

Zone files directory ☒ Automatic (under /zones) ☐ Selected directory

Initial IP address ☒ None ☐

Real network interface ▼

Install after creating? ☒ Yes ☐ No

Setup Webmin after installing? ☒ Yes ☐ No

Additional package directories

Done

Webmin GUI

Create Zone - Web Browser

File Edit View Go Bookmarks Tools Window Help

Web... Sun docs... eSc... Nam... The ... SAP ... Sun Sun ... Sun docs... Web... Cre...

Configure new zone? ☒ Yes, using settings below ... ☐ No, configure manually at console

Initial hostname ☒ Same as zone name

Initial root password ☒ Same as global system

Initial time zone ☒ Same as global system (US/Eastern)

Initial language locale ☒ Same as global system (C)

Terminal type for console ☒ Default (VT100)

Initial name service ☐ None ☒ DNS ☐ NIS ☐ NIS+

DNS or NIS domain name

DNS or NIS name servers

Default router ☐ None ☒

Create Now

Done

Webmin GUI

Edit Zone - Web Browser

FileEditViewGoBookmarksToolsWindowHelp

Wel...Sun docs...eSc...Nam...The ...SAP ...Sun Sun ...Sun docs...Web...Edit ...

CONSOLEVERSIONLOG OUTHELP

User: root Server: unknown

Edit Zone



Sun Microsystems, Inc.

In zone zone4

Solaris Zones > Edit Zone

General zone configuration

Zone name

zone4

Current status

Running

Zone directory

/zones/zone_roots/zone4

Start at system boot?

No

Resource pool

☒ None
☐

Save

Reboot

Shutdown

Install Webmin

Delete Zone

Network interfaces for zone

Add a new network interface

IP address	Interface name	Netmask	Broadcast address
129.149.171.230	bge0:1	255.255.0.0	129.149.255.255

Add a new network interface

Container Architecture Topics

Solaris 10 Privileges

Privileges Restricted to Global Zone

The security model for containers is partially implemented by limiting the use of these privileges to the global zone.

dtrace_*	net_rawaccess
proc_clock_highres	proc_lock_memory
proc_priocntl	proc_zone
sys_config	sys_devices
sys_ipc_config	sys_linkdir
sys_net_config	sys_res_config
sys_suser_compat	sys_time

Solaris 10 System Calls

System Calls that Behave Differently in a Container

Privilege restrictions cause changes in usage of system calls...

adjtime	chmod [S_ISVTX]
ioctl [I_POP, STREAMS]	link/unlink [directory]
memcntl [MC_LOCK, et al.]	mknod
msgctl [IPC_SET && msg_qbytes]	ntp_adjtime
p_online	priocntl
priocntlset	pset_*
shmctl [SHM_{UN}LOCK]	socket [SOCK_RAW]
stime	swapctl [SC_ADD, SC_REMOVE]
uadmin [HALT, REBOOT, ...]	

Solaris 10 Library Calls

Libraries and Library Calls that Behave Differently in a Container

...and library calls.

libdevinfo	libcfgadm
libpool	libkvm
libtnfctl	
clock_gettime	cpc_bind_cpu
mlock / munlock / mlockall / munlockall / plock	
timer_create	t_open [/dev/rawip]
settimeofday	

Solaris 10 Commands

Commands that Behave Differently in a Container

Device access limitations, and changes to library and system call usage, affect commands.

add_drv / rem_drv

cfgadm

devlinks

drvconfig

intrstat

plockstat

poolbind

prtdiag

share

trapstat

arp

cpustat

dispadmin

dtrace

modload / modunload

pooladm

ports

psrset

snoop

date

autopush

devfsadm

disks

fssnap

ipf

poolcfg

prtconf

route

tapes

nca

Solaris 10 Containers

Effects of Security and Isolation - Processes

- Processes
 - > All of a container's processes live in one resource pool
 - > Processes in one container cannot detect or interact with processes in other containers except through the network or a shared file system
- IPC
 - > System V shared memory between containers: No
 - > File system IPC between containers: Maybe

Solaris 10 Containers

Effects of Security and Isolation - Network

- Manage these from the global zone
 - > IPQoS
 - > IPsec
 - > IPMP
 - > IPFilter (also, no inter-zone filters yet)
 - > Routing
 - > NCA
- DHCP not yet available for containers
- Snooping not yet available for containers

Solaris 10 Containers

Effects of Security and Isolation – Network (2)

- Global zone can be an NFS server, containers cannot
- IKE not supported yet in container

Solaris 10 Containers

Effects of Security and Isolation – Storage

- Some limits to device access
 - > SVM:
 - > Solaris 10 3/05: an **SVM metadvice** can be imported into a container and used as a **raw device**, but a **filesystem** cannot be mounted on it *from within the container* – the mounting must be done *from the global zone*
 - > Solaris 10 1/06 will enable the container to mount a filesystem on an SVM metadvice
 - > VxVM metadvice cannot be imported into a container (yet)
 - > Traffic Manager (MPxIO)?

Solaris 10 Containers

Effects of Security and Isolation – Other Devices

- Some limits to device access
 - > No dtrace, kmem, ksyms, kmdb, trapstat, lockstat, uscsi, hme, ...
 - > No /devices
 - > /dev has small set of safe links
 - > Each container has a /dev/console which is a symlink to /dev/zconsole

Solaris 10 Containers

DTrace Synergy

- DTrace can be used from the GZ to monitor containers
 - > Unique ability to watch inter-'node' activity *in chronological order*
 - > Sample:

```
syscall::write:enter
/zonename=="sender" && execname=="sender" && valid==0/
{ valid      = 1;
  sendtime = timestamp; }
```

```
syscall::recv:return
/zonename=="receiver" && execname=="receiver" && valid==1/
{ latency = timestamp-sendtime;
  valid    = 0;
  @l[pid] = lquantize (latency, 0, 500000, 10000); }
```

Solaris 10 Containers

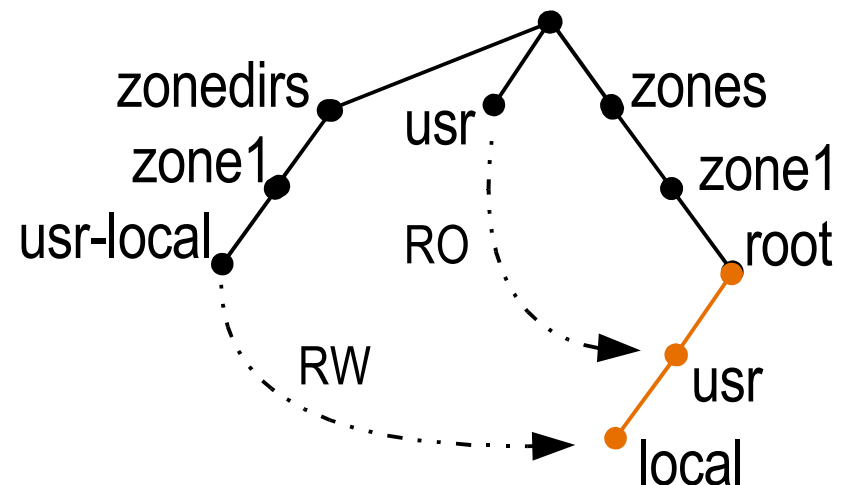
Configuring Containerized Systems – Sizing Resources

- Initial sizing should consider
 - > Peak of sum of CPU usage
 - > Sum of memory needed to avoid paging
 - > “Private” HBAs or multiplexed HBAs
 - > Traffic Manager for higher storage availability
 - > “Private” NICs or multiplexed NICs
 - > IPMP or Trunking for higher network availability
 - > Inter-zone network bandwidth and latency
- Tune sizing: monitor usage with projects, other tools

Solaris 10 Containers

Configuring Containerized Systems – File Systems

- File access vs. device access
- Sparse-root vs. whole-root
- Shared vs. private
- Read-write vs. read-only
- Backups



Solaris 10 Containers

File System Access - Miscellaneous

Type	Mount from GZ into LZ	Mount from LZ
autofs	No	Yes
cachefs	No	No
fdfs	Yes	Yes
hsfs	Yes	Yes
mntfs	No	Yes
pcfs	Yes	Yes
procfs	No	Yes
tmpfs	Yes	Yes
udfs	Yes	Yes
xmemfs	Yes	Yes

Solaris 10 Containers

The Big Picture: What *Should* You Do?

- Storage
 - > Use FS instead of devices where possible
 - > DAS, SAN, or NAS?
 - > Multiple containers running one app can share a RO NAS copy
 - > Secure public data with FS mounted RW in GZ, RO in container
 - > Generate snapshot of a container's files from GZ, backup snapshots
 - > Use shared FS for very primitive inter-zone IPC mechanism

Solaris 10 Containers

The Big Picture: What *Should* You Do?

- Network
 - > Balance load across multiple NICs
 - > Fine-tune bandwidth with IPQoS – managed from GZ
 - > VNI/Trunking/IPMP to balance load and improve availability
 - > Manage routing from GZ
 - > Each container can have its own default gateway
 - > Use IP Filter for traffic to/from other systems
 - > If inter-zone filtering needed, send all of a container's traffic to router
 - > VLANs can be used for containers, but must be plumbed in the GZ
 - > Monitor network traffic from GZ

Solaris 10 Containers

The Big Picture: What *Should* You Do?

- Resource Management
 - > FSS is more flexible than Dynamic Resource Pools
 - > Use DRP when needed to reduce application licensing costs, manage expectations
 - > To limit VM (swap space) usage, use Projects
 - > To limit physical memory usage, use Resource Capping Daemon
 - > To avoid fork bombs, use max-lwps

Solaris 10 Containers

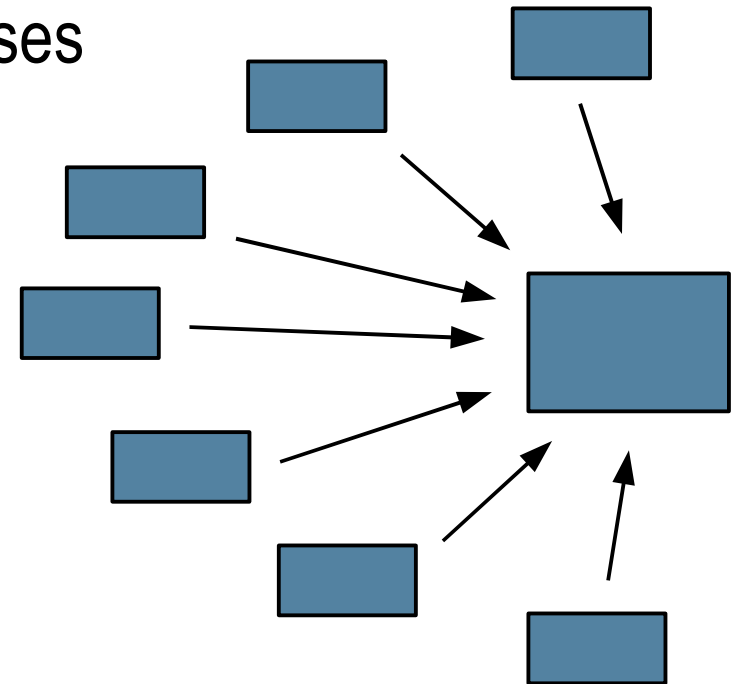
The Big Picture: What *Should* You Do?

- Namespace
 - > Use network services for:
 - > Hostnames
 - > Users, except when necessary
 - > Projects
 - > Network-wide application information

Consolidating with Containers and Dynamic System Domains

Goals of Consolidation

- Reduce TCO
 - Reduce Hardware / Increase Resource Utilization
 - Reduce / Simplify Management Tasks
 - Reduce Operating System Licenses
 - Reduce Environment Costs
 - Reduce Support Fees
- Maintain Service Levels
- Maintain Architectural Flexibility



Goals of Consolidation

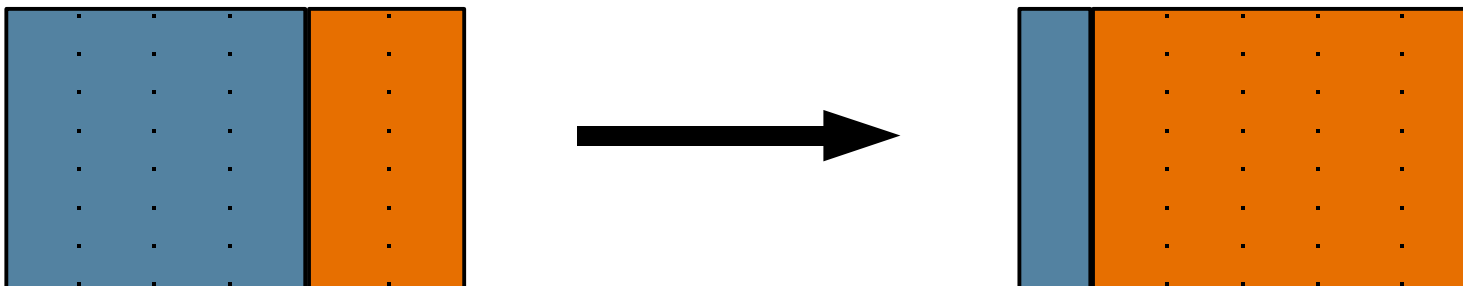
Factors

- Availability characteristics
 - > Redundant components
 - > Horizontally scaled tiers need fewer redundant components
 - > Vertically scaled nodes need more redundancy
 - > Spectrum: no redundancy $\longleftrightarrow$ multiple power utilities
 - > Characterize fault isolation – “domino effect?”
- Application characteristics
 - > Will consolidated software “play well” together?

Sun Dynamic System Domains

Review

- Mid-range and high-end Sun servers support DSD
- First introduced on Sun Enterprise 10000 in 1997
- Ability to separate a pool of hardware resources into sections, electrically isolated from each other
 - > Extremely high fault isolation
 - > Flexibility to “move” hardware from one domain to another by simply entering a command

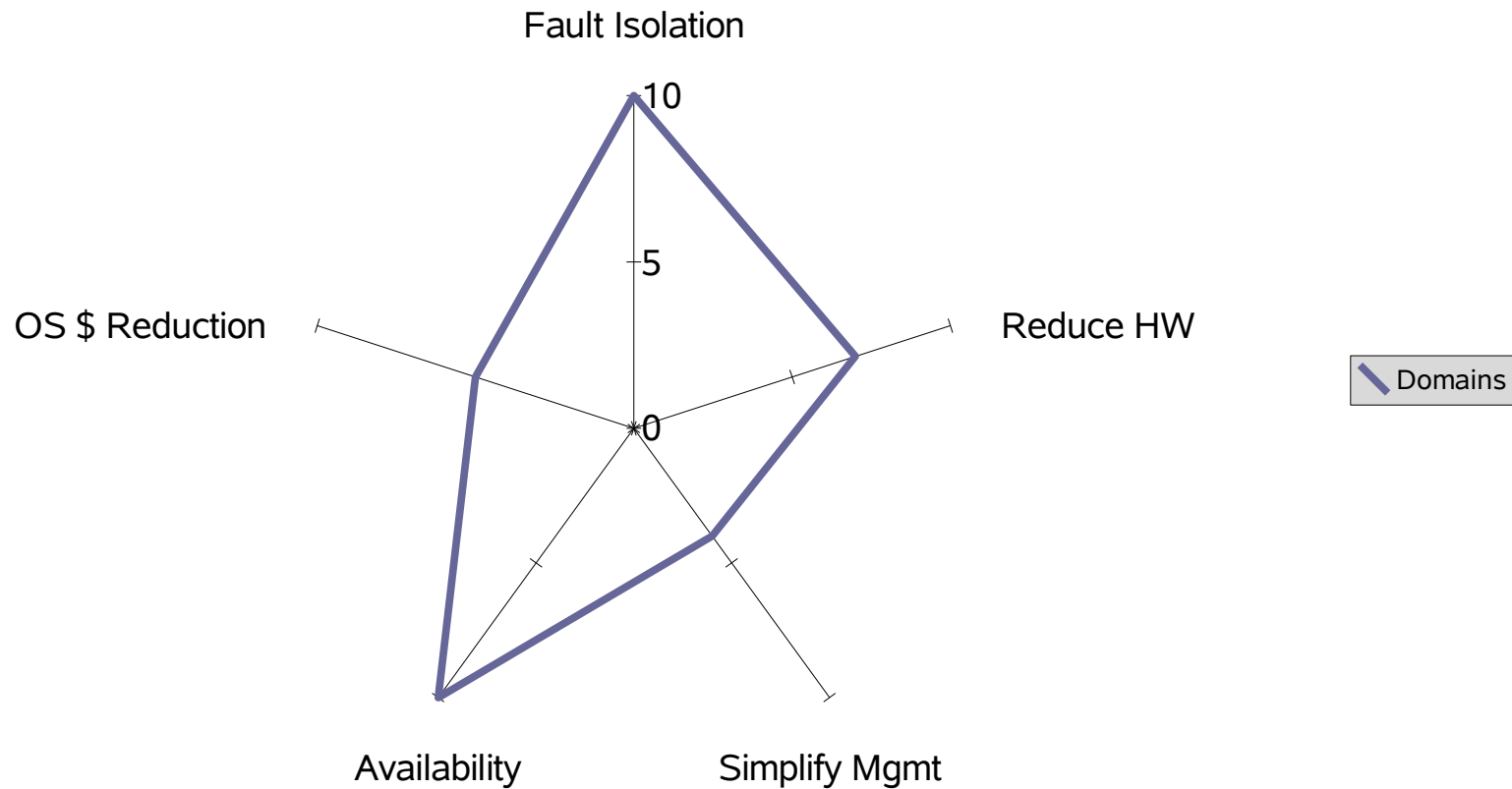


Dynamic System Domains

Achieving Consolidation Goals

- Reduce Hardware
 - > Two smaller systems can be combined into a larger, flexibly-sized domain, reducing hardware, OS, and mgmt costs
 - > A system with a first-shift load can be combined with a second-shift system to reduce total resources needed, including floor space and reduce hardware support costs
- Maintain Service Levels
 - > Service levels will increase through improved hardware redundancy
- Maintain Architectural Flexibility
 - > Flexibility will increase by using the dynamic features of DSD

Goal Achievement - Domains

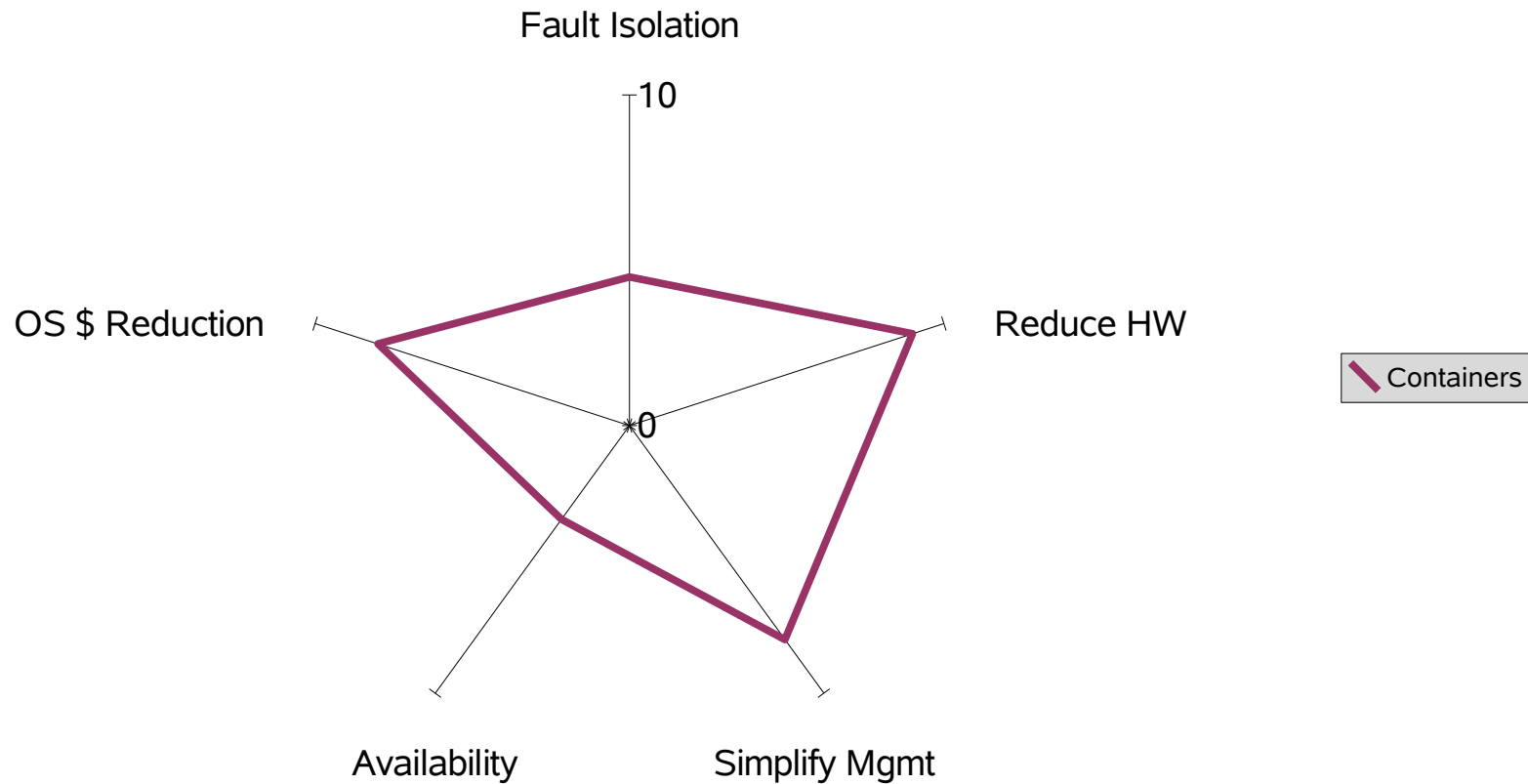


Solaris 10 Containers

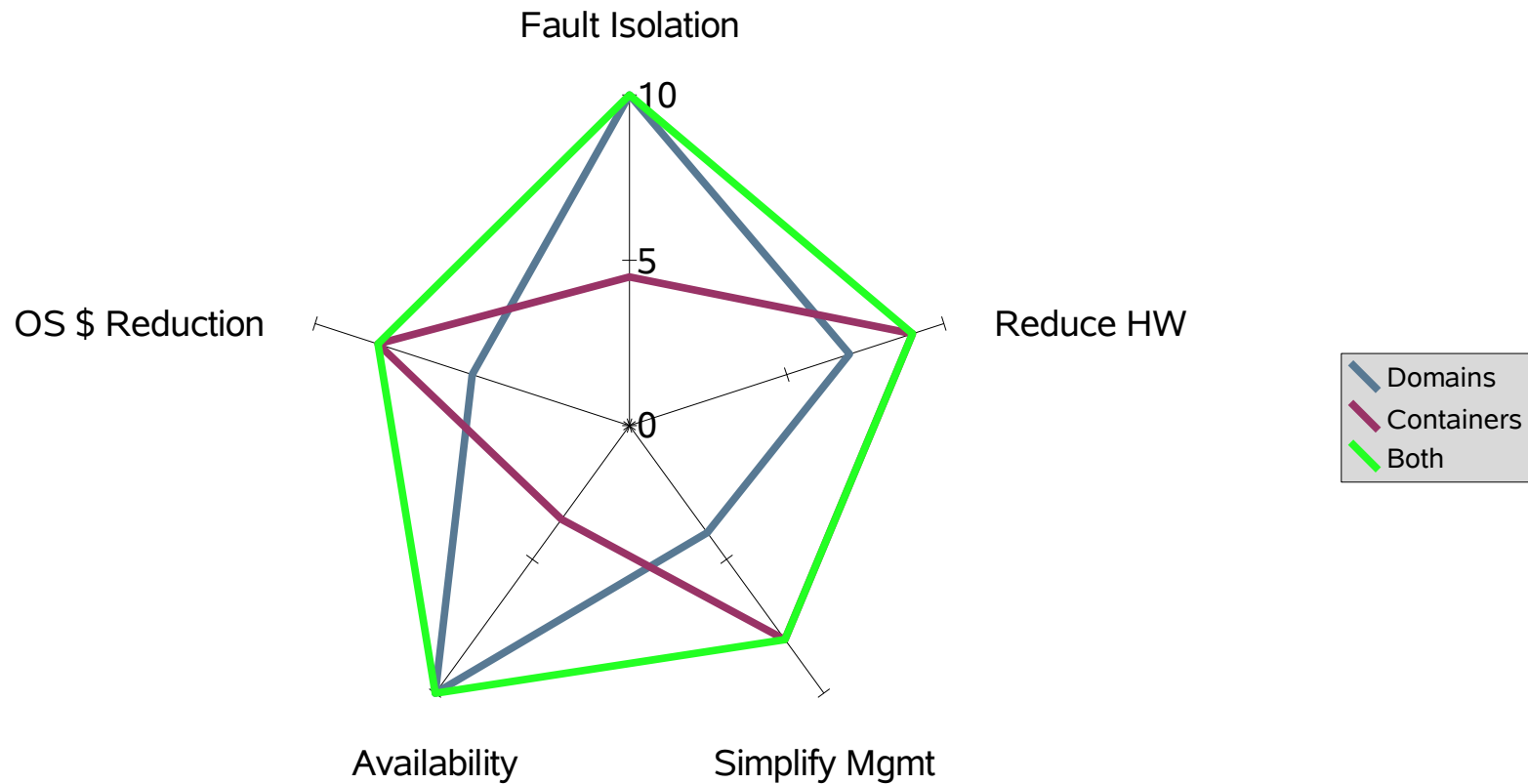
Achieving Consolidation Goals

- Reduce Hardware
 - > Many systems currently at low utilization levels can be combined
- Reduce Software Costs
 - > Minimize OS licensing and support costs
- Maintain Service Levels
 - > Service levels may be increased through data center simplification
 - > Overall service levels may be decreased if too many eggs in 1 basket
- Maintain Architectural Flexibility
 - > Architectural flexibility increased with creative use of containers
 - > Response times can be finely tuned through use of resource mgmt

Goal Achievement - Containers



Goal Achievement: DSD & Containers



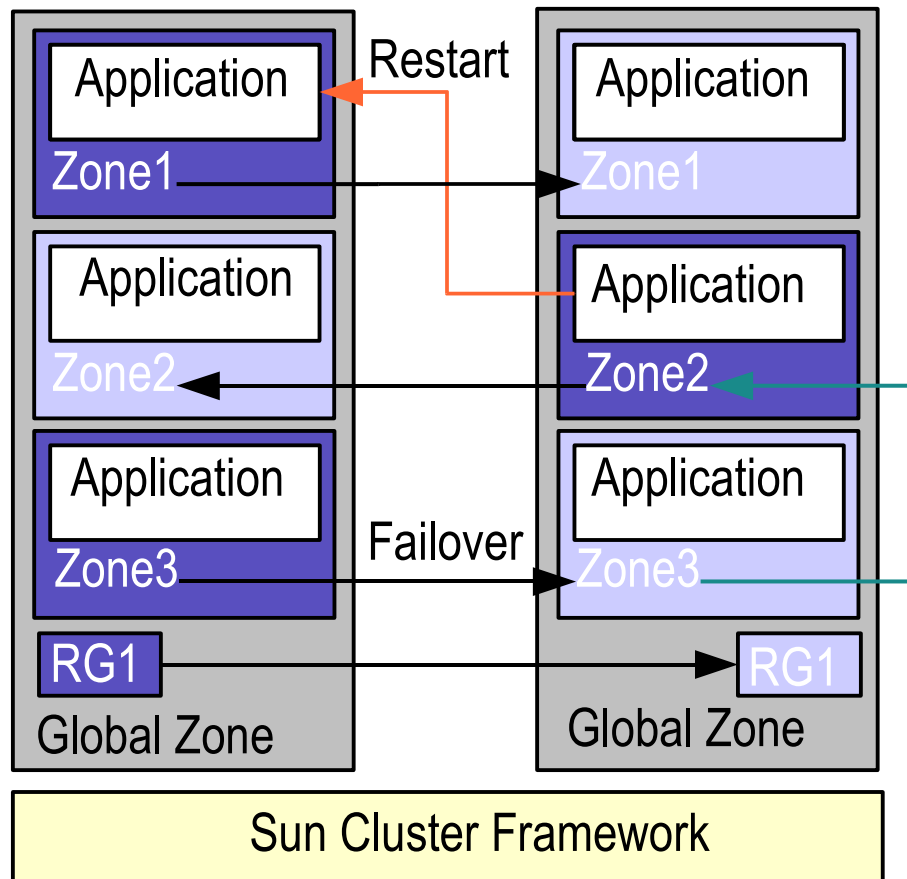
Solaris Containers

Clustering Containers

Sun Cluster 3.1 8/05

Sun Cluster and Solaris Containers

Multiple highly available Solaris Containers



- Failover zones
- Dependencies between Applications across zones across nodes
- ++/-- Affinities or Offload zones across nodes
- Integration with Service Management Facility

Sun Cluster and Solaris Containers

Architectural Details

- Sun Cluster installed only in the global zone
- Sun Cluster allows containers to coexist
- Sun Cluster container agent
 - > Manages the container
 - > Manages the applications within the container through:
 - > Command/Scripts to start/stop and probe
 - > SMF with optional application probing

Container Architectures

Sample Uses

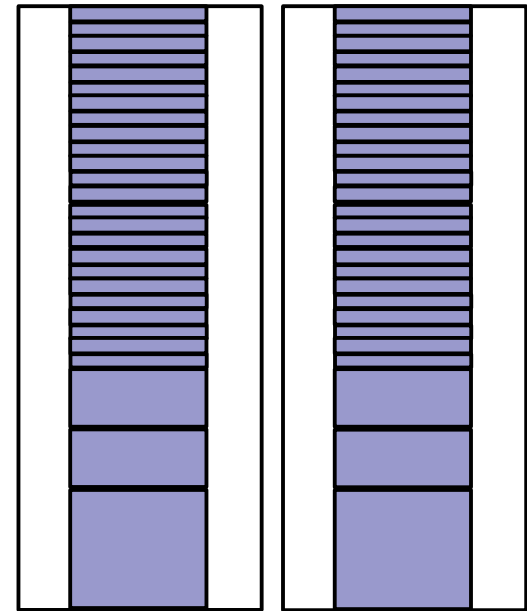
- Server consolidation
- Honeypot
- Multi-tier architecture, esp. combined with DTrace
- Group of applications that need a security boundary
- Group of applications that use a unique resource, e.g. port 80
- Utility Computing Model
- Test/POC environments: simulate a set of production systems
- Developer sandbox
- “No P.O.” POC, e.g. N-tier application
- Per-CPU-hour chargeback

Sample Use

Web Server Consolidation

Before

- 52 web servers, 2-4 Xeon each (700-3066MHz)
- 1RU-8RU each - two racks (~25,000 W)
- Mix of Solaris, Linux, Windows
- Apache
- 5-10% CPU utilization



After

- Target: 80% CPU utilization
- Two V40z, 4x Opteron (2.6GHz) each
- 2x 3RU = 6RU (1500 W total)
- Solaris 10, 52 zones
- Apache, same IP addresses, port numbers

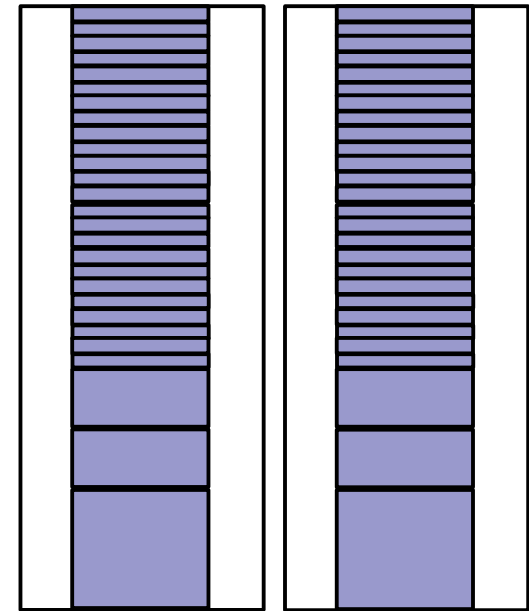


Sample Use

Web Server Consolidation

Before

- 52 web servers, 2-4 Xeon each (700-3066MHz)
- 1RU-8RU each - two racks (~25,000 W)
- Mix of Solaris, Linux, Windows
- Apache
- 5-10% CPU utilization

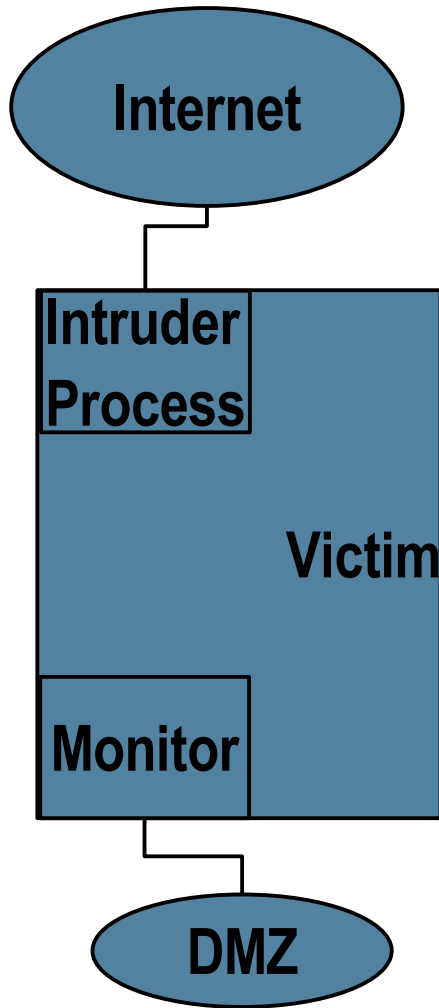


After

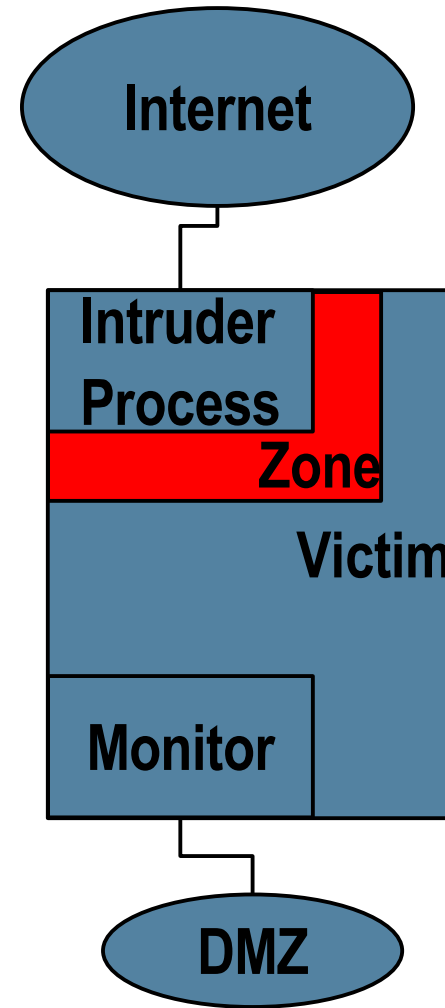
- Target: 70% CPU utilization
- Two T2000, 1x UltraSPARC T1 (1.2GHz) each
- 2x 2RU = 4RU (1100 W total)
- Solaris 10, 52 zones
- Apache, same IP addresses, port numbers



Honeypot

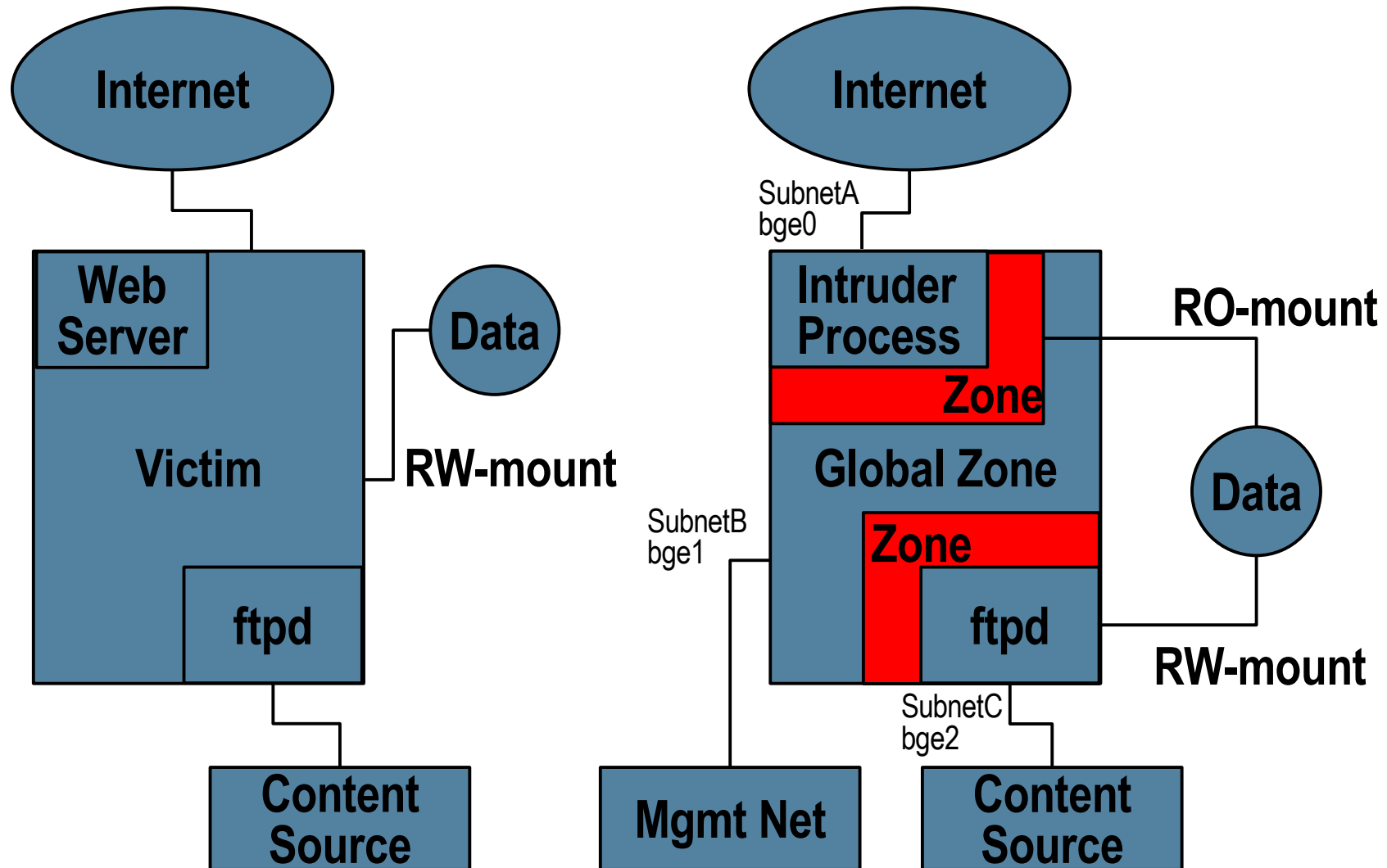


Intruder and Victim
can 'see' each other



Intruder cannot 'see' processes
outside of its zone

Secure Web Content



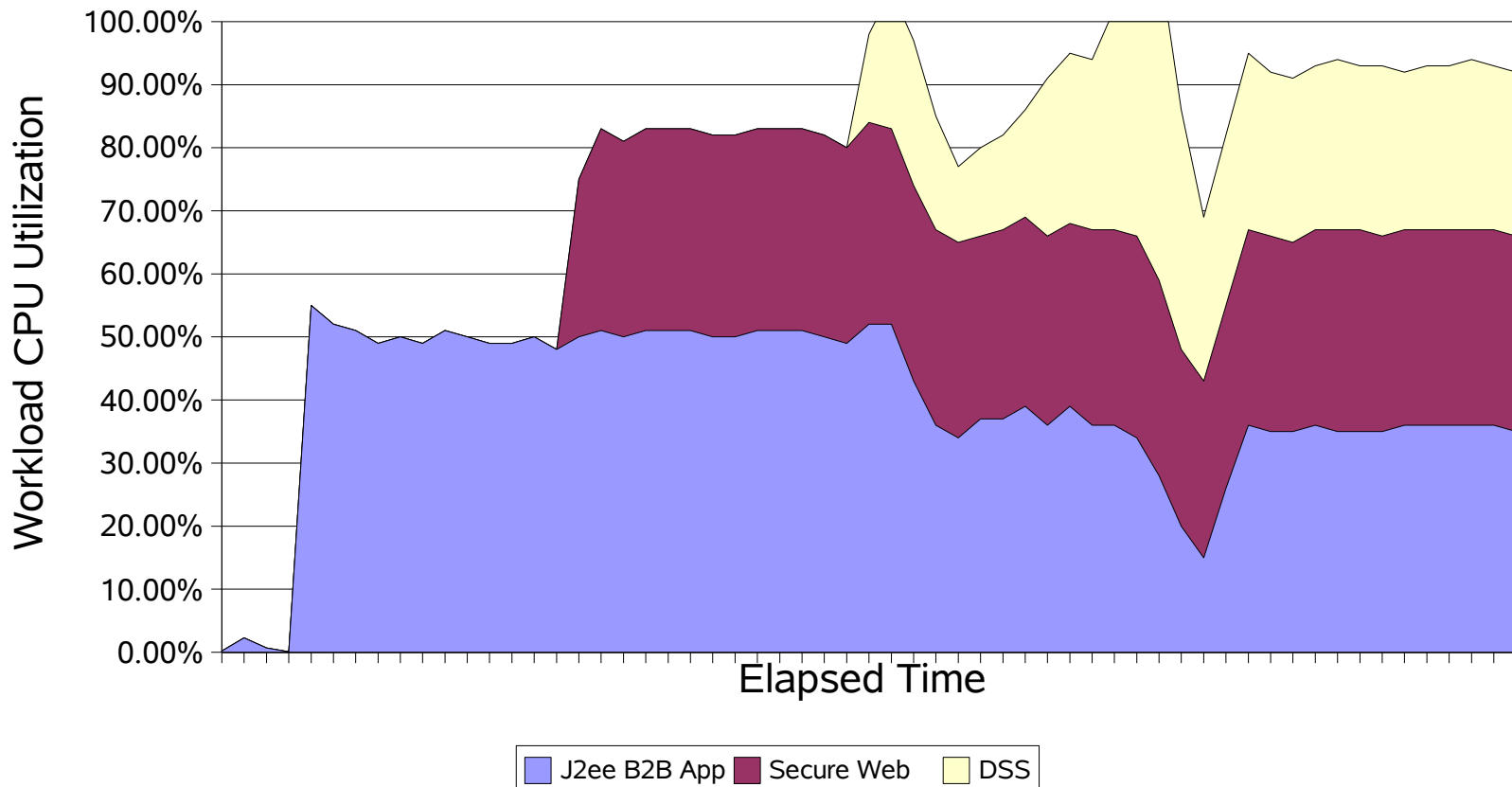
Server Consolidation on Sun Fire V890



- Consolidated 3 diverse, unrelated workloads using Solaris Containers:
 - > Multiple users access J2EE application server that emulate order/inventory and B2B supply chain environment
 - > Web serving workload where secure HTTP requests over a wide range of file sizes using SSL
 - > A RDBMS-based DSS batch workload using complex SQL queries on a large database
- Solaris Zones used to create private environments to isolate applications for each workload
- Resource Pools used to manage CPU consumption to ensure predictable application service levels even at HIGH system utilization
- Dynamically adjust CPU resources to meet changes in workload demand

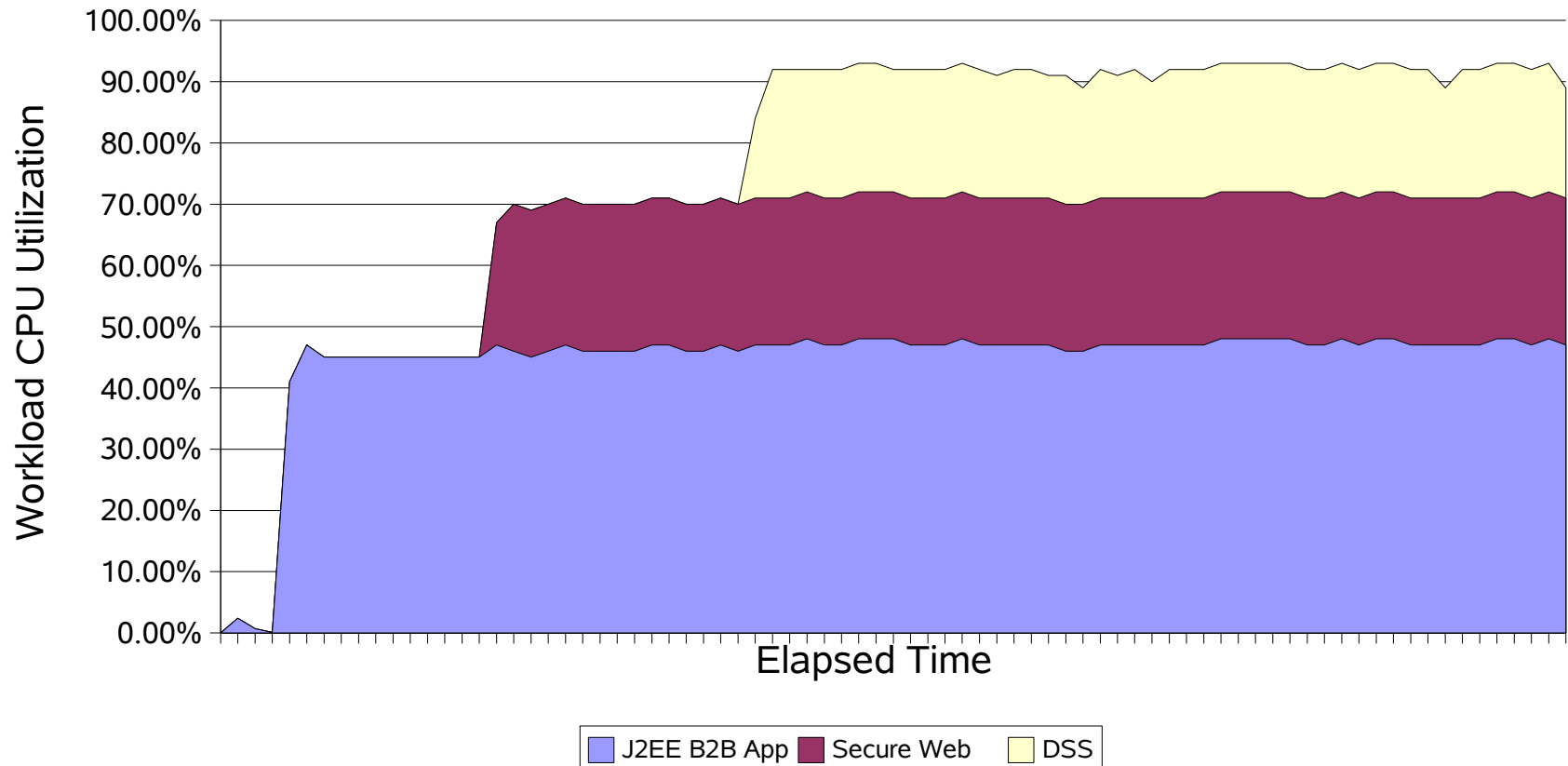
Mixed Workload Performance - Baseline

Mixed Workload CPU Utilization



Mixed Workload Performance Using Solaris 10 Containers

Mixed Workload CPU Utilization



Sun Fire V890 Server

Consolidation Preliminary Results

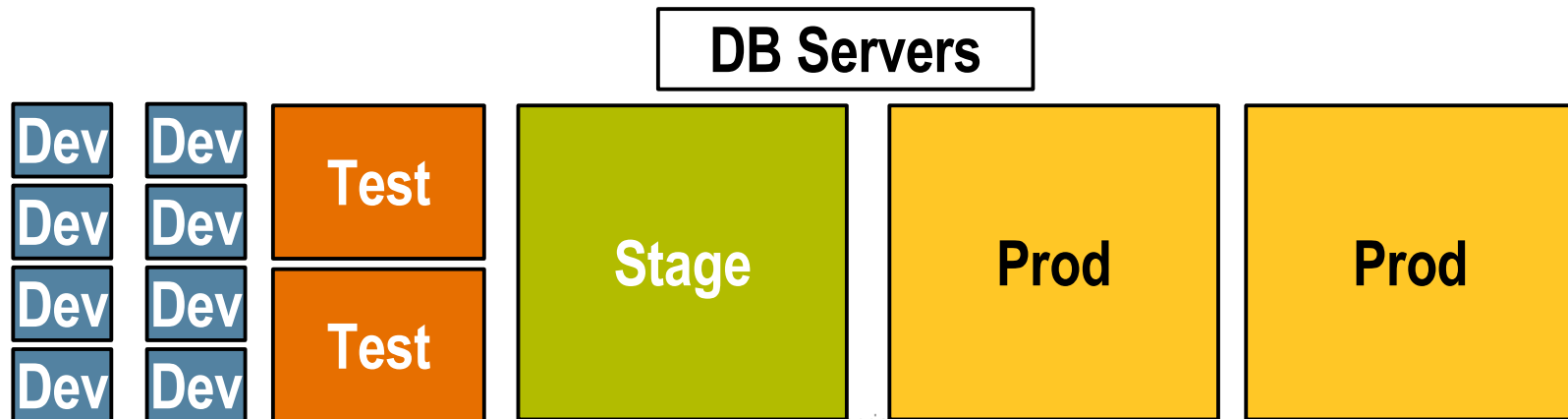
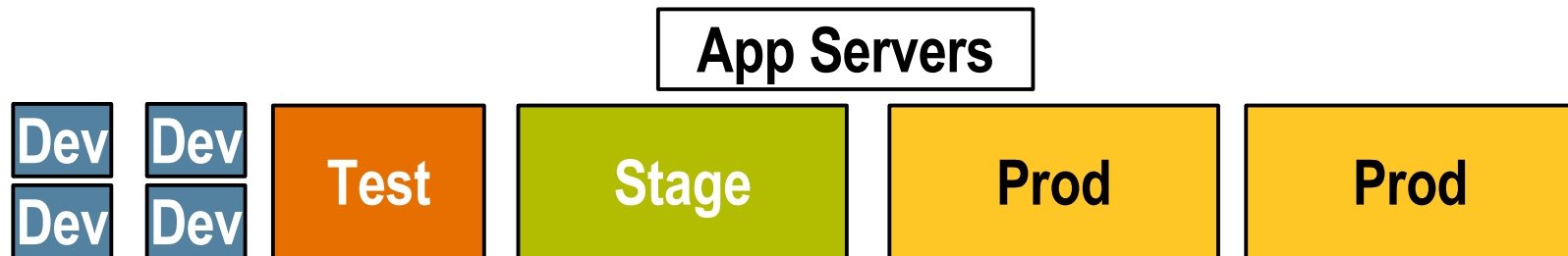
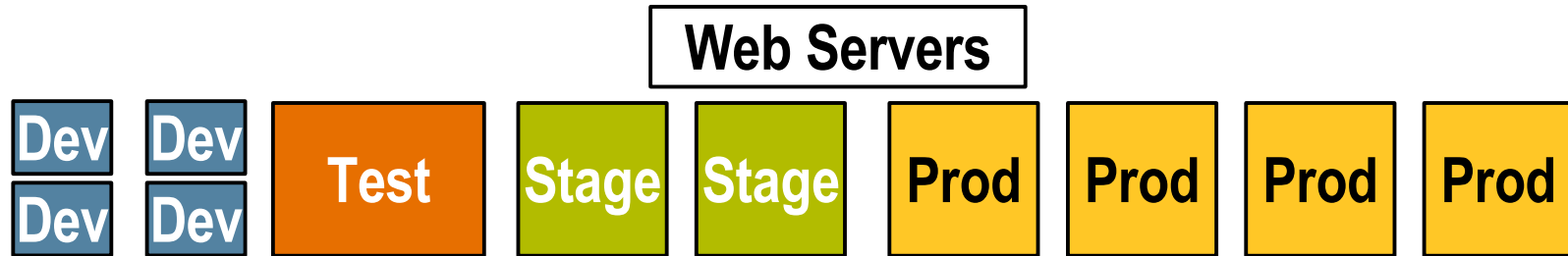
Baseline Performance

Workload	# users	CPU Used	Throughput (avg)	Service Level (avg)
J2EE Application	650	50% -> 36%	46 Txns/sec	5.6 sec/txn
Secure Web	700	31.00%	2311 Txns/sec	0.3 sec/Txn
Batch DSS	10	27.00%	83 MB/sec	6 min/query

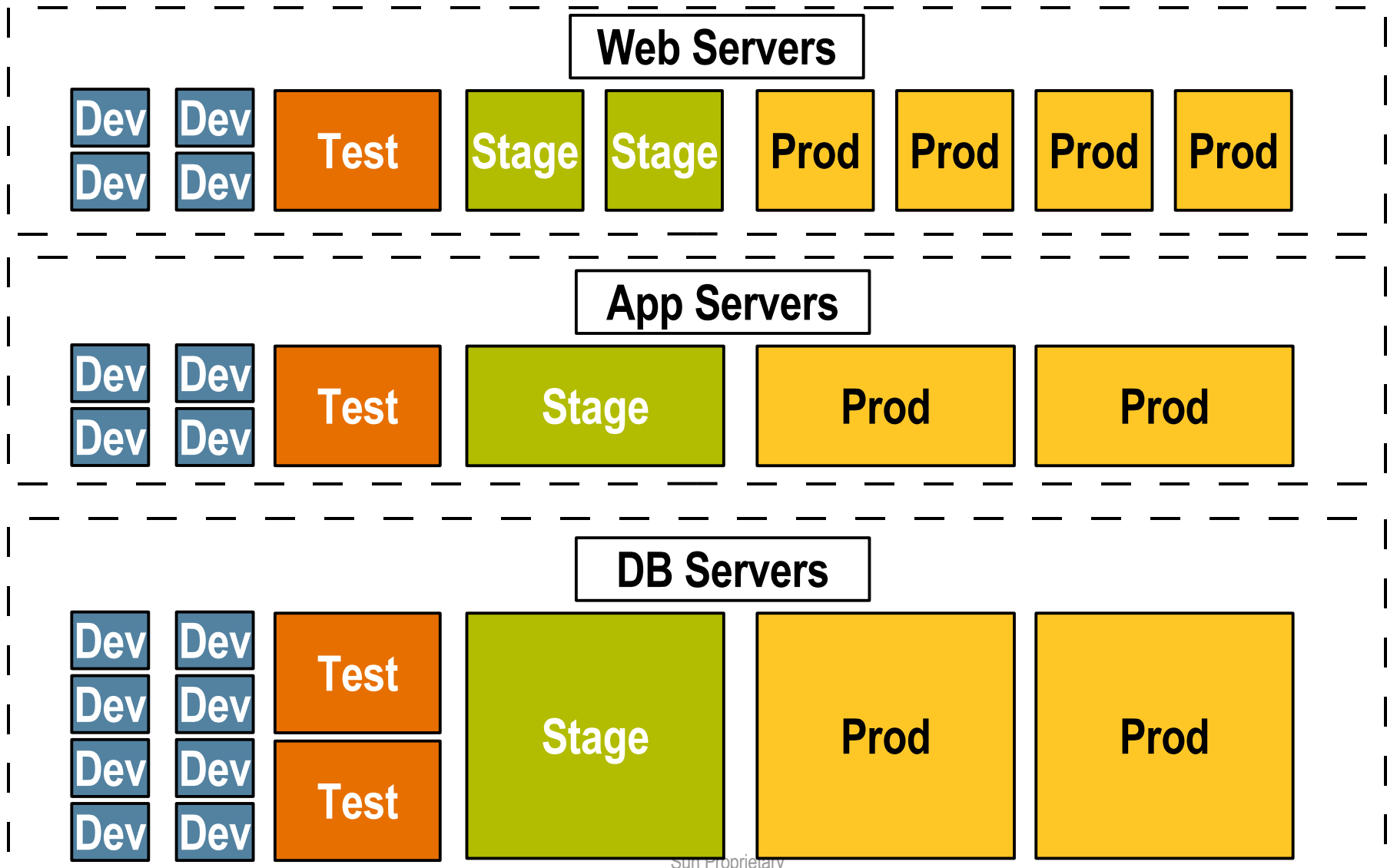
Workload Performance using Solaris Containers

Workload	#users	CPU Alloc	Throughput (avg)	Service Level (avg)
J2EE Application	650	50.00%	64 Txns/sec	0.98 sec/Txn
Secure Web	700	25.00%	2102 Txns/sec	0.33 sec /Txn
Batch DSS	10	25.00%	67 MB/sec	20 min/query

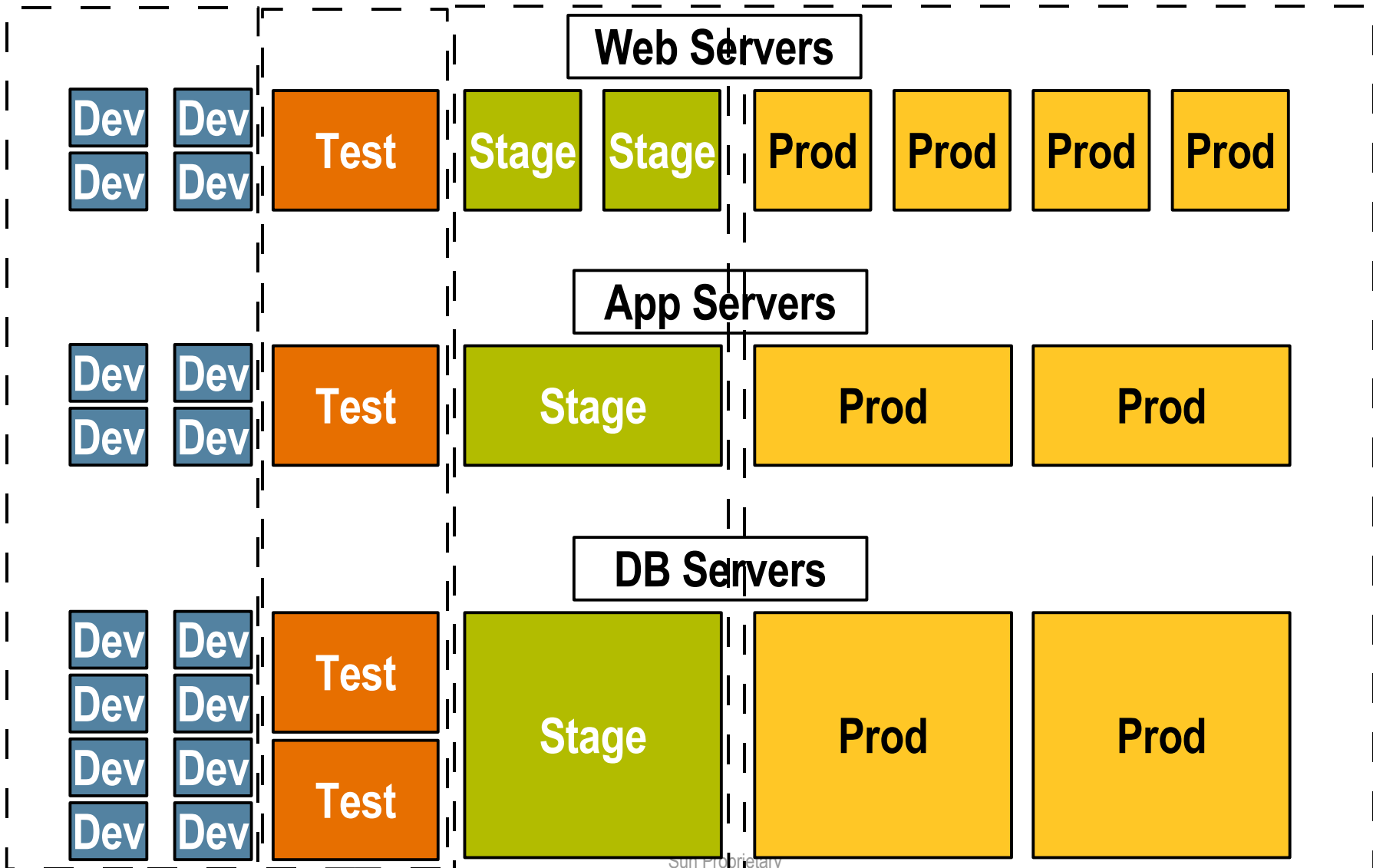
Example: 3-Tier Architectures



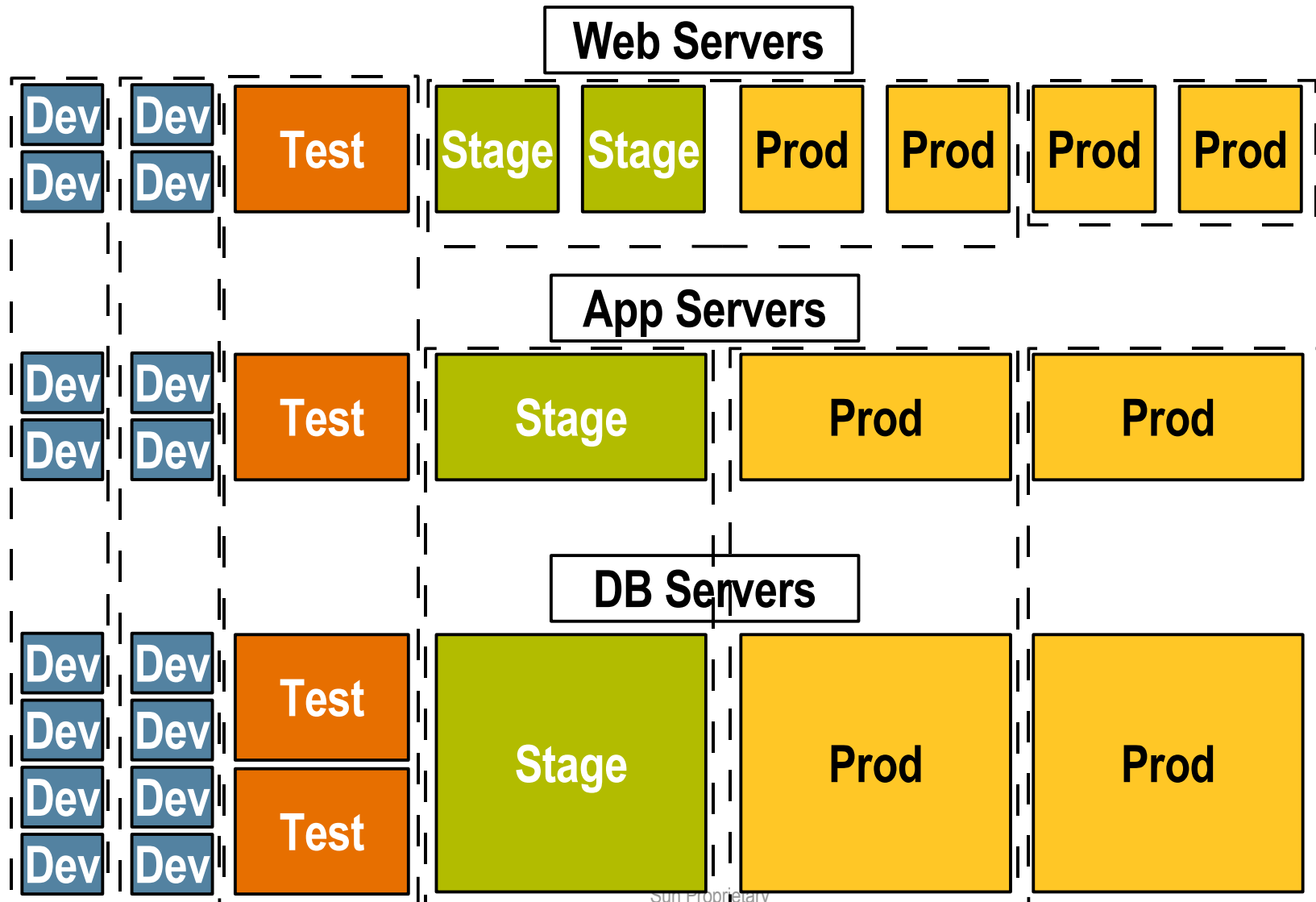
Consolidate by Tier?



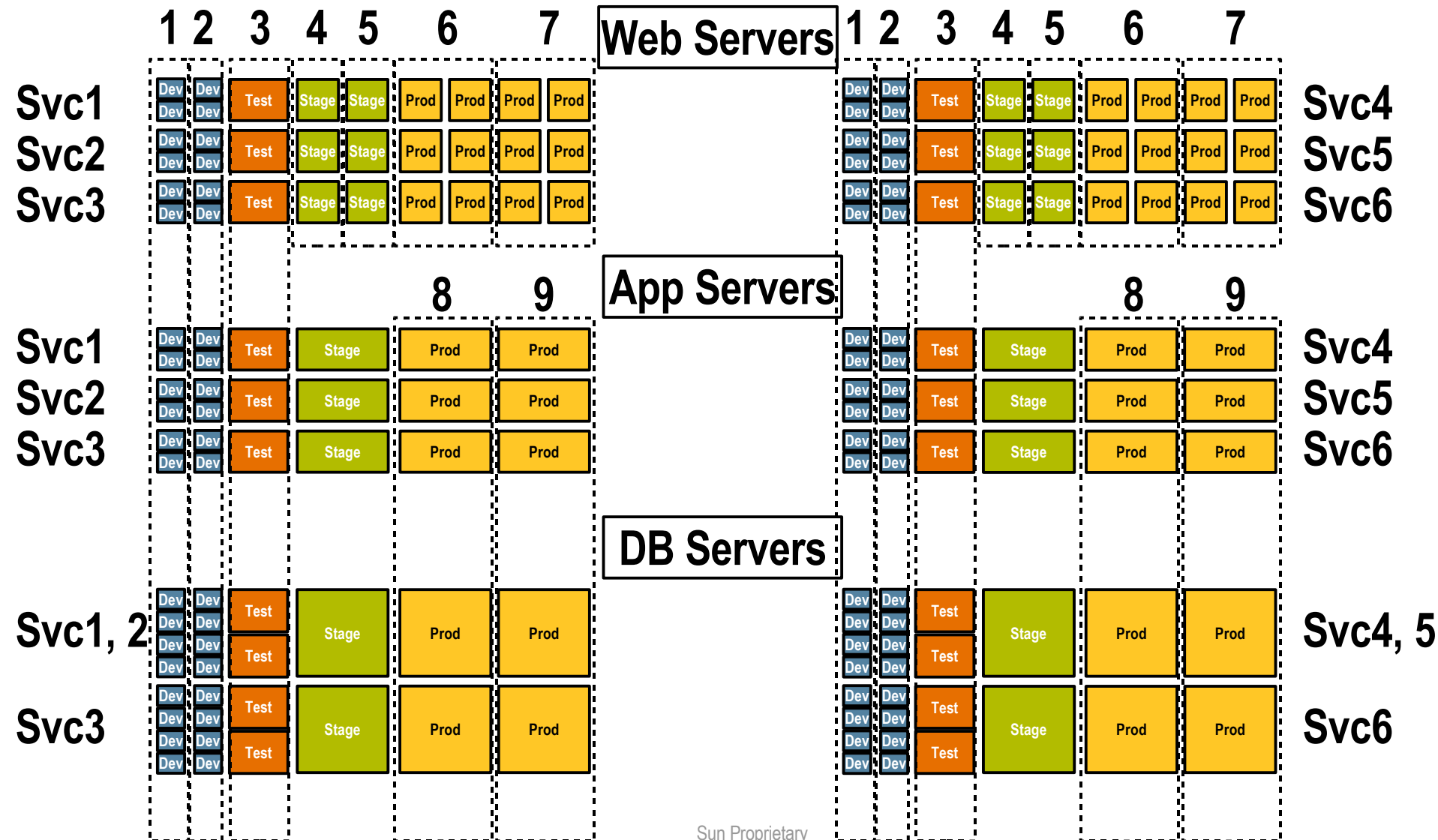
Consolidate by Lifecycle Phase?



Practical Consolidation



Example: 3-Tier Architectures



Sun Server Virtualization

Synergy and Best Practices

- Application availability requirements
- App dependency, independency, interdependency
- Patch dependencies
- Software installation flexibility
- Software lifecycle
- Horizontally vs. vertically scalable apps
- N-Tier architectures
- Resource usage patterns

Sun Server Virtualization

Synergy and Best Practices

- Security boundaries
- Organizational entities
- Government regulations
- Physical location
- Disaster Recovery
- Backups
- Naming Conventions

Sun Server Virtualization

Synergy and Best Practices

- “Manage Globally, Run Locally”
 - > Especially hardware!

ZFS Integrated with Containers

- File system can be exported as usual
 - > Global administrator controls properties
 - > Allow sharing of file system data between containers
- Datasets can be exported using `zonecfg` (1M)
 - > From within the container, file systems, snapshots and clones can be created
 - > Most properties can also be changed
 - > However, neither storage pools nor volumes can be created or modified

ZFS Dataset Example

- Creating a dataset with a 16GB quota and then exporting it into a non-global container

```
global# zpool create poolA c1t0d0 c1t0d1 c1t0d2 c1t0d3
global# zfs create poolA/web2
global# zfs set quota=16G poolA/web2
global# zonecfg -z web2
zonecfg:web2> add dataset
zonecfg:web2:dataset> set name=poolA/web2
zonecfg:web2:dataset> end
zonecfg:web2> commit
zonecfg:web2> exit
global# zoneadm -z web2 boot
```

```
web2# zfs create poolA/web2/opt
web2# zfs set mountpoint=/opt poolA/web2/opt
```

Resources

- FAQ: <http://www.opensolaris.org/os/community/zones/faq/>
- Bigadmin: <http://www.sun.com/bigadmin/content/zones/>
- Online Docs: <http://docs.sun.com/app/docs/doc/817-1592>
- Internal: <http://onestop.central.solaris/containers.shtml>
- Support: <http://www.sun.com/software/solaris/support.jsp>
- Whitepapers:
<http://www.sun.com/software/solaris/whitepapers.xml>
- Blogs: <http://blogs.sun.com>

Jeff Victor

OS Ambassador

jeff.victor@sun.com

Sun Microsystems, Inc.

Secure Web Content

