

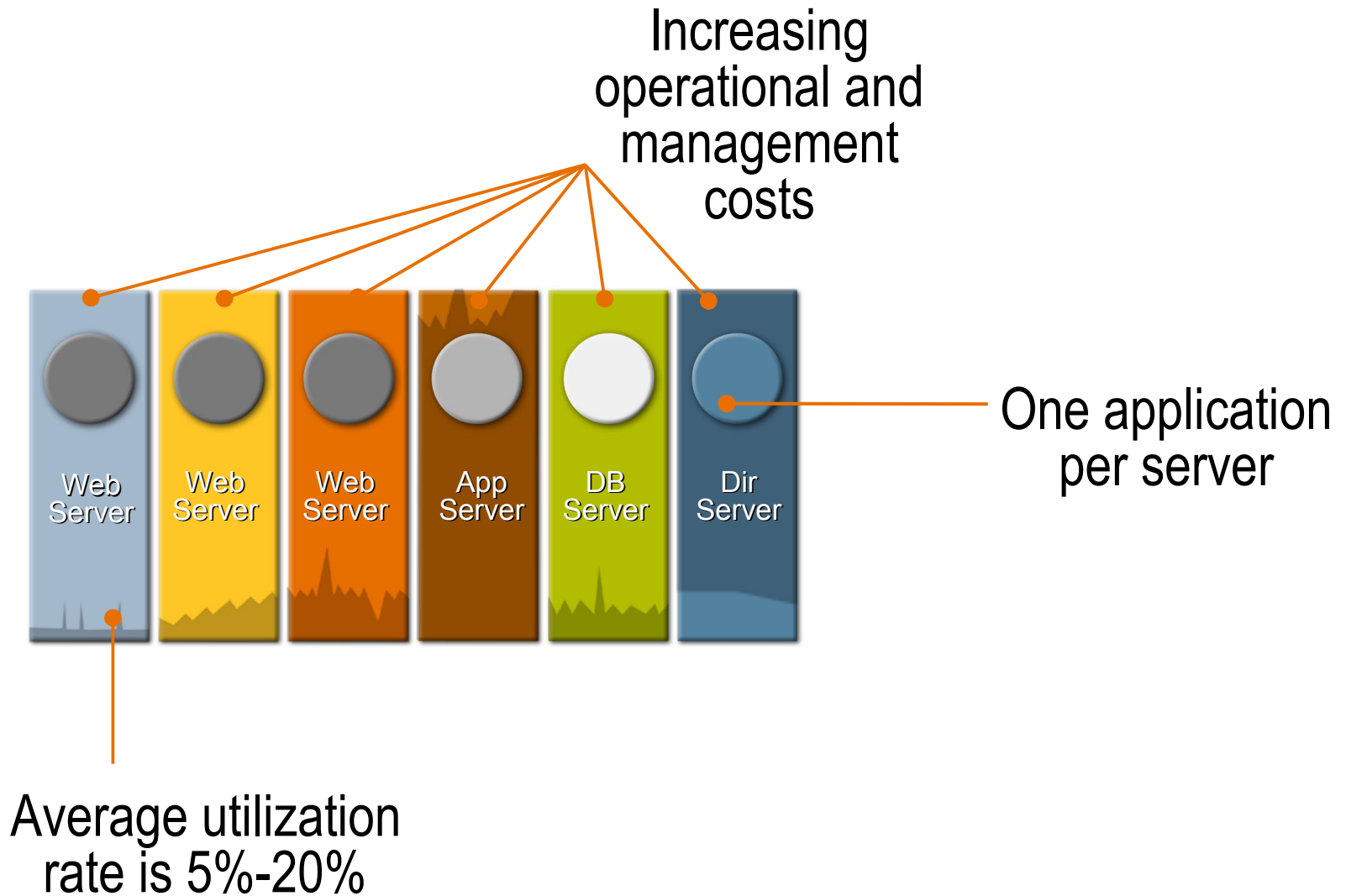


Solaris Containers

Peter Dennis

Sun Microsystems

Situation Today

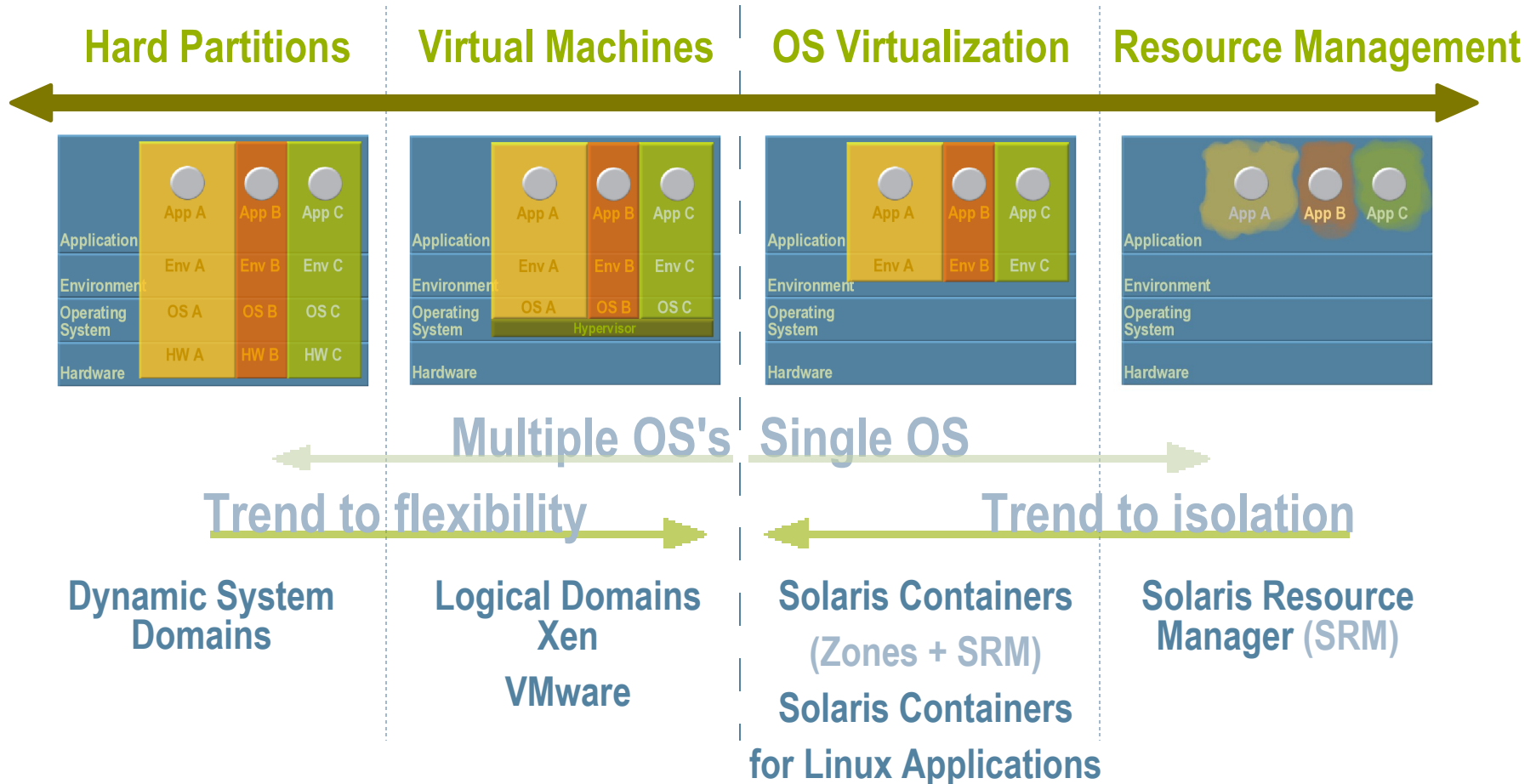


Server Consolidation Goals

- Save Money!
- Reduce costs by running multiple workloads on same system
 - > Better hardware utilization
 - > Reduced infrastructure overhead
 - > Lower administration costs (admins/workload)
- Requires support from system
 - > Resource controls
 - > Security isolation
 - > Failure containment
 - > Delegated administrative control
 - > Software package administration

Virtualization Solutions

- It's all about **Customer Choice**



Solaris Containers Overview

- Introduced in Solaris 10, further enhancements ongoing
- Basic concept: isolated execution environment within a Solaris instance
 - > Resource, security and fault isolation
 - > Lightweight, flexible, efficient
 - > One OS to manage

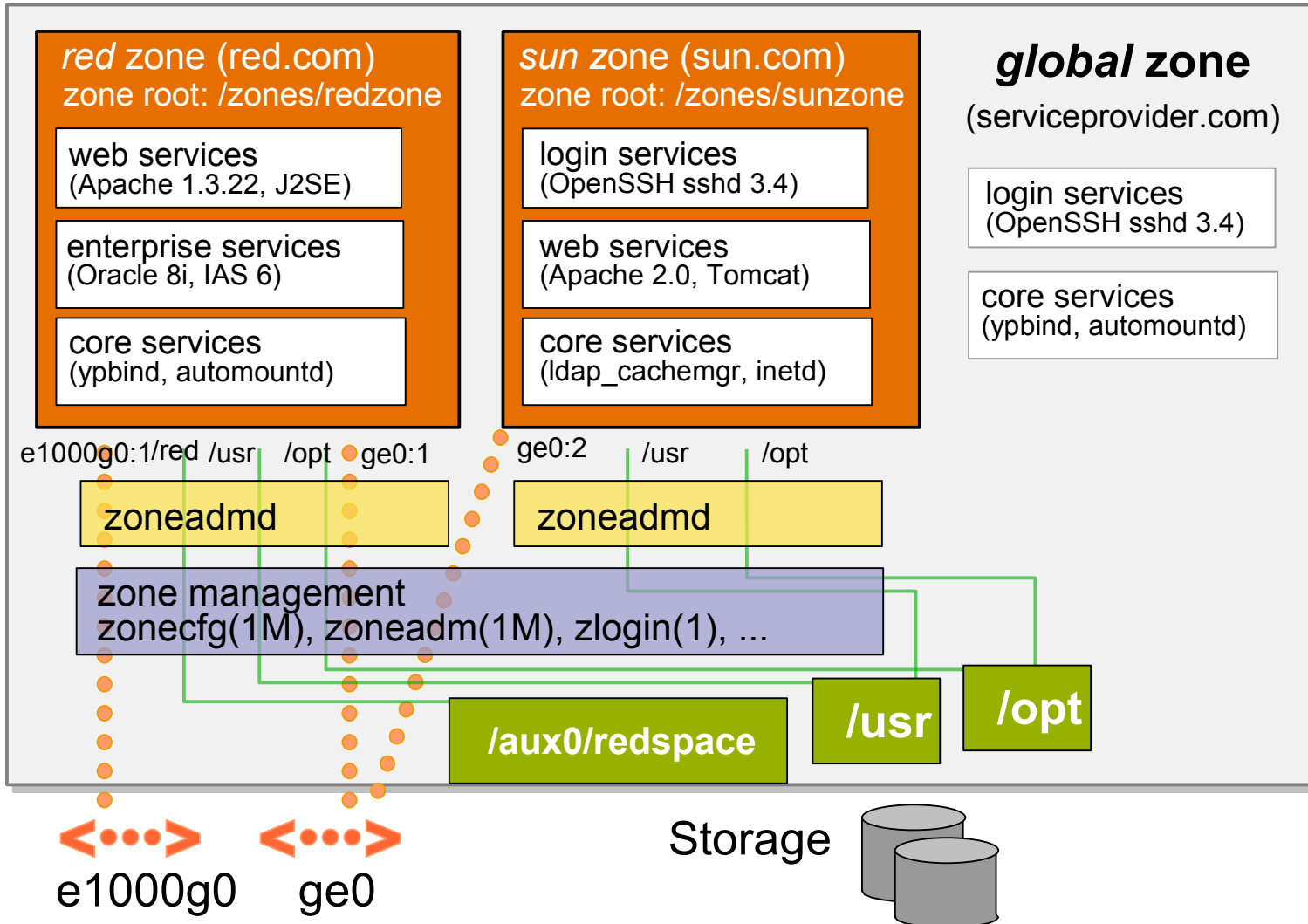
Example Uses

- Data center workload consolidation
- Software development
 - > Test vs. Production (allows the running of different application versions)
- Hostile or untrusted applications
- Hosting environments
- WAN-facing services
 - > Break-in containment

Solaris Zones

- Virtualizes OS layer: file system, devices, network, processes
- Provides:
 - > *Privacy*: can't see outside zone
 - > *Security*: can't affect activity outside zone
 - > *Failure isolation*: application failure in one zone doesn't affect others
- Lightweight, granular, efficient
- Delegated and simplified administration
- Complements resource management
- No porting for most apps; ABI/APIs are the same

Zones Block Diagram



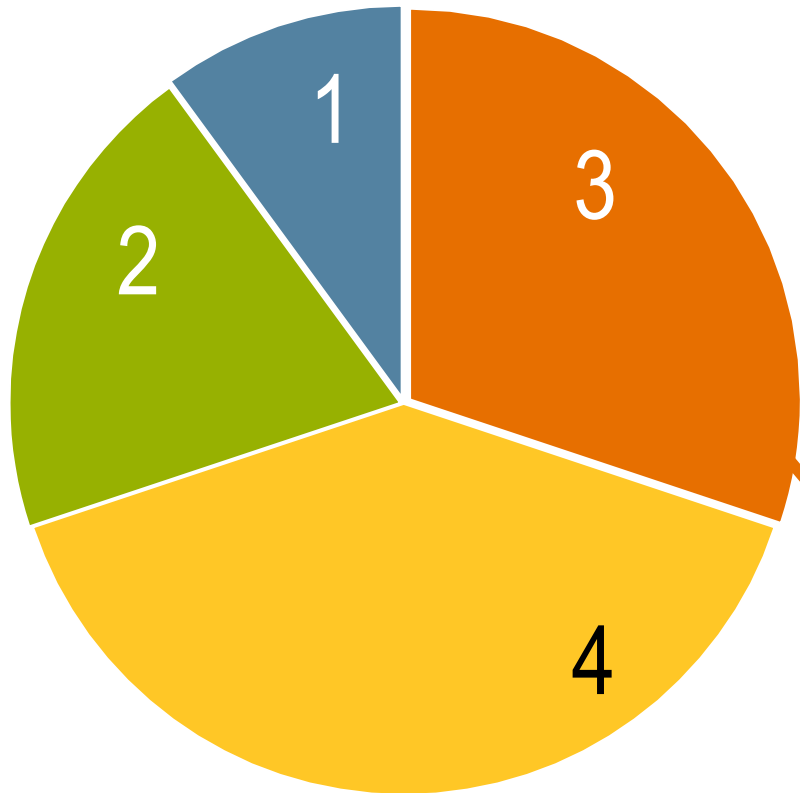
Security

- Each zone has a security boundary
- Runs with subset of `privileges` (5)
- Compromised zone unable to escalate its privileges
- Important name spaces are isolated (`/dev`, `/devices`)
- Processes running in a zone are unable to affect activity in other zones
- Zone-aware audit:
 - > Global zone administrator can specify whether auditing should be global or per-zone
 - > If per-zone, each zone administrator can configure and process their audit trails independently

Resource Management

- Combined with resource management, zones provide a more complete isolated environment
- Zones can be bound to a resource pool to isolate their effect on system resources
- Multiple zones can share a resource pool
- To meet service level guarantees, a single zone can be bound to a specific pool
- CPU time can be divided up with arbitrary granularity using the fair share scheduler (FSS(7))
- Resource limits can be set on a zone as well

Resource Management: Fair Share Scheduler



$$\frac{3}{(1 + 2 + 3 + 4)} = \frac{3}{10} = 30\%$$

Shares Allocated to Zones

Will my applications run on Zones?

- Application which runs as non-root user, runs on Solaris Zones without modification
- Application which accesses the network and files, and performs no other I/O, should work without any problems
- Application which requires direct access to other devices, e.g. disk partition, will usually work if the container is configured correctly. However, in some cases this may increase security risks.
- Application which requires direct access to `/dev/kmem` devices cannot run on non-global zones

Command line interfaces

- zonecfg – set up a zone configuration
- zoneadm – used to administer zones
- zlogin – enter a zone irrespective of any networking configured

zonecfg(1M)

- Operates in either interactive or non-interactive mode
- Subcommands
 - > add, cancel, commit, create, delete, end, export, help, info, remove, select, set, verify, revert, exit
- Top-Level Properties of a zone
 - > zonename, zonepath, autoboot, pool
- Resources for a zone
 - > fs, inherit-pkg-dir, net, device, rctl, attr, dataset

Configuring a zone example

- Creating a zone named `redzone`

```
global# zonecfg -z redzone
my-zone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:my-zone> create
zonecfg:my-zone> set zonepath=/zones/redzone
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=e1000g0
zonecfg:my-zone:net> set address=192.168.0.91
zonecfg:my-zone:net> end
zonecfg:my-zone> exit
```

Listing the zone configuration

- Use the info subcommand

```
global# zoncfg -z redzone info
zonename: redzone
zonepath: /zones/redzone
autoboot: false
pool:
limitpriv:
inherit-pkg-dir:
    dir: /lib
inherit-pkg-dir:
    dir: /platform
inherit-pkg-dir:
    dir: /sbin
inherit-pkg-dir:
    dir: /usr
net:
    address: 192.168.0.91
    physical: e1000g0
```

zones and smf(5)

- `svc:/system/zones:default`
- Each zone has a `autoboot` property whose value can be true or false
- The smf service must be online for the property to be honoured

```

global# svcs zones
STATE          STIME          FMRI
online         Feb_21         svc:/system/zones:default
global# zonecfg -z redzone set autoboot=true

```

Zone Types

- Sparse
 - > Inherit-pkg-dir
 - /lib, /platform, /sbin, /usr
 - > the inherit-pkg-dir are read only lofs(7D) mounts
 - > By default zones are sparse
- Whole-Root. All packages are copied. No lofs mounts. Must delete the inherit-pkg-dir entries during the configuration using zonecfg


```
remove inherit-pkg-dir dir=/lib
remove inherit-pkg-dir dir=/platform
remove inherit-pkg-dir dir=/sbin dir=/usr
```

 - > Increases diskspace usage and time to install the zone

zoneadm(1M) and zlogin(1M)

- zoneadm Subcommands
 - > help, boot, halt, ready, reboot, list, verify, install, uninstall, clone, move, detach, attach
- zlogin
 - > enter the zone (-C use the zone console)

Installing/booting a zone example

- Install the zone/boot and log into the zone

```
global# zoneadm -z redzone install
```

```
Preparing to install zone <redzone>.
```

```
Creating list of files to copy from the global  
zone.
```

```
Copying <6414> files to the zone.
```

```
....
```

```
global# zoneadm -z redzone boot
```

Completing the configuration

- Log into the console and complete the zone configuration process (sysidtool(1M))

```

global# zlogin -C redzone
[Connected to zone 'redzone' console]
[NOTICE: Zone booting up]
SunOS Release 5.10 Version Generic_118855-36
32-bit
Copyright 1983-2006 Sun Microsystems, Inc.
All rights reserved.
Use is subject to license terms.
Hostname: redzone
Loading smf(5) service descriptions: 116/116
...
What type of terminal are you using?
....

```

Alternatively use a /etc/sysidcfg file

- Put in the zone's <zonepath>/root/etc/ a sysidcfg file

```
system_locale=C
terminal=dtterm
network_interface=primary {
    hostname=redzone
}
security_policy=NONE
name_service=NIS {
    domain_name=red.com

name_server=serviceprovider.com(192.168.0.90)
}
timezone=US/Central
root_password=a&hj3f9
```

zonecfg(1M) sub commands

- Rename a zone (zone must be shutdown)

```
global# zonecfg -z redzone
zonecfg:redzone> set zonename=bluezone
zonecfg:bluezone> exit
```

- Move a zone, this changes the zonepath in the configuration

```
global# zoneadm -z bluezone move /ufs2-
zones/bluezone
Moving across file-systems; copying zonepath
/zones/redzone...
Cleaning up zonepath /zones/redzone...
```

zonecfg(1M) sub commands

- Copying zones

- > use the -t option to create to use a previously configured zone as a template

```
global# zonecfg -z redzone
redzone: No such zone configured
Use 'create' to begin configuring a new
zone.
```

```
zonecfg:s10u3-z3> create -t bluezone
zonecfg:s10u3-z3> set
zonepath=/zones/redzone
zonecfg:s10u3-z3> exit
```

- > clone the existing zone

```
global# zoneadm -z redzone clone bluezone
WARNING: network address '192.168.0.91' is
configured in both zones.
Cloning zonepath /ufs2-zones/bluezone...
```

zoneadm(1M) subcommands

- ability to move zones between machines
- Zone must either be configured on the system or use `zonecfg create -a`

```
global# zoneadm -z redzone list -p
-:redzone:installed:/zones/redzone
global# zoneadm -z redzone detach
global# zoneadm -z redzone list -p
-:redzone:configured:/zones/redzone
```
- Make the `/zone` path available on another node (`zfs export/zfs import` for example)
- Now on another node (sharing the same storage)

```
global# zoneadm -z redzone attach
global# zoneadm -z redzone list -p
-:redzone:installed:/zones/redzone
```

OpenSolaris: zones and zfs(1M)

- zfs filesystems created when installing on a zpool

```
global# zoneadm -z snv58-z2 install
```

```
A ZFS file system has been created for this  
zone.
```

```
Preparing to install zone <snv58-z2>.
```

OpenSolaris: zones and zfs(1M)

- cloning uses zfs snapshots

```

global# zonecfg -z snv58-z3
snv58-z3: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:snv58-z3> create -t snv58-z2
zonecfg:snv58-z3> set zonepath=/zfs-
zones/snv58-z3
zonecfg:snv58-z3> exit
global# zoneadm -z snv58-z3 clone snv58-z2
Cloning snapshot zfs-zones/snv58-z2@SUNWzone1
Instead of copying, a ZFS clone has been
created for this zone.

```

zones and dtrace(1M)

- Solaris 10: dtrace is not allowed to be used in the non-global zone
- OpenSolaris: `dtrace_user` and `dtrace_proc` are allowed privileges in non-global zones
`set limitpriv=default,dtrace_user,dtrace_proc`
- `dtrace_kernel` is not allowed in non-global zones because it would break the security model (ie observing other zones)
- Dtrace has a variable `zonename` which can be used within the predicates

The Global Zone

- Can see all processes: `ps -eZ`

```

global      527  ?           0:00  sshd
global      830  pts/1      0:00  ksh
redzone     6379 ?         0:00  kcfcd
redzone     6622 ?         0:00  utmpd
redzone     6710 ?         0:00  syslogd
redzone     6603 ?         0:00  nscd
global      660  ?         0:00  dtlogin
global      817  ?         0:28  java

```

- `ifconfig` shows zone information:

```

e1000g0:1:
flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,I
Pv4> mtu 1500 index 2
        zone redzone
        inet 192.168.0.91 netmask ffffffff00
broadcast 192.168.0.255

```

Resource Management

- Zone & Resource Management == Containers
- Resources configured via `zonecfg(1M)` using the controls: `pool`, `rctl`
- Pools – a 'container' for a resource management configuration – see `poolcfg(1M)`, `pooladm(1M)`
- Rctl – zone specific resource controls:
 - > Solaris 10: `cpu-shares`, `max-lwps`
 - > OpenSolaris: `max-swap`, `max-locked-memory`, `max-shm-memory`, `max-shm-ids`, `max-sem-ids`, `max-msg-ids`
(see `resource_controls(5)` for definitions)

Setting the Rctl

- Limiting the amount of cpu time (Solaris 10):

```
zonecfg:redzone> add rctl  
zonecfg:redzone:rctl> set name=zone.cpu-shares  
zonecfg:redzone:rctl> set  
value=(priv=privileged,limit=20,action=deny)  
zonecfg:redzone:rctl> end
```

- The global zones scheduler should be set to use FSS (`dispadm -d FSS`)
- OpenSolaris is simply `set cpu-shares=20`

pools: partition system resources

- Pool: persistent processor set configuration and (optional) scheduling class
- See `pooladm(1M)`, `poolcfg(1M)`, `poolbind(1M)`, `poolstat(1M)` for details.

- **Example:**

```
poolcfg -c 'create pset rz-pset (uint pset.min  
= 1; uint pset.max = 2)'  
poolcfg -c 'create pool rz-pool'  
poolcfg -c 'associate pool rz-pool (pset rz-  
pset)'  
zonecfg -z redzone set pool=rz-pool
```

- When the zone boots it will be given the required number of CPUs

OpenSolaris And Resources

- Realised that the setting of resource controls for zones was too hard
- If `cpu-shares` is configured then the zone will use FSS
 - > Set via `set cpu-shares=20` in `zonecfg`
- New resources added to the `zonecfg` options:
 - `dedicated-cpus` – a pool will be configured at zone boot for the cpus (processor set)
 - `capped-memory` – the capacity limits on memory for the zone `rcapd(1M)`

Branded Zones

- Zones that contain non-native operating environments
- Current brand is the lx(5) (see brands(5) as well)
 - > supports Centos 3.x and RedHat Enterprise Linux 3.x
 - > Linux 2.4.21 kernel & glibc 2.3.2
- Configure the branded zone using the -t option to create:

```
zonecfg:lx-zone> create -t SUNWlx
```

More information

- Active discussion in the Zones community
 - > zones-discuss@opensolaris.org
- Your participation encouraged
- Community Page
 - > <http://www.opensolaris.org/os/community/zones/>
 - > <http://www.opensolaris.org/os/community/brandz/>
 - > <http://www.opensolaris.org/os/project/rm/>

–

Acknowledgements

- Jerry Jelinek
 - > initial slide set



Solaris Containers

Peter Dennis

peter.dennis@sun.com