Zones ("N1 Grid Containers") in Solaris 10

Harry J. Foxwell, Ph.D.
Senior System Engineer
Sun Microsystems

Related Technologies

- Sun Server Domains
- IBM mainframe LPAR
- IBM AIX WorkLoad Manager
- HP vPar (virtual partition)
- HP PRM (Process Resource Manager)
- VMWare
- Linux
  - http://user-mode-linux.sourceforge.net/
  - http://sourceforge.net/projects/xen

Resources

- www.sun.com/solaris/10
- http://www.sun.com/bigadmin/content/zones/
- http://www.blastwave.org/docs/Solaris-10-b51/DMC-0002/dmc-0002.html

Zones can be used for Server Consolidation

- Run multiple applications securely and in isolation on the same system
- Utilize the hardware resources more effectively
- Allow delegated administration of the application environment
- Streamline the effort in maintaining the system

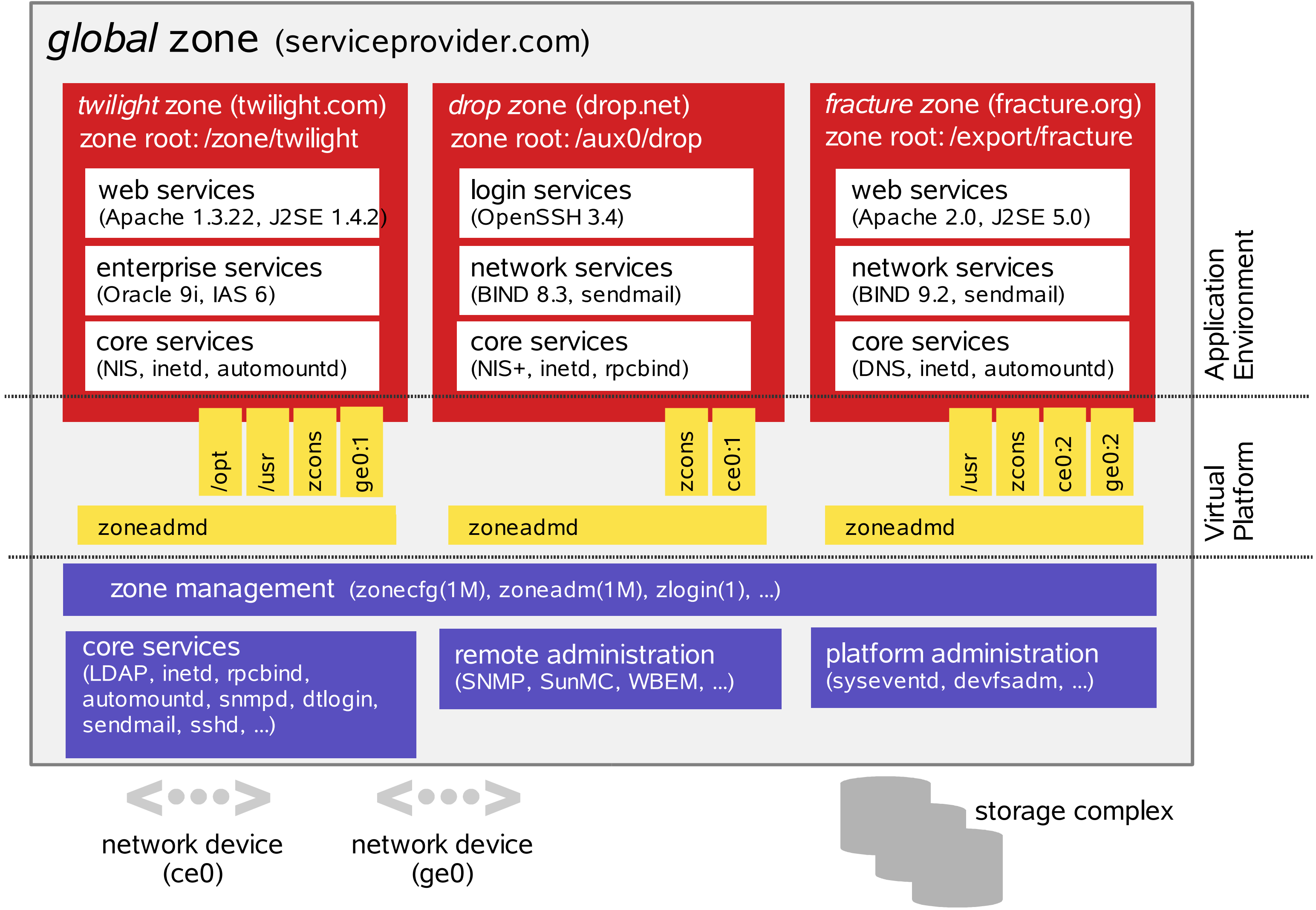
Zones Summary

- Isolated application environments within a single Solaris instance
- Resource, name space, security and failure isolation
- Efficient and granular using a lightweight OS layer
- Delegated, simplified administration
- No porting as ABI/APIs are the same

Typical Uses for Zones

- Consolidating data center workloads such as multiple databases
- Hosting untrusted or hostile applications or those that require global resources like IP port space
- Hosting “complete” environments
- Deploying Internet facing services
- Software development

Zones Block Diagram



global zone (serviceprovider.com)

twilight zone (twilight.com)
zone root: /zone/twilight

web services
(Apache 1.3.22, J2SE 1.4.2)

enterprise services
(Oracle 9i, IAS 6)

core services
(NIS, inetd, automountd)

/opt
/usr
zcons
ge0:1

zoneadmd

drop zone (drop.net)
zone root: /aux0/drop

login services
(OpenSSH 3.4)

network services
(BIND 8.3, sendmail)

core services
(NIS+, inetd, rpcbind)

zcons
ce0:1

zoneadmd

fracture zone (fracture.org)
zone root: /export/fracture

web services
(Apache 2.0, J2SE 5.0)

network services
(BIND 9.2, sendmail)

core services
(DNS, inetd, automountd)

/usr
zcons
ce0:2
ge0:2

zoneadmd

Application Environment

Virtual Platform

zone management (zonecfg(1M), zoneadm(1M), zlogin(1), ...)

core services
(LDAP, inetd, rpcbind, automountd, snmpd, dtlogin, sendmail, sshd, ...)

remote administration
(SNMP, SunMC, WBEM, ...)

platform administration
(syseventd, devfsadm, ...)

network device
(ce0)

network device
(ge0)

storage complex

Security

- Each zone has a security boundary around it
- Runs with subset of `privileges(5)`
- A compromised zone is unable to escalate its privileges
- Important name spaces are isolated
- Processes running in a zone are unable to affect activity in other zones

Processes

- Certain system calls are not permitted or have restricted scope inside a zone
- From the global zone, all processes can be seen but control is privileged
- From within a zone, only processes in the same zone can be seen or affected
- `proc(4)` has been virtualized to only show processes in the same zone

File Systems

- Each zone is allocated its own root file system and cannot see that of others
- Unlike with `chroot(2)`, processes cannot escape out of a zone
- File systems like `/usr` can be inherited in a read-only manner
- File systems such as `autofs(4)` and NFS have been virtualized per zone

Networking

- Single TCP/IP stack for the system so zones are shielded from configuration details for devices, routing and IPMP
- Each zone can be assigned IPv4/IPv6 addresses and has its own port space
- Applications can bind to INADDR_ANY and will only get traffic for that zone
- Zones cannot see the traffic of others

Identity

- Each zone controls its node name, RPC domain name, time zone, locale and naming service like LDAP and NIS
  - `sysidtool(1M)` can set this up
- Separate `/etc/passwd` files means that root can be delegated to the zone
- User ids may map to different names when domains differ (as with NFS now)

Interprocess Communication

- Expected IPC mechanisms such as System V IPC, STREAMS, sockets, `libdoor(3LIB)` and loopback transports are available inside a zone
- Key name spaces virtualized per zone
- Inter-zone communication is available using the network (software loopback)
- Global zone can setup rendezvous too

Devices

- Zones see an subset of “safe” pseudo devices in their `/dev` directory
  - Devices like `/dev/random` are safe but others like `/dev/ip` are not
- Zones can modify the permissions of their devices but cannot `mknod(2)`
- Physical device files like those for raw disks can be put in a zone with caution

Resource Management

- Zones do not require dedicated hardware resources
- CPUs can be partitioned with an arbitrary granularity using `FSS(7)`
- Multiple zones can be multiplexed over a resource pool or a zone can be bound to a pool for service guarantees
- Resource limits can be set on a zone

Configuration/Administration

- `zonecfg(1M)` is used to specify resources (such as IP interfaces) and properties (such as a resource pool)
- `zoneadm(1M)` is used to perform administrative steps for a zone such as list, install, (re)boot, halt, et cetera
- Installation creates a root file system with factory-default editable files

Additional Features

- Support for read-only `lofs(7FS)`
- “nodevices” `mount(2)` option
  - All NFS file systems in a zone are mounted as such
- Configuration stored in a private XML file
- Zone ids are dynamically assigned at zone boot
- `ptree(1)` can displays a zone's process tree
- `traceroute(1M)` supported inside a zone

- Updates to `zonecfg(1M)`
  - Grammar changes with support for complex property values
  - `inherit-pkg-dir` resource specifies a global zone file system to export read-only into a zone
  - `rctl` resource specifies a zone resource control
  - `attr` resource specifies a generic attribute
  - `autoboot` property specifies action at global boot
  - `pool` property specifies name of pool to bind to

-
  - NFSv4 client support
  - `nfsstat(1M)` virtualized per-zone
  - Additional updates to `zonecfg(1M)`
    - Disk-based file systems can (again) be configured
    - Command line editing and history
  - `ps(1)` can display processes from a list of zones or add a `ZONE` column to other reports
  - Support for `-p` option to `prtconf(1M)`

- CPU visibility improvements
  - Only take effect when resource pools are enabled
  - Traditional commands and APIs that deal with processors will provide a “virtualized” view based on the pool (processor set) the zone is bound to
    - Including `iostat(1M)`, `mpstat(1M)`, `prstat(1M)`, `psrinfo(1M)`, `sar(1)` and `vmstat(1M)`
    - Including `sysconf(3C)` (when detecting number of processors configured/online) and `getloadavg(3C)`
    - Including numerous `kstat(3KSTAT)` values from the `cpu`, `cpu_info` and `cpu_stat` publishers

- `zones.max-lwps` zone resource control
  - This resource control can be further subdivided within the zone itself using `project.max-lwps`
- Zone-aware auditing
  - 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

-
  - Support for `-l` and `-s` options to `swap(1M)`
  - Zones can be booted in single-user mode
  - Support for `sysdef(1M)` from within a zone
  - Zones where no `inherit-pkg-dir` resources have been defined are supported

Discussion

- How/Why would you use server virtualization technologies?
- Advantages?
- Disadvantages?

Zones (N1 Grid Containers) Engineering Update

Harry.Foxwell@Sun.COM