

open



USE



IMPROVE



EVANGELIZE

Virtualización Zonas/Contenedores

Roger Jordan

<http://es.opensolaris.org>

開放的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
•••••
открыт
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
ఁవఱిపఁపఱఱ

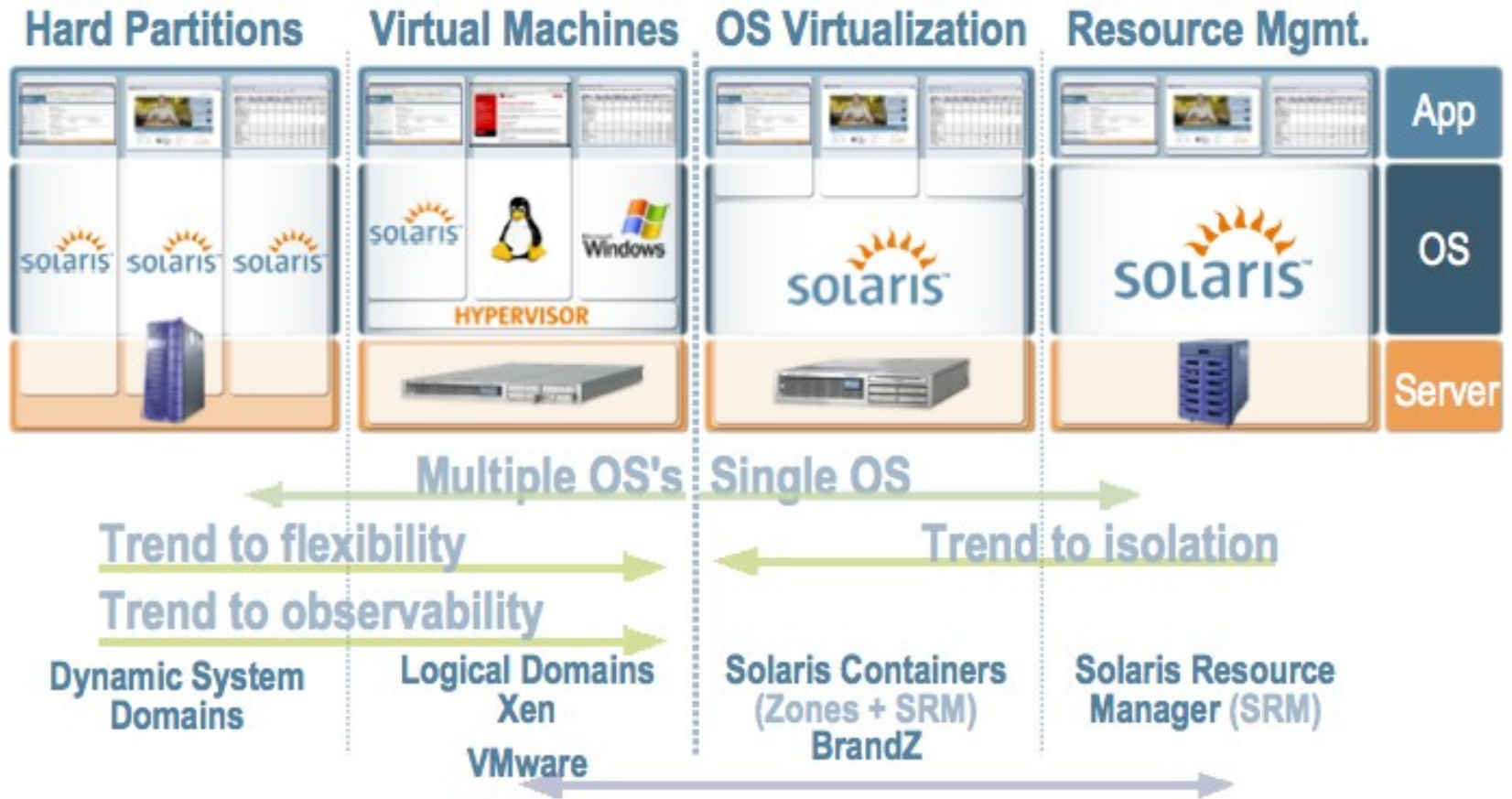


열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
ᄀᄀᄀᄀ
πᄀᄀᄀ
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
ఁవఁలిపపఁడఁ

Virtualización



Introducción





Introducción

- **Particiones físicas.**
La máquina se puede dividir en distintos dominios.
- **Máquinas virtuales.**
En una máquina podemos tener corriendo varias máquinas virtuales con distintos SO.
- **Virtualización de SO.**
Un SO genera varias imágenes de el mismo para dar la impresión de tener varios SO distintos.
- **Control de recursos.**
Solaris dispone de una serie de herramientas para controlar los recursos del sistema. CPU, memoria, tiempo de CPU



Objetivos

- Consolidación de servidores.
- Entornos de pruebas.
- Despliegue rápido de entornos de desarrollo.
- Aprovechar los recursos disponibles.
- Seguridad



열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
:::~::~
गोप
オープン
livre
ανοικτό
offen
otevřený
öppen
ОТКРЫТЫЙ
ఁవఱిపఁపఱఱ

OpenSolaris Zonas



Introduccion a las Zonas

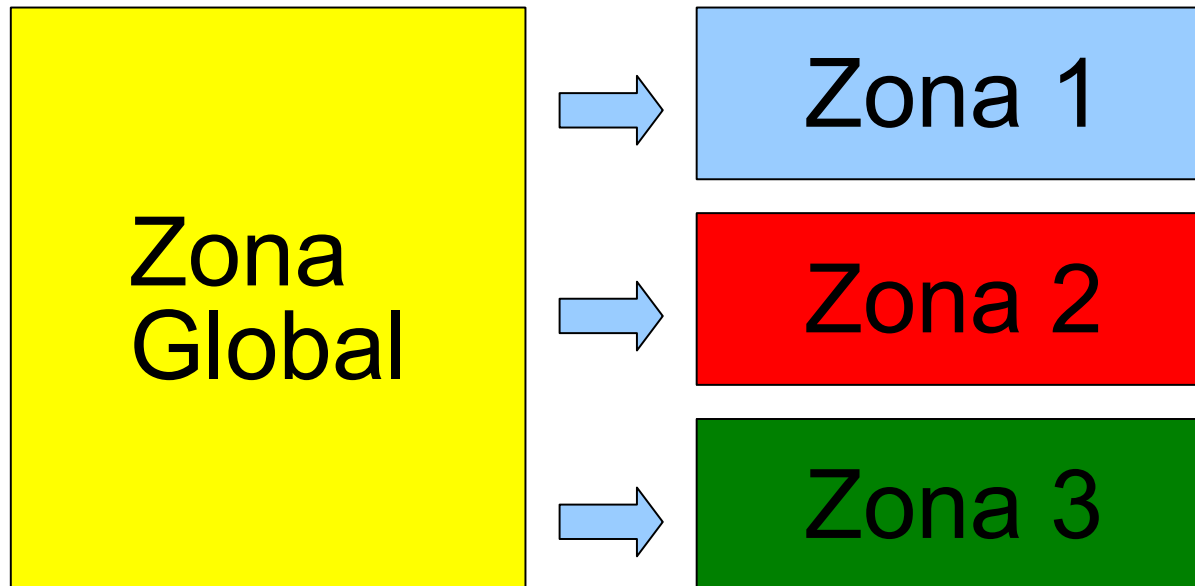
- Es una tecnología de virtualización.
- Permite ejecutar varias “instancias” del sistema operativo a la vez.
- Todas las instancias se ejecutan en un único kernel.
- No existe una capa intermedia de virtualización de hardware. Aunque algunos dispositivos si son virtuales.
- Cada una de ellas es independiente de las demás.

Características de una Zona

- La zona está identificada por un nombre y un id numérico.
- Dispone de sus propios procesos `init`, `(z)sched`, etc.
- Cada zona dispone de los servicios necesarios para su funcionamiento (`/proc`, `/dev/console`, etc.)
- Los procesos de una zona ignoran la existencia de los demás.
- La seguridad de las zonas impide que sus procesos interactúen.

La Zona global

- Se crea cuando el sistema arranca.
- Su id es el 0 y su nombre es global
- Tiene acceso a los procesos de las demás zonas.
- Idealmente debería usarse solo para administración.



Commandos habituales

`zonecfg -z zonename`

Crear, eliminar, reconfigurar.

`zoneadm list -cv`

listar zonas en el sistema.

`Zoneadm -z zonename install`
zona.

Empezar la instalación de una

`zoneadm -z zonename boot`

arrancar zona.

`zoneadm -z zonename halt`

parar zona.

`zoneadm -z zonename reboot`

reiniciar zona.

`zlogin zonename`

logearse al sistema.

`zlogin -C zonename`
zona.

Acceder a la consola de la

Pasos para la creación de una zona.

- Crear la zona con el comando `cfgadm`.
- Definir el `zonepath`.
- Añadir la configuración adicional que deseemos. (interfaces de red, containers, privilegios, ...)
- Crear el directorio que contendrá la zona y darle permisos 700.
- Iniciar la copia de los binarios con `zoneadm`.
- Logearnos a la consola y seguir el asistente para terminar la configuración.

Ejercicio 1: Ver procesos entre zonas

Guión:

- 1.- Arrancamos el sistema y dos zonas adicionales.
- 2.- Lanzamos un proceso ksh en cada una de las zonas.
- 3.- Comprobamos que desde cada una de las zonas no vemos el proceso ksh de la otra.



Ejercicio 2: Usando la Zona Global para administración.

Guión:

Ejecutamos un #pkill -9 ksh, comprobamos como los procesos de las zonas no global mueren.



Tipos de Zonas.

- Small Zone.
- Big Zone.
- BrandZ.

Características una Small Zone.

- Comparten los directorios de sistema con la zona global. (no tiene permisos para modificarlo.)
- Tiene su propio /var y /etc
- Ocupa muy poco espacio en el disco (300MB aprox.).

Ventajas:

- Optimización de recursos

Inconvenientes:

- Poca flexibilidad

Ejercicio 3: Instalación de una Small Zone

```
bash-3.2# zonecfg -z viernes13
viernes13: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:viernes13> create
zonecfg:viernes13> set zonepath=/viernes13
zonecfg:viernes13> commit
zonecfg:viernes13> exit
bash-3.2# mkdir /viernes13
bash-3.2# chmod 700 /viernes13
bash-3.2# zoneadm -z viernes13 install
Preparing to install zone <viernes13>.
Creating list of files to copy from the global zone.
```

Características una Big Zone

- Tiene sus propios directorios de sistema. No comparte ningún directorio con la global zone.

Ventajas:

- Flexibilidad.

Inconvenientes:

- No es tan optimo desde el punto de vista de recursos

Ejercicio 4: Instalación de una Big Zone

```
bash-3.2# zonecfg -z bigzone
```

```
bigzone: No such zone configured
```

```
Use 'create' to begin configuring a new zone.
```

```
zonecfg:bigzone> create
```

```
zonecfg:bigzone> set zone
```

```
set zonename= set zonepath=
```

```
zonecfg:bigzone> set zonepath=/bigzone
```

```
zonecfg:bigzone> remove inherit-pkg-dir dir=/sbin
```

```
zonecfg:bigzone> remove inherit-pkg-dir dir=/usr
```

```
zonecfg:bigzone> remove inherit-pkg-dir dir=/platform
```

```
zonecfg:bigzone> remove inherit-pkg-dir dir=/lib
```

```
zonecfg:bigzone> commit
```

```
zonecfg:bigzone> end
```



Configurando una zona

- **Variables de configuración** #zonecfg:bigzone>
set autoboot=true
- **Atributos** #zonecfg:bigzone> add attr
- **Interfaces de red** #zonecfg:bigzone> add net
- **Filesystems** #zonecfg:bigzone> add fs
- **Control de recursos** #zonecfg:bigzone> add rctl
- **Dispositivos** #zonecfg:bigzone> add device
- **Dataset** #zonecfg:bigzone> add dataset



Ejercicio 5: Modificando la configuración

Guión:

- ◆ Añadir red
- ◆ Añadir filesystem
- ◆ Añadir control de recursos



Contenedores I

- Permiten limitar el consumo de recursos de cada una de las zonas.
- Se configuran usando la herramienta zonecfg.
- Permiten aumentar el nivel de seguridad, prevención DOS.

Contenedores II

- Set max-lwps, max-sem-ids, max-shm-memory, limitpriv, etc
- Capped-cpu (ncpus, podemos asignar fracciones de cpu)
- Capped-memory (locked, swap, physical)
- Dedicated-cpu (ncpus)



Ejercicio 6: Limitando los recursos

Guión:

- ◆ Probar fork-bomb (`perl -e 'fork while 1;'`)
- ◆ Agotar la memoria
- ◆ Llenar tmp
- ◆ etc.

Seguridad I

Hay tareas que no está permitido realizar dentro de una zona:

- ◆ Modificar interfaces de red o tablas de rutas.
- ◆ Acceder al dispositivo `/dev/kmem`.
- ◆ Rebotar o apagar todo el sistema.
- ◆ Cargar módulos personalizados del kernel.

Seguridad II

- La seguridad en OpenSolaris es granular basado en privilegios, en contra del todo o nada del unix tradicional.
- Los procesos dentro de una zona no disponen de todos los privilegios, para evitar que puedan interactuar con los de otras.
- Ni siquiera el usuario root de una zona (id=0) tiene todos los privilegios

Seguridad III

- La suma de privilegios más la configuración de la zona permitiendo el acceso solo a los recursos necesarios es la que garantiza la seguridad de esta.
- El objetivo no es engañar al usuario humano, sino impedir que los procesos de distintas zonas interactúen.

Ejercicio 7: Seguridad

Guión:

- ◆ Ejecutar ppriv en la zona global y en la zona1.
- ◆ Añadir privilegios para permitir el uso de `dtrace` (`set limitpriv="default,dtrace_proc,dtrace_user"`)
- ◆ Tratar de asignarnos el privilegios para ver procesos de otras zonas (`set limitpriv="default,priv_proc_zone"`)
- ◆ Cualquier trastada que se le ocurra a la audiencia.

Performance

- Las distintas zonas comparten kernel, no hay tareas duplicadas (callout table).
- Las páginas de memoria de texto (binario y librerías) son compartidas por todas las zonas.
- La cache de nombres de directorio, DNCL, es compartida entre todas las zonas.



Ejercicio 8: Performance

Guión:

- ♦ Comprobar el estado de la memoria tras sucesivos arranques de zonas.

Procesos de una Zona I

- Zoneadmd

- ◆ Crear las estructuras del kernel necesarias y iniciar un proceso zsched.
- ◆ Configurar el control a los recursos de dicha zona (como pools de CPU, memoria o privilegios).
- ◆ Configurar los dispositivos de la zona con el comando devfsadm.
- ◆ Crear y destruir las interfaces de red virtuales.
- ◆ Montar los filesystems.
- ◆ Proporcionar un servidor de consola para el comando zconsole.
- ◆ Ejecutar el proceso init de la zona.
- ◆ Proporcionar un Door Server, los clientes como el zoneadm o el propio kernel se conectan a el para enviar mensajes de cambio de estado a la zona como halt, reboot, ...



Procesos de una Zona II

- Zsched
 - ♦ Es el padre de todos los threads del kernel de su zona.
 - ♦ Lanza el proceso init de la zona cuando arranca.



Ejercicio 9: Verificando procesos

Guión:

Lanzamos un ps para comprobar los procesos ejecutándose en el sistema.



열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
:::~
πινρ
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
ఁవఱిపఁపఱఱ

OpenSolaris BrandZ

BrandZ I

- Permite ejecutar binarios no nativos en un kernel OpenSolaris.
- Los binarios deben tener formato ELF
- El proyecto está aun en desarrollo
- Actualmente funciona de forma estable RHEL 3, y CentOS 3.

BrandZ II

- Dado que el formato de los ejecutables de Linux y OpenSolaris es el mismo (ELF) idealmente deberíamos poder procesarlos sin más, sin embargo estos esperan unos servicios por parte del kernel, a los que suelen acceder mediante llamadas al sistema que, lógicamente, son distintas en ambos S.O.. En el caso de Linux el problema va incluso un paso más allá, ya que existen diferencias entre las distintas distribuciones dependiendo del nivel de kernel y de la versión de glibc.



BrandZ III: Técnica de interposición

- Se crea una macro que captura los puntos de entrada de un proceso en el kernel (system calls, tratamiento de señales, etc)
- Antes de ser ejecutados por el kernel son tratados por un módulo específico que hace de traductor.
- Una vez procesados se entregan de nuevo a dicho módulo que reformatea la salida a lo que espera el proceso.

BrandZ IV: Ciclo de vida de un proceso

- λ El proceso Linux ejecuta la interrupción 80.
- La macro `BRAND_CALLBACK()` comprueba si la interrupción tiene origen en un Brandz.
- Opensolaris pasa el control del thread al módulo específico para brandz.
- El módulo `lx` lanza la librería de emulación.
- Dicha librería modifica la llamada al sistema para que sea compatible con OpenSolaris.
- El kernel de OpenSolaris procesa la llamada y devuelve la salida a la librería.
- Otra vez se realizan las operaciones oportunas, ahora para adaptar la salida a lo que espera el proceso Linux.
- La librería devuelve el resultado al proceso.

Ejercicio 10: Instalación Brandz

Guión:

```
# zonecfg -z linux
linux: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:linux> create -t SUNWlx
zonecfg:linux> set zonepath=/linux-zone
zonecfg:linux> commit
zonecfg:linux> exit
# mkdir /linux-zone
# chmod 700 /linux-zone
bash-3.2# zoneadm -z linux install -d /centos_fs_image.tar
Installing zone 'linux' at root directory '/linux-zone'
```

open



USE



IMPROVE



EVANGELIZE

<OpenSolaris Hispano>

<http://es.opensolaris.org>

Lista de distribución

<http://mail.opensolaris.org/mailman/listinfo/ug-sposug>

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
⋮
πινρ
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
ఠెఱిపఱఠె