Les zones dans (Open)Solaris

- Mise en oeuvre
- (Open)Solaris
- Avantages/Inconvénients
- Exemples
- Outils de monitoring

Mise en oeuvre

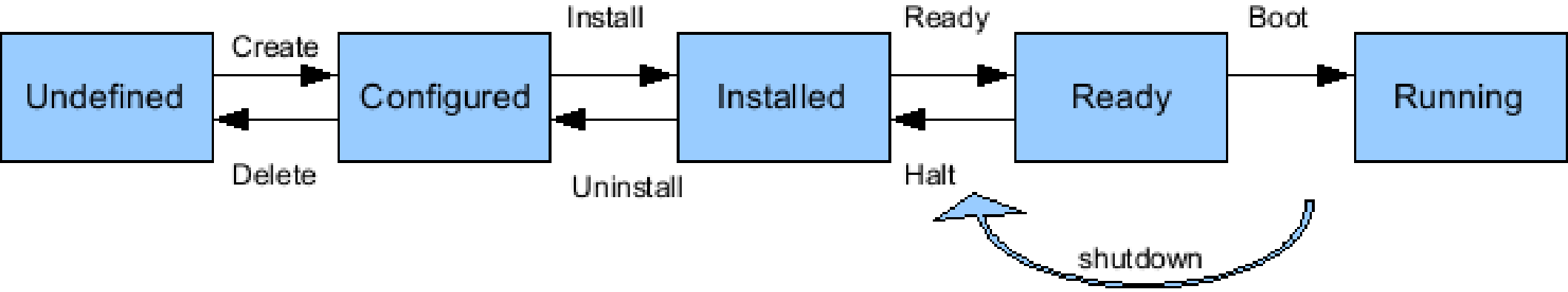
- Créer une zone:

  “zonecfg -z zone1”

  “create” pour créer la zone

  Au moins le chemin de la zone est requis -> zonepath
- Installer la zone: “zoneadm -z zone1 install”
- “zoneadm list -iv”

Mise en oeuvre (2)



- Utiliser un fichier de réponses: sysidcfg

  “cp sysidcfg /zones/zone1/root/etc”

  “cp site.xml /zones/zone1/root/var/svc/profile/”

  “cp -p S99postinstall /zones/zone1/root/etc/rc3.d/”
- Démarrer la zone: “zoneadm -z zone1 boot”

Principales commandes

- zonecfg: Permet d'établir et modifier la configuration de la zone.
- zoneadm: Permet d'administrer la zone (démarrage, arrêt, installation et dé-installation)
- zlogin: Permet de se connecter au terminal de la zone (l'équivalent de la console d'une vrai machine).

Le réseau

- Pour ajouter une interface réseau:

```
yakari-root% zonecfg -z zone1
zonecfg:zone1> info
zonecfg:zone1> add net
zonecfg:zone1:net> set  address=192.168.0.50
zonecfg:zone1:net> set physical=bge0
```

(Open)Solaris

- OS UNIX créé par Sun Microsystems
- La majorité du noyau a été publié en tant que code source libre en juin 2005 sous licence CDDL en tant qu'OpenSolaris
- Code source disponible sur opensolaris.org

Pourquoi (Open)Solaris

- Stabilité, fiabilité, compatibilité binaire ascendante
- Zones
- ZFS
- DTrace

Concepts

- Idée des “BSD jails” (ou “chroot”?)
- Mieux intégré à l'OS + sécurité
- Processes dans une zone ne peuvent affecter les processes hors de cette zone
- similaire à une machine virtuelle, mais “overhead” minimal

Utilisations

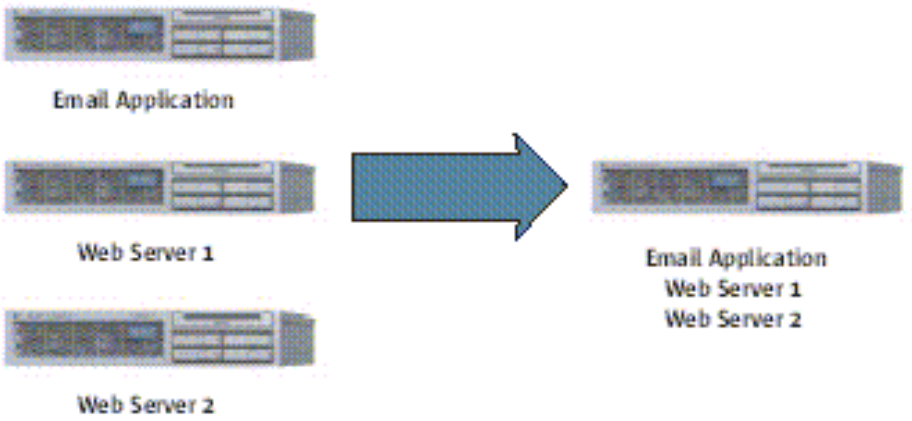
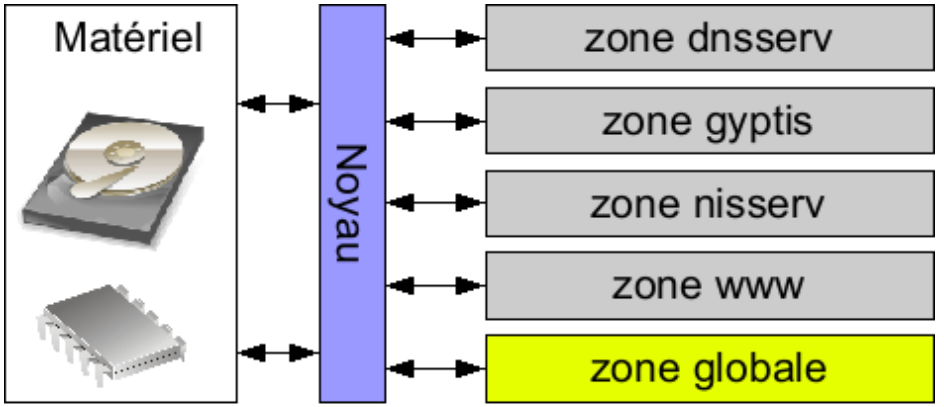


Figure 1—With server virtualization, applications can be safely consolidated onto a fewer number of servers.

Consolider les serveurs

Créer des “sandbox” pour les applis

Vue d'ensemble



- Une seule instance du noyau
- Une et une seule zone globale
- 8192 zones “non globales”
- Accès à toutes les zones depuis la zone globale

Fonctionnement des zones

- Utilisation d'un membre zoneid dans les structures noyau
  - la gestion des processus
  - la pile IP
  - les services RPC (ex: NFS)
  - les IPC
- L'ensemble du noyau est développé en connaissance des zones, d'où une intégration parfaite
- Évolution en cours : crossbow

Restriction de privilèges

- Exemple de privilèges interdits:

  PRIV_NET_RAWACCESS (accès brut aux interfaces réseaux)

  PRIV_PROC_ZONE (envoyer un signal vers un processus d'une autre zone)

  PRIV_SYS_DEVICES (création de périphériques)

  PRIV_SYS_NET_CONFIG (modification de la configuration réseau)

Avantages / Inconvénients

- une seule instance du noyau
- très grande facilité d'administration
- configuration très souple
- consommation de ressources faibles
- isolation interzones

- une seule instance du noyau
- une table de routage unique
- NameServices switch

Exemple d'utilisation de zoneid

Extrait de procfs.h

```
272 typedef struct psinfo {
273         int     pr_flag;        /* process flags (DEPRECATED; do not use) */
274         int     pr_nlwp;        /* number of active lwps in the process */
275         pid_t   pr_pid;         /* unique process id */
276         pid_t   pr_ppid;        /* process id of parent */
277         pid_t   pr_pgid;        /* pid of process group leader */
278         pid_t   pr_sid;         /* session id */
279         uid_t   pr_uid;         /* real user id */
280         uid_t   pr_euid;        /* effective user id */
281         gid_t   pr_gid;         /* real group id */
282         gid_t   pr_egid;        /* effective group id */
283         uintptr_t pr_addr;      /* address of process */
284         size_t  pr_size;        /* size of process image in Kbytes */
285         size_t  pr_rssize;      /* resident set size in Kbytes */
286         size_t  pr_pad1;
[..]
304         taskid_t pr_taskid;     /* task id */
305         projid_t pr_projid;     /* project id */
306         int     pr_nzomb;       /* number of zombie lwps in the process */
307         poolid_t pr_poolid;     /* pool id */
308         zoneid_t pr_zoneid;     /* zone id */
309         id_t    pr_contract;    /* process contract */
310         int     pr_filler[1];   /* reserved for future use */
311         lwpsinfo_t pr_lwp;      /* information for representative lwp */
312 } psinfo_t;
```

typedef int id_t; /* A process id, */
typedef id_t zoneid_t;

Les types de zones

- les zones “sparse”

  Héritage des principaux systèmes de fichiers depuis la zone globale :

  - /usr (rappel: /bin est un lien vers /usr/bin)
  - /lib
  - /platform
  - /sbin

- les zones full (“whole” zone)

  Une zone full dispose de son propre système de fichiers global (aucun héritage)

Clonage des zones

Migration de zones

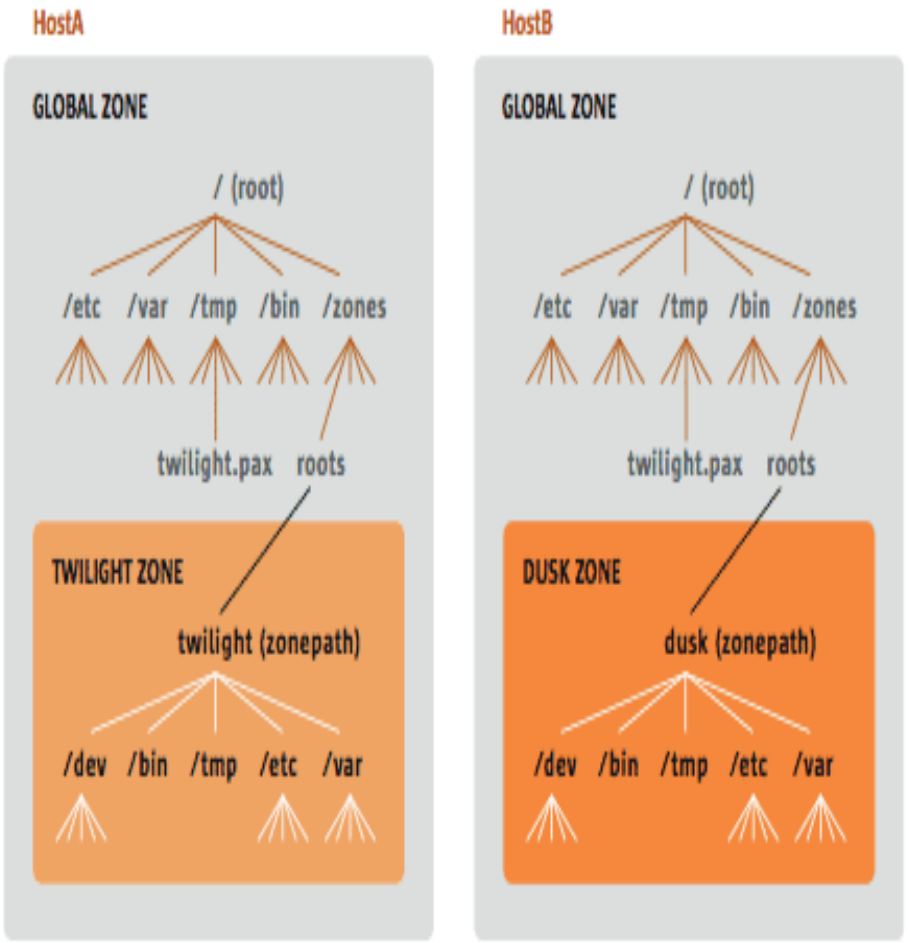


Figure 8—After Container 'dusk' has been attached to HostB

- Possible si les zones globales “identiques”
- “détacher” la zone
- Déplacer les fichiers
- “attacher” la zone

Exemples

- Migration serveur web

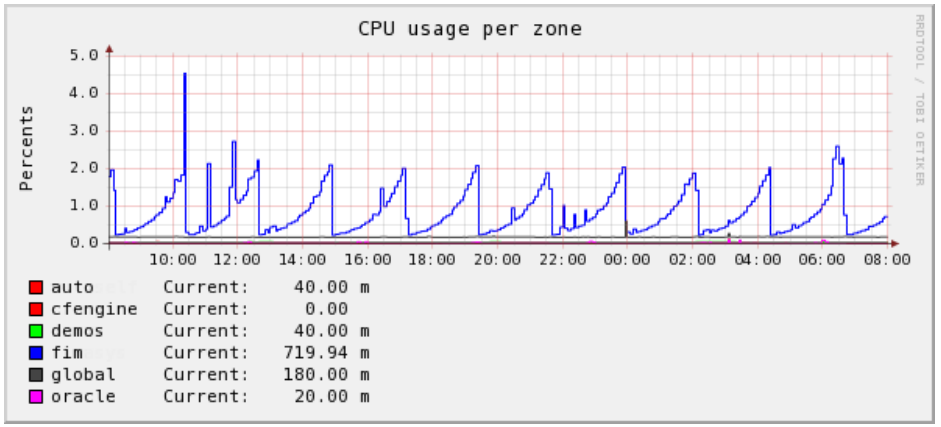
  Déplacement des binaires apache

- Migration serveur mail

  Déplacement des binaires sendmail, mailscanner, etc...

  Gain de la migration -> machine CMT

Outil de monitoring



- Zonestats:
  - http://asyd.net/home/projects/zonestats
- Pré requis: perl et modules RRD