Introduction à ZFS

Matthieu Herrb

Capitoul, 26 février 2009

http://www.laas.fr/~matthieu/talks/capitoul-zfs.pdf

Agenda

Agenda

Solaris / OpenSolaris

version commerciale de l'OS de Sun.
Actuellement version 10/08.
Favorise la stabilité.

version Open-Source (CDDL).
Développement de nouvelles fonctionnalités.
Version courante 2008.11.
OS orienté machines de bureau (Gnome)
Live CD

Présentation

Zetta (10^{21}) byte file system.

- Développé par Sun Microsystems
- Introduit dans Solaris 10
- Licence CDDL
- Porté sur Mac OS X, Linux, FreeBSD,. . .
- Dernières versions dans OpenSolaris

Principes de conception

Système de fichiers pour les 20 prochaines années:

- Pas de limites pratiques (taille des disques, fichiers, ...)
- Garantir la sécurité des données (intégrité, disponibilité)
- Administration simplifiée
- Gestionnaire de volume intégré
- Performances élevées
- Indépendant de l'architecture matérielle

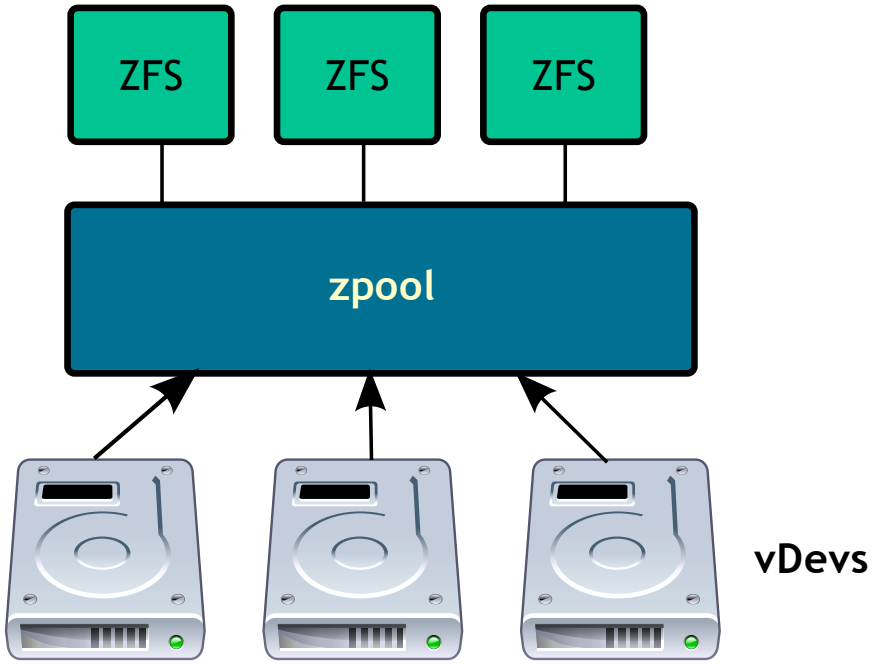
Fonctionnalités

- Snapshots
- Clones
- Quotas et réservation d’espace
- Compression
- Duplication
- Export/Import

Agenda

zpool

Un pool est un ensemble de périphériques qui fournissent de l'espace pour le stockage et la duplication des données.



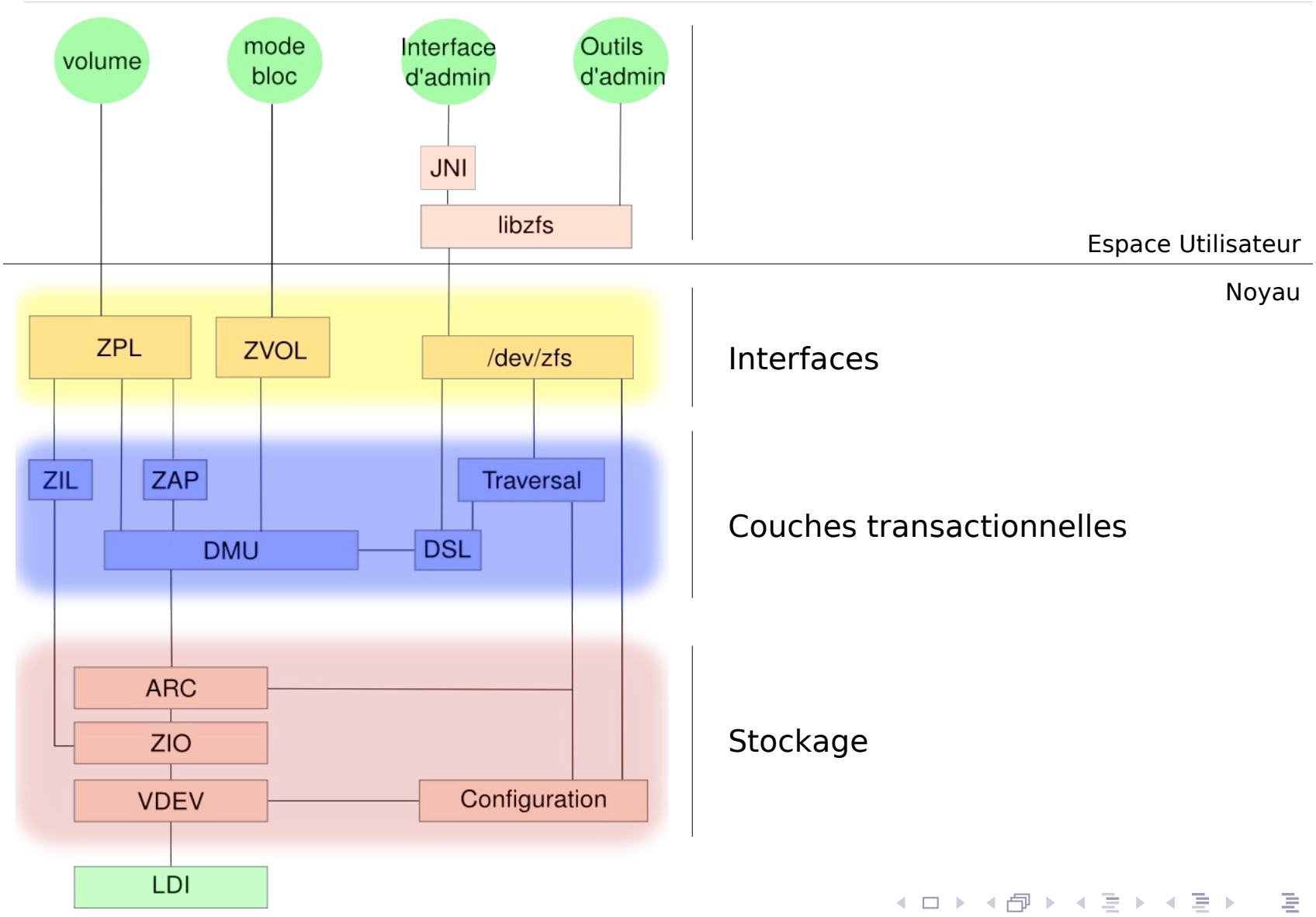
- Construit à partir de périphériques virtuels (vdevs)

vdevs

Unité de base de stockage de données

- disques : entiers ou juste une partition
- fichiers dans un autre système de fichiers
- miroirs : 2 (ou plus) disques, partitions ou fichiers
- raid-z : plusieurs disques, variante de RAID-5

Architecture interne



Plus en détails

- ZPL: Posix Layer: interface classique du système de fichiers.
- DMU: Data management unit: coeur de la gestion des données dans les pools. Modèle transactionnel.
  - ZAP attribute processor
  - DSL Data snapshot layer
  - ZIL Intention Log
  - ARC Adaptative replacement Cache
  - ZIO Couche d’entrées sorties vers les vDevs.4

ZMirror

Miroir classique
Utilise les mécanismes de checksum pour valider les lectures sur un composant,
Bascule sur le second composant si détection d'erreur,
Correction du composant défaillant (si possible).

RaidZ

- Similaire au RAID-5 (parité distribuée
- Utilise les checksums (SHA-256 + fletcher)
- Rpose sur le copy on write : supprime le "write-hole".

Si beaucoup de disques : RaidZ2 (double la parité).

Copy on Write (CoW)

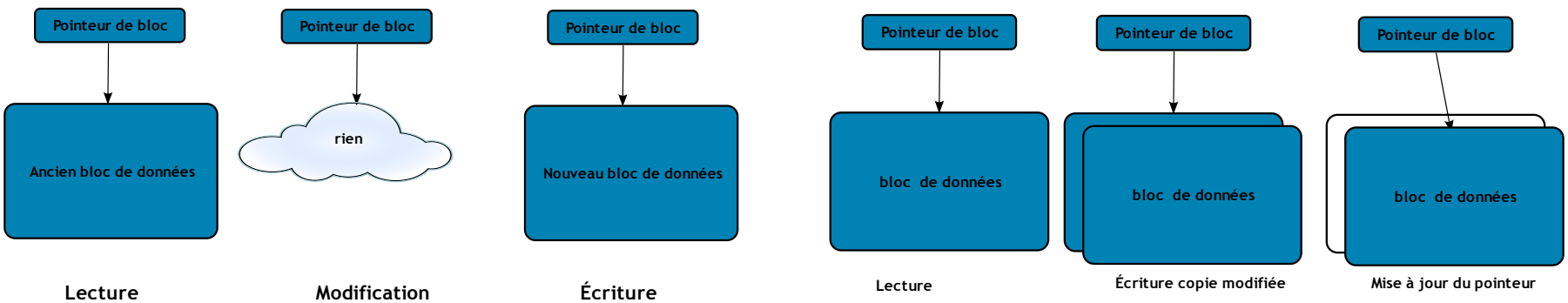
Modification d'un bloc de données

Méthode traditionnelle

Read/Modify/Write

Copy on write

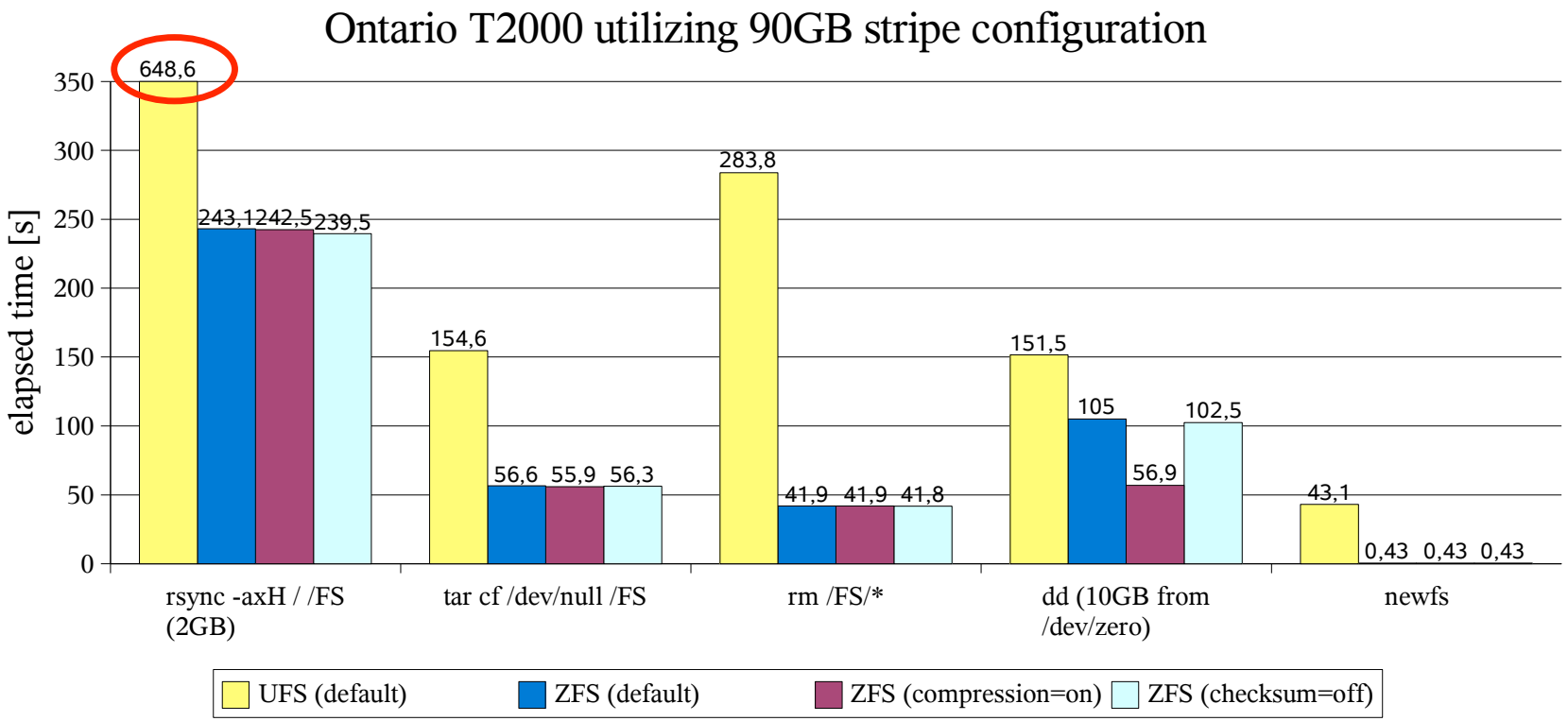
Read/Write/update



Gain de performance, meilleure résistance aux coupures.

Performances

[Nau 06]



Agenda

Commande zpool

Gestion des zpools

- Création
- Destruction
- Import/Export
- Ajout de stockage
- Visualisation état, performances

Commande zfs

Gestion des systèmes de fichiers

- Création/destruction
- Montage
- Gestion des attributs (export NFS, compression, etc.)
- Snapshots/Clones
- Sauvegardes

Exemple: creation d'un pool

Raidz

```
zpool create mypool raidz disk0 disk1 disk2 disk3 disk4
```

miroir

```
zpool create mypool mirror disk0 disk1
```

concaténation

```
zpool create mypool disk0 disk1
```

Exemple: création de systèmes de fichiers

Création de home directories

```
zfs create mypool/home
zfs create mypool/home/alice
zfs create mypool/home/bob
zfs create mypool/home/charlie
```

Définition d'un point de montage

```
zfs set mountpoint=/home mypool/home
```

Exemple: quotas et reservation

Définition d'un quota

```
zfs set quota=10G mypool/home/alice
zfs list
```

Définition d'une réservation

```
zfs set reservation=5G mypool/home/alice
zfs list
```

Exemple: snapshots

Création d'un snapshot

```
zfs snapshot -r mypool/home@jeudi_12h15
zfs list -t snapshot
ls -l .zfs/snapshot/
```

Restauration d'un snapshot

```
zfs rollback mypool/home@jeudi_12h15
```

Exemple: exportation

Utile pour la gestion des supports amovibles.

Exportation avant déconnexion

```
zpool export mykey
```

Importation d'un support après reconnexion

```
zpool import mykey
```

Exemple: suivi de l’état

Commandes d’inspection

```
zpool status
zpool iostat 10
zpool scrub mypool
```

Agenda

Un système de stockage pour l'avenir

- Prendre en compte la capacité croissante des supports et des données
- Simplification de l'administration.
- Meilleures performances: moins besoin de systèmes dédiés
  (ou meilleure utilisation de ces derniers)
- Techno mûre, en pleine expansion.
- Concurents : ext4, btrfs, hammer...

Intégration avec le système

OpenSolaris:

- Gestion des “Environnements de boot”:
  création d’un clone avant une mise à jour: permet de revenir en arrière.
  Commande `beadm`: intégration avec `grub` pour sélectionner le clone à booter.
- Time Slider : “clone” de time machine,
  implémenté sous forme de service.
  Extension nautilus (file manager de gnome) pour explorer les snapshots.
- Utilisation de clones avec zones ou machines virtuelles: déduplication.

Futur

ZFS est ouvert à des extensions - nouvelles fonctionnalités.

En cours:

- Retrait d'un device d'un pool
- Déduplication
- Chiffrement
- ...vDevs distants

Bibliographie

Nau06 Thomas Nau, *Peta, Exa, Zetta: looking at 18+ months of ZFS Experience*, SunHPC 2006, Singapour, mai 2006.

Bonfils08 Bruno Bonfils, *OpenSolaris: fonctionnement des zones et ZFS*, Conférence Guses, Toulouse, octobre 2008.

Banham07 J. Banham & J. Nash, *ZFS Under the Hood*, London Solaris User's Group, mai 2007.