Retour d'expérience ZFS dans un laboratoire CNRS

Jérôme COLOMBET

https://homepages.lcc-toulouse.fr/colombet/respire_retour_exp_zfs.pdf

14 octobre 2021

Le GT-ZFS

- **G**roupe de **T**ravail ZFS, rattaché à Resinfo ⇒ son objectif, promouvoir le ZFS dans l'ESR
- 6 membres, répartis sur toute la France avec des besoins et les contraintes du terrain
- Le groupe de travail ZFS a pour objectif de fournir :
  - des documentations techniques en corrélation avec le marché Matinfo5,
  - conseils, bonnes pratiques, scripts,
  - tutoriels, démonstrations et formations...
- Rassembler les usagers autour de la liste @ : stockage@groupes.renater.fr

L'entité LCC - chimie Toulousaine

- **L**aboratoire de **C**himie de **C**oordination (chimie métaux de transition)
- Proche de la nouvelle attraction touristique Téléo ; Téléphérique Urbain Toulouse
- Le LCC **UPR 8241**, installé sur un campus propre CNRS 205
  - 17 équipes de recherche autour des axes Catalyse, Matériaux et Santé
  - 18 services scientifiques et administratifs en soutien
  - C'est 3 informaticiens pour **270** personnes et un bâtiment de 11000m^2
  - **R**essources **I**nformatiques et **C**alcul **S**cientifique
- Grands équipements techniques **RMN**, **RX**, Spectrométrie IR, Microscopie Electronique
- Volume de données exponentielles, critique et reparti

Rappel du contexte technique

Figure – DC 2013 - 2021

- 3 salles serveurs reparties sur le campus 205
- Réseau 10G sur l'ensemble des bâtiments, 600 prises
- Parc utilisateurs multi-os (380 postes)
- Parc scientifique vieux, hétérogène et critique (50 postes)
- 3 clusters HA-PRA sur la solution Proxmox VE 6/7
- 2 clusters HPC-OAR (Centos 6, Debian 10)
- 3 stockages centralisés ZFS (data, VMs, mails, ...)
- Volumétrie consolidée de **0,5 Po**

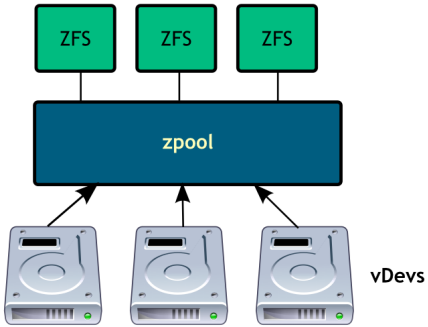
Zetta byte file system

- Développé par Sun Microsystems (2004)
- Introduit dans Solaris 10 (2005), licence CDDL = incompatible avec la GNU GPL
- Porté sur Mac OS X, Linux, FreeBSD,...
- Fonctionnalités
  - Snapshots
  - Clones
  - Quotas et réservation d'espace
  - Compression
  - Dé-duplication
  - Export/Import
  - Chiffrement intégré

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

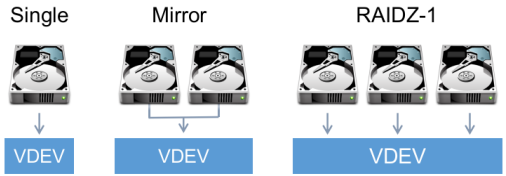
Pool, Fs, Raid-z

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



zpool est construit à partir de périphériques virtuels **vdevs**, c'est l'unité de base de stockage de données

- disques : entiers ou juste une partition
- fichiers dans un autre système de fichiers
- miroirs disques, partitions ou fichiers
- raid-z : plusieurs disques vs raid5



Les 2 commandes à connaître

zpool pour la gestion des pools

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

zfs pour la gestion des systèmes de fichiers

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

Exemple : installation sur debian 10

dkms

```
## buster backports
deb http://deb.debian.org/debian buster-backports main contrib non-free
deb-src http://deb.debian.org/debian buster-backports main contrib non-free
```

dkms

```
$ apt install -t buster-backports dkms spl-dkms -y
$ apt install -t buster-backports zfs-dkms zfsutils-linux -y
$ reboot
```

lister vos disques physiques

```
$ ls -lh /dev/disk/by-id/
$ ata-ST2000LM007-1R8174_WABC13DE -> ../../sda
```

Exemple : creation d'un pool

raidz

```
zpool create hpool raidz ata-ST2000LM007-1R8174_WABC13DE disk1 disk2 disk3
```

miroir

```
zpool create hpool mirror disk0 disk1
```

concaténation

```
zpool create hpool disk0 disk1
```

Exemple : création de systèmes de fichiers

Création d'un home directory

```
$ zfs create hpool/home
$ zfs create hpool/home/colombet
```

Suppression d'un home directory

```
$ zfs destroy hpool/home/colombet
```

Définition d'un point de montage

```
$ zfs set mountpoint=/home hpool/home
$ zfs get mountpoint hpool
 hpool mountpoint /hpool default
```

Exemple : quotas

Définition d'un quota

```
$ zfs set quota=10G hpool/home/colombet
$ zfs list hpool/home/colombet
NAME USED AVAIL REFER MOUNTPOINT
hpool/home/colombet 140K 10.0G 140K /home/colombet
```

Visualisation d'un quota

```
$ zfs get quota hpool/home/colombet
hpool/home/colombet quota 10.0G local
```

Suppression d'un quota

```
$ zfs set quota=none hpool/home/colombet
```

Exemple : reservation

Définition d'une réservation

```
$ zfs set reservation=5G hpool/home/colombet
$ zfs list
NAME USED AVAIL REFER MOUNTPOINT
hpool/home/colombet 140K 3.48T 140K /home/colombet
```

Visualisation d'une réservation

```
$ zfs get reservation hpool/home/colombet
hpool/home/colombet quota 5.0G local
```

Exemple : snapshots

Création d'un snapshot

```
$ zfs snapshot -r hpool/home@zfs-auto-snap-$(date +%Y-%m-%d-%H%M%S)
$ zfs list -H -o name -t snapshot
hpool/home@zfs-auto-snap-2021-01-05-070033

$ ls -l .zfs/snapshot/
```

Restauration d'un snapshot

```
$ zfs rollback hpool/home@zfs-auto-snap-2021-01-05-070033
```

Exemple : exportation

Utile pour la gestion des supports amovibles ou en HA via @IP flottante

Exportation avant déconnexion

```
$ zpool export hpool
```

Importation d'un support après reconnexion

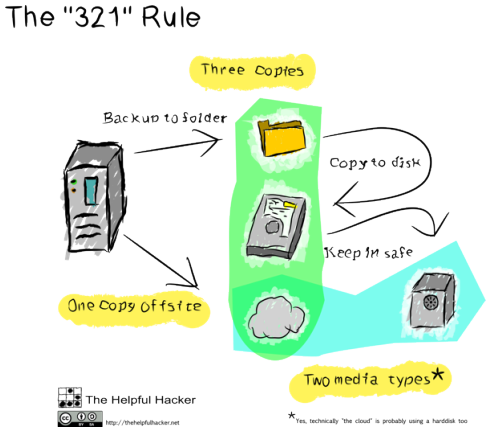
```
$ zpool import hpool
```

Exemple : Externaliser une copie ZFS via SSH

- disposer de trois copies de vos données
- stocker ces copies sur deux supports différents
- conserver une copie de la sauvegarde hors site

Externaliser un snapshot ZFS via SSH

```
$ zfs list -H -o name -t snapshot

hpool/snapshot-old

hpool/snapshot-new



$ zfs send -i snapshot-old snapshot-new| pv -b | ssh myserver

"mbuffer -m 1G | zfs receive bpool/backup -F"
```



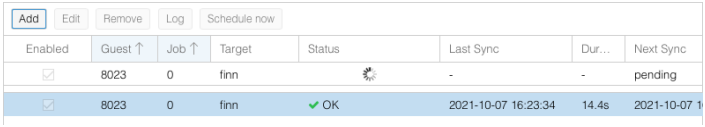
Exemples d'utilisation de ZFS au LCC

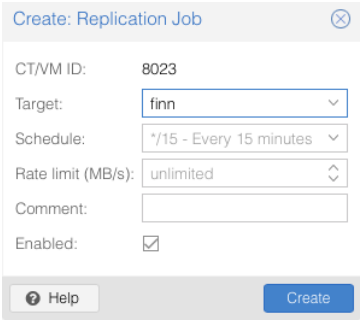
Exemple : PVESR - Proxmox VE Storage Replication

- Depuis la version 3 de PVE il est possible d'utiliser nativement le ZFS, mais pourquoi ?
- Faire un cluster HA pour les pauvres ...
- Oui depuis la version 4 il faut obligatoirement 3 noeuds

pvesr

```
$ pvesr create-local-job 8023-0 finn

$ pvesr list

JobID Target Schedule Rate Enabled

8023-0 local/finn */15 - yes

$ pvesr delete 8023-0 --force
```





Exemple : Proxmox - ZFS over iSCSI

sur stockage distant

créer un dataset dans le pool tank

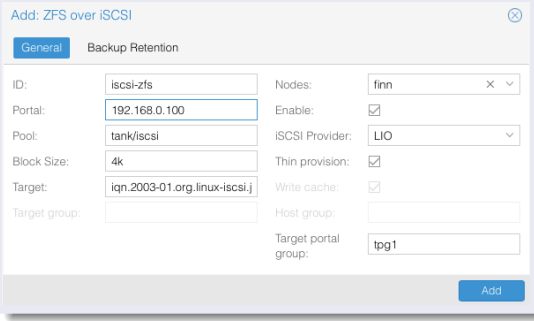
```
$ zfs create tank/iscsi
```

créer une target et les acls vers ce dataset

```
$ targetcli

/> create

Created target

iqn.2003-01.org.linux-iscsi.nas.x8664:sn.2c0c3e76710e

/> cd

iqn.2003-01.org.linux-iscsi.nas.x8664:sn.2c0c3e76710e/tpg1/acls

/> create iqn.1993-08.org.debian:01:1ae0ad6ebb5f
```

sur cluster proxmox

```
$ ssh-keygen -f /etc/pve/priv/zfs/192.168.0.100_id_rsa

$ ssh-copy-id -i /etc/pve/priv/zfs/192.168.0.100_id_rsa.pub

root@192.168.0.100
```



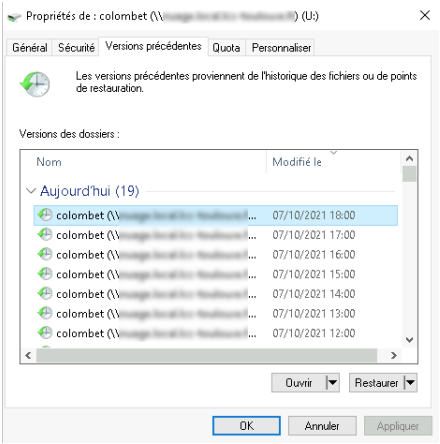
Exemple : SaMBa4 - versions précédentes

cron snapshots toutes les heures

```
$ zfs snapshot -r hpool/home@zfs-auto-snap-$(date
+%Y-%m-%d-%H%M%S)
$ zfs list -H -o name -t snapshot
hpool/home@zfs-auto-snap-2021-01-05-070000
$ zfs list -H -o name -t snapshot
```

/etc/samba/smb.conf ⇒ shadowcopy

```
shadow: snapdir = .zfs/snapshot
shadow: sort = desc
shadow: format = -%Y-%m-%d-%H%M%S
shadow: snapprefix = ^zfs-auto-snap
shadow: delimiter = -20
vfs objects = shadow_copy2
```



Exemple : SaMBa4 - gestion des quotas

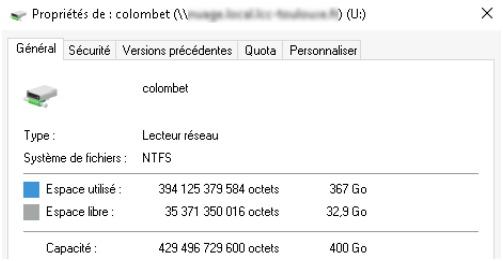
quota zfs + quota utilisateur colombet

```
$ zfs get quota hpool/home

hpool/home quota 15T local



$ zfs get -H userused@colombet hpool/home

hpool/home userused@colombet 367G local



$ zfs get -H userquota@colombet hpool/home

hpool/home userquota@colombet 400G local
```

/etc/samba/smb.conf ⇒ script bash quota zfs

```
get quota command = /opt/scripts/samba_quotazfs.sh %U
```

```
#!/bin/sh
username=$1
if [ ! -z  "$username" ]; then
  smbpath=${PWD}
  dataset=`/bin/df -l ${smbpath} | /usr/bin/tail -n 1 | /usr/bin/awk '
  infoused=`/sbin/zfs get -Hp userused@$username $dataset`
  infoquota=`/sbin/zfs get -Hp userquota@$username $dataset`
  usedbytes=`echo ${infoused}| /usr/bin/awk '{ printf "%.f", $3/1024 }
  quotabytes=`echo ${infoquota}| /usr/bin/awk '{ if ( $3 == "none" ) {
  echo 2 $usedbytes $quotabytes $quotabytes $usedbytes $quotabytes $qu
fi
exit
```



Pour conclure : ZFS est un système de stockage pour l'avenir

- Prendre en compte la capacité croissante des supports et des données
- Simplification de l'administration (zpool & zfs)
- Performances au rendez-vous même sur de simple configuration serveur
- C'est une technologie mûre, en pleine expansion et à promouvoir dans l'ESR
- La concurrence : ext4, btrfs, ceph ...
- Sans oublier le GT-ZFS

Questions

: jerome.colombet@lcc-toulouse.fr
: https://homepages.lcc-toulouse.fr/colombet/
: https://github.com/jeromecolombet
: https://twitter.com/neoclimb