

# ZFS

ZFS<sup>1</sup> est un système de fichier développé par SUN pour SOLARIS et récemment partiellement porté sous Linux. Pour des raisons de licences, ZFS ne peut être intégré au noyau Linux. Aussi, ce système de fichier pourra être utilisé pour stocker des données mais pas pour héberger le système.

ZFS pour Linux<sup>2</sup> utilise FUSE<sup>3</sup> pour l'interfacer au système.

ZFS présente de très nombreux avantages en termes de protection des données, volumétrie et de performance<sup>4</sup>. Mais, le point que nous allons mettre en avant dans le cadre de cet article est la facilité qu'il apporte à la gestion d'une infrastructure de disque comparativement à celle que nous venons de développer avec « mdadm » et « LVM ».

## 1.1 Installation ZFS

L'installation de ZFS sous Debian Lenny est réalisée avec la série de commande ci-dessous:

```
apt-get install -y fuse-utils libaio1
wget http://cdimage.at.debian.org/grml/bin/pool/main/z/zfs-
fuse/zfs-fuse_0.5.hg20081120-1_i386.deb
dpkg -i zfs-fuse_0.5.hg20081120-1_i386.deb
```

Ensuite, il faut éditer le fichier « /etc/default/zfs-fuse » afin de modifier la ligne ENABLE\_ZFS=no pour « ENABLE\_ZFS=yes ». Ainsi les « pools » ZFS seront montés automatiquement t avec le chargement du module.

```
nano /etc/default/zfs-fuse
```

Relancez le module ZFS pour FUSE afin de prendre en compte les dernières modifications.

```
/etc/init.d/zfs-fuse start
```

## 1.2 Création d'un disque virtuel

Comme ZFS permet de créer et monter un système de fichier depuis une seule ligne de commande, créez tout d'abord le point de montage « /storage ».

```
mkdir /media/zfs
```

Vous pouvez désormais créer votre système de fichiers ZFS avec la commande:

```
zpool create vhd -m /media/zfs raidz /dev/sdd /dev/sde
/dev/sdf
```

Dans cette commande:

- « vhd » est le nom du disque virtuel,
- « /media/zfs » est le point de montage de ce disque,
- « raidz » est le niveau de redondance des données entre les disques « /dev/sdd », « /dev/sde » et « /dev/sdf ». Ce niveau de redondance peut être de trois types:
- « mirror » - les données seront recopiées sur chaque disque
- « raidz » - les données seront réparties et recopiées entre les disques à l'image d'un RAID5. La perte d'un disque est tolérée sans pertes d'informations.

1 <http://www.sun.com/software/solaris/zfs.jsp>

2 [http://www.wizy.org/wiki/ZFS\\_on\\_FUSE](http://www.wizy.org/wiki/ZFS_on_FUSE)

3 <http://fuse.sourceforge.net/>

4 [http://opensolaris.org/os/community/zfs/docs/zfs\\_last.pdf](http://opensolaris.org/os/community/zfs/docs/zfs_last.pdf)

- « raidz » - identique à « raid » mais avec un niveau de redondance supérieur de sorte que deux disques peuvent être défectueux sans perte de données.

Vous noterez que tout le travail fastidieux de préparation des disques physique et le formatage du RAID n'est plus nécessaire!

La commande « zpool list » reporte la taille de votre disque virtuel (« 23,9G »). Le taux d'utilisation (« 156K » et « 0% ») et la qualité de l'espace de stockage est « ONLINE ».

```

NAME      SIZE      USED      AVAIL      CAP      HEALTH      ALTROOT
vHD       23,9G     156K     23,9G      0%     ONLINE     -

```

La commande « zpool status » reprends un peu les mêmes informations mais en les détaillant disque par disque.

```

pool: vHD
state: ONLINE
scrub: none requested
config:

      NAME                STATE          READ  WRITE  CKSUM
      vHD                  ONLINE         0     0     0
          raidz1           ONLINE         0     0     0
              sdd          ONLINE         0     0     0
              sde          ONLINE         0     0     0
              sdf          ONLINE         0     0     0

errors: No known data errors

```

La commande « zfs get all vHD » liste la totalité des paramètres de l'espace de stockage ZFS.

```

zfs get all vHD
NAME      PROPERTY          VALUE          SOURCE
vHD      type              filesystem     -
vHD      creation          ven mai 15 12:40 2009 -
vHD      used              97,9K         -
vHD      available         15,7G         -
vHD      referenced        24,0K         -
vHD      compressratio     1.00x         -
vHD      mounted           yes           -
vHD      quota             none          default
vHD      reservation       none          default
vHD      recordsize        128K         default
vHD      mountpoint        /media/zfs    local
vHD      sharenfs          off           default
vHD      checksum          on            default
vHD      compression       off           default
vHD      atime             on            default
vHD      devices           on            default
vHD      exec              on            default
vHD      setuid            on            default
vHD      readonly          off           default
vHD      zoned             off           default
vHD      snapdir           hidden        default
vHD      aclmode           groupmask     default
vHD      aclinherit        restricted     default

```

```

vHD   canmount           on           default
vHD   shareiscsi         off          default
vHD   xattr              on           default
vHD   copies             1           default
vHD   version            3           -
vHD   utf8only           off          -
vHD   normalization     none        -
vHD   casesensitivity    sensitive   -
vHD   vscan              off          default
vHD   nbmand             off          default
vHD   sharesmb           off          default
vHD   refquota           none        default
vHD   refreservation    none        default
vHD   primarycache      all         default
vHD   secondarycache    all         default
vHD   usedbysnapshots   0           -
vHD   usedbydataset      24,0K      -
vHD   usedbychildren    73,9K      -
vHD   usedbyreservation 0           -

```

La commande « zpool status » affiche l'état des pools.

```

pool: vHD
state: ONLINE
scrub: scrub completed after 0h0m with 0 errors on Fri May
15 12:44:00 2009
config:

    NAME                STATE          READ  WRITE CKSUM
    vHD                  ONLINE         0     0     0
    raidz1               ONLINE         0     0     0
    sdd                  ONLINE         0     0     0
    sde                  ONLINE         0     0     0
    sdf                  ONLINE         0     0     0

errors: No known data errors

```

La commande « zpool scrub vHD » force le contrôle et le cas échéant la réparation d'un disque ZFS.

La commande « zpool history » liste l'historique des actions réalisées sur les disques virtuels. History for 'vHD':

```

2009-05-15.12:40:47 zpool create vHD -m /media/zfs raidz
/dev/sdd /dev/sde /dev/sdf
2009-05-15.12:44:00 zpool scrub vHD

```

### 1.3 Création d'une partition virtuelle

Vous allez maintenant créer quatre partitions virtuelles dans le disque ZFS « vHD ».

```

zfs create vHD/vPart1
zfs create vHD/vPart2
zfs create vHD/vPart3
zfs create vHD/vPart4

```

La commande « zfs list » affiche l'intégralité des partitions du disque virtuel.

```

zfs list

```

NAME	USED	AVAIL	REFER	MOUNTPOINT
vHD	243K	15,7G	29,3K	/media/zfs
vHD/vPart1	24,0K	15,7G	24,0K	/media/zfs/vPart1
vHD/vPart2	24,0K	15,7G	24,0K	/media/zfs/vPart2
vHD/vPart3	24,0K	15,7G	24,0K	/media/zfs/vPart3
vHD/vPart4	24,0K	15,7G	24,0K	/media/zfs/vPart4

Vous noterez que comme le disque virtuel est déjà monté en « /media/zfs », les partitions virtuelles sont montées automatiquement dans des sous-répertoires de « /media/zfs ».

## 1.4 Modifier le point de montage d'une partition

Lorsque vous créez un disque virtuel ou une partition virtuelle ZFS, les espaces sont montés automatiquement. Par défaut, le disque virtuel « vHD » est monté en « /media/zfs » et ses partitions sont montées en « /media/zfs/vPart1 », « /media/zfs/vPart2 ».... Il est toutefois possible de modifier les points des disques et des partitions. La séquence de commande ci-dessous illustre comment monter « /media/zfs/vPart1 » sous « /vPart1 ».

```
mkdir /vPart1
zfs set mountpoint=/vPart1 vHD/vPart1
zfs unmount /vHD/vPart1
zfs mount /vHD/vPart1
zfs mount
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
vHD	256K	15,7G	30,6K	/media/zfs
vHD/vPart1	24,0K	15,7G	24,0K	/vPart1
vHD/vPart2	24,0K	15,7G	24,0K	/media/zfs/vPart2
vHD/vPart3	24,0K	15,7G	24,0K	/media/zfs/vPart3
vHD/vPart4	24,0K	15,7G	24,0K	/media/zfs/vPart4

## 1.5 Gestion de la défaillance d'un disque

Si la commande « zpool status » révèle un disque défectueux, la première opération à réaliser est de le retirer de « vHD ». Pour cela, utilisez la commande suivante:

```
zpool detach vHD /dev/sdd
```

Remplacez physiquement le disque et réintégrez le dans le pool avec la commande:

```
zpool attach vHD /dev/sdd
```

« zfs status » vous permettra de suivre la progression du travail de reconstruction du pool. Le disque « vHD » se reconstruira automatiquement en tâche de fond. Je n'ai pas trouvé de fonction native à ZFS pour être alerté par email en cas de dégradation d'un pool

## 1.6 Gestion de la taille des partitions

Contrairement à LVM, les partitions ZFS ne sont pas limitées en taille individuellement. Ceci signifie que toutes les partitions partagent la totalité du disque. L'espace est alloué en fonction des besoins. Toutefois, il peut être nécessaire de limiter la taille d'une partition.

Pour cela, vous pouvez utiliser la commande « zfs set quota ». Dans l'exemple ci-dessous, la partition « /vHD/vPart1 » sera limitée à une taille de 10 Go.

```
zfs set quota=10G vHD/vPart1
```

La commande « zfs get quto » permet d'afficher les quotas attribués à une partition.

```
zfs get quota vHD/vPart1
```

Inversement, il peut être utile de réserver un minimum d'espace pour une partition et ainsi s'assurer dans l'espace ne viendra pas à manquer le jour où elle sera utilisée. Cette opération peut être réalisée avec la commande « zfs set réservation ». Dans l'exemple ci-dessous, la partition « vHD/vPart1 » se voit réserver 1G sur le disque.

```
zfs set reservation=1G vHD/vPart1
```

Cette approche est beaucoup plus souple que celle proposée par ZFS et la gestion de la taille des partitions ne nécessite pas de démonter le système de fichier.

Tapez la commande « zfs get all vHD/vPart1 » pour s'assurer que les modifications ont bien été appliquées.

NAME	PROPERTY	VALUE	
<b>SOURCE</b>			
vHD/vPart1	type	filesystem	-
vHD/vPart1	creation	ven mai 15 12:44 2009	-
vHD/vPart1	used	24,0K	-
vHD/vPart1	available	10,0G	-
vHD/vPart1	referenced	24,0K	-
vHD/vPart1	compressratio	1.00x	-
vHD/vPart1	mounted	no	-
vHD/vPart1	quota	10G	
<b>local</b>			
vHD/vPart1	reservation	1G	
<b>local</b>			
vHD/vPart1	recordsize	128K	
<b>default</b>			
vHD/vPart1	mountpoint	/vPart1	
<b>local</b>			
vHD/vPart1	sharenfs	off	
<b>default</b>			
vHD/vPart1	checksum	on	
<b>default</b>			
vHD/vPart1	compression	off	
<b>default</b>			
vHD/vPart1	atime	on	
<b>default</b>			
vHD/vPart1	devices	on	
<b>default</b>			
vHD/vPart1	exec	on	
<b>default</b>			
vHD/vPart1	setuid	on	
<b>default</b>			
vHD/vPart1	readonly	off	
<b>default</b>			
vHD/vPart1	zoned	off	
<b>default</b>			
vHD/vPart1	snapdir	hidden	
<b>default</b>			
vHD/vPart1	aclmode	groupmask	
<b>default</b>			

<i>vHD/vPart1</i>	<i>aclinherit</i>	<i>restricted</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>canmount</i>	<i>on</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>shareiscsi</i>	<i>off</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>xattr</i>	<i>on</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>copies</i>	<i>1</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>version</i>	<i>3</i>	-
<i>vHD/vPart1</i>	<i>utf8only</i>	<i>off</i>	-
<i>vHD/vPart1</i>	<i>normalization</i>	<i>none</i>	-
<i>vHD/vPart1</i>	<i>casesensitivity</i>	<i>sensitive</i>	-
<i>vHD/vPart1</i>	<i>vscan</i>	<i>off</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>nbmand</i>	<i>off</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>sharesmb</i>	<i>off</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>refquota</i>	<i>none</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>refreservation</i>	<i>none</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>primarycache</i>	<i>all</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>secondarycache</i>	<i>all</i>	
<i>default</i>			
<i>vHD/vPart1</i>	<i>usedbysnapshots</i>	<i>0</i>	-
<i>vHD/vPart1</i>	<i>usedbydataset</i>	<i>24,0K</i>	-
<i>vHD/vPart1</i>	<i>usedbychildren</i>	<i>0</i>	-
<i>vHD/vPart1</i>	<i>usedbyrefreservation</i>	<i>0</i>	

## 1.7 Création d'un Snapshot

Comme LVM, ZFS propose un mécanisme de SnapShots. La création des SnapShots est quasiment immédiate. Initialement, elle ne consomme pas d'espace disque supplémentaire au sein du pool. Toutefois, à mesure que les données contenues dans le jeu de données actif changent, le SnapShots consomme de l'espace disque en continuant à faire référence aux anciennes données.

Les instantanés ZFS présentent les caractéristiques suivantes :

- Persistance au cours des réinitialisations de système,
- Théoriquement, le nombre maximal d'instantanés est de  $2^{64}$  instantanés,
- Une seule opération, dite atomique, permet de créer rapidement des instantanés récursifs. Ceux-ci sont tous créés simultanément.

Il n'est pas possible d'accéder en écriture aux instantanés de volumes, mais ils peuvent être clonés, sauvegardés, restaurés, etc.

## 1.7.1 Création et destruction d'instantanés ZFS

La commande « `zfs snapshot` » permet de créer les SnapShots. Elle ne prend pour argument que le nom de l'instantané à créer. Le nom de l'instantané est spécifié comme suit :

*volume@snapshotname.*

Dans l'exemple suivant, un instantané de `media/music` nommé « `friday` » est créé.

```
zfs snapshot vHD/vPart1@friday
zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
vHD/vPart1@friday                   0      -   24,0K  -
```

Vous pouvez créer des instantanés pour tous les systèmes de fichiers descendants à l'aide de l'option `-r`. Exemple :

```
zfs snapshot -r vHD @now
zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
vHD@now                             0      -   29,3K  -
vHD/vPart1@friday                   0      -   24,0K  -
vHD/vPart1@now                      0      -   24,0K  -
vHD/vPart2@now                      0      -   24,0K  -
vHD/vPart3@now                      0      -   24,0K  -
vHD/vPart4@now                      0      -   24,0K  -
```

Les propriétés des instantanés ne sont pas modifiables (compression par exemple).

La commande « `zfs destroy` » permet de détruire les instantanés.

```
zfs destroy vHD/vPart1@friday
```

La destruction d'un jeu de données est impossible s'il existe des instantanés du jeu de données.

## 1.7.2 Restauration d'un instantané ZFS

La commande « `zfs rollback` » permet de supprimer toutes les modifications effectuées depuis la création d'un instantané spécifique. Le système de fichiers revient à l'état dans lequel il était lors de la prise de l'instantané. Par défaut, la commande ne permet pas de restaurer un instantané autre que le plus récent.

Pour restaurer un instantané précédent, tous les instantanés intermédiaires doivent être détruits. Vous pouvez détruire les instantanés précédents en spécifiant l'option `-r`.

Si le système de fichiers que vous souhaitez restaurer est actuellement monté, il doit être démonté, puis remonté. L'option `-f` force le démontage du système de fichiers, le cas échéant.

```
zfs rollback vHD/vPart1@friday
cannot rollback to 'vHD/vPart1@friday': more recent
snapshots exist
use '-r' to force deletion of the following snapshots:
vHD/vPart1@now

zfs rollback -r vHD/vPart1@friday
```

## 1.8 Présentation des clones ZFS

ZFS introduit la notion de clone. Un clone est un SnapShot accessible en écriture et dont le

contenu initial est similaire à celui du SnapShot à partir duquel il a été créé. Tout comme pour les instantanés, la création d'un clone est quasiment instantanée et ne consomme initialement aucun espace disque supplémentaire. Vous pouvez d'autre part créer un SnapShot d'un clone.

Les clones se créent uniquement à partir d'un SnapShot. Lors du clonage d'un SnapShot, une dépendance implicite se crée entre le clone et le SnapShot. Même en cas de création d'un clone à un autre emplacement de la hiérarchie, le SnapShot d'origine ne peut pas être détruit tant que le clone existe.

Un clone n'hérite pas des propriétés du jeu de données à partir duquel il a été créé. Les commandes « `zfs get` » et « `zfs set` » permettent d'afficher et de modifier les propriétés d'un jeu de données cloné.

### 1.8.1 Création d'un clone ZFS

Pour créer un clone, utilisez la commande « `zfs clone` » en spécifiant l'instantané à partir duquel créer le clone, ainsi que le nom du nouveau volume ou système de fichiers. Le nouveau volume ou système de fichiers peut se trouver à tout emplacement de la hiérarchie ZFS.

Dans l'exemple suivant, un nouveau clone appelé « `vHD/vPart1/bug123` » avec le même contenu initial que l'instantané « `vHD/vPart1@friday` » est créé.

```
zfs clone vHD/vPart1@friday vHD/vPart1/bug123
zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
vHD	1,00G	14,7G	29,3K	/media/zfs
vHD@now	0	-	29,3K	-
vHD/vPart1	24,0K	10,0G	24,0K	/vPart1
vHD/vPart1@friday	0	-	24,0K	-
vHD/vPart1/bug123	0	10,0G	24,0K	/vPart1/bug123
vHD/vPart2	24,0K	14,7G	24,0K	/media/zfs/vPart2
vHD/vPart2@now	0	-	24,0K	-
vHD/vPart3	24,0K	14,7G	24,0K	/media/zfs/vPart3
vHD/vPart3@now	0	-	24,0K	-
vHD/vPart4	24,0K	14,7G	24,0K	/media/zfs/vPart4
vHD/vPart4@now	0	-	24,0K	-

### 1.8.2 Destruction d'un clone ZFS

La commande « `zfs destroy` » permet de détruire les clones ZFS.

```
zfs destroy vHD/vPart1/bug123
```

Les clones doivent être détruits préalablement à la destruction de l'instantané parent.

## 1.9 Accroître la taille d'un disque

Si l'espace venait à manquer, vous pouvez agrandir votre disque sans avoir à faire appel à des technologies comme LVM. ZFS propose la commande « `zpool add` ». Ainsi, pour ajouter les disques « `/dev/sdg` » et « `/dev/sdh` » montés en miroir à « `vHD` », tapez:

```
zpool add vHD mirror /dev/sdg /dev/sdh
```

Pour ajouter un disque seul à un pool, tapez la commande:

```
zpool add vHD /dev/sdg
```



## 1.10 Gestion de la compression

En plus de LVM, ZFS supporte la compression native des partitions virtuelles. Vous pouvez commander la compression de la partition « vPart1 » du disque « vHD » avec la commande:

```
zfs set compression=on vHD/vPart2
```

Maintenant, rendez-vous dans le point de montage de « vHD » c'est à dire « /media/zfs » et créez y un fichier remplis de zéros.

```
cd /media/zfs  
dd if=/dev/zero of=uncompressetest bs=128k count=100
```

Le fichier est crée à la vitesse de 53mb/s.

Rendez-vous ensuite dans la partition compressée « music » et créez y un fichier identique.

```
cd /media/zfs/vPart2  
dd if=/dev/zero of=compressetest bs=128k count=100
```

Le fichier est écrit à la vitesse de 44.8mb/s. L'impact de la compression sur la vitesse d'écriture est assez faible.

Si vous demandez le listage des deux dossiers, vous obtiendrez un résultat identique:

```
ls -ls /media/zfs  
ls -ls /media/zfs/vPart1
```

Par contre, si vous lancez la commande « zfs list », vous constaterez que le fichier « uncompressedtest » occupe effectivement 12,5mb du disque « media/zfs » alors que « compressedtest » n'occupe pratiquement pas d'espace (27,3K) (le taux de compression est assez exceptionnel puisque que le fichier est plein de zéros).

```
zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
vHD	1,01G	14,6G	12,5M	/media/zfs
vHD/vPart1	24,0K	10,0G	24,0K	/vPart1
vHD/vPart2	27,3K	14,6G	27,3K	/media/zfs/vPart2
vHD/vPart3	24,0K	14,6G	24,0K	/media/zfs/vPart3
vHD/vPart4	24,0K	14,6G	24,0K	/media/zfs/vPart4

## 1.11 Mesurer les performances

La commande « zpool iostat -v vHD » permet de s'étudier la répartition des données entre les disques et les performances moyennes de chaque disque physique. En effet, ZFS mesure en permanence les taux de lecture/écriture de chaque et adapte automatiquement son comportement de façon à offrir les meilleures performances en fonctions de la taille de données à lire/écrire.

```
capacity operations bandwidth
```

pool	used	avail	read	write	read	write
vHD	19,2M	23,9G	0	2	121	31,0K
raidz1	19,2M	23,9G	0	2	121	31,0K
sdd	-	-	0	1	233	16,7K
sde	-	-	0	1	233	16,7K
sdf	-	-	0	1	233	16,7K

## 1.12 Suppression d'une partition

La suppression de la partition « vPart1 » du disque virtuel nommé « vHD » est aussi simple que:

```
zpool destroy vHD/vPart1
```

## 1.13 Suppression d'un disque

La suppression d'un disque virtuel nommé « media » est aussi simple que:

```
zpool destroy vHD
```

## 1.14 Récupération d'un pool ZFS

Il est recommandé de conserver en sécurité une copie des informations affichées par « zpool status » et ainsi pouvoir reconstruire le système ZFS manuellement.

Si un pool est détruit, il est possible de tenter sa récupération automatique.

La commande « zpool import » tente la reconstruction automatique d'un pool détruit.

```
zpool destroy vHD
zpool import -D
pool: media
  id:
  state: DEGRADED (DESTROYED)
  status: One or more devices are missing from the system.
  action: The pool can be imported despite missing or damaged
devices. The
  fault tolerance of the pool may be compromised if
imported. The
  pool was destroyed, but can be imported using the '-
Df' flags.
  see: http://www.sun.com/msg/ZFS-8000-2Q
  config:
```

<i>vHD</i>	<i>DEGRADED</i>
<i>raidz</i>	<i>ONLINE</i>
<i>vPart1</i>	<i>ONLINE</i>

Vous pouvez lancer l'importation forcée du disque dégradé

```
zpool import -Df vHD
zpool status -x
  pool: vHD
  state: DEGRADED
  status: One or more devices could not be opened. Sufficient
replicas exist for
  the pool to continue functioning in a degraded
state.
  action: Attach the missing device and online it using 'zpool
online'.
  see: http://www.sun.com/msg/ZFS-8000-D3
  scrub: resilver completed after 0h0m with 0 errors on Thu
Aug 28 10:01:48 2008
```

```

config:

      NAME                STATE      READ WRITE CKSUM
      vHD                 DEGRADED   0     0     0
      raidz               ONLINE     0     0     0
      vPart1              ONLINE     0     0     0
errors: No known data errors

```

Repassez les disques en ligne

```

# zpool online media c1t2d0
Bringing device music online

# zpool status -x
all pools are healthy

```

Il est important de noter que la structure de ZFS n'est pas liée à l'architecture matérielle. Ainsi, il est possible d'exporter un ZFS, de placer dans un nouveau matériel et de le réimporter.

## 1.15 Les distributions utilisant ZFS

OpenSolaris<sup>5</sup> est un projet de système d'exploitation libre commandité par Sun Microsystems. Il est initialement basé sur un morceau du code source de Solaris dont il conserve le noyau, le support réseau, les bibliothèques et les commandes. La licence utilisée est la CDDL 1.0, proche de la MPL. Cette licence est libre au sens de la FSF mais incompatible avec la GPL. OpenSolaris est installé sur un système ZFS.

NexentaStor<sup>6</sup> est une distribution orientée vers le stockage de données. En cela, elle ressemble un peu à FreeNAS<sup>7</sup> ou OpenFiler<sup>8</sup> sauf que le système de fichiers utilisés est ZFS. Il existe une version gratuite mais limitée à 1 Téraoctets. L'administration des espaces de stockage est réalisée au travers d'une interface WEB extrêmement conviviale et rapide à prendre en main. On retrouve tous les avantages de ZFS avec la possibilité d'exporter les volumes vers le réseau au travers des protocoles NFS/CIFS/RSYNC/FTP.

## Conclusion

L'objectif de cet article n'est de vanter les qualités de ZFS sur mdadm/LVM car il ne faut pas oublier que ZFS est un produit en développement sous Linux. Cette technologie, même si elle a été conçue pour garantir la pérennité des données, ne saurait être utilisée dans un environnement de production. A l'inverse, mdadm et LVM assurent la sécurité d'un grand nombre de serveurs tournant sous Linux.

De plus ZFS est beaucoup plus gourmand en mémoire vive que mdadm/LVM.

Toutefois, la lecture de cet article met clairement en la simplification des tâches d'administration des espaces de stockages que vont apporter les nouveaux systèmes de fichiers. Alors, ZFS est-il l'avenir du système de fichiers sous Linux? Rien n'est moins sûr car déjà BTRFS<sup>9</sup>, un projet développé par Oracle sous licence GPL, reprends les grandes lignes

5 <http://www.opensolaris.com/>

6 <http://www.nexenta.com/corp/>

7 <http://www.freenas.org>

8 <http://www.openfiler.com/>

9 [http://btrfs.wiki.kernel.org/index.php/Main\\_Page](http://btrfs.wiki.kernel.org/index.php/Main_Page)

de ZFS, les améliore et les adapte aux monde Linux<sup>10</sup>.

	ZFS	BTRFS
Copy on Write	oui	oui
Snapshots	oui	oui
Performance degradation at near 98-100% disk usage	oui	oui
Block level compression	oui	Option de montage
Disk encryption	En cours	Prévu
Online resizing	non	oui
Online defragmentation	non	oui
Write checksums	oui	oui
RAID	0,1,10, 5 (raidz), 6 (raidz2)	0, 1, 10
ACL	oui	oui
Direct IO	oui	Writes: oui Reads: non mais prévu
Quotas	oui	oui

---

<sup>10</sup> <http://www.linux-mag.com/id/7308>