

Small Scale ZFS

Tobias Rehbein / blabber

13. März 2012

*Warum hast du ZFS auf deinem Laptop? ZFS macht doch
nur mit vielen Platten Spaß.*

– muzy

Begriffe

`vdev` ein virtuelles Device: disk, file, mirror, raidz, raidz2, raidz3, spare, log, cache.

Begriffe

vdev ein virtuelles Device: disk, file, mirror, raidz, raidz2, raidz3, spare, log, cache.

pool Speicherpool der aus mehreren vdevs besteht.

Begriffe

vdev ein virtuelles Device: disk, file, mirror, raidz, raidz2, raidz3, spare, log, cache.

pool Speicherpool der aus mehreren vdevs besteht.

dataset ein Pool beinhaltet mehrere datasets. Ein dataset ist ein Dateisystem, zvol (virtuelles Blockdevice), snapshot oder clone.

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets
- copy-on-write (Transaktionen)
- snapshots und clones

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets
- copy-on-write (Transaktionen)
- snapshots und clones
- Checksummen (Daten und Metadaten)
- Selbstheilung (Redundanzgruppen)

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets
- copy-on-write (Transaktionen)
- snapshots und clones
- Checksummen (Daten und Metadaten)
- Selbstheilung (Redundanzgruppen)
- Selbstheilung (copies Attribut)
- dataset Kompression

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets
- copy-on-write (Transaktionen)
- snapshots und clones
- Checksummen (Daten und Metadaten)
- Selbstheilung (Redundanzgruppen)
- Selbstheilung (copies Attribut)
- dataset Kompression
- Deduplikation
- riesige Datenmengen (256 Quadrillionen Zettabytes, $256 * 10^{36}$ bytes, 128bit Adressen)

Features

- Storagepools (integrierter Volume Manager)
- dynamisch wachsende datasets
- copy-on-write (Transaktionen)
- snapshots und clones
- Checksummen (Daten und Metadaten)
- Selbstheilung (Redundanzgruppen)
- Selbstheilung (copies Attribut)
- dataset Kompression
- Deduplikation
- riesige Datenmengen (256 Quadrillionen Zettabytes, $256 * 10^{36}$ bytes, 128bit Adressen)

Nachteile

- Performance
- Speicherhunger

Nachteile

- Performance
- Speicherhunger
- Ich habe keinen Plan was passiert wenn mal ein Fehler auftritt.

zfs list

```
> zfs list -o name,referenced,available,mountpoint
NAME                                REFER    AVAIL    MOUNTPOINT
tank0                                25,5M    217G    legacy
tank0/home                           12,1G    217G    /home
tank0/music                          42,9G    217G    /music
tank0/tmp                             1,48M    217G    /tmp
tank0/usr                             5,34G    217G    /usr
tank0/usr/local.texlive              3,19G    217G    /usr/local/t
tank0/var                             148M     217G    /var
```

Attribute

```
zfs set compression=on tank0/tmp
```

Attribute

```
zfs set compression=on tank0/tmp  
zfs set copies=3 tank0/music
```


snapshots

```
zfs snapshot -r tank0@test
```

snapshots

```
zfs snapshot -r tank0@test
```

```
zfs rollback tank0@test
```

```
zfs rollback tank0/usr@test
```

snapshots

```
zfs snapshot -r tank0@test
```

```
zfs rollback tank0@test
```

```
zfs rollback tank0/usr@test
```

```
zfs destroy -r tank0@test
```

snapshots

```
zfs snapshot -r tank0@test
```

```
zfs rollback tank0@test
```

```
zfs rollback tank0/usr@test
```

```
zfs destroy -r tank0@test
```

```
zfs diff tank0@test tank0
```

Datensicherung (voll)

```
zfs snapshot -r tank0@snap1
```

Datensicherung (voll)

```
zfs snapshot -r tank0@snap1
```

```
zpool import extern
```

Datensicherung (voll)

```
zfs snapshot -r tank0@snap1
```

```
zpool import extern
```

```
zfs send tank0@snap1 | \  
zfs receive extern/backup@snap1
```

```
zfs send tank0/usr@snap1 | \  
zfs receive extern/backup/usr@snap1
```

Datensicherung (voll)

```
zfs snapshot -r tank0@snap1
```

```
zpool import extern
```

```
zfs send tank0@snap1 | \  
zfs receive extern/backup@snap1
```

```
zfs send tank0/usr@snap1 | \  
zfs receive extern/backup/usr@snap1
```

```
zpool export extern
```


Datensicherung (inkrementell)

```
zfs snapshot -r tank0@snap2
```

```
zpool import extern
```

```
zfs send -i snap1 tank0@snap2 | \  
zfs receive extern/backup@snap2
```

```
zfs send -i snap1 tank0/usr@snap2 | \  
zfs receive extern/backup/usr@snap2
```

```
zpool export extern
```

chroot klonen

```
zfs snapshot -r tank0@sandbox
```

chroot klonen

```
zfs snapshot -r tank0@sandbox
```

```
zfs clone tank0@sandbox tank0/sandbox
```

```
zfs clone tank0/var@sandbox tank0/sandbox/var
```

```
zfs clone tank0/usr@sandbox tank0/sandbox/usr
```

```
zfs clone tank0/tmp@sandbox tank0/sandbox/tmp
```

chroot klonen

```
zfs snapshot -r tank0@sandbox
```

```
zfs clone tank0@sandbox tank0/sandbox
```

```
zfs clone tank0/var@sandbox tank0/sandbox/var
```

```
zfs clone tank0/usr@sandbox tank0/sandbox/usr
```

```
zfs clone tank0/tmp@sandbox tank0/sandbox/tmp
```

```
zfs set mountpoint=/sandbox tank0/sandbox
```

chroot klonen

```
zfs snapshot -r tank0@sandbox
```

```
zfs clone tank0@sandbox tank0/sandbox
```

```
zfs clone tank0/var@sandbox tank0/sandbox/var
```

```
zfs clone tank0/usr@sandbox tank0/sandbox/usr
```

```
zfs clone tank0/tmp@sandbox tank0/sandbox/tmp
```

```
zfs set mountpoint=/sandbox tank0/sandbox
```

```
mount -t devfs devfs /sandbox/dev
```

chroot klonen

```
zfs snapshot -r tank0@sandbox
```

```
zfs clone tank0@sandbox tank0/sandbox
```

```
zfs clone tank0/var@sandbox tank0/sandbox/var
```

```
zfs clone tank0/usr@sandbox tank0/sandbox/usr
```

```
zfs clone tank0/tmp@sandbox tank0/sandbox/tmp
```

```
zfs set mountpoint=/sandbox tank0/sandbox
```

```
mount -t devfs devfs /sandbox/dev
```

```
chroot /sandbox
```

Zukunft

- Sun ist nicht mehr, Oracle hat ZFS Versionen ab v28 nicht mehr im Quellcode veröffentlicht
 - btw, Oracle entwickelt auch BTRFS
 - ein internes Memo soll aussagen, das zu Solaris Releases die entsprechenden ZFS Quellen veröffentlicht werden.

Zukunft

- Sun ist nicht mehr, Oracle hat ZFS Versionen ab v28 nicht mehr im Quellcode veröffentlicht
 - btw, Oracle entwickelt auch BTRFS
 - ein internes Memo soll aussagen, das zu Solaris Releases die entsprechenden ZFS Quellen veröffentlicht werden.
- momentan Duldungsstarre
 - ZFS fault monitoring and management daemon
 - Grow Mounted Filesystems
 - backports

Zukunft

- Sun ist nicht mehr, Oracle hat ZFS Versionen ab v28 nicht mehr im Quellcode veröffentlicht
 - btw, Oracle entwickelt auch BTRFS
 - ein internes Memo soll aussagen, das zu Solaris Releases die entsprechenden ZFS Quellen veröffentlicht werden.
- momentan Duldungsstarre
 - ZFS fault monitoring and management daemon
 - Grow Mounted Filesystems
 - backports
- Fork falls Sun neue Versionen nicht veröffentlicht (mit illumos?)

Zukunft

- Sun ist nicht mehr, Oracle hat ZFS Versionen ab v28 nicht mehr im Quellcode veröffentlicht
 - btw, Oracle entwickelt auch BTRFS
 - ein internes Memo soll aussagen, das zu Solaris Releases die entsprechenden ZFS Quellen veröffentlicht werden.
- momentan Duldungsstarre
 - ZFS fault monitoring and management daemon
 - Grow Mounted Filesystems
 - backports
- Fork falls Sun neue Versionen nicht veröffentlicht (mit illumos?)
- BTRFS in FreeBSD?
 - eher nicht (GPL)
 - dann doch eher HAMMER (DragonflyBSD)

*Warum hast du ZFS auf deinem Laptop? ZFS macht doch
nur mit vielen Platten Spaß.*

– muzy

*Warum hast du ZFS auf deinem Laptop? ZFS macht doch
nur mit vielen Platten Spaß.*

– muzy

Weil ich es kann... und weil es doch Spaß macht.

– blabber

Ende

Danke für eure Aufmerksamkeit.