

ZFS EINFÜHRUNG

INHALT

1. Geschichte
2. Pools
3. Datasets
 - Volumes
4. Snapshots
6. Caches
7. Besondere Features
 - Deduplikation
 - Kompression

NICHT BEHANDELT

- Austausch von Platten
- Eingebautes Sharing via NFS/CIFS/iSCSI
- Root auf ZFS
- Verschlüsselung

GESCHICHTE

GESCHICHTE

- 2001: Beginn der Entwicklung bei Sun für Solaris
- 2005: Erster Public Release, Code wird Open Source
- 2006-2010: Portierung auf Linux, MacOS, FreeBSD
- Ab 2010: Oracle macht ZFS zu Closed Source, offene Entwicklung weiter als OpenZFS

FEATURES

POOLS

- ZFS ist Dateisystem und Volume Manager gleichzeitig
- Basis: Blockdevices
- Blockdevices können zu vdevs zusammengefasst werden
- Blockdevices und vdevs bilden zusammen einen Storage Pool
- Viele Einstellungen schon hier möglich, werden runter vererbt

DEMO

Speaker notes

- Einfachen Stripe-Pool anlegen: `zpool create -oashift=12 -R/mnt tank /dev/sd{b,c,d,e}`
- "Raid-10" Pool: `zpool create -oashift=12 -R/mnt tank mirror /dev/sd{b,c} mirror /dev/sd{d,e}`
- RAID-Z ("RAID-5") Pool: `zpool create -oashift=12 -R/mnt tank raidz /dev/sd{b,c,d,e}`
- Allgemeine Einstellungen:
 - `zfs get compression tank`
 - `zfs get dedup tank`
 - `zfs get all tank`

DATASETS

- Datasets entsprechen eigentlichen Dateisystemen
- Können verschachtelt werden
- Settings eines Datasets werden vererbt
- Standardmäßig so groß wie das übergeordnete Dataset
- Quotas können gesamt, pro User oder pro Gruppe vergeben werden

VOLUMES

- Spezielle Datasets
- Verhalten sich wie Blockdevices
- Gerne genutzt für VM Images, Swap Space, ...

SNAPSHOTS

- Werden durch CoW ermöglicht
- Halten den exakten Stand fest
- Sind keine Kopie sondern der eingefrorene Zustand
- Können auf andere Rechner, Pools, Datasets übertragen werden

Speaker notes

CoW

Copy on Write, neue Daten überschreiben nicht den vorhandenen Block sondern werden in einen neuen Block geschrieben und Referenzen auf den alten werden bereinigt.

DEMO

Speaker notes

- Dataset anlegen: `zfs create tank/zfsintro`
- Dataset mit Quota: `zfs create -o quota=1G tank/zfsintro/data`
- Quota ändern: `zfs set quota=5G tank/zfsintro`
- Quota für User: `zfs set userquota@backup=1G tank/zfsintro`
- Volume anlegen: `zfs create -b 4096 -V 2G tank/swap`
- Snapshot anlegen: `zfs snapshot tank/zfsintro/data@snap1`
- Snapshot in `$(DATASET)/.zfs/snapshot`, Readonly

CACHES: RAM

- ARC Read-Cache, Default: zwischen $\max(32\text{MB}, 1/32 \text{ RAM})$ und $1/2 \text{ RAM}$
- Write-Cache, schreibt zumindest alle 5-30 Sekunden auf die Platte

CACHES: DISKEN

- Pro Pool
- Schnelle Medien (SSD, M2, ...)
- Lesen: L2ARC (Level 2 ARC), kann auch Dedup-Tabelle halten
- Schreiben: SLOG als Schreib-Cache
 - Sollte gespiegelt sein, sonst Risiko des Datenverlusts
- Können im Betrieb hinzugefügt und entfernt

Speaker notes

ARC

Adaptive Replacement Cache

BESONDERE FEATURES

DEDUPLIKATION

- Prüfsumme über Blöcke, wenn gleich wird nur 1 Kopie gespeichert
- Keine rückwirkende Änderung
- Braucht viel RAM (👍 ·π: 5GB pro 1TB Daten)
- Sinnvoll für Archive

KOMPRESSION

- Wie Dedup auf Block-Level
- Wie Dedup keine rückwirkende Änderung
- Unterschiedliche Verfahren je nach Anforderung vorhanden
- *Sehr* sinnvoll für Datasets mit viele Text-Dateien (Logs, Mail-Spool, ...)

LINKS

LINKS

ZFS Demo VM

<https://cloud.usrspace.at/s/eB4ifqQBNtLz322>

Wikipedia zu ZFS

<https://en.wikipedia.org/wiki/ZFS>

ZFS Administration (Aaron Toponce)

<https://pthree.org/2012/12/04/zfs-administration-part-i-vdevs/>

Präsentation ist [CC-BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

FRAGEN?