Datenträgerverwaltung unter FreeBSD

Robert Clausecker
[Be]LUG e. V.

GEOM

Was ist GEOM?

- Framework zur Datenträgerverwaltung
- verschachtelbar mit beliebiger Topologie
- weitestgehend vollautomatisch

Begrifflichkeiten

Klasse

Ein Modul, dass eine bestimmte Art von Übersetzung vornimmt

Geom

Eine Instanz einer Klasse

Provider

Gerätedatei in /dev, durch die ein Geom seine Dienste anbietet

Consumer

Geom, das an einen Provider angehängt ist

Datenquellen

- GEOM-Klasse **disk**
  - Festplatten **da**(4), **ada**(4), **nda**(4), ...
  - Optische Medien **cd**(4)
  - RAM-Disks **md**(4)
  - Raidcontroller **mfi**(4), ...
  - iSCSI-Targets **iscsi**(4)
  - ZFS-Volumina als zvol/...

RAM-Disks, Loopback

`mdconfig -a -s` *`size`* `[-noSuxyL]`

- legt **md**(4)-Gerät für RAM-Disk der Größe *`size`* an

`mdconfig -a -f` *`file`* `[-nosSuxyL]`

- legt **md**(4)-Gerät für *`file`* an

`mdconfig -d -u` *`unit`*

- zerstört *`unit`*

`mdconfig -r -u` *`unit`* `-s` *`size`*

- ändert die Größe von *`unit`*

`mdconfig -l [-v]`

- zeigt alle **md**(4)-Geräte an

Allgemeine Befehle

geom *class* help

zeigt alle verfügbaren Befehle an

geom *class* list

zeigt alle Geoms dieser Klasse an

geom *class* status

zeigt den Zustand aller Geoms der Klasse an

geom -p *provider*

zeigt Informationen zu *provider* an

geom -t

zeigt GEOM-Hierarchie als Baum an

- viele Klassen haben **g*class***(8) kurz für **geom *class***

Partitionierung

- GEOM-Klasse **part**, siehe auch **gpart**(8)
  - foo0 → foo0a, foo0p0, foo0s0, …

    gpart create -s *scheme provider*
    - `scheme`: `gpt`, `mbr`, `bsd`, `ldm`, …

    gpart add -t *type* -s *size provider*
    - `type`: `freebsd`, `linux`, `fat32`, `ntfs`, …

    gpart show *provider*
    - Partitionstabelle anzeigen

    gpart delete -i *index provider*

    gpart destroy *provider*

Verschlüsselung

- GEOM-Klasse **eli**, siehe auch **geli**(8)
  - foo → foo.eli
  - **`geli init [-aBbdegiJKlPsTVv]`** ***`provider`***
    - `-a algo` – Integritätsprüfung anschalten
      - z. B. mit `-a HMAC/SHA256`
    - `-b` Partition beim booten entschlüsseln
    - `-g` Bootloader von dieser verschlüsselten Partition laden
    - `-J passfile` – Password aus Datei oder Konsole (`-J -`)
    - `-K keyfile` – Schlüssel aus Datei oder Konsole (`-K -`)
    - `-s Sektorgröße` – Sektorgröße wählen
    - `-T` gelöschte Sektoren nicht an die Platte melden

Verschlüsselung

- GEOM-Klasse **eli**, siehe auch **geli**(8)
  - **`geli attach [-Cdjknprv]` *`provider`***
    - `-j passfile` – Password aus Datei oder Konsole (`-j -`)
    - `-k keyfile` – Schlüssel aus Datei oder Konsole (`-k -`)
  - **`geli detach [-fl]` *`provider`***
    - `provider` aushängen
  - **`geli onetime [-adelsT]` *`provider`***
    - wie `geli attach`, aber mit zufälligem Einmalschlüssel (z. B. für Swap)

Stripes

- GEOM-Klasse **stripe**, siehe auch **gstripe**(8)
  - foo → stripe/bar
  - **`gstripe label [-hsv] name prov ...`**
    - erzeuge Stripe *name* aus *prov* ...
    - Metadaten auf den Providern ermöglichen automatische Erkennung beim Boot
  - **`gstripe clear [-v] prov ...`**
    - lösche Stripe-Metadaten auf *prov*

Spiegel

- GEOM-Klasse **mirror**, siehe auch **gmirror**(8)
  - foo → mirror/bar
  - **`gstripe mirror [-bFhsv] name prov ...`**
    - erzeuge Mirror *name* aus *prov* ...
    - Metadaten auf den Providern ermöglichen automatische Erkennung beim Boot
  - **`gstripe clear [-v] prov ...`**
    - lösche Stripe-Metadaten auf *prov*
  - **`gstripe insert [-hipv] name prov ...`**
  - **`gstripe remove [-v] name prov ...`**

Mehr zu RAID/Verschlüsselung

- **geom_linux_lvm**(8)
- **graid**(8) für RAID-BIOS-Unterstützung
- **graid3**(8), **graid5**(8) aus sysutils/graid5
- **gvinum**(8), **gconcat**(8)
- **gbde**(8) alternativ zu **geli**(8)
- **gshsec**(8) für Shared Secrets

Spezielle Anwendungen

- **glabel**(8) für feste Providernamen
- **geom_uzip**(4) mit **mkuzip**(8) für komprimierte Dateisystemabbilder
- **gmultipath**(8) für mehrer Pfade zum gleichen Gerät
- **ggatel**(8), **ggatec**(8) reichen Geoms durchs Netzwerk durch
- **gmountver**(8) für Überbrückung sporadischer Geräteausfälle
- **gvirstor**(8) für Thin Provisioning

Der Bootprozess

FreeBSD booten

- zwei/dreistufiger Bootprozess
  1. BIOS lädt Bootstrapcode **boot**(8), **uefi**(8), oder **gptboot**(8)
  2. Bootstrapcode lädt **loader**(8) oder direkt Kernel
  3. **loader**(8) lädt den Kernel
- die Details hängen von der Konfiguration ab
- Installation des Bootstrapcodes mit **gpart**(8)

```
gpart bootcode -b bootcode provider
```

FreeBSD booten per MBR

BSD-Label per MBR

```
gpart bootcode -b /boot/boot foo
```

BSD-Label in DOS-Partitionstabelle per MBR

```
gpart create -s bsd foos1
gpart set -a active -i 1 foo
gpart bootcode -b /boot/boot0 foo
gpart bootcode -b /boot/boot  foos1
```

GPT-Label per MBR

```
gpart bootcode -b /boot/pmbr foo
gpart add -b 40 -s 472 -t freebsd-boot foo
gpart bootcode -p /boot/gptboot -i 1 foo
```

FreeBSD booten per UEFI

DOS-Partitionstabelle per UEFI

```
gpart add -t efi -s 1600 foo
cat /boot/boot1.efifat >/dev/foos1
```

GPT-Partitionstabelle per UEFI

```
gpart add -t efi -s 1600 foo
cat /boot/boot1.efifat >/dev/foop1
```

ZFS

Was ist ZFS?

- Copy-on-Write (CoW)-Dateisystem aus dem Hause SUN Microsystems, weiterentwickelt als OpenZFS
- ersetzt (fast) hier genannten Komponenten durch eine vollautomatische Lösung
  - weite Teile von **geom**(4)
  - **fstab**(5)
  - **mksnap_ffs**(8) Snapshots

Begrifflichkeiten

Zpool

Sammlung von Datenträgern, die Speicher und Redundanz für Datensätze bilden

Datensatz (dataset)

Entität in einem Zpool, die durch einen Pfad eindeutig bezeichnet ist. Kann ein *Dateisystem*, ein *Volumen*, oder ein *Snapshot* sein

Volumen (volume)

Datensatz, der eine virtuelle Festplatte darstellt

Snapshot

Datensatz, der den momentanen Zustand eines anderen Datensatzes festhält

Zpools

`zpool create` *`pool vdev`* `...`

- Erzeugt Zpool *pool* bestehend aus *vdev* ...
- Jeder Zpool ist ein Stripe aus mehreren Vdevs,
- mögliche Vdevs:
  - GEOM-Provider (Festplatte, Partition, ...) oder Datei
  - Spiegel **mirror** ***dev dev ...***
  - RAID-Z **raidz** ***dev dev dev ...*** (auch **raidz1**, **raid2, raidz3**)
  - Hotspare **spare** ***dev***, benötigt **zfsd**(8)
  - Lesecache **cache** ***dev***
  - Log/Schreibcache **log** ***dev***

Datensätze

`zfs create [-pou] pool/filesystem`

- erzeugt Dateisystem *filesystem* in *pool*
- Platz wird nach Bedarf aus dem Pool genommen

`zfs create [-bpos] -V size pool/volume`

- erzeugt Volumen der Größe *size* in *pool*
- maximal benötigter Platz wird reserviert
  - `-s` unterlasse Reservierung (Thin Provisioning)

`zfs destroy [-fnpRrv] pool/dataset`

`zfs list [-dHoprSst] [dataset ...]`

Snapshots

- Zugriff auf Snapshots über *mountpoint*/.zfs/snapshot

  `zfs snapshot [-r] `*`dataset@snap`

  - legt Snapshot *snapshot* von *pool/dataset* an

  `zfs rollback [-rRf] `*`dataset@snap`

    setzt *dataset* auf den Stand von *snap* zurück

weitere Features

- Snapshots senden/empfangen: **zfs send**/**receive**
- Rechte delegieren: **zfs allow**
- transparente Kompression
- Platzstatistiken und Quotas pro Nutzer/Gruppe
- Online Konsistenzprüfung mit **zpool scrub**
- ...
- einfach mal **zpool**(8) und **zfs**(8) lesen