BTRFS

Das Linux Filesystem der nächsten Generation

1. btrfs

Was ist btrfs?

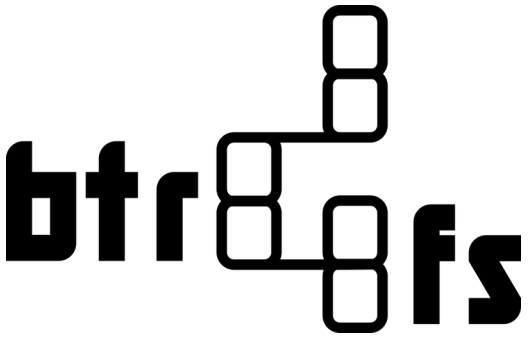
- Modernes Filesystem mit erweiterten Features
- Ziel: Bessere Skalierbarkeit und Robustheit als traditionelle Linux Dateisysteme

- Oracle 2007 (Chris Mason)
- Im Linux Kernel ab 2.6.29 (März 2009)
- Ab Kernel 3.16 (Oktober 2014) 'stable'
  (production-ready mit Einschränkungen)
- Andrew Morton sieht *btrfs* langfristig als Standard-FS für Linux

- btrfs auch als 'ZFS for Linux'
  (ZFS Sun Microsystems ab 2001, leider Lizenzprobleme mit Linux)

Wie spricht man das eigentlich aus?

- Be-Te-Er-Eff-Es
- Better File System
- Butter File System
- B-Tree File System



B-Trees zentrale Datenstruktur für btrfs:

- Root Tree (Wurzel)
- FS Tree (Subvolume)
- Extent Tree (Daten / Metadaten)
- Chunk Tree (Mapping)
- Dev Tree (Block devices)
- Checksum Tree, Log Tree, Reloc Tree...

<=> FAT File System, File Allocation Table als verkettete Liste

Features:

- Maximale Größe File / Volume 16 EiB
- Umwandlung von ext3 / ext4 nach btrfs
- Copy-On-Write
- Kompression
- Prüfsummen und Fehlerkorrektur
- Subvolumes
- Atomic Snapshots
- RAID integriert
- Online Vergrößern / Verkleinern
- Online Hinzufügen / Entfernen von Block Devices

Btrfs Dateisystem anlegen:

```
mkfs.btrfs <options> <device> [<device> ...]

-L|--label <name>   # assigns a FS label
-f|--force          # forces FS creation
-K|--nodiscard      # no TRIM at FS creation

-d|--data raid[0|1|5|6|10] | single
-m|--metadata raid[0|1|5|6|10] | single | dup
```

Btrfs Dateisystem einhängen:

```
mount -t btrfs -o <options> <device> <dir>

compress[=zlib|lzo|no]   # Compression type
subvol=<path>            # Mount subvolume
subvolid=<id>            # Mount subvolume by ID
ssd|nossd                # Override SSD detect
discard                  # Enable SSD TRIM
skip_balance             # No balancing at mount
degraded                 # Mount with missing
                           devices in RAID
```

Btrfs Dateisystem verwalten:

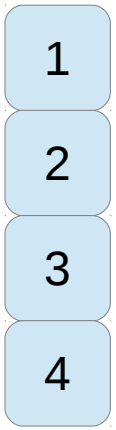
```
btrfs <command> <subcommand> <...>

filesystem | fi     # Filesystem-level control
subvolume  | sub    # Subvolume/Snapshot control
device     | dev    # Multi device management
balance    | bal    # FS rebalancing
check      | ch     # FS checking
scrub      | scr    # FS scrub
```

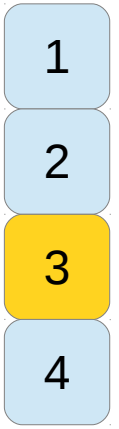
Btrfs Filesystem Kommandos:

```
btrfs fi <subcommand> <...>

df <path>                       # Show space usage
show <path>                     # Show FS info
sync <path>                     # Force sync
defragment -r <file|dir> # Defragment file/dir
resize <size>|max <path> # Resize the FS
label <dev> <newlabel>   # Assign new label
```

Overwrite

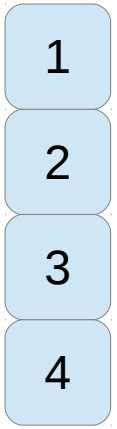
(meiste FS)

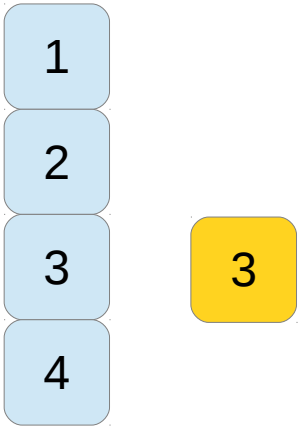




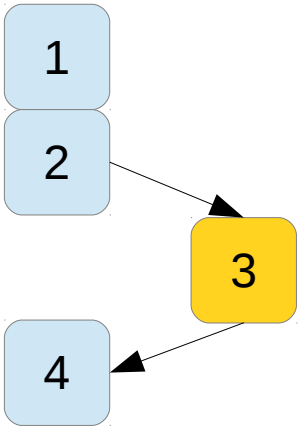
Copy-On-Write

(btrfs, ZFS)





1) Schreibe neuen Block an freie Stelle



2) Aktualisiere Metadaten

Copy-On-Write erhöht Datenintegrität z.B. bei Crash,

aber: Problem Fragmentierung!

Grosse Files mit häufigen kleinen Änderungen -> Performanceverlust (Datenbanken, Virtual Machine Images)

Copy-On-Write deaktivieren:

a) Fürs gesamte FS: Mount-Option `-o nodatacow`

b) Für ein einzelnes File: > `chattr +C <file>`

Btrfs Subvolumes

- Teilen Eigenschaften von Directories und Block Devices
- Jedes Subvolume -> Eigener btrfs B-Tree
- Ein Subvolume hat Namen und ID
- Sehen wie Directories aus
- Können direkt wie Block Device eingehängt werden
- Snapshot: Sonderform eines Subvolumes
  - Abbild zu einem bestimmten Zeitpunkt
  - Speichert nur geänderte Dateien; unveränderte per Zeiger

Btrfs Subvolume Kommandos:

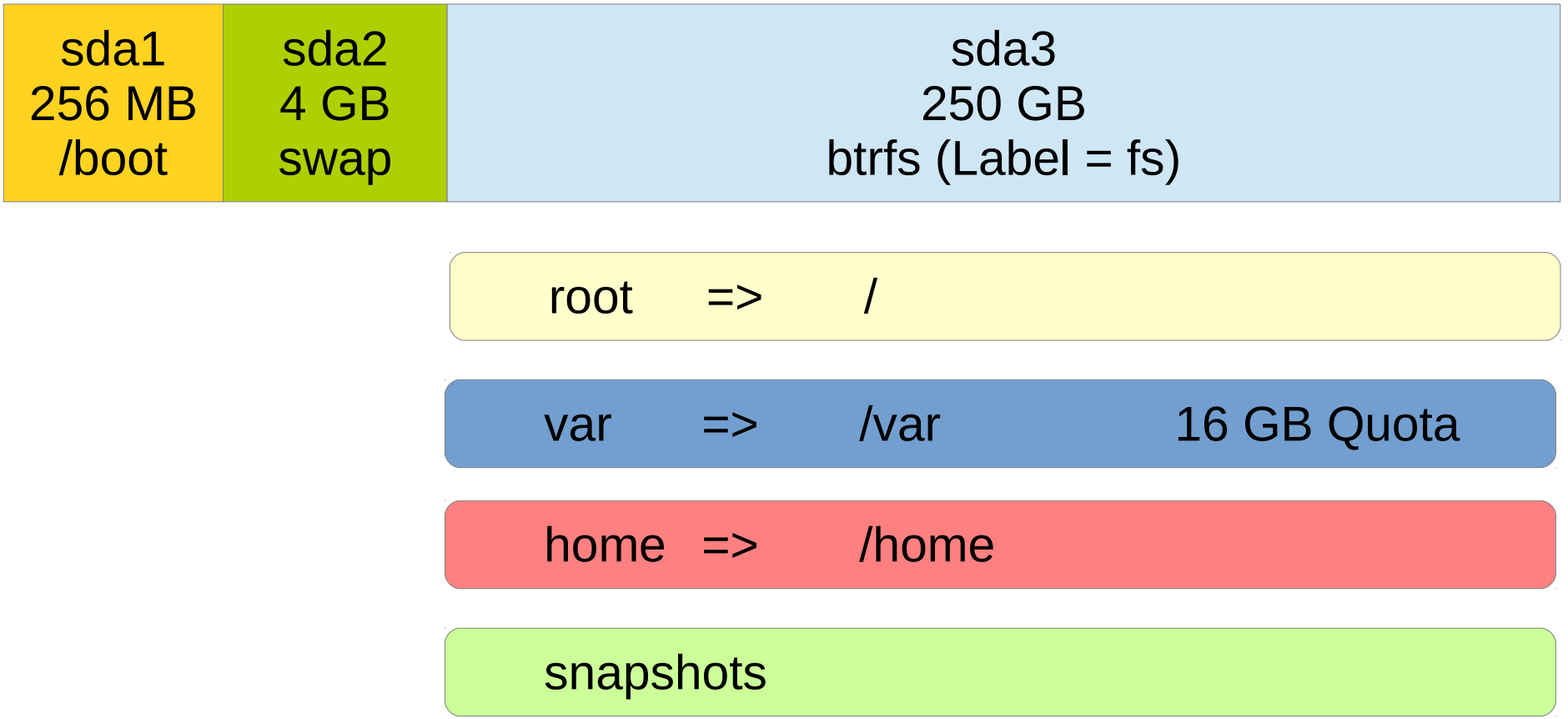
```
btrfs subvolume <subcommand> <...>

create <name>                 # Create a subvolume
delete <subvol>               # Delete a subvolume
list <path>                   # Show subvols in path
show <name>                   # Show subvol info
snapshot <source> <dest> # Create snapshot
set-default <id> <path>  # Set mount default
```

Klassisches Filesystem Layout:

- Trennung in Partitionen
- Feste Partitionsgröße:
  - Vergeudung Speicherplatz
  - Platz kann zu klein sein
- Atomic Snapshot / Rollback schwierig

Btrfs Layout:

- Subvolumes statt Partitionen, mit Quotas
- Extra Subvolume für Snapshots



3. BTRFS Subvolumes: Layout Linux Filesystem

```
# Subvolume structure
root                        * default subvolume
    /media/btrfs            * directory in root subvolume
home
var                         * Quota 16 GB
snapshots/root              * subdirectories for each subvol
         /home
         /var
```

```
# /etc/fstab
LABEL=fs  /              btrfs  defaults
LABEL=fs  /home          btrfs  defaults,subvol=home
LABEL=fs  /var           btrfs  defaults,subvol=var
LABEL=fs  /media/btrfs btrfs  defaults,noauto,subvolid=0
```

3. BTRFS Subvolumes: Layout Linux Filesystem

```
# Subvolume structure
root                        * default subvolume
    /media/btrfs            * directory in root subvolume
home
var                         * Quota 16 GB
snapshots/root              * subdirectories for each subvol
         /home
         /var
```

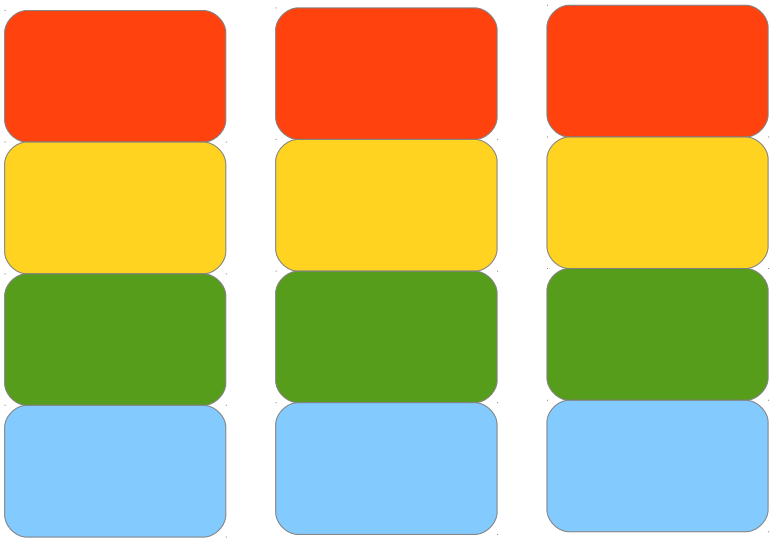
```
> mount /media/btrfs
> cd /media/btrfs
> btrfs sub snapshot root snapshots/root/2015_09_22_001
> cd ~
> umount /media/btrfs
```

Btrfs Device Kommandos:

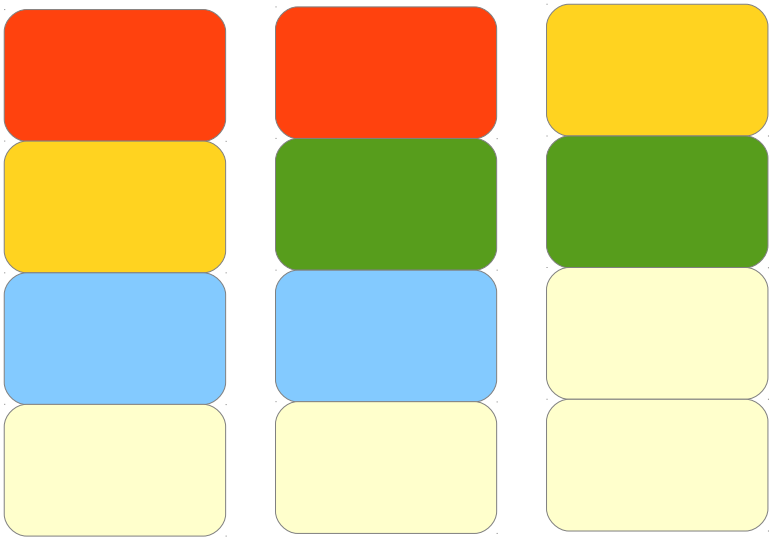
```
btrfs device <subcommand> <...>

add <dev> <path>        # Add a device
delete <dev> <path>     # Remove a device
scan [-d]|<dev>         # Scan for btrfs devices

btrfs balance start <path>
btrfs replace start <srcdev> <tgtdev> <path>
```

3. BTRFS Raid

Traditional RAID1



BTRFS RAID1



http://carfax.org.uk/btrfs-usage/

Defragmentieren: `btrfs fi defrag -r <file|dir>`

Chunks neu verteilen: `btrfs balance start <path>`

Checksummen prüfen: `btrfs scrub start <path>`

Prüfen & Reparieren: `btrfs check <device>`

ENOSPC trotz freiem Speicherplatz

-> Lösung: Rebalancen mittels

```
btrfs balance start -dusage=0 <path>
```

1:1 Kopie mittels dd funktioniert nicht (btrfs verwendet UUID)

-> Lösung: Verwendung von

```
btrfs send [-f <outfile>] <subvol>
btrfs receive [-f <infile>] <mount>
```

Funktioniert, aber noch nicht stable oder performant:

- Quotas
- RAID 5/6
- Hohe Schreiblast (Datenbanken etc.)
- btrfs-check

Geplant:

- Filesystem Verschlüsselung
- Swap Unterstützung

Btrfs Wiki:

https://btrfs.wiki.kernel.org/

Btrfs bei Arch Linux:

https://wiki.archlinux.org/index.php/Btrfs

Btrfs disk usage calculator:

http://carfax.org.uk/btrfs-usage/