

Warum sollten wir BTRFS verwenden?

- LIT2009
 - Motto: It's your turn to go ...
- LIT 2016
 - Backups
- LIT 2018
 - SSH
- LIT 2023
 - BTRFS
 - B-Tree-Filesystem
 - als Unterbau
- Ziel des Vortrages
 - einige Demos zu BTRFS
 - nicht vollständig
 - keine Vergleiche, was ist besser



<https://btrfs.readthedocs.io/en/latest/>



BTRFS als Ergänzung zu SSH, Backups, ...

- SSH, Filesystem, Backups (und Restore), VPN, siehe Vortrag von Joel Hatsch
- BTRFS ist Unterbau
 - mehr als ein ‚Filesystem‘
 - Strukturierung der Daten
 - neue Arbeitsweise mit Mountpoints über Subvolumes
 - schnelle lokale Snapshots
 - online Backups des Systems über readonly Snapshots
- Einige Links
 - <https://archive.kernel.org/oldwiki/btrfs.wiki.kernel.org/index.php/SysadminGuide.html#Subvolumes> (archiv)
 - <https://help.ubuntu.com/community/btrfs>
 - <https://everything.explained.today/Btrfs/>
 - <https://arstechnica.com/information-technology/2014/01/bitrot-and-atomic-cows-inside-next-gen-file-systems/>
 - https://www.usenix.org/system/files/login/articles/bacik_0.pdf
 - ‚The Swiss Army Knife of Storage‘
 - uvam.
 - nicht immer aktuell



- <https://btrfs.readthedocs.io/en/latest/>
- https://archive.kernel.org/oldwiki/btrfs.wiki.kernel.org/index.php/Main_Page.html

- Vorteile
 - CoW, Copy on Write
 - Snapshots
 - Checksums
 - Subvolumes mit Snapshots
 - Optimiert für SSD <https://btrfs.readthedocs.io/en/latest/Trim.html>
 - Optimiert für SMR Disks <https://btrfs.readthedocs.io/en/latest/Zoned-mode.html>

- Nachteile
 - RAID 5/6 noch nicht stabil
 - bei sehr großen Files ist bei Änderungen CoW langsam
 - fragmentiert
 - CoW lokal abschaltbar, chattr +C

 - **How do I undelete a file?**
 - ‚There is no reliable way of doing this, other than **restoring from backups**. ---‘
 - https://archive.kernel.org/oldwiki/btrfs.wiki.kernel.org/index.php/FAQ.html#How_do_I_undelete_a_file.3F

- Let's go to BTRFS



Sehr viel, würde Tage dauern ...

- Mostly self-healing in some configurations due to the nature of copy-on-write
- Online defragmentation and an autodefrag mount option
- Online volume growth and shrinking
- Online block device addition and removal
- Online balancing (movement of objects between block devices to balance load)
- Offline filesystem check[30]
- Online **data scrubbing** for finding errors and automatically fixing them for files with redundant copies
- RAID 0, RAID 1, and RAID 10[31]
- **Subvolumes** (one or more separately mountable filesystem roots within each disk partition)
 - **Snapshots**
- Transparent compression via zlib, LZO[4] and (since 4.14) ZSTD,[5] configurable per file or volume[32] [33]
- Atomic writable (via **copy-on-write**) or read-only[34] Snapshots of subvolumes
- File cloning (relink, **copy-on-write**) via `cp --relink <source file> <destination file>`[35]
- Checksums on data and metadata (CRC-32C[36]).
- **In-place conversion from ext3/4** to Btrfs (with rollback). This feature regressed around btrfs-progs version 4.0, rewritten from scratch in 4.6.[38]
- Union mounting of read-only storage, known as file system seeding (read-only storage used as a copy-on-write backing for a writable Btrfs)[39]
- Block discard (reclaims space on some virtualized setups and improves wear leveling on SSDs with TRIM)
- **Send/receive** (saving diffs between snapshots to a binary stream)
- Incremental backup[40]
- Out-of-band data deduplication (requires userspace tools)
- Ability to handle swap files and swap partitions

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



Kommandos

auch sehr viel,
btrfs + subcommands + ... + mount/device

- **btrfs** filesystem label [<device>|<mountpoint>] [<newlabel>]
- btrfs check /dev/sdh1
- blkid | grep btrfs
- btrfs --help
- send / receive
- btrfs device scan
- btrfs device scan /dev/sdh1
- btrfs filesystem show /dev/sdh1
- btrfs balance start /mnt/gdisk
- btrfs check /dev/sdh1
- btrfs check /dev/sdh1 --force
- btrfs **filesystem df**
- btrfs filesystem df /mnt/adisk
- btrfs filesystem show /dev/sda2
- btrfs scrub start /mnt/adisk
- btrfs scrub status /mnt/adisk
- btrfs subvolume create subvol0
- btrfs subvol list /mnt/cnano/
- btrfs subvolume delete daten/fuse/subvol4
- btrfs subvolume show daten/messen/data/
- btrfs subvolume show /mnt/wd/daten/subvol/
- btrfs filesystem defragment Videos
- btrfs filesystem balance /home/myhome



Was zeigen ?

- Auswahl
 - **Copy on Write** als Basis
 - **Neuformatierung** eines EXT4 Filesystems mit BTRFS ohne die Daten umzukopieren
 - **Datenintegrität** online testen
 - **Subvolumes** und Strukturierung von Daten
 - **Snapshots** großer Datenmengen in Sekunden

Demo, Festplatte mit Partitionen ...

- /mnt/b1 - BTRFS
- /mnt/b2 - BTRFS
- /mnt/c1 - LUKS mit BTRFS
- /mnt/e1 - ext4

Alias und Python-Skript

- **alias** lsblk1='lsblk -o "NAME,TYPE,TRAN,RM,FSTYPE,MOUNTPOINT" -e7' | grep usb -A5
- **ddf** = df mit Filtern (Python)
 - <https://manpages.ubuntu.com/manpages/jammy/man1/discus.1.html>



CoW = Copy on Write

- **Links in Linux**

- softlink = link to file or folder.

- **files -> /b1/files**
- nicht an Filesystem gebunden
- über Partition hinweg
- kann auf ‚nichts‘ zeigen, dead link
- in ‚ls‘ ist ein Pfeil zu sehen

- hardlink = nur für Files, weiterer Inode, der auf die gleichen Daten zeigt

- Referenzzähler zur Verwaltung
- mit ‚ls‘ anzeigbar
- nur im gleichen Filesystem
- rsnapshot verwendet Hardlinks intensiv
- in rsync: --link-dest=DIR
- ändern der Daten ändert alle Files, die durch den Hardlink verbunden sind
- kann gelöscht werden, Daten bleiben erhalten, wenn Referenzzähler > 0



CoW = Copy on Write

- **Links in Linux:**
- reflink = CoW
 - mit ,cp': --reflink=always
 - wird intern verwaltet, mit ,ls' unsichtbar
 - Metadaten werden kopiert, Daten nicht
 - erst bei Änderung der Daten werden diese auch kopiert.
 - atomarer Vorgang
 - keine offenen Files
 - Snapshots möglich
- Vorteile
 - schnell, platzsparend, transaktionssicher
 - Snapshots
 - <https://www.ctrl.blog/entry/file-cloning.html>
- Nachteile
 - belegt Platz, wenn Daten geändert werden
 - nicht mit herkömmlichen FS-Tools sichtbar
- Demo
 - `cd /mnt/b1`
 - `cp -r --reflink=always btrfstestfiles/ abc`
 - `filefrag -v btrfstestfiles/files/hhhhhh.txt`



CoW = Copy on Write

- Abschalten von CoW:
 -
 - `chattr +C new_empty_file`
 - `lsattr -d ./folder`
 - `-----C----- ./folder`
 - `cp -r --reflink=always btrfstestfiles/ abc` funktioniert nicht
- schaltet CoW und Prüfsummen ab
- Sicherheit wie bei ext4
- empfohlen bei Files für virtuelle Maschinen



online Neuformatierung eines EXT4 Filesystems mit BTRFS

- <https://btrfs.readthedocs.io/en/latest/Convert.html>
- **/mnt/e1**
- **# convert from ext3/4 → Btrfs**
- `btrfs-convert /dev/xxx`
- `btrfs-convert /dev/disk/by-partlabel/e1 1m17,585s`
- `lsblk -o "NAME,TYPE,TRAN,RM,FSTYPE,MOUNTPOINT" -e7`
- **# mount the resulting Btrfs filesystem**
- `mount -t btrfs /dev/disk/by-partlabel/e1 /mnt/e1`
- **# mount the ext3/4 snapshot**
- `mount -t btrfs -o subvol=ext2_saved /dev/sdc4 /mnt/ext4_saved`
- **# loopback mount the image file**
- `mount -t ext4 -o loop,ro /mnt/ext4_saved/image /mnt/ext4`
- `diff -r ext4/ e1/`
- `ok`
- **Warnung**
 - instabil bei Stromausfall
 - instabil bei zu wenig freiem Platz
 - ext4 Snapshot nicht immer mountable
- **besser:**
 - Backup, Format mit BTRFS, Restore



Scrub – Kontrolle aller Prüfsummen

- `btrfs scrub start /mnt/b1`
- `btrfs scrub status /mnt/b1`
- `cat /var/lib/btrfs/scrub.status.a3d6bb26-76e9-4f29-b5a8-313293e898ea`

- Ergebnis wird angezeigt
- Ergebnis ist in `/var/lib/btrfs/scrub.status.UUID`
- lange Laufzeit
- mit RAID erfolgt automatische Korrektur
- wöchentlich empfohlen

- Script
 - http://marc.merlins.org/perso/btrfs/post_2014-03-19_Btrfs-Tips_-_Btrfs-Scrub-and-Btrfs-Filesystem-Repair.html
 - `ionice -c 3 nice -10 btrfs scrub start -Bd $mountpoint`

- `tr '|' '\n' < /var/lib/btrfs/scrub.status.56c26d49-490e-424a-bafb-f07736424be9 | g csum_errors`



Subvolumes und Snapshots 1

- Subvolume ?
 - ‚dynamische Partitionen‘
 - ‚Filesystem im Filesystem‘
 - keine Namensregeln, muss selbst gestaltet werden
 - ...
- rekursiv
- mountable
 - statt nur ‚root‘,
- Doku: <https://btrfs.readthedocs.io/en/latest/Subvolumes.html>
 - ‚A BTRFS subvolume is a part of filesystem with its own independent file/directory hierarchy and inode number namespace. Subvolumes can share file extents. A snapshot is also subvolume, but with a given initial content of the original subvolume.‘
 - ‚A subvolume looks like a normal directory, with some additional operations described below.‘
- Namensregeln nötig
 - mount ist nicht mehr fixe Zuordnung von Partitionen, sondern Zuordnung von Arbeitsbereichen
- `tree -L 3 -d` als Hilfe
- manchmal sieht man ‚@‘ als Prefix, kann man, muss man nicht
- Subvolumes nur mit btrfs tools sichtbar
- **btrfs subvolume list folder**
- Wie macht Ubuntu das? VM



Subvolumes und Snapshots 2

- Subvolume ?
- vom Parentfilesystem erreichbar
- Subvolume kann Parent nicht erreichen
 - wie ‚chroot‘
- **Demo**
 - subvol create delete
 - files und tgg erzeugen
 - btrfs subvolume create files
 - cp --reflink=always -r btrfstestfiles/files/* files/
 - mv tgg/ _tgg
 - btrfs subvolume create tgg
 - mv _tgg/* tgg/
 - rmdir _tgg/
-
- **btrfs subvolume list b1**
- **mount -o subvol=files/tgg /dev/disk/by-label/b1 tgg**
- **/proc/self/mountinfo**
- **diff -r tgg b1/files/tgg**
- **du -sh tgg**
- **df -h**



Subvolumes und Snapshots 3

- **send/receive**
- Send/receive, only readonly Subvolumes, nicht rekursive Snapshots
- Quelle: /mnt/b1/snr, Ziel: /mnt/c1
- **btrfs subvolume snapshot -r b1/files b1/files_ro**
- **btrfs send b1/files_ro | btrfs receive c1**
- `diff -r b1/files_ro/ c1/files_ro/`
- **tgg** fehlt
- c1 ist LUKS d.h. BTRFS und LUKS ist ok
- `rsync -av b1/files c1/rsync/ -P`
- **tgg** fehlt nicht
- `touch b1/files/xxxxx` (erzeuge neuen File)
- `btrfs subvolume snapshot -r b1/files b1/files_ro2`
- send diff to c1
- `btrfs send -p b1/files_ro b1/files_ro2 | btrfs receive c1`
- https://aligrant.com/web/blog/2019-02-27_copying_a_btrfs_volume_to_another_disk_the_easy_way



Subvolumes und Snapshots 4

- durch CoW effektiv
- Namensregeln für sich erstellen, ‚@‘ oder andere
- z.B. Erstellen, Backup, delete
- klein, können schnell wachsen
- mit BTRFS möglichst Partitionen nicht zu voll schreiben, testen
- rekursive Subvolumes nicht in Snapshot oder send/receive
- als Filter einsetzen
- Skripte für mount einsetzen
- flexibles mount mit Subvolumes verstehen
- Doku lesen
- <https://btrfs.readthedocs.io/en/latest/>



Danke

- Wie beginnen?
 - **apt install btrfs-progs**
 - formatieren und genau wie ein ,ext4' verwenden
 - Vorteile: Prüfsummen, Copy on Write ...
- Take Home Message
 - ein sicheres Filesystem mit Linux Bordmitteln ist nicht schwer
 - mit BTRFS gibt es eine zuverlässige Lösung
 - noch ohne RAID5/6
- meine Webseite
 - <https://www.görlinux.de>

Vielen Dank für Eure Aufmerksamkeit.

Richard Albrecht

LUG-Ottobrunn



Richard Albrecht

Linux in Görlitz

LIT Augsburg 2023
Richard Albrecht
LUG-Ottobrunn

