btrfs

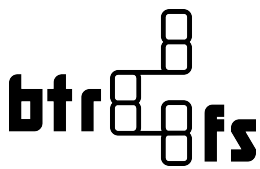
Eine Einführung

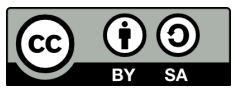
Florian Preinstorfer

http://nblock.org

VALUG

13.12.2013





This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Austria license (CC-BY-SA).

Inhalt

Historie

2007 Usenix 07: Vorstellung einer Copy-On-Write freundlichen B-tree Variante [7].

2007 Chris Mason kündigt btrfs auf der LKML an [8].

2009 Aufnahme in Linux 2.6.29 [8, 3].

20xx Support in Arch, Debian, Fedora, Ubuntu, ...

2012 Kommerzielle Unterstützung in Oracle Unbreakable Linux und SLES 11 SP2 [3].

Aktueller Status

- Linux 3.12: *Btrfs is highly experimental, and THE DISK FORMAT IS NOT YET FINALIZED. You should say NO here unless you are interested in testing Btrfs with non-critical data* [6].[1]
- Ständige Verbesserung mit jeder Kernel Version [3].
- Lizenz: GPLv2

[1] Linux 3.13: The filesystem disk format is no longer unstable, ...

Projektaktivität

- Kernel Modul
  - LOC: ∼75k
  - 376 Commits von 48 Entwicklern im letzten halben Jahr.
  - 8 Entwickler mit mehr als 10 Commits im letzten halben Jahr.
- Userland Tools
  - LOC: ∼45k
  - 274 Commits von 30 Entwicklern im letzten halben Jahr.
  - 8 Entwickler mit mehr als 10 Commits im letzten halben Jahr.

Copy-On-Write

- Geänderte Blöcke werden nicht überschrieben, sondern in einem freien Speicherbereich abgelegt. Anschließend wird der Verweis auf die Datei in den Metadaten aktualisiert [1].
- Kein Journal nötig.
- Dateisystemkonsistenz wird sichergestellt.

Transparente Kompression

- Automatische Kompression auf Dateiebene, wenn wirklich Speicherplatz eingespart werden kann [1].
- Verfügbare Kompressionsalgorithmen: ZLIB oder LZO*
- Automatische Kompression: `mount -o compress`
- Erzwungene Kompression: `mount -o compress-force`

Online Resize*

- Ein btrfs Dateisystem kann im Betrieb sowohl *verkleinert* als auch *vergrößert* werden.
- Auch für mehrere Festplatten möglich.
- `btrfs filesystem resize ...`

Festplatten hinzufügen/entfernen*

- Einem bestehenden btrfs Dateisystem können zur Laufzeit Festplatten hinzugefügt/entfernt werden.
- Hinzufügen: Neue Blöcke werden nach und nach auf der neuen Festplatte allokiert.
- Entfernen: Nicht redundante Daten werden vorab auf bestehende Festplatten kopiert.
- Daten werden *nicht* verteilt (balance).
- `btrfs device add/delete ...`

Online Balance*

- Daten werden gelesen und je nach Allokationsmodus neu auf dem btrfs Dateisystem verteilt.
- Anwendungsfall: Eine Festplatte wurde hinzugefügt/entfernt.
- `btrfs filesystem balance <path>`

RAID*

- btrfs bietet Unterstützung für RAID 0, 1, 10.
- Unterstützung für RAID 5, 6 ist in Arbeit [4].
- Das RAID Level kann für Daten (-d) und Metadaten (-m) separat angegeben werden.
- `mkfs.btrfs -m <metadata raid> -d <data raid> ...`

Prüfsummen

- btrfs berechnet Prüfsummen für Daten und Metadaten.
- Bei jedem Lesezugriff wird geprüft, ob die Daten korrekt gelesen werden konnten.
- Bei Lesefehlern versucht btrfs auf redundant vorhandene Datenblöcke auszuweichen (RAID).
- Algorithmus: derzeit nur crc32c

Scrubbing

- Fehler im Dateisystem werden mit Hilfe von Prüfsummen und redundanten Kopien repariert.
- Srubbing passiert im Hintergrund und kann sehr lange dauern.
- `btrfs scrub <start|cancel|resume|status> <path>`

Subvolumes*

- Subvolumes sind Namespaces innerhalb eines Dateisystems.
- Sie vehalten sich wie Verzeichnisse (keine devices!).
- Subvolumes können gemounted werden.
- Default Subvolume: default (0)
- `btrfs subvolume <command> ...`

Snapshots*

- Snapshots sind Subvolumes.
- Snapshots entstehen als Kopien von Subvolumes.
- Snapshots können auch schreibgeschützt werden.
- Typische Anwendungsfälle: Backup, Experimente, ...
- `btrfs subvolume snapshot ...`

Offline Deduplication

- Redundante Datenblöcke erkennen und eliminieren.
- Offline bedeuted: gemounted aber nicht „aktiv“.
- Seit Kernel 3.12 [5]

Support für SSDs

- . . . kann via `mount -o ssd` aktiviert werden.
- Unterstützung für TRIM ist standardmäßig deaktiviert.
  - Aktivieren via `mount -o discard`
  - Manuell via `fstrim(8)`

Offline Dateisystemprüfung

- Ja, mittlerweile gibt es `btrfsck`.
- Ja, es kann auch ein kaputtes btrfs Dateisystem reparieren.
- `btrfsck /dev/<umounted-device>`

Fazit

- Gute Unterstützung in vielen Distributionen.
- Tools sind nicht ausgereift.
- Schwerwiegende Probleme während meiner Experimente:
  - Kernel 3.11.3: Kernel bug in btrfs balance [2]
  - 3.9.x: btrfs verursacht auf meiner NAS ein Problem: btrfs scrub notwendig
- btrfs wird – irgendwann mal – das „next generation filesystem für Linux“.

Referenzen I

[1] Btrfs - the swiss army knife of storage.
https://www.usenix.org/legacy/publications/login/2012-02/openpdfs/Bacik.pdf.

[2] Btrfs balance bug.
http://www.mail-archive.com/linux-btrfs@vger.kernel.org/msg27694.html.

[3] btrfs changelog.
https://btrfs.wiki.kernel.org/index.php/Changelog.

[4] btrfs faq.
https://btrfs.wiki.kernel.org/index.php/FAQ.

[5] btrfs pull request for kernel 3.12.
http://lkml.indiana.edu/hypermail/linux/kernel/1309.1/02981.html.

[6] Source of linux 3.12.
https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/fs/btrfs/Kconfig?id=refs/tags/v3.12.

[7] Valerie Aurora.
A short history of btrfs.
http://lwn.net/Articles/342892.

[8] Chris Mason.
[announce] btrfs: a copy on write, snapshotting fs.
https://lkml.org/lkml/2007/6/12/242.