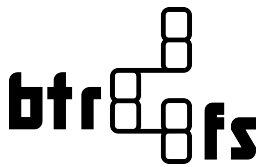
btrfs

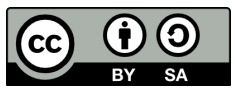
Eine Einführung

Florian Preinstorfer

http://nblock.org

Linuxwochen Linz
30.05.2015





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 [8].

2007 Chris Mason kündigt btrfs auf der LKML an [9].

2009 Aufnahme in Linux 2.6.29 [9, 3].

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

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

Aktueller Status

- Seit Linux 3.13: *The filesystem disk format is no longer unstable, and it's not expected to change unless there are strong reasons to do so.* [7]
- Viele Änderungen mit jeder neuen Kernel Version [3].
- Lizenz: GPLv2

Projektaktivität

Kernelmodul, LOC: ∼86k

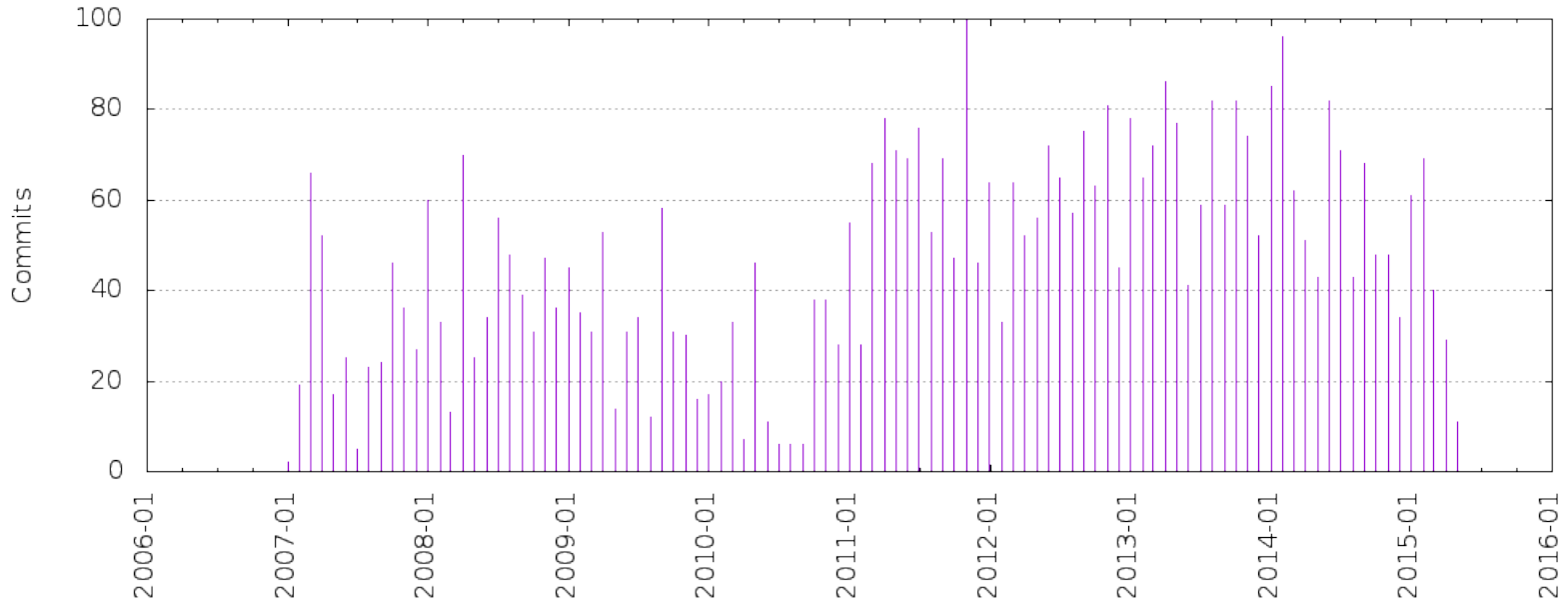


Abbildung: Commits pro Jahr und Monat.

Projektaktivität

Kernelmodul, LOC: ∼86k

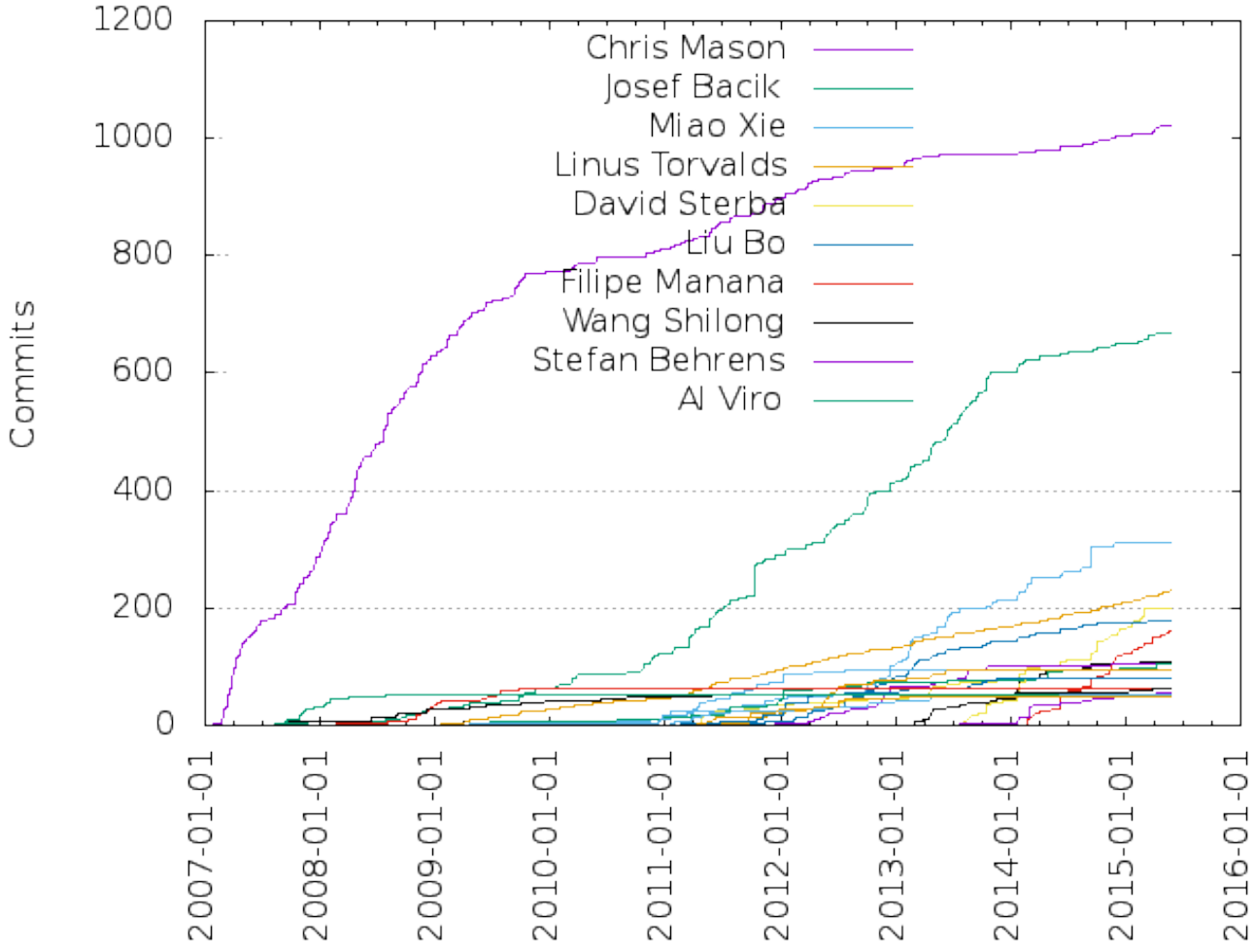


Abbildung: Commits pro Entwickler.

Projektaktivität

Userland tools, LOC: ∼60k

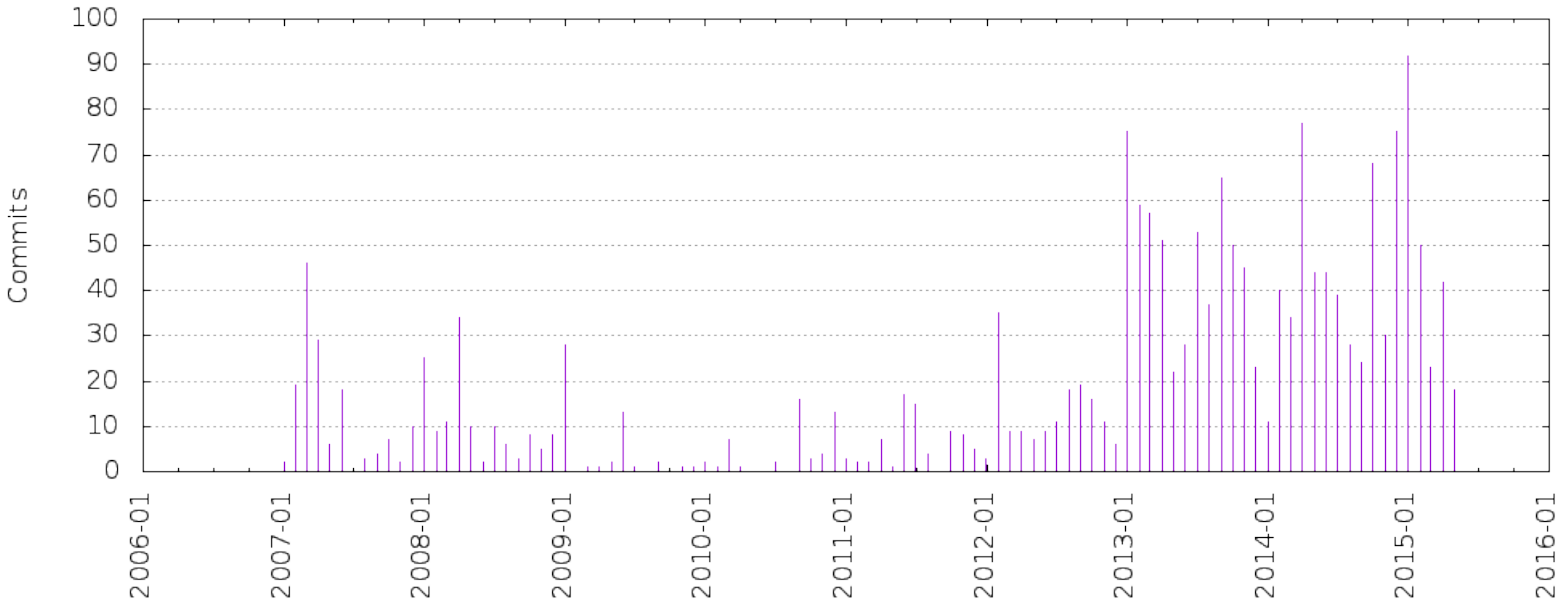


Abbildung: Commits pro Jahr und Monat.

Projektaktivität

Userland tools, LOC: ∼60k

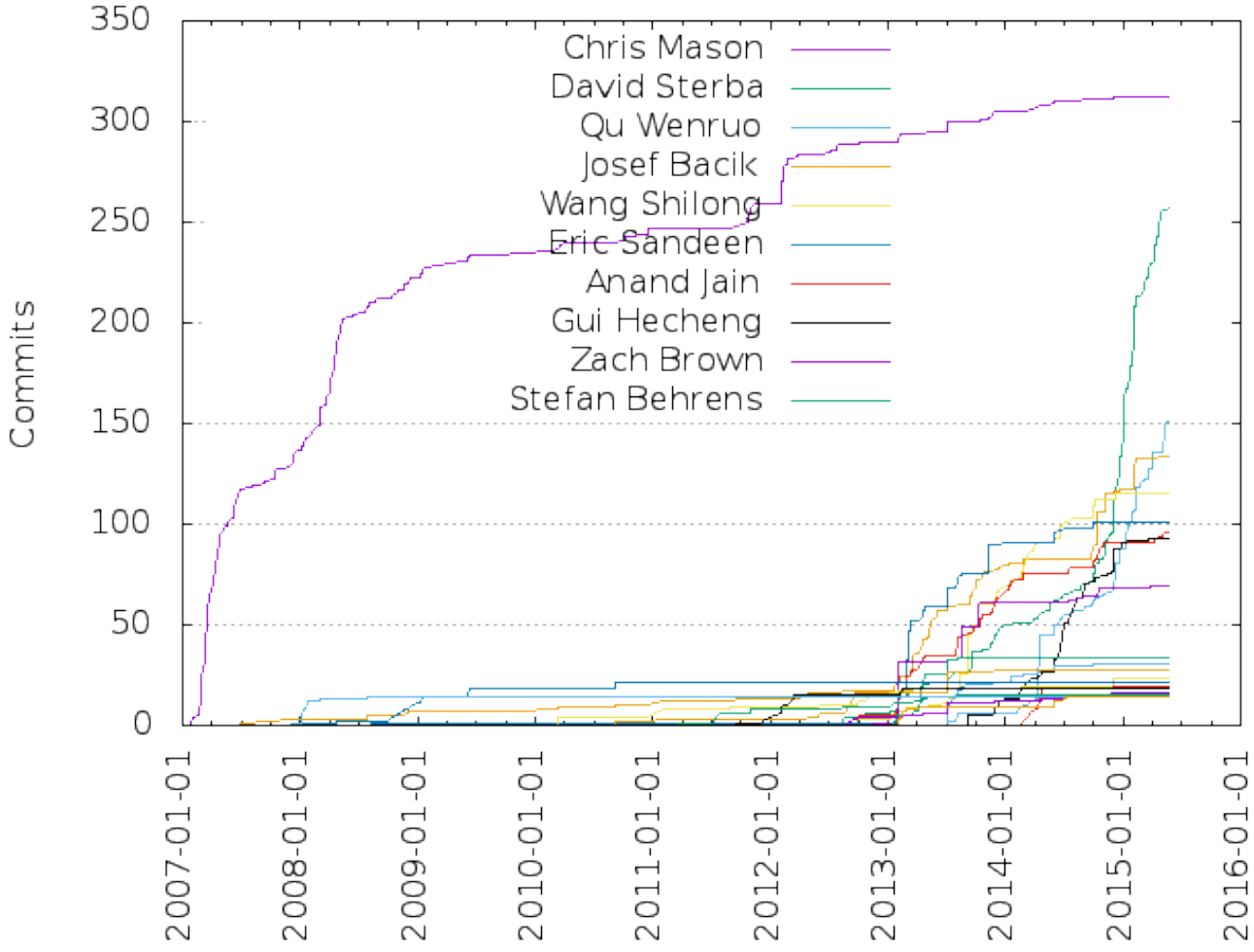


Abbildung: Commits pro Entwickler.

Copy-On-Write

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

Transparente Kompression

- Automatische Kompression auf Dateiebene, wenn 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 balance <path>`

RAID*

- btrfs bietet Unterstützung für RAID 0, 1, 10.
- Seit Linux 3.9: Initiale Unterstützung für RAID 5, 6 [4].
- „Recovery and rebuild“ erst seit Linux 3.19 möglich.
- Das RAID Level kann für Daten (-d) und Metadaten (-m) separat angegeben werden[1].
- `mkfs.btrfs -m <metadata raid> -d <data raid> ...`

[1] Online btrfs disk usage calculator: `http://carfax.org.uk/btrfs-usage/index.html`

Prüfsummen

- btrfs berechnet Prüfsummen für Daten und Metadaten.
- Bei jedem Lesezugriff wird geprüft, ob die Daten korrekt gelesen werden konnten.
- Bei Lesefehlern versucht btrfs die defekten Blöcke mittels unbeschädigter Kopien zu reparieren.
- 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 verhalten sich wie Verzeichnisse (keine devices!).
- Subvolumes können gemounted werden.
- Das default Subvolume (0) ist immer vorhanden.
- `btrfs subvolume <command> ...`

Snapshots*

- Snapshots sind Subvolumes.
- Snapshots können schreibgeschützt werden.
- Typische Anwendungsfälle: Backup, Experimente, ...
- `btrfs subvolume snapshot ...`

Seed devices

- Schreibgeschütztes „Image“ für ein btrfs Dateisystem.
- Änderungen landen nur im beschreibbaren btrfs Dateisystem.
- Seed devices können jederzeit entfernt werden.
- `btrfstune -S 1 <seed-device>`

Support für SSDs

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

Offline Dateisystemprüfung

- Ja, mittlerweile gibt es `btrfs check`[2].
- Ja, es kann auch ein kaputtes btrfs Dateisystem reparieren.
- Ist als letzter Ausweg vor Datenverlust gedacht.
- `btrfs check [options] /dev/<umounted-device>`

[2] `btrfsck` wurde von `btrfs check` abgelöst.

btrfs und ich in den letzten 4-5 Jahren

- 3.9.x: btrfs verliert eine Disk in meiner NAS → btrfs scrub notwendig
- 3.11.3: Kernel bug in btrfs balance [2]
- 3.13.x: Disks können nicht aus einem RAID entfernt werden.
- aktuell (seit 3.18.x): Hängt beim Löschen von Dateien [6]

Fazit

- Gute Unterstützung in vielen Distributionen.
- Tools sind nicht ausgereift.
- Schwerwiegende Probleme während meiner Experimente.
- btrfs wird – hoffentlich bald – das „next generation filesystem für Linux“.
- Einfach mal ausprobieren.

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

[6] Hang on deletion.
https://bugzilla.kernel.org/show_bug.cgi?id=76421.

[7] Source of linux 4.0.
https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/fs/btrfs/Kconfig?id=refs/tags/v4.0.

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

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

Freier Speicherplatz [5]

- Bisher mühselige Berechnung mit:
  - `btrfs filesystem show`
  - `btrfs filesystem df`

```
$ sudo btrfs fi show
Label: none  uuid: 12345678-1234-5678-1234-1234567890ab
  Total devices 2 FS bytes used 304.48GB
  devid    1 size 427.24GB used 197.01GB path /dev/sda1
  devid    2 size 465.76GB used 197.01GB path /dev/sdc1

$ sudo btrfs fi df /mnt
Metadata, single: total=18.00GB, used=6.10GB
Data, single: total=376.00GB, used=298.37GB
System, single: total=12.00MB, used=40.00KB
```

Freier Speicherplatz

- Seit btrfs-progs 3.18 gibt es `btrfs filesystem usage`

```
$ sudo btrfs fi usage /mnt
Overall:
    Device size:         80.00GiB
    Device allocated:    32.02GiB
    Device unallocated:  47.98GiB
    Used:                29.33GiB
    Free (estimated):    24.34GiB  (min: 24.34GiB)
    Data ratio:          2.00
    Metadata ratio:      2.00
    Global reserve:      16.00MiB  (used: 0.00B)
```

Freier Speicherplatz

```
$ sudo btrfs fi usage /mnt    (continued)
Data,RAID1: Size:15.00GiB, Used:14.65GiB
   /dev/sdc1      15.00GiB
   /dev/sdd1      15.00GiB

Metadata,RAID1: Size:1.00GiB, Used:15.59MiB
   /dev/sdc1      1.00GiB
   /dev/sdd1      1.00GiB

System,RAID1: Size:8.00MiB, Used:16.00KiB
   /dev/sdc1      8.00MiB
   /dev/sdd1      8.00MiB

Unallocated:
   /dev/sdc1      23.99GiB
   /dev/sdd1      23.99GiB
```