Snapshots, Checksummen, RAID

Eine Einführung in das Btrfs-Dateisystem

Lenz Grimmer
Oracle Deutschland B.V. & Co. KG
Hamburg, Germany

Schlüsselworte

Oracle Linux, Dateisystem, Btrfs, Checksummen, Snapshots, RAID, Administration

Einleitung

Das Linux-Betriebssystem bietet ausgefeilte und vielfältige Methoden zur Speicherung und Organisation von Daten auf Festplatten, wie z.B. ext3, ext4 oder dem XFS-Dateisystem. Jedes hat seine eigenen Charakteristiken und Fähigkeiten, die es dem Administrator erlauben, das jeweils passende Dateisystem für seine Speicher-Anforderungen und Bedürfnisse auszuwählen.

Das Btrfs-Dateisystem verfügt über eine Reihe von Eigenschaften, die so von keinem anderen Linux-Dateisystem geboten werden. Unter anderem unterstützt es die folgenden Eigenschaften:

- Unterstützung von sehr großen Dateien und Dateisystemen
- Integrierte Funktionalität zum Verwalten von Speicher-Volumen
- Eingebaute RAID-Funktionalität
- Datensicherheit durch Checksummen und Copy-on-Write Speichermethoden
- Snapshots von Verzeichnisstrukturen oder einzelner Dateien (Clones)
- Datenkompression mit verschiedenen Kompressionsmethoden (z.B. zlib, lzo)

Design-Ziele und Historie

Das ursprünglich von Chris Mason bei Oracle entwickelte Btrfs-Dateisystem basiert auf den Konzepten einer Präsentation des IBM-Mitarbeiters Ohad Rodeh zum Thema “copy-on-write friendly B-tree implementations”, die Rodeh auf der USENIX FAST '07 Konferenz vorgestellt hatte.

Mason basierte das Btrfs-Design auch auf seine Erfahrungen mit der Entwicklung des ReiserFS-Dateisystems, zu dem er während seiner Zeit als Entwickler bei SUSE maßgeblich beigetragen hatte. Von diesem stammen z.B. Konzepte wie die extent-basierte Speicherung von Daten oder das Ablegen von kleineren Dateien direkt im inode-Datenblock, die mit der Organisation der Daten und Metadaten in Form von B-Bäumen kombiniert wurden.

Nach einigen Monaten interner Entwicklung wurde Btrfs im Juni 2007 erstmalig der Linux-Community vorgestellt. Ursprünglich als separater Code-Zweig entwickelt, wurde Btrfs im Januar 2009 als fester Bestandteil des mainline Linux-Kerns mit Version 2.6.29 aufgenommen. Die kontiniuerliche Weiterentwicklung des Dateisystems wird weiterhin von Chris Mason koordiniert. Entwickler namhafter Linux-Distributionen und anderen Firmen arbeiten heutzutage hauptberuflich an der Verbesserung und Erweiterung von Btrfs. Zu diesen gehören u.a. Oracle, Red Hat und SUSE, sowie Dream-

host, Fujitsu, Fusion-IO, HP, IBM, Intel oder auch STRATO. Btrfs ist Bestandteil vieler Linux-Distributionen und wird mit fortschreitender Entwicklung weiter an Bedeutung gewinnen.

Oracle war zusammen mit SUSE einer der ersten Linux-Distributoren, die Btrfs als Linux-Dateisystem voll unterstützen und technischen Support dafür anbieten.

Architektur und allgemeine Eigenschaften

Btrfs besitzt die folgenden Eigenschaften:

- Struktur: Btrfs setzt den Großteil der internen Datenstrukturen in B-Bäumen um
- Journaling: Btrfs ist kein journal-basiertes Dateisystem. Schreiboperationen werden nur einmal durchgeführt, was viele Einschränkungen bei der Verwendung eines Journals umgeht, wie z.B. die Journal-Größe und eventueller Verschleiß durch das wiederholte Beschreiben des gleichen Bereichs eines Speichermediums wie z.B. einer Festplatte oder einer Flash-Disk (SSD).
- Copy-on-write: Btrfs verwendet diese Technik, die vermeidet, daß Datenblöcke und Extents an Ort und Stelle überschrieben werden. Stattdessen werden zu verändernde Datenblöcke zuerst an eine neue Stelle kopiert.
- Dateiattribute und Zugriffskontroll-Listen (ACLs): Btrfs bietet Unterstützung für erweiterte Dateiattribute und Access Control Lists (ACLs) nach dem POSIX-Standard.
- Feintuning: Btrfs bietet eine Reihe von Einstellmöglichkeiten, die das Verhalten und die Funktionalität des Dateisystems beeinflussen. Zu den interessanten Optionen gehört z.B. die `-o autodefrag` mount-Option, die eine automatische Defragmentierung des Dateisystems zur Laufzeit aktiviert. Eine weitere ist die Möglichkeit, den Copy-on-write-Mechanismus mit Hilfe der `nocow` mount-Option zu deaktivieren. In diesem Modus werden Datenblöcke an Ort und Stelle überschrieben, wie man es von anderen Dateisystemen kennt. Weitere low-level Tuning-Möglichkeiten bestimmen, wie viele Datenblöcke vorausschauend vom Dateisystem in den Hauptspeicher geladen werden (read-ahead), oder die Anzahl der gleichzeitig abzusetzenden Anfragen an das Speicher-Subsystem.

Administration

Btrfs wird hautpsächlich über die Kommandozeile administriert. Die wenigen verfügbaren grafischen Oberflächen beschränken sich in ihrer Funktionalität meist auf die grundlegende Einrichtung sowie die Installation des Betriebssystems auf einem Btrfs-Dateisystem. Einen Zugriff auf die weiterführenden Möglichkeiten bieten sie üblicherweise nicht.

Btrfs wird nach dem initialen Anlegen des Dateisystems mit dem Kommando `mkfs.btrfs` im wesentlichen über das `btrfs`-Werkzeug administriert. Dieses Kommando vereint alle Funktionen unter einer gemeinsamen Oberfläche – mit ihm lassen sich u.a. Dateisystem-Snapshots erstellen (`btrfs subvolume snapshot`), neue Festplatten zu einem Volume hinzufügen (`btrfs device add`) oder die Größe eines Dateisystems dynamisch verändern (`btrfs filesystem resize`).

Btrfs benötigt keinen Logical Volume Manager oder separaten Software-RAID Layer zur dynamischen und flexiblen Verwaltung des Speicherplatzes – diese Funktionalität wird vom Dateisystem an sich bereit gestellt.

Obwohl technisch möglich, ist es daher nicht notwendig, die verfügbaren Festplatten mit Hilfe des Li-

nux Device Mappers zu einem RAID-Verbund zusammenzufassen und diesen Speicherplatz mit Hilfe von LVM zu verwalten. Btrfs funktioniert am effektivsten, wenn es direkten Zugriff auf die einzelnen verfügbaren Block-Geräte hat (z.B. als „JBOD“ – „just a bunch of disks“), ohne die Daten durch eine weitere Abstraktionsschicht hindurchreichen zu müssen. Auf diese Weise kann es seine Stärken am besten ausspielen.

Hierbei gibt es allerdings eine Ausnahme: falls eine verschlüsselte Speicherung der Daten erforderlich ist, muß dies mittels `dm-crypt` oder LUKS unterhalb des Dateisystems oder mittels eCryptfs innerhalb der Dateisystemstruktur erfolgen – Btrfs bietet im Moment noch keine eingebaute Datenverschlüsselung.

Snapshots und Clones

Aufgrund der von Btrfs verwendeten Copy-on-Write Technologie ist das Anlegen von Schnappschüssen einzelner Dateien oder ganzer Dateisysteme eine sehr „billige“ Operation. Snapshots können in Bruchteilen von Sekunden in beliebiger Anzahl erzeugt werden und verursachen keinen nennenswerten Overhead.

Das Funktionalität zum Anlegen eines Datei-Snapshots (oder „Clone“) ist in das herkömmliche Kopierkommando „`cp`“ integriert: mit dem Befehl „`cp --reflink <quelle> <ziel>`“ wird im ersten Schritt lediglich ein neuer Verweis auf die Quelldatei angelegt (ähnlich einem hard link), die eigentlichen Daten werden nicht kopiert. Erst wenn die Datei unter einem ihrer Namen geöffnet und verändert wird, werden zusätzliche Speicherblöcke benötigt, um die nun enstandene Differenz zwischen den Dateien festzuhalten. Diese sehr speicher-effiziente Methode ist besonders hilfreich, wenn auf einem Btrfs-Dateisystem sehr viele und sehr große Dateien gespeichert werden sollen, die sich aber nur marginal an wenigen Stellen unterscheiden (wie z.B. Disk Images von virtuellen Maschinen).

Auch das Anlegen von Schnappschüssen kompletter Verzeichnis-Strukturen (Subvolumes) erfolgt nach dem gleichen Schema – nur die eigentlichen Differenzen zwischen den Verzeichnissen werden zusätzlich gespeichert.

Snapshots können auch gegen Schreibzugriffe geschützt werden (read-only), so dass sie für Archivierungszwecke eingesetzt werden können.

Praktische Anwendungsfälle für Dateisystem-Snapshots auf Oracle Linux

Die Snapshot-Technik wird zum Beispiel beim Einsatz von Linux Containers (LXC) verwendet: das Oracle Linux-Template für LXC klont bestehende Container-Verzeichnisse unter Zuhilfenahme von Btrfs-Snapshots. Auf diese Weise können neue Container-Instanzen innerhalb kürzester Zeit erstellt und hochgefahren werden, ohne viel zusätzlichen Plattenplatz zu belegen.

Für das Paket-Management-Tool `yum` gibt es einen „Filesystem snapshot plugin“ der bewirkt, dass vor jeder Ausführung einer Aktion (z.B. Installation, Update oder Entfernung von Software-Paketen) automatisch ein Snapshot des Dateisystems erstellt wird. Auf diese Weise ist es jederzeit möglich, per Neustart auf den vorherigen Zustand zurück zu gehen, falls ein Installationsvorgang unerwünschte Nebeneffekte gehabt haben sollte.

Datenintegrität: Checksummen und Scrubbing

Btrfs generiert Checksummen für Daten- und Metadatenblöcke, um die Integrität der Daten zu gewährleisten. Diese Checksummen werden beim Abspeichern eines Datenblocks generiert und bei je-

dem nachfolgenden Lesevorgang des Blocks von der Festplatte überprüft. Falls dabei ein Checksummenfehler festgestellt wird, versucht Btrfs automatisch eine Kopie des Blocks von einem anderen Gerät einzulesen, wenn Datenspiegelung (RAID1) verwendet wird. Wird eine intakte Kopie gefunden, wird diese stattdessen zurückgeliefert und der beschädigte Block korrigiert. Dieser Vorgang geschieht transparent im Hintergrund; der Systemadministrator wird über dieses Ereignis via `syslog` informiert.

Es ist auch möglich, manuell eine Konsistenzprüfung des gesamten Dateisystems (Daten und Metadaten) anzustoßen. Mit dem Kommando „`btrfs scrub start <filesystem>`“ wird ein im Hintergrund laufender Kernel-Thread gestartet, der nun die gesamte Dateisystemstruktur traversiert und die Konsistenz sämtlicher Datenblöcke anhand ihrer Checksummen überprüft. Dieser Vorgang prüft nur die tatsächlich vom Dateisystem belegten Blöcke und ist damit meist deutlich schneller abgeschlossen als ein herkömmlicher „surface scan“, bei dem die gesamte Festplatte nach defekten Blöcken durchsucht wird (unabhängig davon, ob sie vom Dateisystem benutzt werden). Der Fortschritt der Überprüfung läßt sich jederzeit mit Hilfe des Kommandos „`btrfs scrub status <filesystem>`“ anzeigen.

Falls der Scrub-Vorgang Checksummenfehler aufdeckt, wird zuerst versucht, den defekten Block durch eine intakte Kopie zu ersetzen. Anderenfalls wird ein nachfolgender Zugriff auf die beschädigte Datei mit einer Fehlermeldung („I/O Error“) quittiert; der Rest des Dateisystems ist davon nicht betroffen. In jedem Fall wird ein gefundener Fehler (ob korrigierbar oder nicht) im Syslog vermerkt.

Redundanz: Unterstützung mehrerer Festplatten

Neben der schlichten Erweiterung des vorhandenen Plattenplatzes kann Btrfs zusätzliche Festplatten auch dazu verwenden, mit Hilfe von Redundanz die Verfügbarkeit von Daten und Metadaten zu erhöhen.

Im Gegensatz zum „klassischen“ RAID auf Festplatten werden bei Btrfs allerdings nicht komplette Block-Geräte gespiegelt; das Dateisystem verteilt Blockgruppen (sogenannte „Chunks“) nach bestimmten Methoden. Im Moment unterstützt Btrfs auf Oracle Linux RAID0 (Striping), RAID1 (Spiegelung) und RAID10 (Striped and Mirrored). Weitere RAID-Level (5 und 6) befinden sich im Moment in der Entwicklung; ein erster experimenteller Support dafür wurde in den Linux Kernel 3.9 aufgenommen, ist aber ausdrücklich noch nicht für den produktiven Einsatz geeignet.

Der RAID-Level eines Dateisystems läßt sich sowohl beim Anlegen des Dateisystems als auch zur Laufzeit ändern, dies kann für Daten und Metadaten getrennt konfiguriert werden. Standardmäßig wird für Daten RAID0 (Striping) verwendet, während Metadaten redundant gespeichert werden, selbst wenn nur eine einzelne Festplatte verwendet wird.

Mit dem folgenden Kommando wird Beispielsweise RAID1 (Spiegelung) sowohl für Daten als auch Metadaten für ein bestehendes Dateisystem konfiguriert:

```
btrfs balance start -dconvert=raid1 -mconvert=raid1 <filesystem>
```

Das „balance“-Kommando verteilt daraufhin die Chunks entsprechend nach dem gewählten RAID-Modus. Das Kommando kann auch dazu verwendet werden, Datenblöcke gleichmäßig auf alle Festplatten zu verteilen, um eine ausgewogene Auslastung zu erreichen, falls zur Laufzeit eine weitere Platte zum Dateisystem hinzugefügt wurde.

Das Hinzufügen einer weiteren Festplatte läßt sich jederzeit mit Hilfe des Kommandos „`btrfs device add <device> <filesystem>`“ bewerkstelligen. Der zusätzliche Speicherplatz läßt

sich umgehend ansprechen und verwenden.

Analog dazu läßt sich mit „`btrfs device delete <device> <filesystem>`“ eine (intakte) Festplatte aus dem Verbund entfernen. Btrfs nimmt in diesem Fall automatisch eine Umverteilung der auf dieser Platte befindlichen Chunks vor, falls diese nicht bereits redundant auf einer anderen Platte vorhanden sind. Der dafür notwendige Speicherplatz muß natürlich noch auf den verbleibenden Laufwerken zur Verfügung stehen. Weiterhin muß die frei zu gebende Festplatte unbedingt bis zum Abschluß dieser Operation verfügbar bleiben, anderenfalls droht Datenverlust!

Kompression

Btrfs unterstützt Kompression auf Subvolumen-Basis. Diese kann jederzeit aktiviert werden – ab diesem Zeitpunkt versucht das Dateisystem automatisch, Daten vor der Speicherung auf der Festplatte zu komprimieren. Dabei prüft Btrfs, ob eine Kompression überhaupt sinnvoll ist – falls eine Datei nicht weiter komprimierbar ist, wird sie entsprechend markiert und ohne weitere Kompressionsversuche abgespeichert. Als Kompressions-Algorithmus stehen im Moment LZO oder zlib zur Verfügung; weitere Kompressionsmethoden wie Snappy oder LZ4 sind ebenfalls in der Entwicklung. Damit kann abhängig von den zu speichernden Daten die dafür effektivste Kompressionsmethode gewählt werden.

Defragmentierung

Aufgrund der von Btrfs verwendeten Copy-on-Write Methode tendiert das Dateisystem dazu, Daten im Laufe der Zeit über die gesamte Festplatte zu verteilen. Auf herkömmlichen Festplattenlaufwerken kann diese Fragmentierung zu Performance-Einbußen durch exzessive Lesekopfbewegungen führen.

Btrfs bietet hierzu die Möglichkeit, das Dateisystem mit Hilfe der „`-o autodefrag`“ Mount-Option automatisch im Hintergrund zu defragmentieren. Wenn ein Datenblock kopiert und auf Platte geschrieben wird, wird dieser Teil der Datei zur Defragmentierung vorgemerkt und im Hintergrund an den dafür zuständigen Kernel-Prozess weitergereicht, der in regelmäßigen Abständen eine Re-Organisation der Dateisystemblöcke vornimmt.

Alternativ zu dieser automatischen Defragmentierung kann der Defragmentierungsvorgang auch manuell auf Datei- oder Dateisystemebene mit Hilfe des Kommandos „`btrfs filesystem defrag <file|filesystem>`“ angestoßen werden.

Optimierungen für SSDs

Btrfs ist „SSD-aware“ – es kann seine Zugriffsmuster an das verwendete Speichermedium anpassen. So umgeht es bei Schreiboperationen auf Flash Disks jegliche für herkömmliche Festplatten notwendigen Optimierungen zur Vermeidung von Schreib-/Lesekopfbewegungen und reicht Schreiboperationen in größeren Blöcken an das Gerät weiter, auch wenn sie in Zugriffe auf verschiedene Dateien resultieren.

Optional kann zusätzlich die Unterstützung für „TRIM/Discard“ aktiviert werden. In diesem Modus markiert das Dateisystem frei gewordene Datenblöcke, die dann vom Speichermedium (typischerweise ein Solid-State-Laufwerk) anderweitig wieder verwendet werden können.

Konvertierung von bestehenden ext3 und ext4 Dateisystemen

Ein weiteres interessantes Merkmal von Btrfs ist die Möglichkeit, ein bestehendes ext3- oder ext4-Dateisystem „on the fly“ ohne Datenverlust in ein Btrfs-Dateisystem umzuwandeln. Btrfs hat nur sehr wenige on-disk Strukturen, die an einer bestimmten Stelle auf der Festplatte angelegt werden müssen,

was eine ideale Voraussetzung für eine solche Konversion ist. Weiterhin legt Btrfs eine Snapshot-Kopie des ursprünglichen ext4-Dateisystems an, so dass eine Konvertierung sogar wieder rückgängig gemacht werden kann.

Ressourcen

Weitere Informationen zu Btrfs:

- Btrfs Wiki: https://btrfs.wiki.kernel.org/
- IBM Research Report: BTRFS: The Linux B-tree Filesystem
  http://ibm.co/MFE73u
- Kapitel über Btrfs im Oracle Linux Administrator's Solutions Guide for Release 6:
  http://docs.oracle.com/cd/E37670_01/E41138/html/ol_localfs.html
- Kapitel über Btrfs im Oracle Linux Administrator's Solutions Guide for Release 6:
  http://docs.oracle.com/cd/E37670_01/E37355/html/ol_btrfs.html
- Oracle Linux Hands-on lab: Hands-on lab - Storage Management with Btrfs
  https://wikis.oracle.com/display/oraclelinux/Hands-on+lab+-+Storage+Management+with+Btrfs
- OTN Artikel: How I Got Started with the Btrfs File System for Oracle Linux
  http://www.oracle.com/technetwork/articles/servers-storage-admin/gettingstarted-btrfs-1695246.html
- OTN Artikel: How I Use the Advanced Capabilities of Btrfs
  http://www.oracle.com/technetwork/articles/servers-storage-admin/advanced-btrfs-1734952.html
- OTN Blog: Save disk space on Linux by cloning files on Btrfs and OCFS2
  https://blogs.oracle.com/OTNGarage/entry/save_disk_space_on_linux
- Btrfs Wiki: Conversion from ext3
  https://btrfs.wiki.kernel.org/index.php/Conversion_from_Ext3

Kontaktadresse:
Lenz Grimmer
Oracle Deutschland B.V. & Co. KG
Riesstraße 25
D-80992 München

Telefon: +49 (0) 40 41267308
E-Mail lenz.grimmer@oracle.com
Internet: http://www.oracle.com/linux