Intro to Btrfs: CoW for All!

Neal Gompa
(Conan Kudo [ニール・ゴンパ])

Who am I?

- Professional technologist
- Humble maintainer of a handful of packages in the Fedora Project
- Diligent follower of telecommunications industry
- Associate SQA Engineer at Datto, Inc

Contact Points:

- Twitter: @Det_Conan_Kudo
- Google+: +NealGompa

So… What is Btrfs?

From the Btrfs wiki:

- *Btrfs is a new copy on write (CoW) filesystem for Linux aimed at implementing advanced features while focusing on fault tolerance, repair and easy administration.*

Err, what is Copy on Write?

The label Copy on Write (CoW) refers to a type of filesystem optimization strategy where each modification to the filesystem is written in a new location while the original remains preserved.

By doing this, it’s possible to preserve each instance of the filesystem and move back and forth through the instances.

It is ***NOT*** a replacement for proper backups, but it does provide some safety that isn’t possible in traditional filesystems.

How big can Btrfs get?

64-bit filesystem, max volume size is 16 EiB

- Approximately 18,446,744 terabytes!
- At 100GB per 4K (QHD) full-length film, it’d take 184,467,441 copies to fill the whole volume at max size!
- This is more data than what is even possible (or even desired!) to record today on any single disk or disk array.

What are the features of Btrfs?

- Space efficient storage/packing of small files
- Space efficient indexing of directories
- Subvolumes & quota support for subvolumes
- Read-only and writable snapshots
- Sending/receiving snapshot data
- SSD awareness and SSD-specific optimizations
- Integrated disk management & multiple disk support
  - RAID 0, 1, 5, 6, 10 support
  - Dynamic resizing (shrink/grow) arrays/volumes after initial array creation
- Transparent on-disk compression
- Seeding from other filesystems
- And much more...

Subvolumes? Snapshots?

Subvolumes are subsections of a volume that can be independently managed. This is useful if you want to have different snapshotting schedules for portions of your volume.

Snapshots are instances of (sub)volumes that are preserved. With the appropriate tools and configuration, snapshots can be used as a means to provide “Time Machine” style data recovery or even to save a system from a bad software install/upgrade.

This sounds like ZFS…?

Btrfs:

- Designed for Linux
- Uses Linux facilities
- Integrated into kernel mainline
- Actively maintained in the kernel by a number of people and companies
- 16 EiB max volume size
- Been in rapid development since 2007, stabilized last year
- Disk arrays can be altered after initial creation
- Can be seeded from other filesystems

ZFS:

- Transplant from Solaris
- Uses own facilities
  - Uses more than double the system memory as a consequence
- Must be retrieved and built separately
- No longer actively maintained by original developer (Sun/Oracle)
  - OpenZFS Project has taken over development
- 256 ZiB max volume size (262,144 EiB)
- Developed in 2005, active development resumed from 2010 onward with illumos Project and later OpenZFS Project
- Disk arrays are frozen after creation
- No seeding capabilities

Both are CoW filesystems with nearly identical capabilities!

Who made Btrfs?

It is principally developed by:

So why use Btrfs?

As a filesystem that is developed within the mainline kernel, it takes advantage of facilities provided in the kernel to be more efficient at doing operations on devices.

Linux distributions also have support for Btrfs out of the box, and can be used with minimal effort.

Isn’t Btrfs too new to be stable?

I’ve been using Btrfs for over two months with no issues. Marc Merlin at Google has been using it for *three years*!

That said, Btrfs did finalize its on-disk format and implemented all the features expected in CoW filesystems last year, so articles and papers about the filesystem before spring 2015 on reliability and features should be taken with a grain of salt.

The traditional definition of filesystem stability is that the code has remained basically the same for *x* number of years, but that doesn’t mean the code and filesystem itself isn’t stable long before that. Even “stable” filesystems can get massive corruption bugs. Also, there are several companies using Btrfs in production today.

Who uses Btrfs?

It is used in production by:

Using features of Btrfs is hard, right?

Not at all! There has been a lot of work done to make Btrfs easy to use and maintain.

I’ve been using Btrfs for two months and taking advantage of the capabilities of the filesystem with nearly no effort!

The Btrfs user's best friend: Snapper

Snapper was created by SUSE to make it easier to trigger snapshot generation and manage snapshots. It's a user-friendly way for creating, manipulating, and deleting snapshots for Btrfs volumes/subvolumes.

It also integrates with other tools to be able to generate them on events. It supports generating snapshots on configurable time periods and predefined events, too.

It's available for a number of distributions, but openSUSE 13.2+/Tumbleweed and Fedora 22+ both offer integration with Snapper to use it for snapshotting on events automatically (such as software updates).

Setting up Snapper with integration

Fedora 22+

- Assuming you've set up Fedora with Btrfs for / and /home during installation, run the following ***as root***:
  1. [~]# dnf install *dnf-*snapper
  2. [~]# snapper create-config -c <root-subvol-cfg-name> /
  3. [~]# snapper create-config -c <home-subvol-cfg-name> /home
     - This is optional, and only if you want snapshots of your home subvolume
- Done! It'll automatically take snapshots whenever DNF is triggered, as well as regular temporal snapshots.

openSUSE 13.2+/Tumbleweed

- If you installed openSUSE 13.2+/Tumbleweed with Btrfs (the default), then it's already pre-configured for you for subvolumes other than /home. You can use YaST to configure Snapper, or use the snapper CLI tool.

More information on how to use Snapper is on its website.

How do I manage subvolumes?

Subvolume management is done through the btrfs program, which must be run as root.

- To create a subvolume:
  - [~]# btrfs subvolume create <path>
    - The path you pass must not exist as a directory or file, and must be on a Btrfs volume.
- To delete a subvolume:
  - [~]# btrfs subvolume delete <path>
    - The deletion process is lazy for data blocks, but immediate for volume structure.
- List subvolumes:
  - [~]# btrfs subvolume list <path>
    - This will list subvolumes from the parent declared in <path>. I usually use “/” to see all of them. You can also list from another (sub)volume to see subvolumes within the (sub)volume.

What about all the other features?

The functions for various actions for the filesystem are managed through the btrfs program (part of the btrfs-progs package in most distributions). Unlike Snapper, the btrfs program is less intuitive to use, but it is certainly usable and capable.

Documentation on btrfs-progs is available on the Btrfs Wiki.

What’s the best way to get started?

The best way to start using Btrfs is to install a Linux distribution that has a recent kernel (4.x.x preferred) with Btrfs from the beginning and go from there.

Personally, I recommend either Fedora 22 or openSUSE Tumbleweed. Fedora 22 arrived on May 26, 2015 with kernel 4.0.4, and openSUSE Tumbleweed is a “rolling release” distribution, so it always has the latest working code. Both distributions also provide excellent supporting functionality, as I mentioned earlier.

Some additional resources...

- Btrfs Wiki: https://btrfs.wiki.kernel.org/
- Marc Merlin: *Why you should consider using btrfs ... like Google does.*
  - YouTube recording: https://www.youtube.com/watch?v=6DplcPrQjvA
  - PDF slides: http://marc.merlins.org/linux/talks/2015/Btrfs-LCA2015/Btrfs.pdf
- Snapper website: http://snapper.io/
- Matthias G. Eckermann: *Using btrfs Snapshots for Full System Rollback*: http://events.linuxfoundation.org/sites/events/files/slides/EndUserSummit_2014_06_btrfs_full_system_rollback.pdf
- openSUSE wiki portal on Snapper: https://en.opensuse.org/Portal:Snapper
- Tutorial on how to set up Btrfs on Fedora 21 (still useful for Fedora 22): http://www.linuxbsdos.com/2014/12/11/how-to-install-fedora-21-workstation-cinnamon-on-a-btrfs-filesystem/
- Fedora Project: https://getfedora.org/
- openSUSE Tumbleweed: https://en.opensuse.org/openSUSE:Tumbleweed

The End

Any Questions?