Introducing Btrfs

Let Linux “have a CoW!”

Neal Gompa

Velocity Limitless

Who am I?

- Open Source Advocate
- Owner and Principal Consultant at Velocity Limitless
- Contributor to Fedora, CentOS, openSUSE, and Mageia
  - Member of Fedora Engineering Steering Committee (FESCo)
  - Member of Fedora Workstation, Fedora Cloud, and Fedora Server WGs
  - Chair of Fedora KDE SIG
  - Co-chair of CentOS Hyperscale SIG
  - Member of CentOS Alternative Images SIG
  - Former member of the openSUSE Board
  - Member of the openSUSE Heroes (Infrastructure)
  - Member of the Mageia Council and Mageia.org Board
- Contributor to RPM, DNF, KIWI, Koji, Open Build Service, and various other similar projects
- KDE contributor
  - Member of KDE, e.V. and X.Org Foundation
- Co-host of the Sudo Show podcast

Velocity
Limitless

Btrfs?

What is Btrfs?

From the Btrfs wiki:

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

What is Copy-on-Write (CoW)?

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 much Btrfs?

Btrfs is a 64-bit filesystem with a max volume size of 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.

Velocity
Limitless

Features?!

- Space efficient storage/packing of small files
- Space efficient indexing of directories
- Subvolumes & quota support for subvolumes
- Read-only and writable snapshots
- Sending/receiving volume data with efficient deltas
- 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
- Cryptographically secure filesystem authentication and tamper protection
- Seeding from other filesystems
- And much more...

Subvolumes? Snapshots? Authentication?

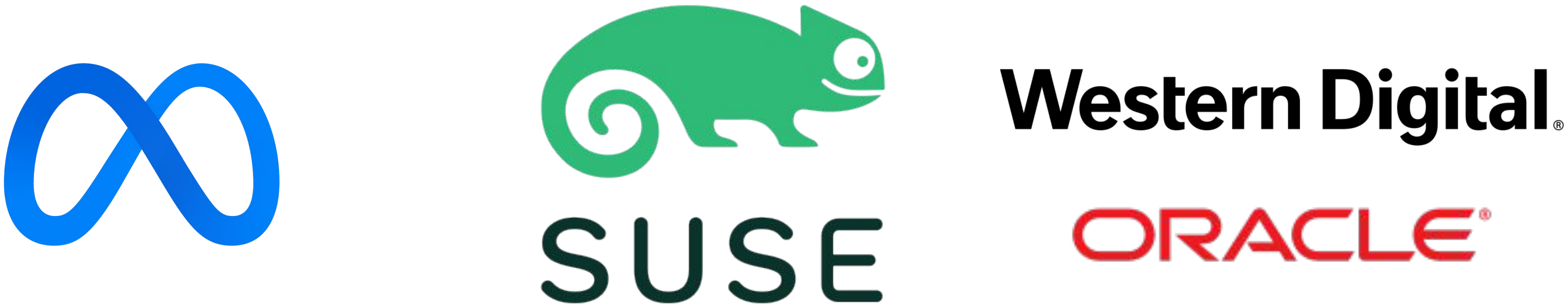
Subvolumes are subsections of a volume that can be independently managed. This is useful if you want to have different snapshotting schedules for portions of your volume.

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.

Authentication refers to the capability to have cryptographically strong checksums of data used to verify the integrity of the data on every access. This is used to implement tamperproofing for read-only data.

The makers...

It is principally developed by:



… And the shakers!

It is used in production by:

openSUSE

Rockstor

Synology

fedora

NETGEAR

So why?

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.

There is also a large and continuously growing community of practitioners that offer creative solutions to various problems by leveraging features in the Btrfs filesystem.

Speaking of distributions…

Btrfs and openSUSE

Current state on openSUSE

- Btrfs introduced in 2012 to SUSE Linux Enterprise
- Btrfs is the default for the operating system data since 2014
  - Landed in openSUSE 13.2 and SUSE Linux Enterprise 12
- Btrfs is the default for ***all*** data since 2018
  - Landed in openSUSE Tumbleweed in fall of 2018
- Full system snapshots and rollbacks with Btrfs since 2014
- No compression
- Disk encryption uses LUKS
  - LUKS with Btrfs means only full disk encryption is possible
- Read-only root filesystem with transactional updates built on Btrfs
  - Shell implementation with Zypper
  - C++ plugin for libdnf (MicroDNF, PackageKit)

Btrfs and Fedora Linux

Velocity
Limitless

Current state on Fedora Linux

- Anaconda has been configured to install non-server variants with Btrfs
- Disk images of desktop variants provide Btrfs-based images
- Cloud images use Btrfs
- Zstd compression by default
- Full support in all commonly used image build tools
    - KIWI, osbuild, mkosi
- /boot is not on Btrfs currently by default
    - Installation *is* possible with /boot as Btrfs subvolume or separate Btrfs volume
- Disk encryption uses LUKS
    - LUKS with Btrfs means only full disk encryption is possible

Velocity Limitless

Future plans (Fedora 44/45?)

- /boot on Btrfs by default
  - Proposed change for cloud images for F44
  - Pending coordination with bootloader team for non-cloud environments
- Online/Live full or partial disk encryption using Btrfs native encryption
  - Pending upstream work
  - Will require moving /boot to Btrfs
- Simpler setup for full system snapshotting and boot-to-snapshot
  - Pending coordination with bootloader team, the DNF team, and boom+snapm developers
- Upcoming work tracked in fedora-btrfs/project Pagure project

Btrfs and AlmaLinux

Velocity
Limitless

Current state on AlmaLinux

Available since AlmaLinux OS 10.1

- Anaconda offers the option of Btrfs at install-time with a similar configuration to Fedora with all the same features and caveats
- Operating system storage stack (UDisks, *et al*) have Btrfs support enabled

Future plans for AlmaLinux

- Enabling Btrfs support throughout the full package set in AlmaLinux where previously disabled
- Btrfs based alternative disk and cloud images
- Building communities of practice around Btrfs

The End

www.velocitylimitless.com