BTRFS & ZFS

蕭從恩 (tsn) 2016.01.28
Computer Center, Dept. of CS, NCTU

Overview

- btrfs: stable version since 2014 by Oracle.
  - GPL
  - Native Linux kernel support
  - In active development
- zfs: released as a part of OpenSolaris in 2005 by sun.
  - Sun's CDDL (not compatible with GPL).
  - Can't be distributed as a part of Linux.
  - Linux support by userspace driver or kernel patch.

Common Features

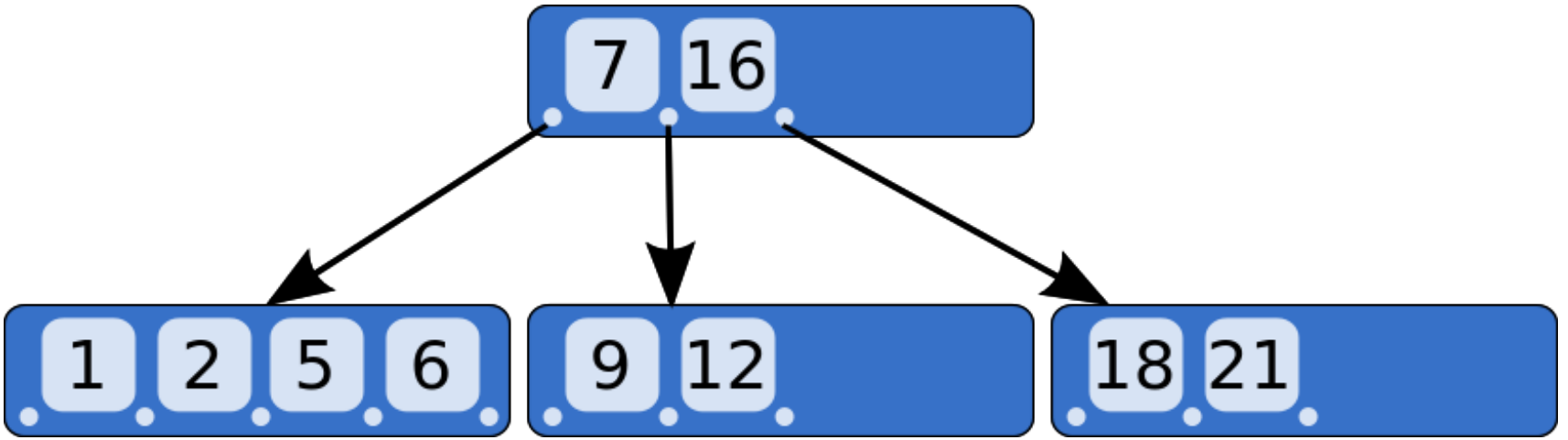
- copy on write
- subvolumes
- storage pool (dynamically add/remove partitions)
- RAID[0,1] support
- snapshot
- cp --reflink
- transparent compression
- scrub (online file system check)

Common Features

- data checksum
- data encryption
- send/receive volumes
- ...and lots of features

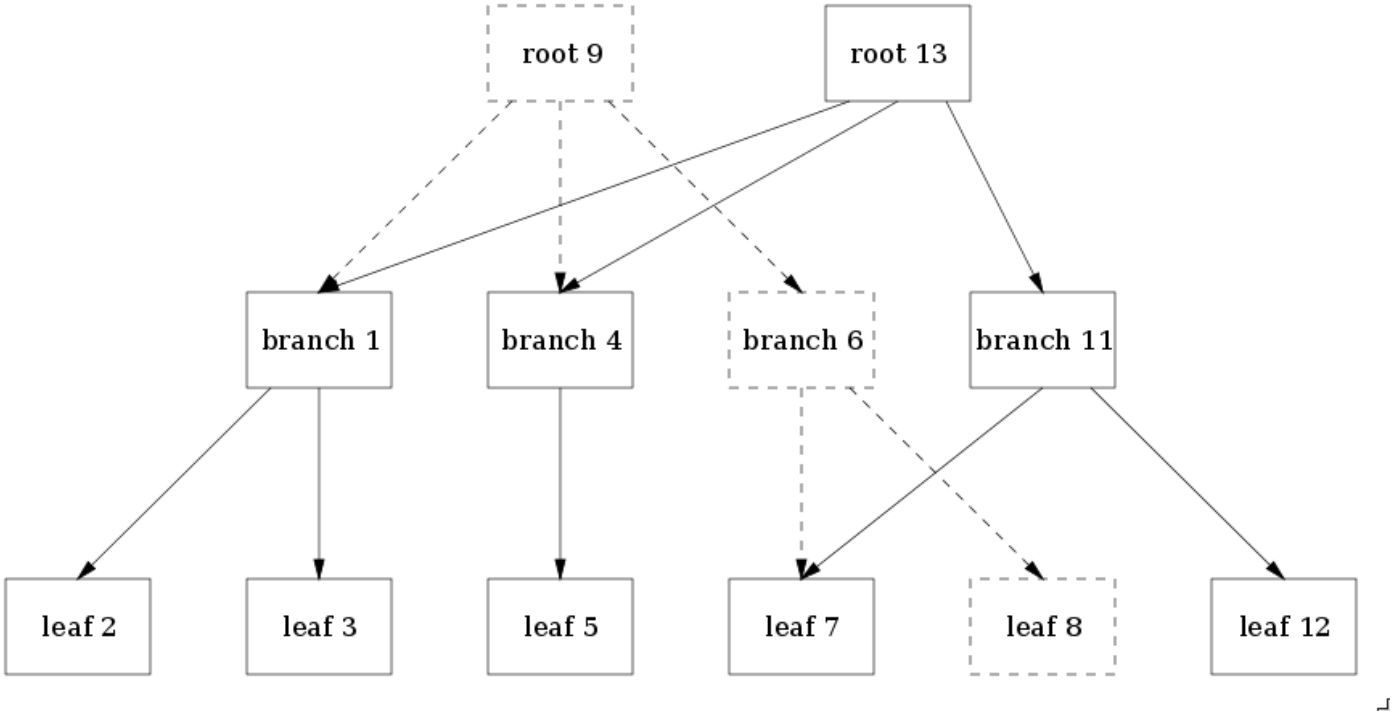
BTRFS: B-tree

- Generalized binary tree (not just two childs)
-



BTRFS: Copy on Write

- how the append-only btree works
-



BTRFS: Turn off Copy on Write

- For a file or directory or volume that is to be write very frequently from random location in it.
- We can turn off CoW for specific file or directory, especially database files, vm disk images, browser profiles...etc

BTRFS: Turn off Copy on Write

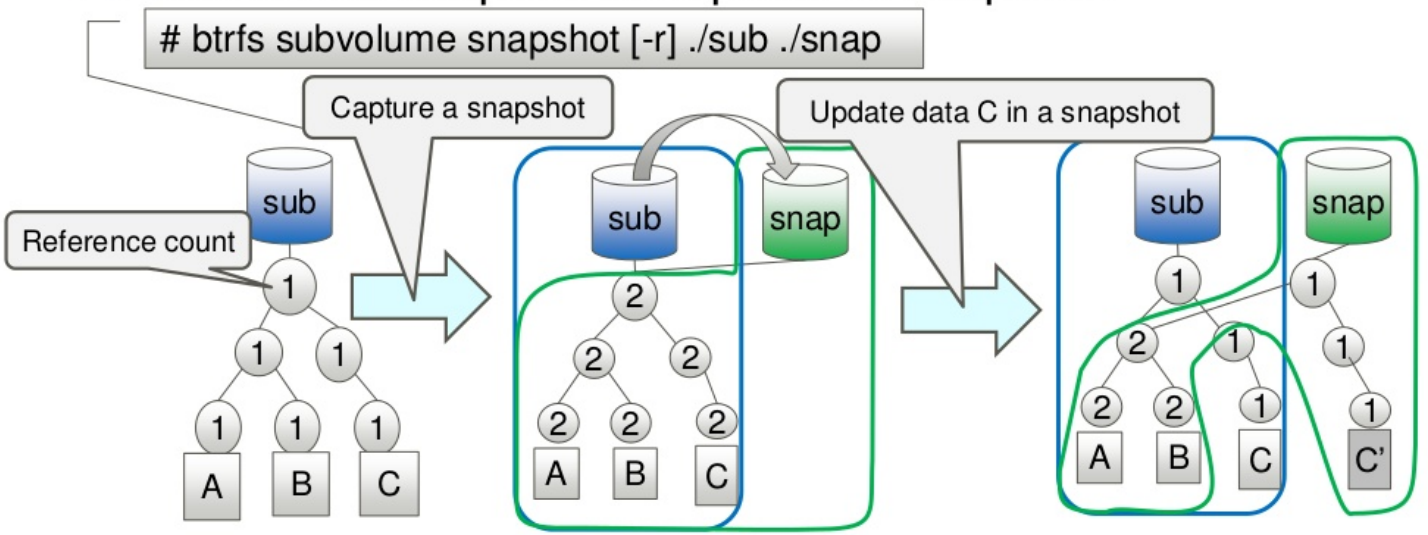
- example
  - touch vm-image.raw
  - chattr +C vm-image.raw
  - fallocate -l10g vm-image.raw
- Mount option: nodatacow

BTRFS: Snapshot

Snapshot

FUJITSU

- Copy of a subvolume
  - Far faster than LVM
    - Not a full copy, but only update metadata in CoW style
  - Readonly snapshot: with –r option
  - Incremental snapshot: snapshot of snapshot



BTRFS: Snapshot

- Snapshots are not backups!
- Snapshot takes almost no additional space on disks
- When you delete a file which is in a snapshot, it won't free any space. (just decrease the reference count.)

BTRFS v.s. ZFS

Reasons to Use BTRFS

- GPL
- Native Linux Support
- Love(?) from Oracle
- Faster than ext4 in most tests.
- Less memory usage than zfs
- Google Admin Encourages Trying Btrfs, Not ZFS On Linux
- Active development by lots of big companies.

Reasons to Use BTRFS

- btrfs-convert: In-place convert to ext3/4 (there are some issues in kernel 4.0+)
- Design for general purposes, not just high-end servers

Reasons NOT to Use BTRFS

- Some features are still in development or not stable (RAID 5/6)
- Need to be rebalanced
- Need to be defragmented

Reasons to Use ZFS

- Mature and more robust
- More features than btrfs
  - RAID 5/6
  - Online deduplication
- Design for large servers
- Better support in FreeBSD
- https://rudd-o.com/linux-and-free-software/ways-in-which-zfs-is-better-than-btrfs
- http://www.osdevcon.org/2009/slides/zfs_internals_uli_graef.pdf

Reasons NOT to Use ZFS

- No official support in Linux kernel.
- Large memory usage in Linux

My Choice is BTRFS

- I will use btrfs instead of ZFS
  - we use Linux
  - 後勢看漲
- When on a VM, we can still use btrfs without raid.

Something even newer: BcacheFS

- A New Linux File-System Aims For Speed While Having ZFS/Btrfs-Like Features
- SSDs for caching, HDDs for storage.
- Based on Linux kernel block layer: https://wiki.archlinux.org/index.php/Bcache
- Merged in Linux kernel in 2013.

Reference

- LinuxCon 2014 slide
- Wikipedia: btrfs, ZFS,
- Btrfs Wiki
- http://louwrentius.com/the-sorry-state-of-cow-file-systems.html
- BTRFS: The Linux B-Tree Filesystem
- how the append-only btree works
- How I Use the Advanced Capabilities of Btrfs
- http://www.slideshare.net/fj_staoru_takeuchi/btrfs-current-status-andfutureprospects