Btrfs Update

Liu Bo

Introduction

- Liu Bo
  - Working on btrfs
  - Member of James Morris's upstream team
  - Background
    - Work on btrfs since 2010
    - Made many contributions to the project, both on features and performance side

Btrfs community

- Filesystems span many different use cases
- Btrfs has contributors from many different companies(including Oracle, Redhat, Strato, SUSE, Fujitsu, Intel, IBM) and many individuals
- Broad community ensures btrfs is full of unique features

Btrfs features overview

- Copy on write on data and metadata
- Efficient writable/readonly snapshots
- Internal Raid with restriping
- Online device management
- Online scrub
- Transparent compression
- Crcs for data and metadata
- Etc.

What's new in the last year(2012)?

Kernel: features

- Preliminary Raid 5/6 support
- Snapshot-aware defrag
- Get/set filesystem label
- Send/receive
- Online Device replace
- Remove limitation on hardlinks in a single directory
- Subvolume-aware quotas
- Device IO error statistics
- Bigger metadata

Kernel: features, cont

- Restriper
- Recovery mode
- Online Filesystem scrub
- Auto defragmentation
-

Raid 5/6

- This builds on David Woodhouse's original Btrfs raid5/6 implementation.
- read/modify/write cycle is done after the higher levels of the filesystem have prepared a given bio, it's actually done when we map bios down to the individual drives.
- Scrub and discard doesn't (yet) work on raid5/6.
- plugging
- its performance is overall better than MD.

Snapshot-aware defrag

- As we defragment files, we break any sharing from other snapshots, which is not good.

- Update other snapshots' references to new blocks after defragment

Get/set filesystem label

- Mount filesystem by 'LABEL='
- Btrfs filesystem label

Send/receive

- Determine difference between snapshots
- Make a file which consists of a stream of instructions meant to be replayed 1:1 on the receiving side.
- Only the send side is happening in-kernel. Receive is happening in user-space.
- On experimental stage

Online device replace

- you don't need to unmount it or stop active tasks
- Safe to crash or power loss
- Instead of adding a new disk & deleting an old one, just replace!

Remove limitation on hardlinks

- The limitation of hardlinks is from btree leaf size as hardlinks just live in btree leaf.

- This introduces Extended refs,
  - It doesn't replace the existing ref array.
  - An inode gets an extended ref for a given link only after the ref array has been filled.

- Incompatible with old kernels
  - BTRFS_FEATURE_INCOMPAT_EXTENDED_IREF

Subvolume-aware quota

- Similar to directory quota(used in ext4, xfs)
- qgroups only apply to subvolumes/snapshots;
- set limits on a per-subvolume basis or create quota groups and toss multiple subvolumes into a big group.

Device IO error statistics

- The goal is to detect
  - when drives start to get an increased error rate,
  - when drives should be replaced soon.
- IO errors:
  - read, write and flush
  - checksum errors and corrupted blocks

Bigger metadata

- We do have the max metadata block size, 64K.
  - This limit is somewhat artificial, but the memmove costs go through the roof for larger blocks.
- leafsize=16K performs best in most workloads.

restriper

- do selective profile changing and selective balancing
- pausing/resuming
- Report progress to users

Recovery mode

- record information about most of the roots in the last 4 commits.
- With -o recovery, use the root history log when we're not able to read the root of some vital trees

Online filesystem scrub

- Scrubbing verifies data and metadata integrity
- Duplicate copies are checked in parallel

Auto defragment

- detect small random writes into files and queue the up for an auto defrag process
- Benefits Random write performance

Kernel: performance and improvement

- Direct IO speedup (lockless read/write)
- Fsync speedup
- Scrub speedup thanks to new read-ahead infrastructure
- Improved error handling
  - go readonly gracefully on errors instead of crashing
- A great amount of bug-fixes and cleanups

btrfs-progs

- Btrfsck with a lot of fixes
- Btrfs list snapshot becomes more flexible
- Related commands of new features

Ongoing

- Online/offline deduplication
- Send speedup by caching subvolumes' uuid
- Hot relocation
-

Questions?