Btrfsck

From btrfs Wiki

Before trying fsck

If you have a broken filesystem, you should probably look at the recovery or repair tools or you can read Marc MERLIN's page that explains the different ways to check and fix a btrfs filesystem (http://marc.merlins.org/perso/btrfs/post_2014-03-19_Btrfs-Tips_-Btrfs-Scrub-and-Btrfs-Filesystem-Repair.html) .

In a nutshell, you should look at:

- btrfs scrub to detect issues on live filesystems
- look at btrfs detected errors in syslog (look at Marc's blog above on how to use sec.pl to do this)
- mount -o ro,recovery to mount a filesystem with issues
- btrfs-zero-log might help in specific cases. Go read Btrfs-zero-log
- btrfs restore will help you copy data off a broken btrfs filesystem. See its page: Restore
- btrfs check --repair, aka btrfsck is your last option if the ones above have not worked.

btrfs check --repair

btrfs check --repair (used to be called btrfsck) checks consistency of a btrfs filesystem, and optionally repair some types of breakage. It can only be run on an unmounted FS.

Note that while this tool should be able to repair broken filesystems, it is still relatively new code, and has not seen widespread testing on a large range of real-life breakage. It is possible (but highly unlikely in the most recent versions) that it may cause additional damage in the process of repair. btrfs check --repair received significant updates prior to v4.0. You are strongly recommended to use a version after v4.0 when trying to recover a filesystem.

It is highly recommended to get the latest version of btrfs-tools and recover important data with the steps listed above (mount -o recovery,ro and btrfs restore) *before* trying btrfs check --repair

```
btrfs check --repair <device>
```

See the man page for exact up to date usage info: Manpage/btrfs-check

If all goes well, it should look something like this:

```
gargamel:/mnt# btrfs check --repair /dev/mapper/space  2>&1 | tee /tmp/repair
Fixed 0 roots.
checking extents
checking free space cache
checking fs roots
checking csums
checking root refs
enabling repair mode
Checking filesystem on /dev/mapper/space
UUID: 77077815-4ca2-44cd-a83a-b1aeb4607df4
cache and super generation don't match, space cache will be invalidated
found 294976426877 bytes used err is 0
total csum bytes: 636680648
total tree bytes: 6981910528
total fs tree bytes: 5149491200
total extent tree bytes: 895369216
btree space waste bytes: 1860661732
file data blocks allocated: 651971338240
 referenced 676102934528
Btrfs v3.17
```

btrfsck

> **Deprecated** The tool btrfsck functionality has been merged to 'btrfs check' command. See Manpage/btrfs-check.

For comparison, the old obsolete btrfsck looks like this:

```
gargamel:/mnt/btrfs_pool1# btrfsck /dev/mapper/raid0d1
Checking filesystem on /dev/mapper/raid0d1
UUID: 7f0c0b97-befe-4183-ab33-dba5b0059c3a
checking extents
checking free space cache
checking fs roots
checking csums
checking root refs
found 15735927605 bytes used err is 0
```

```
total csum bytes: 626165256
total tree bytes: 3327885312
total fs tree bytes: 2231566336
total extent tree bytes: 352927744
btree space waste bytes: 499663388
file data blocks allocated: 641351032832
 referenced 648040566784
```

My personal experience with the old btrfsck is that it could run for 24H or more on a filesystem of a few terabytes, and look like it's never going to complete. In my experience it's also not unusual to get tens to over a hundred thousand lines of output showing lines like:

```
Extent back ref already exists for 2148837945344 parent 0 root 257
leaf parent key incorrect 504993210368
bad block 504993210368
```

FIXME: are those real problems or unfortunate real problems that easily happen btrfs filesystems?

Retrieved from "https://btrfs.wiki.kernel.org/index.php?title=Btrfsck&oldid=29446"

- This page was last modified on 6 July 2015, at 22:16.