NAME

btrfs-scrub − scrub btrfs filesystem, verify block checksums

SYNOPSIS

btrfs scrub <*subcommand*> <*args*>

DESCRIPTION

btrfs scrub is used to scrub a mounted btrfs filesystem, which will read all data and metadata blocks from all devices and verify checksums. Automatically repair corrupted blocks if there's a correct copy available.

Note

Scrub is not a filesystem checker (fsck) and does not verify nor repair structural damage in the filesystem. It really only checks checksums of data and tree blocks, it doesn't ensure the content of tree blocks is valid and consistent. There's some validation performed when metadata blocks are read from disk but it's not extensive and cannot substitute full *btrfs check* run.

The user is supposed to run it manually or via a periodic system service. The recommended period is a month but could be less. The estimated device bandwidth utilization is about 80% on an idle filesystem. The IO priority class is by default *idle* so background scrub should not significantly interfere with normal filesystem operation. The IO scheduler set for the device(s) might not support the priority classes though.

The scrubbing status is recorded in */var/lib/btrfs/* in textual files named *scrub.status.UUID* for a filesystem identified by the given UUID. (Progress state is communicated through a named pipe in file *scrub.progress.UUID* in the same directory.) The status file is updated every 5 seconds. A resumed scrub will continue from the last saved position.

Scrub can be started only on a mounted filesystem, though it's possible to scrub only a selected device. See **scrub start** for more.

SUBCOMMAND

cancel <*path*>|<*device*>

If a scrub is running on the filesystem identified by *path* or *device*, cancel it.

If a *device* is specified, the corresponding filesystem is found and **btrfs scrub cancel** behaves as if it was called on that filesystem. The progress is saved in the status file so **btrfs scrub resume** can continue from the last position.

resume [−BdqrR] [−c <*ioprio_class*> −n <*ioprio_classdata*>] <*path*>|<*device*>

Resume a cancelled or interrupted scrub on the filesystem identified by *path* or on a given *device*. The starting point is read from the status file if it exists.

This does not start a new scrub if the last scrub finished successfully.

Options

see **scrub start**.

start [−BdqrRf] [−c <*ioprio_class*> −n <*ioprio_classdata*>] <*path*>|<*device*>

Start a scrub on all devices of the mounted filesystem identified by *path* or on a single *device*. If a scrub is already running, the new one will not start. A device of an unmounted filesystem cannot be scrubbed this way.

Without options, scrub is started as a background process. The automatic repairs of damaged copies is performed by default for block group profiles with redundancy.

The default IO priority of scrub is the idle class. The priority can be configured similar to the **ionice**(1) syntax using *−c* and *−n* options. Note that not all IO schedulers honor the ionice settings.

Options

−B
do not background and print scrub statistics when finished

−d
print separate statistics for each device of the filesystem (*−B* only) at the end

−r
run in read−only mode, do not attempt to correct anything, can be run on a read−only filesystem

−R
raw print mode, print full data instead of summary

−c <*ioprio_class*>
set IO priority class (see **ionice**(1) manpage)

−n <*ioprio_classdata*>
set IO priority classdata (see **ionice**(1) manpage)

−f
force starting new scrub even if a scrub is already running, this can useful when scrub status file is damaged and reports a running scrub although it is not, but should not normally be necessary

−q
(deprecated) alias for global *−q* option

status [options] <*path*>|<*device*>
Show status of a running scrub for the filesystem identified by *path* or for the specified *device*.

If no scrub is running, show statistics of the last finished or cancelled scrub for that filesystem or device.

Options

−d
print separate statistics for each device of the filesystem

−R
print all raw statistics without postprocessing as returned by the status ioctl

−−raw
print all numbers raw values in bytes without the *B* suffix

−−human−readable
print human friendly numbers, base 1024, this is the default

−−iec
select the 1024 base for the following options, according to the IEC standard

−−si
select the 1000 base for the following options, according to the SI standard

−−kbytes
show sizes in KiB, or kB with −−si

−−mbytes
show sizes in MiB, or MB with −−si

−−gbytes
show sizes in GiB, or GB with −−si

−−tbytes
show sizes in TiB, or TB with −−si

EXIT STATUS

btrfs scrub returns a zero exit status if it succeeds. Non zero is returned in case of failure:

1
: scrub couldn’t be performed

2
: there is nothing to resume

3
: scrub found uncorrectable errors

AVAILABILITY

btrfs is part of btrfs−progs. Please refer to the btrfs wiki **http://btrfs.wiki.kernel.org** for further details.

SEE ALSO

mkfs.btrfs(8), **ionice**(1)