How can I check for bad blocks on an LVM physical volume?

When you're using ext4, you can check for badblocks with the command `e2fsck -c /dev/sda1 # or whatever` . This will "blacklist" the blocks by adding them to the bad block inode.

What is the equivalent of this for an LVM2 physical volume? The filesystem on it is ext4, but presumably, the bad blocks that are detected will become invalid as the underlying LVM setup moves data around on the physical disk.

In other words, how can I check for bad blocks to not use in LVM?

/ lvm / badblocks

asked Sep 23 '13 at 20:19

strugee
7,508 11 49 90

3 Answers

> When you're using ext4, you can check for badblocks with the command `e2fsck -c /dev/sda1` or whatever. This will "blacklist" the blocks by adding them to the bad block inode.

`e2fsck -c` runs `badblocks` on the underlying hard disk. You can use the `badblocks` command directly on a LVM physical volume (assuming that the PV is in fact a hard disk, and not some other kind of virtual device like an MD software RAID device), just as you would use that command on a hard disk that contains an ext file system.

That won't add any kind of bad block information to the file system, but I don't really think that that's a useful feature of the file system; the hard disk is supposed to handle bad blocks.

Even better than `badblocks` is running a SMART selftest on the disk (replace `/dev/sdX` with the device name of your hard disk):

```
smartctl -t long /dev/sdX
smartctl -a /dev/sdX | less
```

The test ifself will take a few hours (it will tell you exactly how long). When it's done, you can query the result with `smartctl -a` , look for the self-test log. If it says "Completed successfully", your hard disk is fine.

> In other words, how can I check for bad blocks to not use in LVM?

As I said, the hard disk itself will ensure that it doesn't use damaged blocks and it will also relocate data from those blocks; that's not something that the file system or the LV has to do. On the other hand, when your hard disk has more than just a few bad blocks, you don't want something that relocates them, but you want to replace the whole hard disk because it is failing.

edited Sep 23 '13 at 20:48

answered Sep 23 '13 at 20:38

Martin von Wittich
9,118 2 27 54

2 You might want to check the e2fsck manpage and see what `-c` does before calling something complete nonsense. – derobert Sep 23 '13 at 20:40

@derobert oops... – Martin von Wittich Sep 23 '13 at 20:42

@derobert TIL. I've rewritten the wrong section. Thanks for the feedback! – Martin von Wittich Sep 23 '13 at 20:49

Indeed, rather than flag the blocks so the filesystem does not use them on modern disks you should simply write new data to the block and the disk will automatically remap the sector to a spare if it really is physically damaged. You can do that with `dd` . More often than you would think, the medium is actually fine and the data was just corrupted, so writing over it works fine without the need to remap. – psusi Sep 23 '13 at 23:09

"You can do that with `dd` " - but you still probably shouldn't. If you have an `md` raid, it can take care of the issue for you. @derobert will probably know what to do when the disk is not part of an `md` raid :) – Martin von Wittich Sep 24 '13 at 10:40

I'm pretty sure LVM doesn't handle bad blocks; it expects the underlying storage to. And most, if not all, modern hard disks do. You may need to perform a write to the sector, but the disk should remap it. (You may need it to do an offline surface scan first, with e.g., `smartctl /dev/sda -t offline`).

That said, LVM doesn't actually move data around unless you ask it, with e.g., `pvmove` . So you can use the ext4 badblocks feature; you'll just have to re-check for bad blocks if run `pvmove` . No common operation (such as `lvextend`) moves data.

Extend doesn't move data because LVM keeps a map saying "logical extents 0–99 are physical extents 200–299", and then when you extend it, it just adds "logical extents 100–199 are physical extents 100–199". Or even "logical extents 100–149 are physical extents 50–99; logical extents 150–199 are physical extents 140–189". LVM doesn't care that the physical extents aren't in order, or aren't contiguous.

answered Sep 23 '13 at 20:39

derobert
62.9k 7 128 187

`pvck` can check LVM metadata, after that consistency is the job of the filesystem. LVM is only about volume management so it doesn't need to care if the space constituting a particular extent is bad since higher level software catches those issues. LVM metadata only takes up the first (optionally also the last sector) of the physical volume anyways.

If just the first and last sectors of a reasonably large PV (such as you'd see in production) just happen to fail simultaneously, you basically have the sh*ttiest luck in the world since that's so astronomically unlikely. Otherwise, if the admin knows multiple sectors of the drive have been failing, most people are alright with just filing such things as this under "hard drive failed permanently and needs to be replaced."

If `pvck` returns an error, you can check to see if your LVM metadata is backed up in `/etc/lvm` somewhere. If it is you can do `pvcreate` specifying the backup copy to `--restorefile`

Syntax:

```
pvcreate --uuid "<UUID-of-target-PV>" --restorefile <Path-To-Metadata-Backup-File>
<path-to-PV-block-device>
```

Example:

```
pvcreate --uuid "2VydVW-TNiN-fz9Y-ElRu-D6ie-tXLp-GrwvHz" --restorefile
/etc/lvm/archive/vg_raid_00000-1085667159.vg /dev/sda2
```

If the restore doesn't work (for example, if the first sector is bad) you can re-do the above, but set `--metadatacopies 2` (or you might just go straight to doing that) which will attempt to write the metadata to the first and last sectors on the PV. When `pvscan` does its thing on boot it will check both places and if it finds metadata it will verify them against a checksum. If the checksum fails on the first sector but succeeds on the last sector you'll get a non-fatal error message.

Kind of manual and a pain, but then again this is part of the reason why people are excited to get a volume management redux with BTRFS. Most of the time it's not really that much of an

issue for the reasons derobert mentioned, and because the people who absolutely positively need to ensure continuity of data will usually do RAID and have a backup strategy.

edited Sep 24 '13 at 1:36

answered Sep 24 '13 at 1:28

Bratchley

11.2k 6 37 80