Wednesday, April 3. 2013

And no ... zfs scrub isn't a hidden fsck

I've got quite a number of tweets and mails with the question "But is zfs scrub" not something like fsck. And the answer is "Well ... no".

It's zpool scrub, not zfs scrub. So it can't be a fsck. Perhaps it's a pock, but not fsck (Okay, that's a really lame argument.)
Scrub is not part of the ZFS Posix layer, the entity that let you access a zpool with the semantics of a filesystem. It's done much deeper in ZFS and while it's following a tree there, this tree has nothing to do with your filesystem structure.(Ah, not much better reasoning.)
You don't run zfs scrub on a unimportable pool, you run run zpool clear -F, which starts the txg rollback to make the pool importable again.(Don't know if this one is more convincing)
A scrub works as well on a ZFS Emulated Volume.It can't do a filesystem check on it. Obviously, as Solaris has no idea for example how to check a ext2 file system that you are writing via iSCSI into a zvol. However the scrub has still to work.(Perhaps a little better.)
I've got one tweet saying that it finds silent corruption in the data, thus it's a filesystem check. What I find interesting about this, is the point that this definition would exclude other fsck from being fsck, as they just check the validity of the metadata and not the validity of the data. zpool scrub is a data validity checker, not a filesystem checker. It checks if you are reading the data from a location that you have once written to this location. Whatever this data is. As far as i understand the source, it even doesn't understand the concept of "filesystems" at all.
What is zpool scrub? It is not much more than reading everything. However the repair mechanism is the same as the one when you read a block and the checksum on-disk doesn't match the checksum computed. The scrub has no own repair code. In the case an error is detected , for example for RAID1 the known good copy is written in the place of the bad copy. Whereas the repair while normal reading is more or less accidental (you've stumbled over an error and reading at one location doesn't mean that other locations are correct, you have to check all locations to be sure). Scrub is the forced search for such incorrectnesses on all copies of a block stored in a pool. With a scrub you ensure that all redundancies are correct. At end scrub checks if the data on-disk is still the data you have written once by other means (ZPL or ZVOL) on it. It simply doesn't care about the structure, as it doesn't know about the structure. It's like memory scrubbing or RAID scrubbing. When your trigger a scrubbing in your RAID array the array gives nothing what's on it.

So, no zfs scrub is no fsck, it has a different job. It might look like a fsck, but it isn't. Forget everything how you expect filesystems internals from your pre-ZFS knowledge. I can give just one important advice in regard of understanding ZFS. An advice that was already used on the first presentations about ZFS: Free your mind!

Posted by Joerg Moellenkamp in English, Operating Systems, Solaris at 22:54

Interesting and educational. Thankfully, zpool scrub is one of those features which I have never really had cause to use. It is interesting to note that the significant function of the tool is to read every active block on the volume.

I can see this being very useful in a situation were large amounts of data rarely are read and are for the most part static. It will become part of my periodic maintenance on my spinning rust.
Anonymous on Apr 4 2013, 00:10

Hi,

As always great post. I however have a question. Does the zpool scrub command also try and re-balance the data over the available disks (if multiple disk are in the pool strip or raidZ) ?
Anonymous on Apr 4 2013, 10:08