OpenZFS

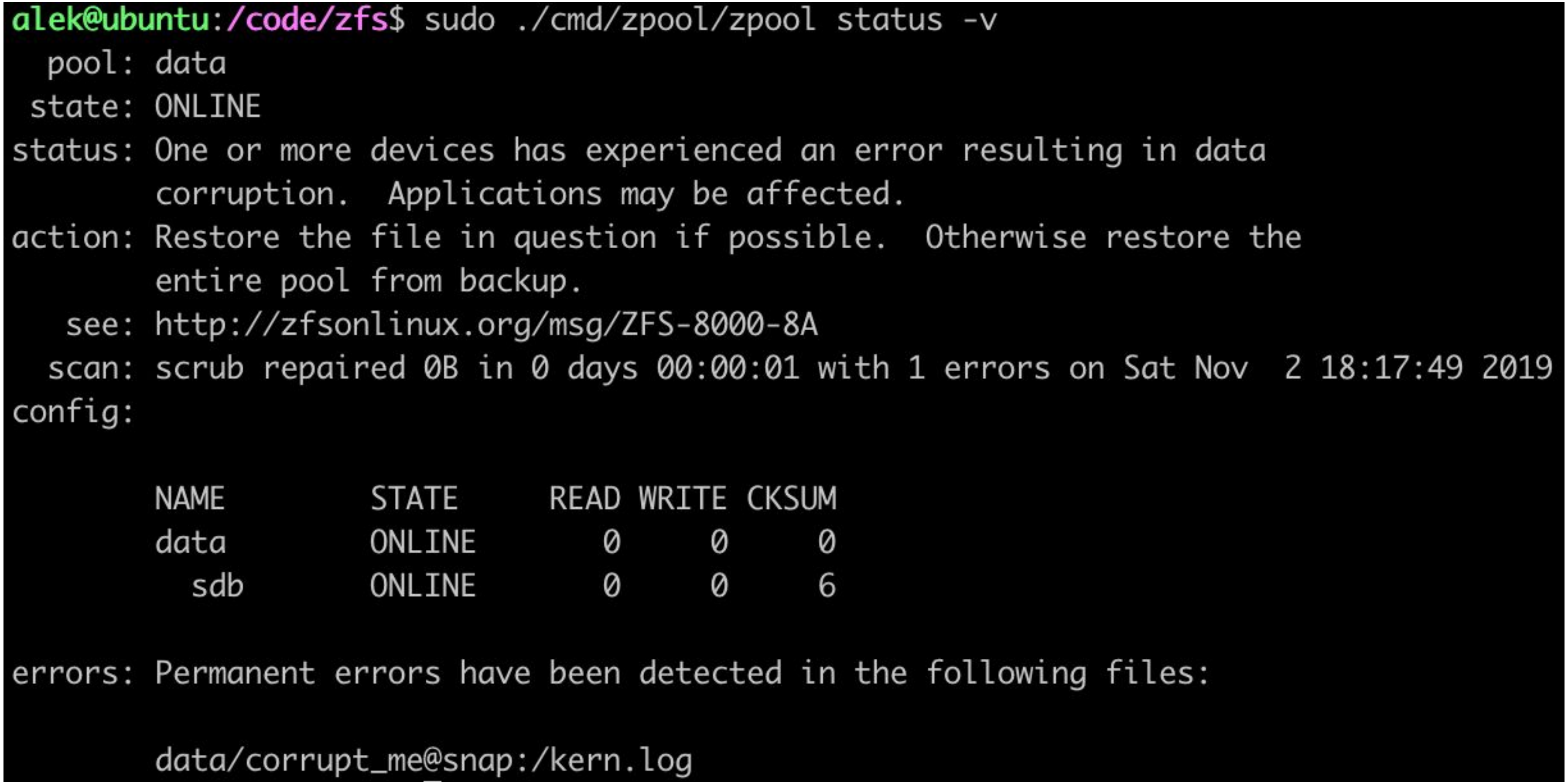
Healing data corruption w/ ZFS receive

•••

Alek Pinchuk
2019 OpenZFS Dev Summit

@alek_says
apinchuk@axcient.com

zpool status -v data



```
alek@ubuntu:/code/zfs$ sudo ./cmd/zpool/zpool status -v
  pool: data
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://zfsonlinux.org/msg/ZFS-8000-8A
  scan: scrub repaired 0B in 0 days 00:00:01 with 1 errors on Sat Nov  2 18:17:49 2019
config:

        NAME        STATE     READ WRITE CKSUM
        data        ONLINE       0     0     0
          sdb       ONLINE       0     0     6

errors: Permanent errors have been detected in the following files:

        data/corrupt_me@snap:/kern.log
```

Corrective (-c) receive motivation

OpenZFS

- datto has > 600 PB stored in “OpenZFS on Linux” pools
- Thanks to send/recv remote copies of zfs data are common
- Currently permanent data corruption can’t be fixed
- Tom suggested implementing send stream based healing
- Corrective? Why not healing receive?
    - ‘zfs recv -h’ was taken for receiving holds

How corrective recv works p.1

- https://github.com/zfsonlinux/zfs/pull/9323
- zfs recv -c pool/dataset@snap < /tmp/sendfile
- Sendfile contain GUID of the snapshot that was used to make the sendfile
    - Check the GUID of @snap to make sure it matches GUID in the sendfile
        - Send stream data can be used for healing dataset

How corrective recv works p.2

- Each DRR_WRITE and DRR_SPILL send stream record
    - Includes object set, object, offset, size and data
    - get the corresponding block pointer for the on-disk data
- Read the corresponding block from disk
- If the read returns ECKSUM, then use the good data from the send stream to reconstruct the bad block
- **Checksum the reconstructed block to make sure it has the same checksum as the one on disk**
- If the checksums matched - issue a zio_rewrite() of the bad block with the reconstructed block.

How corrective recv works p.3

- After rewrite is done re-read the block to make sure corruption was fixed
- Finally remove the healed data errors from the list of errors
- All reads async, rewrite currently a sync write

Limitations

OpenZFS

- GUIDs must match between snapshot and send stream
- Data encrypted on-disk but send stream is not encrypted
  - Need to re-encrypt send stream block - WiP
- Metadata cannot be healed
  - DRR_WRITE & DRR_SPILL records have all needed data to reconstruct block
  - Metadata (DRR_OBJECT etc) block info like birthtime (TXG #) is not in send stream

Future work

- “provide a way for a corrupted pool to tell a backup system to generate a minimal send stream in such a way as to enable the corrupted pool to be healed with this minimal send stream”
    - Needs communication between corrupted ⇐⇒ replica datasets

Testing

OpenZFS

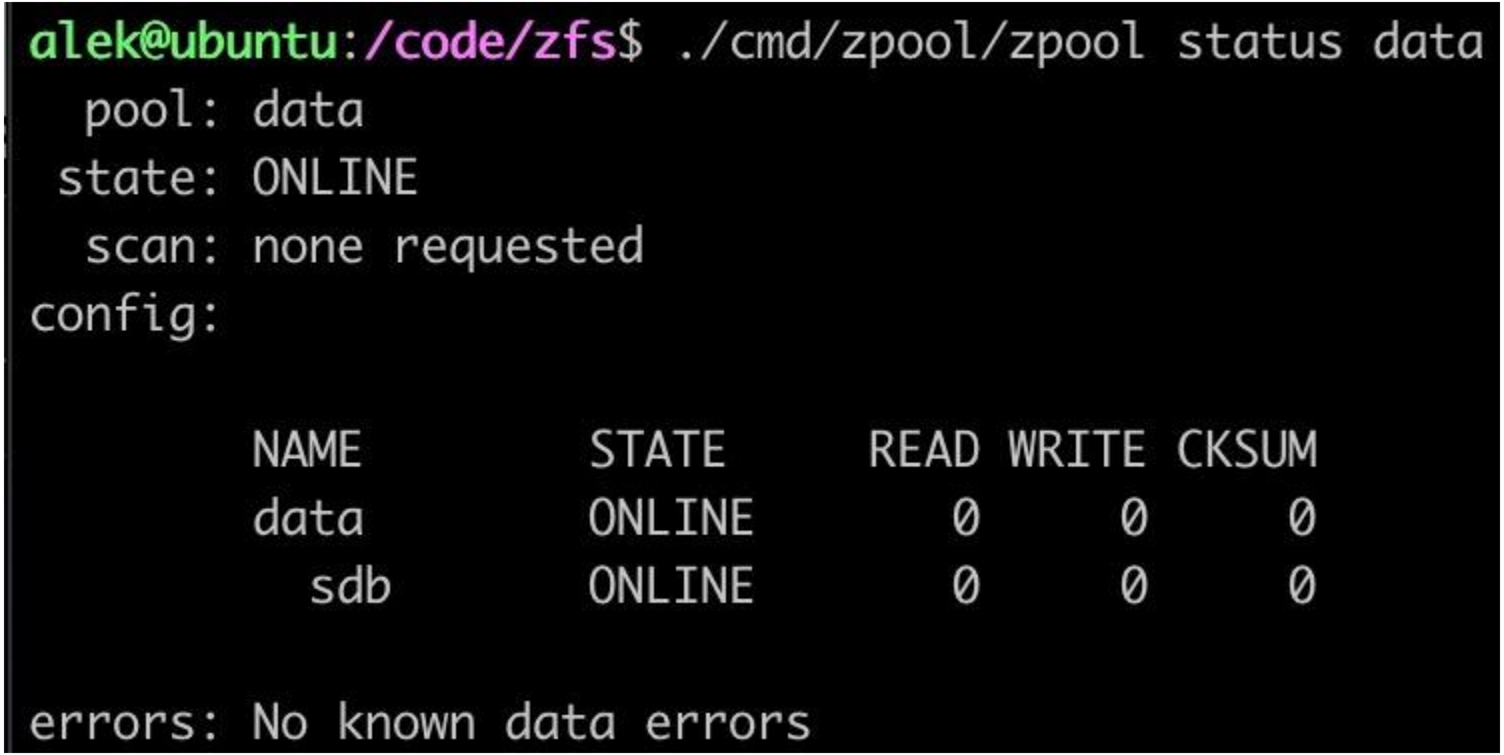
- Currently
  - full send stream healing
  - incremental send stream healing
  - raw send stream healing
  - on-disk & send stream have different compression algos
  - on-disk is encrypted & send stream is not - WiP
- Todo
  - Spill block healing testing in zfs-tests

Demo

OpenZFS

Thank you

- Questions?



```
alek@ubuntu:/code/zfs$ ./cmd/zpool/zpool status data
  pool: data
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        data        ONLINE       0     0     0
          sdb       ONLINE       0     0     0

errors: No known data errors
```