

PANIC: rpool: blkptr at 00000000a44c5bb3 DVA 0 has invalid OFFSET 18388167655883276288

3 similar errors caught:

- 2021-04-18: PANIC: rpool: blkptr at 000000003a3d1018 DVA 0 has invalid OFFSET 18388167655883276288
- 2021-04-19: PANIC: rpool: blkptr at 000000009897c6f0 DVA 0 has invalid OFFSET 18388167655883276288
- 2021-05-09: PANIC: rpool: blkptr at 00000000a44c5bb3 DVA 0 has invalid OFFSET 18388167655883276288

DVA (DataVirtualAddress) is made up of:

- a 32bit integer representing the VDEV
- followed by a 63bit integer representing the offset.

Important steps

Stop scrub to prevent error loop (if scrub reads corrupted data, kernel will panic again):

```
zfs scrub -s rpool
```

(Option) Turn ZFS PANIC into WARNING

```
echo 1 > /sys/module/zfs/parameters/zfs_recover
```

Dump whole history of pool:

```
zdb -hh rpool
```

- -bb: Display statistics regarding the number, size (logical, physical and allocated) and deduplication of blocks. (Verbosity 2)
- -c: Verify the checksum of all metadata blocks while printing block statistics (see: -b). If specified multiple times, verify the checksums of all blocks.
- -s: Report statistics on zdb I/O. Display operation counts, bandwidth, and error counts of I/O
- -L: Disable leak detection and the loading of space maps. By default, zdb verifies that all non-free blocks are referenced, which can be very expensive.
- -AAA: Do not abort if asserts fail and also enable panic recovery.

```
zdb -AAA -bbcsvL <zfs_pool>
```

And sometimes it shows some errors during reading:

```
zdb_blkptr_cb: Got error 52 reading <259, 75932, 0, 17>
```

```
DVA[0]=<0:158039e9000:6000> [L0 ZFS plain file] fletcher4 lz4 unencrypted LE
contiguous unique single size=20000L/6000P birth=62707L/62707P fill=1
cksum=516dd1ace1c:414cbfc202333b:af36411a2766c4f:7bc4d6777673687b --
skipping
```

Find problematic file / volume

Try to read all ZFS volumes

```
zfs list
```

And for each volume, try to read it:

```
zfs send rpool/data/vm-703-disk-1 | pv > /dev/null
```

Catch PANIC and reboot system by sysrq to prevent IO lock. This problematic volume is a replicated (received) volume from another ZFS node.

```
echo s > /proc/sysrq-trigger
echo b > /proc/sysrq-trigger
```

After power up delete problematic zfs volume. During deletion PANIC happens again. Deletion is stored ZFS journal, so during mounting ZFS tries to replay pending deletion which cause PANIC again. System stuck in bootloop.

- Boot from Live USB with ZFS Support (Ubuntu has 0.8.3 ZOL).
- stop zfs-zed service
- `rmmod zfs`
- `modprobe zfs zfs_recover=1`

It doesn't help. After hit ZFS warning instead of panic, ZFS informs about unrecoverable error and pool is suspended.

Last possibility is to boot from Live system from USB and copy all data to other zpool and recreate rpool:

```
mkdir /rpool
zpool import -f -R /rpool rpool -o readonly=on
```

- -e operate on exported pool
- -L Disable leak detection and the loading of space maps. By default, zdb verifies that all non-free blocks are referenced, which can be very expensive.

```
zdb -e -bcsvL rpool
```

Resoruces

- <https://github.com/openzfs/zfs/issues/3990>
- <https://github.com/google/rowhammer-test>
- ztest zdb hits zdb_blkptr_cb error leading to leak and zdb error exit with no core file
- [zfs] Panic during data access or scrub on 12.0-STABLE r343904 (blkptr at <addr> DVA 0 has invalid OFFSET)
- ZFS kernel panic: Solaris(panic): blkptr at 0xfffffe006bf0d4a8 DVA 0 has invalid OFFSET 70368744177664

ZFS scrub repairs only blocks with bad checksums or otherwise unreadable blocks.

ZFS scrub does not analyze and thus does not fix any logical inconsistencies.

If bad data has a correct checksum, then at present ZFS cannot fix it.

Sometimes it can recognize that the data is bad and report an error, sometimes it has no option but to panic, but sometimes it cannot even tell if it's bad data.

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

https://niziak.spox.org/wiki/linux:fs:zfs:issues:dva_0_has_invalid_offset

Last update: **2021/05/24 19:10**

