RAI 1

What's new in OpenZFS 2.3

Rob Norris

Hello!

- OpenZFS developer
- Recovering Linux sysadmin
- FreeBSD non-committer

- 🏡 despairlabs.com
- 📝 robn.au
- 🛝 robn.au/openzfs-23

💾

OpenZFS? what even

💾 What is OpenZFS?

- a filesystem
- a volume manager
- ❤️ *together at last* ❤️

💾 Copy-on-write

- never overwrite stored data
- (but free it later when nothing is using it)
- block pointer tree

- *snapshots*: contents of a filesystem at a point in time
- *replication*: send a snapshot to another pool
- *clone*: make a new filesystem based on a snapshot
- *rollback*: discard changes on a filesystem since the snapshot was taken

💾 Data integrity

- checksums on all blocks
- checksum all read data, compare
- repair block if checksums don't match

- 📢 TELL YOU ABOUT IT 📢

💾 Storage pools

- *What if disks were like RAM?*
- All your disks in one "pool"
- No partitions
- Shared among all filesystems
- Run out of space? Add more disks!

💻 Create a pool

```
computer# zpool create tank loop0 loop1 loop2 loop3

computer# zpool status
  pool: tank
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          loop0     ONLINE       0     0     0
          loop1     ONLINE       0     0     0
          loop2     ONLINE       0     0     0
          loop3     ONLINE       0     0     0

errors: No known data errors
```

💻 Create some filesystems

```
computer# zfs create tank/movies
computer# zfs create tank/photos

computer# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
tank          176K   192M    25K  /tank
tank/movies    24K   192M    24K  /tank/movies
tank/photos    24K   192M    24K  /tank/photos

computer# df
Filesystem     1K-blocks   Used Available Use% Mounted on
overlay          8198820 614544   7584276   8% /
none             8196484      0   8196484   0% /dev
tank              196480    128    196352   1% /tank
tank/movies       196480    128    196352   1% /tank/movies
tank/photos       196480    128    196352   1% /tank/photos
```

💻 Add some data

```
computer# ls -l /tank/movies
-rw-r--r-- 1 root root 104857600 Nov 23 00:23 absolute_casserole.avi

computer# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
tank          100M  91.7M    25K  /tank
tank/movies   100M  91.7M   100M  /tank/movies
tank/photos    24K  91.7M    24K  /tank/photos

computer# df
Filesystem     1K-blocks   Used Available Use% Mounted on
overlay          8198820 614544   7584276   8% /
none             8196484      0   8196484   0% /dev
tank               93952    128     93824   1% /tank
tank/movies       196352 102528     93824  53% /tank/movies
tank/photos        93952    128     93824   1% /tank/photos
```

💻 Take a snapshot

```
computer# zfs snapshot -r tank@today

computer# zfs list -t all
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank                101M  90.8M    25K  /tank
tank@today            0B      -    25K  -
tank/movies         100M  90.8M   100M  /tank/movies
tank/movies@today     0B      -   100M  -
tank/photos          24K  90.8M    24K  /tank/photos
tank/photos@today     0B      -    24K  -
```

💥 Oh no!

```
computer# rm /tank/movies/absolute_casserole.avi
computer# ls -l /tank/movies/

computer# zfs list -t all -d 1 -o space tank/movies
NAME               AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
tank/movies        91.2M   100M      100M     24K             0B         0B
tank/movies@today      -   100M         -       -              -          -
```

💻 Restore from snapshot

```
computer# ls -l /tank/movies/.zfs/snapshot/today
-rw-r--r-- 1 root root 104857600 Nov 23 00:23 absolute_casserole.avi

computer# cp /tank/movies/.zfs/snapshot/today/absolute_casserole.avi /tank/movies

computer# zfs list -t all -d 1 -o space tank/movies
NAME               AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
tank/movies        91.0M   200M      100M    100M             0B         0B
tank/movies@today      -   100M         -       -              -          -

computer# zpool get all | grep bclone
tank  bcloneused                     100M                           -
tank  bclonesaved                    100M                           -
tank  bcloneratio                    2.00x                          -
```

💻 Rollback to snapshot

```
computer# ls -l /tank/movies

computer# zfs rollback tank/movies@today

computer# ls -l /tank/movies
-rw-r--r-- 1 root root 104857600 Nov 23 00:23 absolute_casserole.avi

computer# zfs list -t all -d 1 -o space tank/movies
NAME               AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
tank/movies        91.6M   100M       13K    100M             0B         0B
tank/movies@today      -    13K         -       -              -          -
```

💾 Virtual devices

```
computer# zpool create tank loop0 loop1 loop2 loop3

computer# zpool status
  pool: tank
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          loop0     ONLINE       0     0     0
          loop1     ONLINE       0     0     0
          loop2     ONLINE       0     0     0
          loop3     ONLINE       0     0     0

errors: No known data errors
```

💾 Virtual devices: mirror

```
computer# zpool create tank mirror loop0 loop1 mirror loop2 loop3

computer# zpool status
  pool: tank
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            loop0   ONLINE       0     0     0
            loop1   ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            loop2   ONLINE       0     0     0
            loop3   ONLINE       0     0     0

errors: No known data errors
```

💾 Virtual devices: raidz

```
computer# zpool create tank raidz1 loop0 loop1 loop2 loop3

computer# zpool status
  pool: tank
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          raidz1-0  ONLINE       0     0     0
            loop0   ONLINE       0     0     0
            loop1   ONLINE       0     0     0
            loop2   ONLINE       0     0     0
            loop3   ONLINE       0     0     0

errors: No known data errors
```

NEW

OpenZFS 2.3

🍔

RAIDZ Expansion

🍔 RAIDZ Expansion

```
computer# zpool status
  pool: tank
 state: ONLINE
config:

        NAME          STATE     READ WRITE CKSUM
        tank          ONLINE       0     0     0
          raidz1-0    ONLINE       0     0     0
            loop0     ONLINE       0     0     0
            loop1     ONLINE       0     0     0
            loop2     ONLINE       0     0     0
            loop3     ONLINE       0     0     0
```

```
computer# download /tank/movies/le_goose.avi
download: failed to open '/tank/movies/le_goose.avi': No space left on device
```

🍔 RAIDZ Expansion

Pre-2.3

```
computer# zpool attach tank raidz1-0 loop4
cannot attach loop4 to raidz1-0: can only attach to mirrors and top-level disks
```

🍔 RAIDZ Expansion

2.3+

```
computer# zpool attach tank raidz1-0 loop4
```

🍔 RAIDZ Expansion

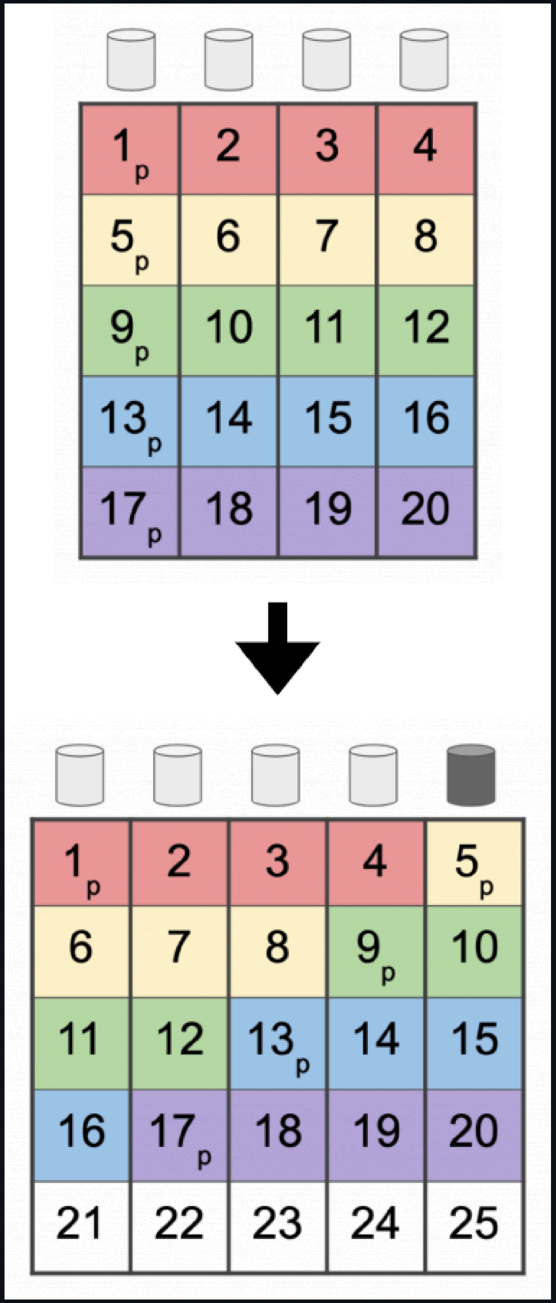
```
computer# zpool status
  pool: tank
 state: ONLINE
  scan: scrub repaired 0B in 00:00:00 with 0 errors on Sat Nov 23 02:31:43 2024
expand: expanded raidz1-0 copied 165M in 00:00:02, on Sat Nov 23 02:31:43 2024
config:

        NAME          STATE     READ WRITE CKSUM
        tank          ONLINE       0     0     0
          raidz1-0    ONLINE       0     0     0
            loop0     ONLINE       0     0     0
            loop1     ONLINE       0     0     0
            loop2     ONLINE       0     0     0
            loop3     ONLINE       0     0     0
            loop4     ONLINE       0     0     0

errors: No known data errors
```

🍔 RAIDZ Expansion

- Under development since 2017
- RAIDZ-1/2/3 (not dRAID)
- Reflowed data retains data:parity ratio
- New data uses new data:parity ratio
- `zpool attach <pool> <raidz vdev> <device>`



1_p	2	3	4
5_p	6	7	8
9_p	10	11	12
13_p	14	15	16
17_p	18	19	20

1_p	2	3	4	5_p
6	7	8	9_p	10
11	12	13_p	14	15
16	17_p	18	19	20
21	22	23	24	25

🏎️

Direct IO

🏎️ Direct IO

- `O_DIRECT` to `open(2)` to bypass cache (ARC)
    - Databases (own caching)
    - read/write-once data
    - low-latency/high-bandwidth devices (NVMe)
- Fully integrated (no limitations on other features)
    - compression, encryption, RAIDZ reflow, cloning, snapshots, etc, etc
- Uses user data as far as possible (avoiding a copy)
- ARC coherent (that is, mixed buffered + direct IO gets same view)
- `zfs set direct=<disabled|standard|always>`

📠

JSON output

🕵️ Sysadmin-friendly

```
$ zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
crayon                428G  3.18G   192K  none
crayon/dump          46.8M  3.18G  45.6M  /dump
crayon/home           418G  3.18G   200K  /home
crayon/home/robn      416G  3.18G   392G  /home/robn
crayon/home/root     1.50G  3.18G  1.13G  /root
crayon/root          9.60G  3.18G   192K  none
crayon/root/debian   9.60G  3.18G  8.69G  /
crayon/var           73.9M  3.18G   192K  /var
crayon/var/cache     61.0M  3.18G  54.7M  /var/cache
crayon/var/log       11.4M  3.18G  7.63M  /var/log
crayon/var/tmp       1.25M  3.18G   744K  /var/tmp
```

Sysadmin-friendly

```
$ zfs list -H -p -o refer,mounted
196608  no
47775744        yes
204800  yes
421021003776    yes
1217708032      yes
196608  no
9325654016      yes
196608  no
57380864        yes
7999488 yes
761856  yes
```

🕵️ Sysadmin-friendly

```
$ zfs list -H -p -o refer,mounted | \
    awk '/yes$/ { u += $1 } END { print u }'
431678521344
```

🕵️ Sysadmin-friendly

```
$ zfs list -H -p -o refer,mounted | \
    awk '/yes$/ { u += $1 } END { print u }' | \
    numfmt --to=iec
403G
```

👮 Sysadmin-unfriendly

```
$ zpool status shed
  pool: shed
 state: ONLINE
  scan: scrub repaired 0B in 04:29:15 with 0 errors on Sat Nov  2 11:42:28 2024
config:

        NAME        STATE     READ WRITE CKSUM
        shed        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            da0     ONLINE       0     0     0
            da1     ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            da2     ONLINE       0     0     0
            da3     ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            da4     ONLINE       0     0     0
            da5     ONLINE       0     0     0

errors: No known data errors
```

📠 Machine-friendly

```
$ zpool status -j shed
```

```
{"output_version":{"command":"zpool status","vers_major":0,"vers_minor":1},"pools":{"shed":{"name":"shed","state":"ONLINE","pool_guid":"1660
1227028685247571","txg":"5163851","spa_version":"5000","zpl_version":"5","status":"Some supported and requested features are not enabled on
the pool.\n\tThe pool can still be used, but some features are unavailable.\n","action":"Enable all features using 'zpool upgrade'. Once thi
s is done,\n\tthe pool may no longer be accessible by software that does not support\n\tthe features. See zpool-features(7) for details.\n",
"scan_stats":{"function":"SCRUB","state":"FINISHED","start_time":"Sat Nov  2 07:13:13 2024","end_time":"Sat Nov  2 11:42:28 2024","to_examin
e":"4.80T","examined":"4.81T","skipped":"10.8M","processed":"0B","errors":"0","bytes_per_scan":"0B","pass_start":"1731314497","scrub_pause":
"-","scrub_spent_paused":"0","issued_bytes_per_scan":"0B","issued":"4.81T"},"vdevs":{"shed":{"name":"shed","vdev_type":"root","guid":"166012
27028685247571","class":"normal","state":"ONLINE","alloc_space":"5.87T","total_space":"21.8T","def_space":"21.8T","read_errors":"0","write_e
rrors":"0","checksum_errors":"0","vdevs":{"mirror-0":{"name":"mirror-0","vdev_type":"mirror","guid":"13397193233085620674","class":"normal",
"state":"ONLINE","alloc_space":"2.47T","total_space":"7.27T","def_space":"7.27T","rep_dev_size":"7.27T","read_errors":"0","write_errors":"0"
,"checksum_errors":"0","vdevs":{"da0":{"name":"da0","vdev_type":"disk","guid":"2155107347182380136","path":"/dev/da0","class":"normal","stat
e":"ONLINE","rep_dev_size":"7.27T","phys_space":"7.28T","read_errors":"0","write_errors":"0","checksum_errors":"0","slow_ios":"0"},"da1":{"n
ame":"da1","vdev_type":"disk","guid":"14087237141989434741","path":"/dev/da1","class":"normal","state":"ONLINE","rep_dev_size":"7.27T","phys
_space":"7.28T","read_errors":"0","write_errors":"0","checksum_errors":"0","slow_ios":"0"}}},"mirror-1":{"name":"mirror-1","vdev_type":"mirr
or","guid":"13126522904398015453","class":"normal","state":"ONLINE","alloc_space":"2.67T","total_space":"7.27T","def_space":"7.27T","rep_dev
_size":"7.27T","read_errors":"0","write_errors":"0","checksum_errors":"0","vdevs":{"da2":{"name":"da2","vdev_type":"disk","guid":"2043805469
087768923","path":"/dev/da2","class":"normal","state":"ONLINE","rep_dev_size":"7.27T","phys_space":"7.28T","read_errors":"0","write_errors":
"0","checksum_errors":"0","slow_ios":"0"},"da3":{"name":"da3","vdev_type":"disk","guid":"9435675405080108327","path":"/dev/da3","class":"nor
mal","state":"ONLINE","rep_dev_size":"7.27T","phys_space":"7.28T","read_errors":"0","write_errors":"0","checksum_errors":"0","slow_ios":"0"}
}},"mirror-2":{"name":"mirror-2","vdev_type":"mirror","guid":"7175701775352017511","class":"normal","state":"ONLINE","alloc_space":"739G","t
otal_space":"7.27T","def_space":"7.27T","rep_dev_size":"7.27T","read_errors":"0","write_errors":"0","checksum_errors":"0","vdevs":{"da4":{"n
ame":"da4","vdev_type":"disk","guid":"15399265605284752209","path":"/dev/da4","class":"normal","state":"ONLINE","rep_dev_size":"7.27T","phys
_space":"7.28T","read_errors":"0","write_errors":"0","checksum_errors":"0","slow_ios":"0"},"da5":{"name":"da5","vdev_type":"disk","guid":"10
023359214663812931","path":"/dev/da5","class":"normal","state":"ONLINE","rep_dev_size":"7.27T","phys_space":"7.28T","read_errors":"0","write
_errors":"0","checksum_errors":"0","slow_ios":"0"}}}}}}},"error_count":"0"}}}
```

📠 Machine-friendly

```
$ zpool status -j shed | jq .
```

```
{
  "output_version": {
    "command": "zpool status",
    "vers_major": 0,
    "vers_minor": 1
  },
  "pools": {
    "shed": {
      "name": "shed",
      "state": "ONLINE",
      "pool_guid": "16601227028685247571",
      "txg": "5163851",
      "spa_version": "5000",
      "zpl_version": "5",
...
```

📠 Machine-friendly

```
$ zpool status -j shed | \
    jq -r '.pools[].vdevs[]'
```

```
{
  "name": "shed",
  "vdev_type": "root",
  "state": "ONLINE",
  "vdevs": {
    "mirror-0": {
      "name": "mirror-0",
      "vdev_type": "mirror",
      "state": "ONLINE",
      "vdevs": {
        "da0": {
          "name": "da0",
          "vdev_type": "disk",
          "state": "ONLINE",
          "path": "/dev/da0",
...
```

📠 Machine-friendly

```
$ zpool status -j shed | \
    jq -r '.pools[].vdevs[] | recurse(.vdevs[]?) |
           select(.vdev_type == "disk")'
```

```
{
  "name": "da0",
  "vdev_type": "disk",
  "path": "/dev/da0",
  "state": "ONLINE",
}
{
  "name": "da1",
  "vdev_type": "disk",
  "path": "/dev/da1",
  "state": "ONLINE",
}
...
```

📠 Machine-friendly

```
$ zpool status -j shed | \
    jq -r '.pools[].vdevs[] | recurse(.vdevs[]?) |
           select(.vdev_type == "disk") | [.path,.state]'
```

```
[
  "/dev/da0",
  "ONLINE"
]
[
  "/dev/da1",
  "ONLINE"
]
[
  "/dev/da2",
  "ONLINE"
]
...
```

📠 Machine-friendly

```
$ zpool status -j shed | \
    jq -r '.pools[].vdevs[] | recurse(.vdevs[]?) |
           select(.vdev_type == "disk") | [.path,.state] | @tsv'
```

```
/dev/da0        ONLINE
/dev/da1        ONLINE
/dev/da2        ONLINE
/dev/da3        ONLINE
/dev/da4        ONLINE
/dev/da5        ONLINE
```

🎨 Pretty pictures

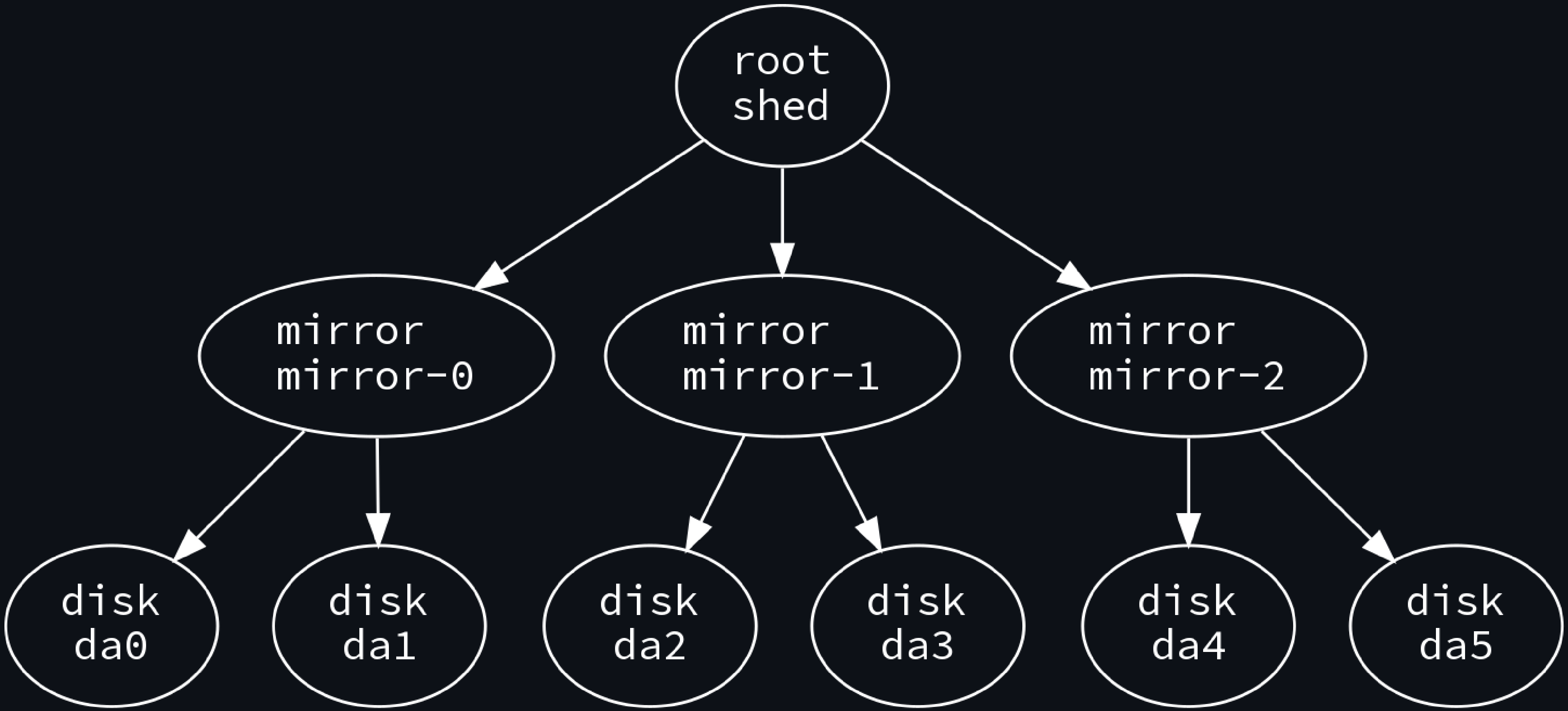
```
$ (
    echo 'digraph {'
    zpool status -j shed | \
        jq -r '.pools[].vdevs[] | recurse(.vdevs[]?) |
               "\(.guid) [label=\"\(.vdev_type)\\l\(.name)\"]",
               (select(.vdevs) | "\(.guid) -> \(.vdevs[].guid)")'
    echo '}'
)
```

🎨 Pretty pictures

```
digraph {
16601227028685247571 [label="root\lshed"]
16601227028685247571 -> 13397193233085620674
16601227028685247571 -> 13126522904398015453
16601227028685247571 -> 7175701775352017511
13397193233085620674 [label="mirror\lmirror-0"]
13397193233085620674 -> 2155107347182380136
13397193233085620674 -> 14087237141989434741
2155107347182380136 [label="disk\lda0"]
14087237141989434741 [label="disk\lda1"]
13126522904398015453 [label="mirror\lmirror-1"]
13126522904398015453 -> 2043805469087768923
13126522904398015453 -> 9435675405080108327
2043805469087768923 [label="disk\lda2"]
9435675405080108327 [label="disk\lda3"]
7175701775352017511 [label="mirror\lmirror-2"]
7175701775352017511 -> 15399265605284752209
7175701775352017511 -> 10023359214663812931
15399265605284752209 [label="disk\lda4"]
10023359214663812931 [label="disk\lda5"]
}
```

🎨 Pretty pictures

```
$ (
    echo 'digraph {'
    zpool status -j shed | \
        jq -r '.pools[].vdevs[] | recurse(.vdevs[]?) |
               "\(.guid) [label=\"\(.vdev_type)\\l\(.name)\"]",
               (select(.vdevs) | "\(.guid) -> \(.vdevs[].guid)")'
    echo '}'
) | \
    dot -Tpng -Gdpi=200 -Nfontname='SourceCodeVF Regular' \
        -Gcenter=true -Gpad=0.5 | \
    display -
```


root
shed
mirror
mirror-0
mirror
mirror-1
mirror
mirror-2
disk
da0
disk
da1
disk
da2
disk
da3
disk
da4
disk
da5

📠 JSON output

- available in most commands
  - `zpool`: `version`, `list`, `get`, `status`
  - `zfs`: `version`, `list`, `get`, `mount`, `program`

🧱🧱

Block cloning

🧱🧱 Block cloning

- aka "reflinks"
- make a copy-on-write "clone" of a file
  - like hard links: same data blocks on disk
  - unlike hard links: when one file changes, the contents are "forked" from the original and modified
- GNU coreutils: `cp --reflink=auto` (default since 9.2)
- FreeBSD `/bin/cp` (since 13.0)

🧱🧱 Block cloning

- "copy offloading"
  - instead of reading and writing the file yourself, ask the system to copy it for you
- Syscall: `copy_file_range()`, `ioctl(FICLONERANGE)` (Linux)
- NFS + Samba
- Any program that copies files

🧱🧱 Block cloning (2.3)?

- Actually a 2.2 feature, but had some … issues 😬
- Disabled by default on FreeBSD in 2.2.0, Linux in 2.2.1
- Stabilised by 2.2.4, but still disabled out of an abundance of caution
- Enabled by default in 2.3.0

✌️

Fast dedup

✌️

Slow dedup?

✌️ Traditional dedup

- Online block-level duplication
- A table of checksums → [on-disk location of data, refcount]
- When data is written, look up the checksum
  - If it's there, don't do the write, just bump the refcount
  - If it's not, do the write, and add it to the table

🖕 Bad dedup

- Every write requires a read-update-write cycle to the on-disk dedup table
- Table quickly fills with "unique" entries
- ⚠️ DO NOT USE ⚠️

✌️ Fast dedup

- Co-exists with traditional deduplication
- Log (journal) to minimise update overheads
- Prune blocks unlikely to be deduplicated
  - `zpool ddtprune <-d days | -p percentage> <pool>`
- Set quota on dedup tables
  - `zpool set dedup_table_quota=<amount> <pool>`
- Load the dedup table into RAM ahead of time
  - `zpool prefetch -t ddt <pool>`

- ⚠️ DO NOT USE ⚠️
  - (probably)

🎉

Mixed bag

🛤️ Parallel pool import/export

- Import/export can take a while for large/busy pools
- For multiple pools, do them at the same time

⚡ Enclosure power control

- Enclosures with independent slot power management
    - SCSI Enclosure Services (SES)
- `zpool online|offline --power <pool> <vdev>`

👿 ZED: ZFS Event Daemon

- monitors disks, takes actions based on config
    - Demote failed/slow disk
    - Promote spare disk

👿 ZED: ZFS Event Daemon

- Power off enclosure slots
- Consider peer disks when there are multiple faults
    - Don't demote drive if related drives also acting up
    - Maybe a flaky controller or cable?

📛 Long file names

- Names up to 1023 chars
- Mostly useful for SMB shares

🖼️ Disable `.zfs` snapshot directory

```
computer# ls -l /tank/movies/.zfs/snapshot/today
-rw-r--r-- 1 root root 104857600 Nov 23 00:23 absolute_casserole.avi
```

🖼️ Disable `.zfs` snapshot directory

```
computer# ls -la /tank/movies
drwxr-xr-x 2 root root 2 Nov 23 23:57 .
drwxr-xr-x 3 root root 3 Nov 23 23:57 ..

computer# ls -la /tank/movies/.zfs
drwxrwxrwx 1 root root 0 Nov 23 23:57 .
drwxr-xr-x 2 root root 2 Nov 23 23:57 ..
drwxrwxrwx 2 root root 2 Nov 23 23:57 shares
drwxrwxrwx 3 root root 2 Nov 23 23:57 snapshot

computer# zfs get snapdir tank/movies
NAME          PROPERTY  VALUE     SOURCE
tank/movies   snapdir   hidden    default
```

🖼️ Disable `.zfs` snapshot directory

```
computer# zfs set snapdir=visible tank/movies

computer# ls -la /tank/movies
drwxr-xr-x 3 root root 2 Nov 23 23:57 .
drwxr-xr-x 3 root root 3 Nov 23 23:57 ..
drwxrwxrwx 1 root root 0 Nov 23 23:57 .zfs

computer# ls -la /tank/movies/.zfs
drwxrwxrwx 1 root root 0 Nov 23 23:57 .
drwxr-xr-x 3 root root 2 Nov 23 23:57 ..
drwxrwxrwx 2 root root 2 Nov 23 23:57 shares
drwxrwxrwx 3 root root 2 Nov 23 23:57 snapshot
```

🖼️ Disable `.zfs` snapshot directory

```
computer# zfs set snapdir=disabled tank/movies

computer# ls -la /tank/movies
drwxr-xr-x 2 root root 2 Nov 23 23:57 .
drwxr-xr-x 3 root root 3 Nov 23 23:57 ..

computer# ls -la /tank/movies/.zfs
ls: cannot access '/tank/movies/.zfs': No such file or directory
```

☎ Rewritten IO assembly/handoff

- Longstanding IO alignment issues under extreme memory/disk pressure
  - Fragmented pool/memory could cause incorrect splitting (esp SCSI core)
  - IO pages not aligned to memory pages rejected by `dm-crypt` (LUKS)
- Up to 10% faster on stress-test workloads

🧠 Memory management

- Friendlier to Linux reclaim (OOM)
  - especially since 6.1 (MGLRU)
- Improve dbuf cache accounting
  - 🚨 *critical* on datasets with tens of millions of files

🥼 Developer quality-of-life

- Improved userspace debugging facilities
- Improved taskq stats
- Dropped support for Linux < 4.18, FreeBSD < 13
- Require extra "yes I’m really sure" build flag for bleeding-edge kernels

📅

Release day

📆 Release day

- 2.3.0-rc3 (9 November)
- Still some bugs to shake out
- Expecting by Christmas 🎄
  - (2024)
- "When it's ready"

📆 Release day

- Official release site
  - tarball download (github.com/openzfs/zfs/releases)
  - DKMS & kmod for Linux, kmod for FreeBSD
- Fedora, Red Hat: LLNL repository (download.zfsonlinux.org)
- Ubuntu 25.04? 25.10? 26.04 LTS?
- Arch, NixOS, Void Linux: "immediately"
- FreeBSD 15 (mid-late 2025)

💸

Support independent software development

💸 Support independent software development

- I work on OpenZFS full-time
- Income from two main sources
  - Business contracts for feature development, support and maintenance
  - Individual sponsorships

- My family like food, internet, clothes, etc.

💸 Support independent software development

klara

- OpenZFS and FreeBSD
- Feature development
- Storage system design and development
- One-off health & performance checks
- Ongoing support contracts
- Commitment to upstreaming everything

klarasystems.com

💸 Support independent software development

Individual sponsorships

- General maintenance 🔧
- Bug hunting 🐜
- Ad-hoc support (forums, email, IRC, etc) ⛑️
- Development hardware 🧰
- Conference travel ✈️
- Future design and planning 🏗️

despairlabs.com/sponsor

OpenZFS