VDEV Properties

OpenZFS Developer Summit 2019

allan@klarasystems.com

Summary & Introductions

1

Allan Jude

FreeBSD Core Team
OpenZFS Developer

2

Klara Inc.

FreeBSD Professional
Services and Support

Properties

- ZFS makes it very easy to configure your storage since most all objects in ZFS have properties
  - Pools, Filesystems, Volumes, Snapshots
- This should extend to other ZFS concepts: VDEVs
- Additionally, many tunables are system wide. Should more of these become per-pool or per-disk?

Motivations

- Make it easier to get data from ZFS programmatically
  - `zpool status`
  - `zpool list -v` of one disk only
- Express administrative intent
  - `zpool set noalloc=on pool vdev3` # Removing this disk soon
  - `zpool remove pool vdev2`
- Extend existing features: Special VDEVs

Implementation

- Device Evacuation feature added per-VDEV ZAP
- Each vdev now has a a key-value store
- Applies to both top-level vdevs (stripe, mirror, RAID-Z*) and the leaf vdevs they contain
- Many read-only properties are not stored in the ZAP, but just read directly from `vdev_t`

Two Types of Properties

- System properties (declared with a type)
  - Integer, Index (Table), and String
- User Properties (always a string)
  - Namespaced by the user to avoid collisions
  - Work the same as dataset user properties

Interface

- After toying with:
- `zpool get prop@vdev pool`
- I was introduced to the new ZoL `zpool iostat`
- Generalized that code a bunch to get
- `zpool get prop pool vdev1 vdev2`
- All the other stuff you love about zfs/zpool get
    - -H = scriptable, -p = parsable, -o = select columns

Output

```
# zpool status
  pool: mypool
 state: DEGRADED
config:
       NAME                     STATE   READ WRITE CKSUM
       mypool                   DEGRADED   0   0   0
         ada0p4                 ONLINE     0   0   0
         mirror-1               DEGRADED   0   0   0
           ada1p3            ONLINE     0   0   0
           14611559395453898841  UNAVAIL   0   0   0
```

Output

```
# zpool get state,frag,parent,read-errors mypool ada0p4 ada1p3
NAME       PROPERTY      VALUE   SOURCE
ada0p4     state         ONLINE  -
ada0p4     fragmentation  0%     -
ada0p4     parent        mypool  -
ada0p4     read-errors   0       -
ada1p3     state         ONLINE  -
ada1p3     fragmentation  0%     -
ada1p3     parent        mirror-1   -
ada1p3     read-errors   0       -
```

Properties

- Name
- Capacity
- State
- GUID
- Allocated Size
- Physical Size
- Ashift
- Size
- Free
- Allocated
- Readonly
- Comment
- Expand Size
- Fragmentation
- Boot size
- Parity
- Path
- DevID
- PhysPath
- EncPath
- FRU
- Parent
- Children
- NumChildren
- Errors Counts
- IOP Counts
- Byte Counts

Ideas

- Extend zpool properties
  - get each of the messages from `zpool status`
    - status, action, scan, wait, removing, etc.
- Possibly make many of these read-only properties hidden to avoid cluttering `zpool get all`
- Maybe make special `zpool get everything`, or a flag for get to unmask hidden properties

Future Items

- ZCP (Channel Program) Intergration
- Caching (libzfs does this for pool properties…)
- Inheritance (previously thought to avoid this)
- Persistent Counters (track total writes to SSD etc)
    - How to flush these “frequently enough” but not too often
- How many properties is too many?

klara

QUESTIONS

???

klarasystems.com

Contact

- BSDNow.tv - Weekly video podcast on BSD & ZFS
  - Looking for ZFS developers to interview
- @allanjude on twitter
- allan@klarasystems.com