Bacula and ZFS

Great tools for use with PostgreSQL

Chapter 2: ZFS

Dan Langille
PGCon 2018

Why bother with a choice of filesystem?

- The filesystem sits between you and the storage.
- Different FS excel at different things
- Like anything, not everything can be everything to everyone

ZFS

- Think of is as "combined file system and logical volume manager"
- solid
- robust
- reliable

https://en.wikipedia.org/wiki/ZFS

Some features

- scales like mad
- protection from data corruption
- great compression
- snapshots
- clones
- One goal: uninterrupted continual use even during self checking and self repair

vdevs

- physical devices (eg HDDs/SSDs) are organized into vdevs
- Each vdev can be one of:
  - a single device, or
  - multiple devices in a mirrored configuration, or
  - multiple devices in a ZFS RAID ("RaidZ") configuration.

pool

- top level of data management
- can define multiple pools
- consists of one or more vdevs
- vdevs can be of any type
- raidz[1-3]
- collections

one device multiple pools

- It is possible to break up a device into multiple pools
- e.g. two large SSD
  - partition it up as you want
  - create two zpool mirrors
  - part for database server
  - part for working copy of my code

mirror

- zpool create zroot mirror ada0p3 ada1p3
  - name = zroot
  - mirror of two devices
    - partition 3 of ada0
    - partition 3 of ada1

raidz3

- zpool create tank_fast raidz3 ada0p3 ada1p3 ada2p3 ada3p3 ada4p3 ada5p3 ada6p3 ada7p3 ada8p3 ada9p3 ada10p3
- can survive three concurrent drive failures
- If 4th drive dies, all gone

snapshots

- zfs create snapshot tankfast@2018.05.29
- readonly - ideal for backups
- can be mounted, readonly (e.g. for taking a copy or backing up)
- can be restored instantly (e.g. ransomware)

filesystems

- zfs create recordsize=128K tank_data/pg01
  - mountpoint /tank_data/pg01
  - recordsize 128K
- that's just one filesystem, can create more

filesystems

- also known as datasets
- zfs create tank_data/pg01/freshports
  - inherits attributes from parent
  - separate dataset
  - can snapshot separately

filesystems

- Use the same dataset for $PGDATA/ and pg_xlogs/
- One dataset per database

PostgreSQL options

- zfs set atime=off tank_data/pg01
- zfs set recordsize=16K tank_data/pg01
- zfs set compression=lz4 tank_data/pg01
- reasonable to expect ~3-4x pages worth of data in a single ZFS record

ZFS checksums

- Checksum errors are an early indicator of failing disks
- ZFS Always has your back
- ZFS will checksum every read from disk

Anecdotes and Recommendations

- Performed better in most workloads vs ZFS's prefetch
- Disabling prefetch isn't necessary, tends to still be a net win
- Monitor arc cache usage

https://www.slideshare.net/SeanChittenden/postgresql-zfs-best-practices

- primarycache=metadata
- metadata instructs ZFS's ARC to only cache metadata (e.g. dnode entries), not page data itself
- Default: cache all data

Two different recommendations based on benchmark workloads

- Enable primarycache=all where working set exceeds RAM
- Enable primarycache=metadata where working set fits in RAM

ARC

- adaptive replacement cache
- very fast RAM-based cache
- Cap max ARC size ~15%-25% physical RAM + ~50% RAM shared_buffers

initdb

- Do not use PostgreSQL checksums
  - -k —data-checksums
- Don't do compression within PostgreSQL, let ZFS do it instead
- Same with pg_dump etc, I reckon ZFS will do it better

snapshots

- many tools to manage snapshots
- automated
- light-weight, reliable
- Use for copies
- Use for backups

ZFS scrub

- scheduled event
- recommended weekly
- reads *all* data and fixes any checksum issues

ZFS & Bacula

- RunBeforeClientJob: snapshot
- FileSet: specifies the snapshot to backup
- RunAfterClientJob: destroy snapshot

Why snapshot not pg_dump?

- block-level
- read-only
- time to pg_dump

Why pg_dump not snapshot?

- pg_dump exercises the whole database
- can test pg_dump via pg_restore
- don't trust file-level backups of live db

With snapshots:

- you can do a pg_dump from the snapshot... if you load it up into another DB, because you need a live database for that.
- can't do this on a snapshot, have to do some zfs magic to change snapshot to readable, it's not that hard.

With snapshots: