ZFS Tuning

This is **not** a definitive guide. The information on tuning a ZFS file system is very scattered and non-definitive. This is only my working notes as I try to make sense of it. Definitely read the references at the bottom and use them directly.

Tuning

Loader Variables

This is basically variables which are loaded at boot time, so a reboot is required after setting them, as far as I can tell. You can use the sysctl command to see the current settings on the. For example, to see the current setting for vm.kmem_size

```
sysctl -a vm.kmem_size
```

which on my machine returns **vm.kmem_size: 66849464320**. The number is in bytes, so 66849464320/1024/1024/1024 is 62G of RAM. As far as I know, all values are entered as the number of bytes (no modifiers, but maybe I'm wrong).

These can be set in /boot/loader.conf (or in a file in /boot/loader.conf.d/, oe in /boot/loader.conf.local)

vfs.zfs.arc.meta_min

If deduplication in effect, vfs.zfs.arc.meta_min should be set high enough that the dedup tables are in memory at all times, meaning whatever “normal” metadata may be used, plus the memory size of the dedup Calculate the size of dedup metatable with zpool status -D [pool name] dedup: DDT Entries * core size, ie dedup: DDT entries 25868427, size 881B on disk, 284B in core means total dedup size in memory is 25868427 entries * 284 bytes each = 6415369896 bytes, or 5.97 Gigabytes in memory, so if you allocate less than that, the DDT table will have to be swapped out. However, even allocating 6G will not guarantee the dedup table remains in memory since other metadata must be loaded, but this is a good starting point.

vfs.zfs.arc_max

ZFS is a pig, and wants to own all of memory. It will definitely cache anything and everything it can, but this can interfere with opertion of other things. For example, it takes a while to swap out enough space to compare checksums when starting an rsync job.

vfs.zfs.arc_max allows you to limit the amount of memory availble for this cache. NOTE this must be smaller than vm.kmem_size, which is the amount of memory availble to the kernel (ARC is considered to be a kernel memory allocation). If you mess with vm.kmem_size, you must make sure is smaller.

Setting vfs.zfs.arc_max to 50% of vm.kmem_size appears to be a good starting point. On my TrueNAS

machine, this is set to all of memory, from what I can see.

kern.maxvnodes

References

- https://forums.freebsd.org/threads/zfs-and-memory-requirements-for-good-performance.33902/
- https://wiki.freebsd.org/ZFSTuningGuide
- https://forums.freebsd.org/threads/vfs-zfs-arc_max-size-recommendations.69994/

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/unix/freebsd/system_builds/zfstuning

Last update: **2023/01/08 22:13**