ECE 4310
Operating Systems for Embedded Application

Extra Credit 2

Choi Tim Antony Yung

May 13, 2021

1 What were the design goals of ZFS?

- strong data integrity
- simple administration
- immense capacity

2 Would you typically use a separate "RAID" system with ZFS? Why or why not?

Because of the abstraction resulted in the use of storage pool, the duty of RAID (mirroring/striping) are already taken by the storage pool allocator. Therefore, it does not typically require a separate RAID system.

3 Explain how ZFS implements it's copy-on-write data model?

All data in the pool is part of a tree of indirect blocks, with the data blocks as the leaves of the tree. The block at the root of the tree is called the uberblock. Whenever any part of a block is written, a new block is allocated and the entire modified block is copied into it. Since the indirect block must be written in order to record the new location of the data block, it must also be copied to a new block. Newly written indirect blocks "ripple" all the way up the tree to the uberblock.

4 What is the role of the Data Management Unit (DMU) in ZFS?

The DMU consumes blocks from the SPA and exports objects (flat files). Objects live within the context of a particular dataset. A dataset provides a private namespace for the objects contained by the dataset. Objects are identified by 64-bit numbers, contain up to 264 bytes of data, and can be created, destroyed, read, and written. Each write to (or creation of or destruction of) a DMU object is assigned to a particular transaction by the caller.

5 What is the role of the Storage Pool Allocator (SPA) in ZFS?

The Storage Pool Allocator (SPA) allocates blocks from all the devices in a storage pool. One system can have multiple storage pools, although most systems will only need one pool. Unlike a volume manager, the SPA does not present itself as a logical block device. Instead, it presents itself as an interface to allocate and free virtually addressed blocks — basically, `malloc()` and `free()` for disk space. Virtual addresses of disk blocks are called data virtual addresses(DVAs). Using virtually addressed blocks makes it easy to implement several of our design principles. First, it allows dynamic addition and removal of devices from the storage pool without interrupting service. None of the code above the SPA layer knows where a particular block is physically located, so when a new device is added, the SPA can immediately start allocating new blocks from it without involving the rest of the file system code. Likewise, when the user requests the removal of a device, the SPA can move allocated blocks off the disk by copying them to a new location and changing its translation for the blocks' DVAs without notifying anyone else.

6 How is error correction and detection handled in ZFS? Why is the error correction and detection mechanisms built-in to ZFS better than what is provided by other storage systems, such as RAID controllers and/or other file systems?

To protect against data corruption, each block is checksummed before it is written to disk. A block's checksum is stored in its parent indirect block. The uberblock is the only block that stores its checksum in itself. Keeping checksums in the parent of a block separates the data from its checksum on disk and makes simultaneous corruption of data and checksum less likely. It makes the checksums self-validating because each checksum is itself checksummed by its parent block. Another benefit is that the checksum doesn't need to be read in from a separate block, since the indirect block was read in to get to the data in the first place. The checksum function is a pluggable module; by default the SPA uses 64-bit Fletcher checksums. Checksums are verified whenever a block is read in from disk and updated whenever a block is written out to disk. Since all data in the pool, including all metadata, is in the tree of blocks, everything ZFS ever writes to disk is checksummed.

7 Explain in your own words why you think ZFS may be a better choice than using other physical volume management systems (ie RAID) and other more common file systems such as ext3/4, JFS, XFS?

ZFS integrates volume management into file system to support mirroring and striping so that these management can be configured after the disks are formatted to ZFS. ZFS gives more flexibility to the user because the user no longer need to worry about the actual block devices but can create and manage volume and let ZFS handle the checksum and the hardware configuration of mirroring and striping.