Recitation 17 — ZFS

ZFS' Goals

- Data integrity
- Simple administration
- Scalability

Modules

- Storage pool allocator
    - Allows for separation of filesystem specification and creation from disk allocation. Filesystem size can be dynamic.
    - Metadata stored as a tree with uberblock at root.
- Data Management Unit
    - Keeps on-disk data consistent using copy-on-write: allocate a new block every time there's a write, including for indirect blocks; atomically switch from old tree of blocks to new tree assuming everything goes well (if not, we go back and correct)
        - Figures 6-8 illustrate this process
- ZFS Posix Layer
    - Gives us the abstraction of a "normal" filesystem (POSIX is a family of standards that many UNIX-like OSes adhere to)
- Note that the bottom-most and top-most layers in Figure 3 are the same for ZFS and UNIX. Both build on top of device drivers (at the bottom) and export a set of system calls (at the top).

Discussion

- What are the performance tradeoffs here? Especially for copy-on-write. What do we gain by using copy-on-write?
- How does this relate to lecture? After all, we are not talking about filesystems in lecture right now.
- There's no evaluation section in this paper. What would you evaluate about ZFS?