CS 134
Operating Systems

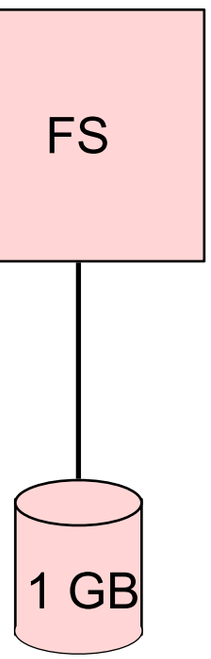
April 1, 2019

ZFS

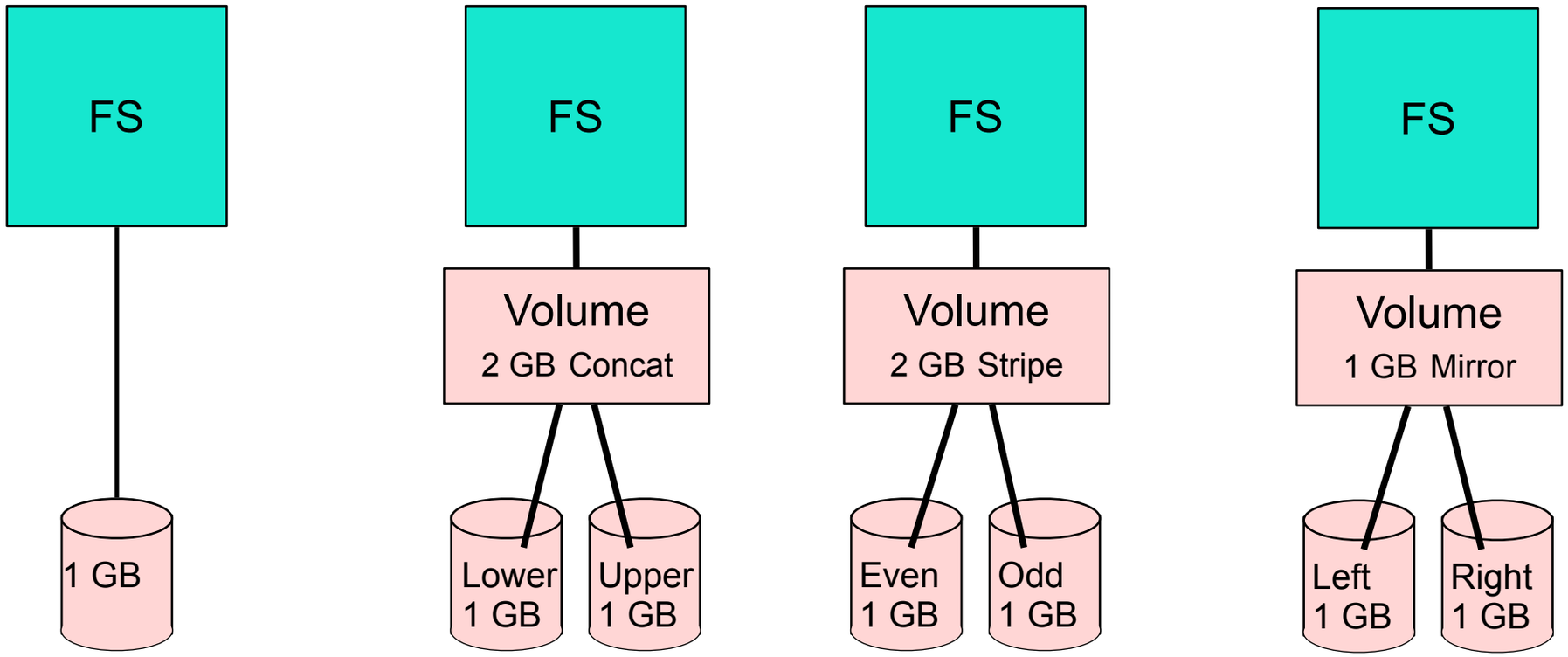
ZFS

- Developed in early 2000's at Sun (now Oracle)

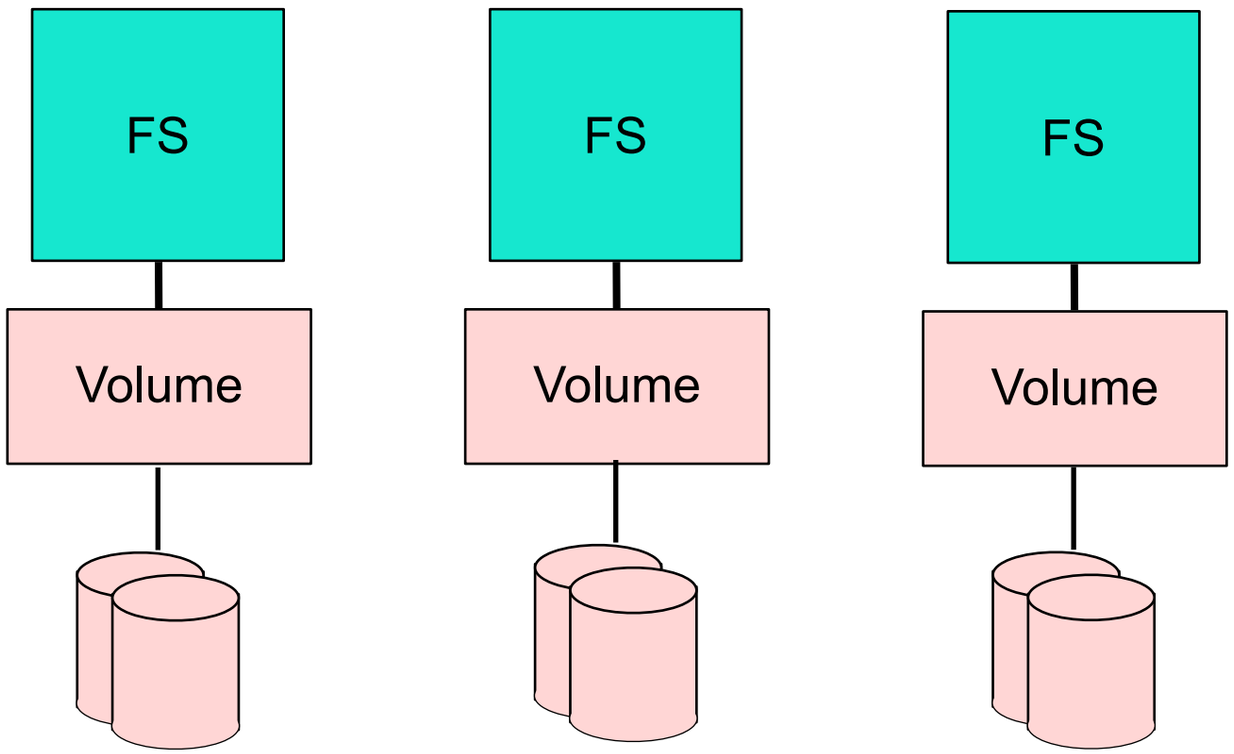
Traditional FS: One/disk

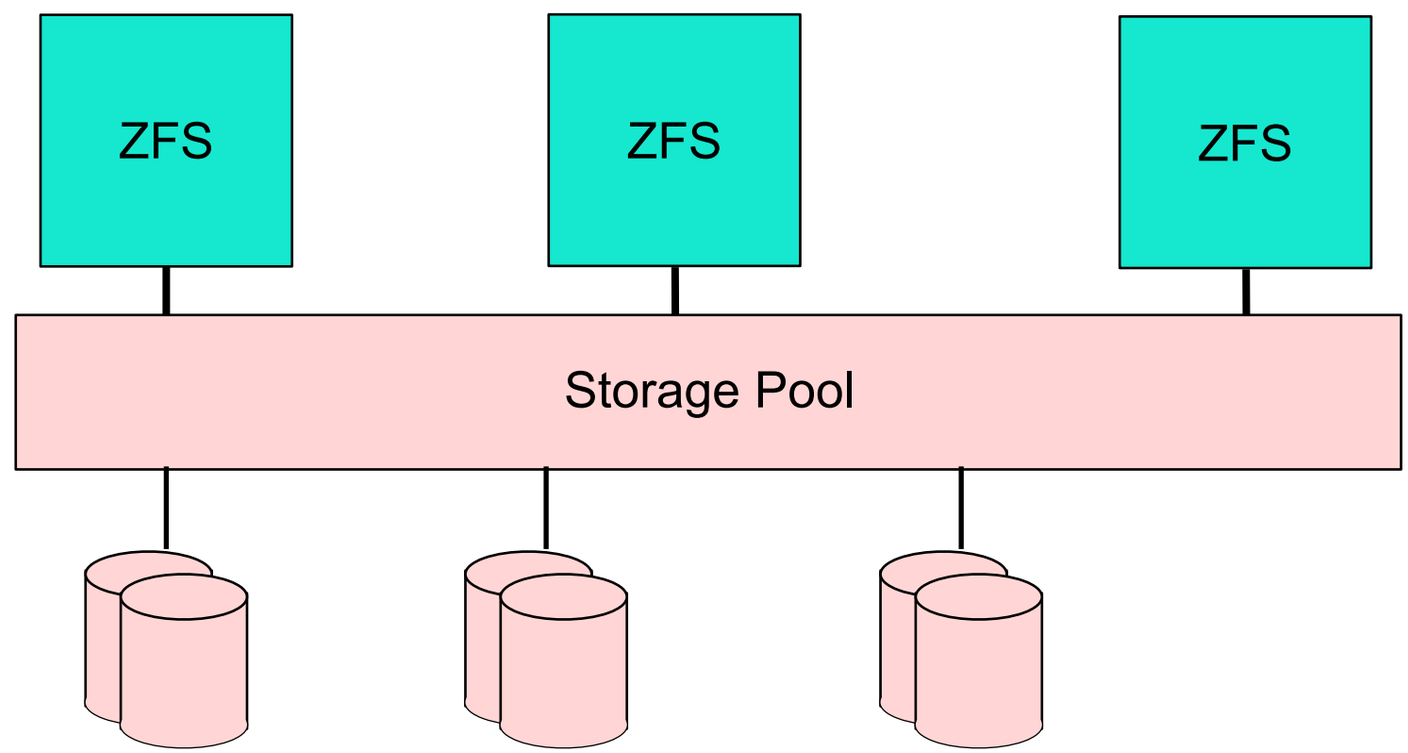


Larger FS: One/ many disk



Pool approach





Advantages of pooling

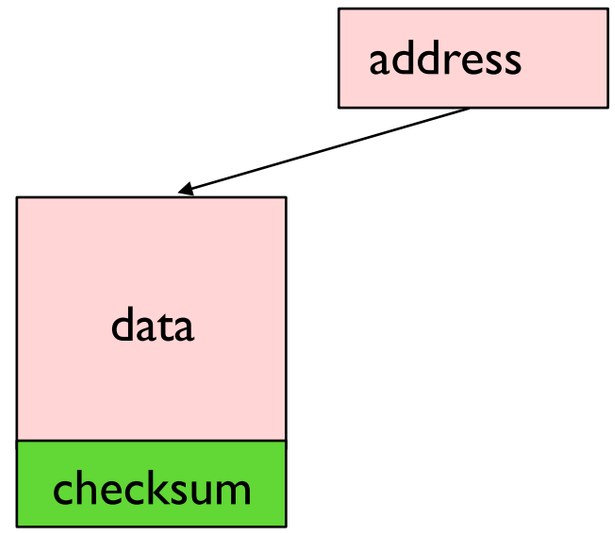
- Dynamic filesystem size
- All storage in the pool is shared
- Easily add new drives to the pool (dynamic pool size)

Data integrity

- Checksumming
- Copy-on-write
- Transactional

Checksumming

- Which of the following errors are caught if we store a checksum in each block?



Bit rot

Phantom writes

Misdirected reads and writes

Memory errors (cosmic ray)

Driver bugs

Accidental overwrite

Checksumming

- Which of the following errors are caught if we store a checksum along with a pointer to the block?

Bit rot

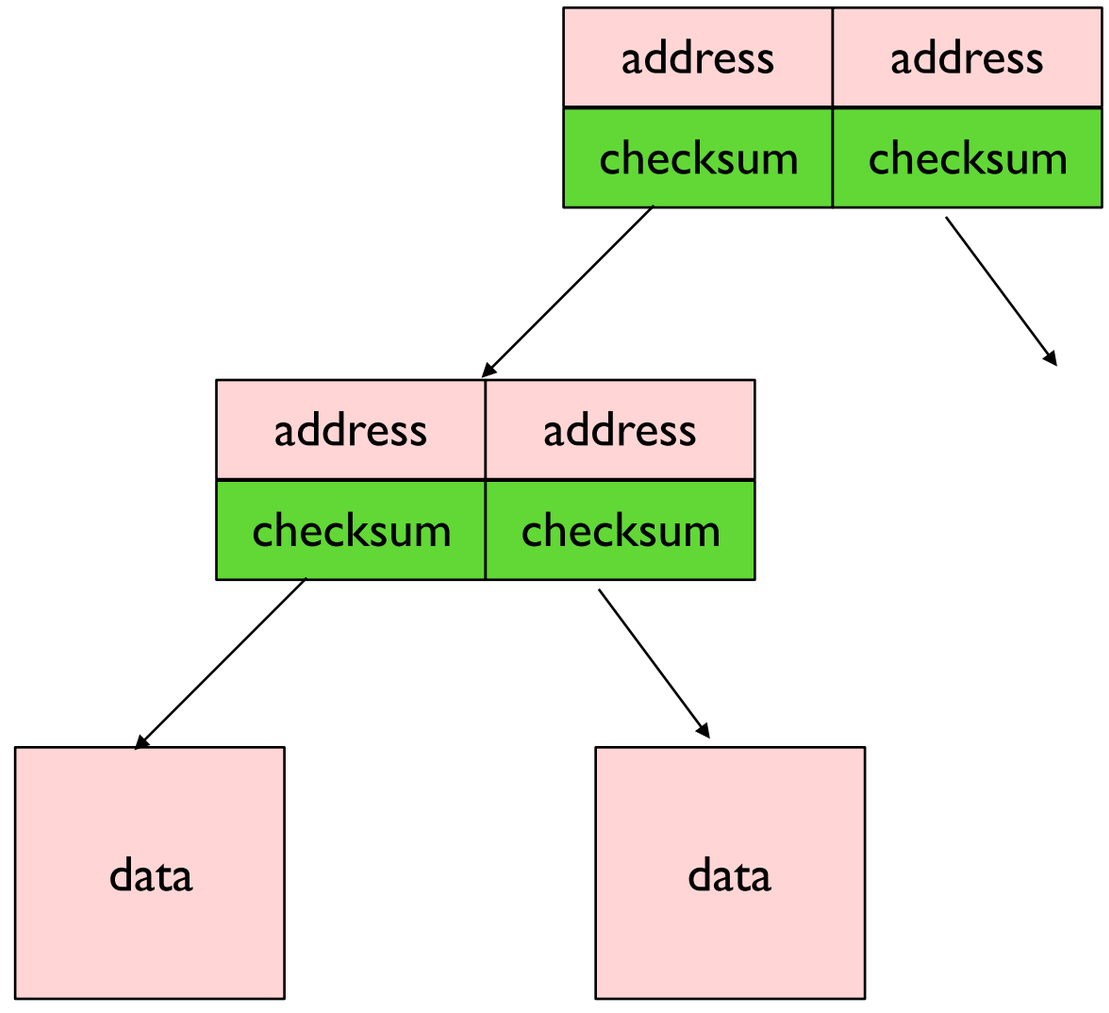
Phantom writes

Misdirected reads and writes

Memory errors (cosmic ray)

Driver bugs

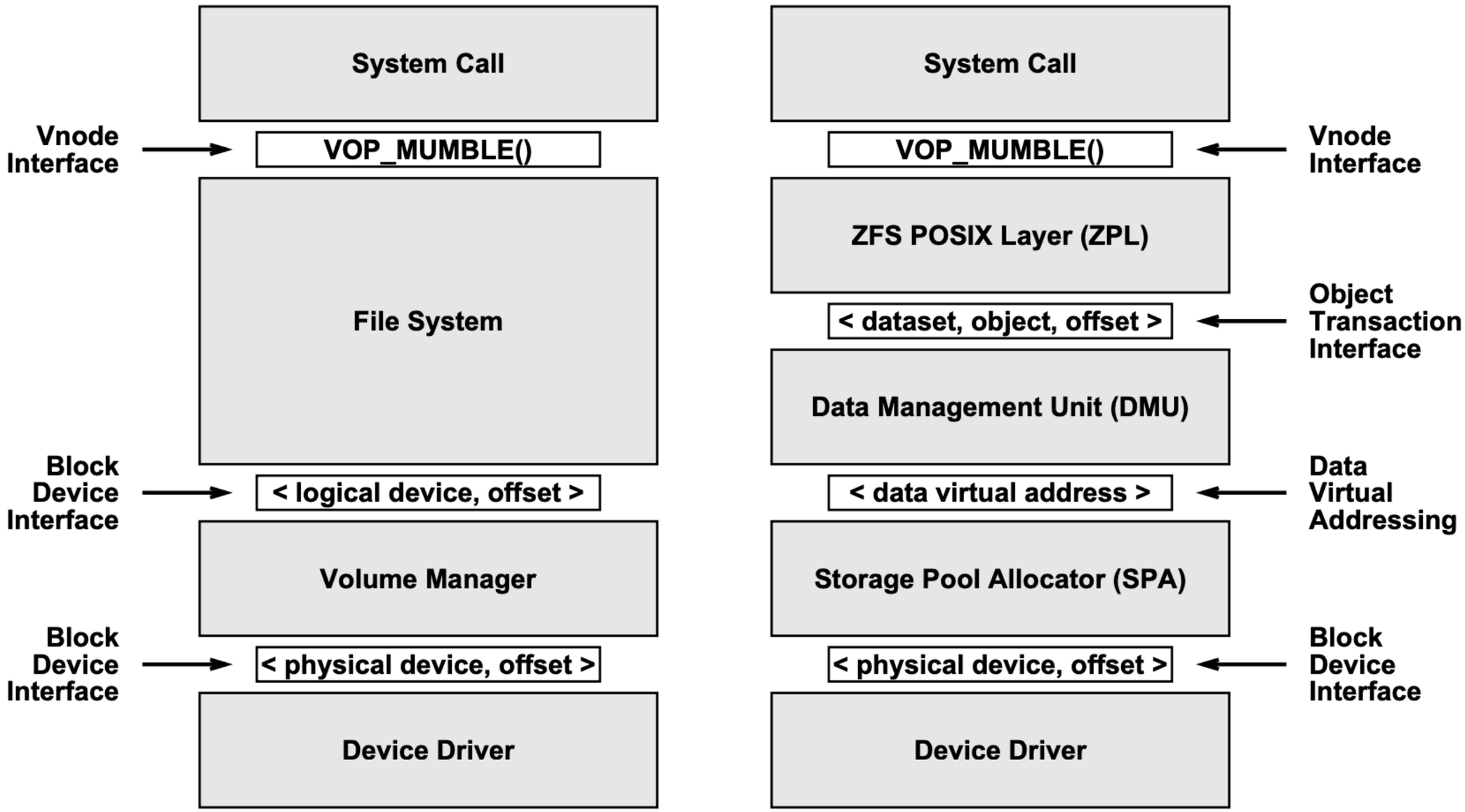
Accidental overwrite



Fundamental Theorem of Software Engineering

- All problems in computer science can be solved by another level of indirection

Block diagram



Traditional ZFS

Figure from The Zettabyte File System, Bonwick et al.

Data virtual address

- Storage pool provides malloc/free for disk space

- Can allocate (variable-size) disk blocks and receive back data virtual addresses (128 bit!)

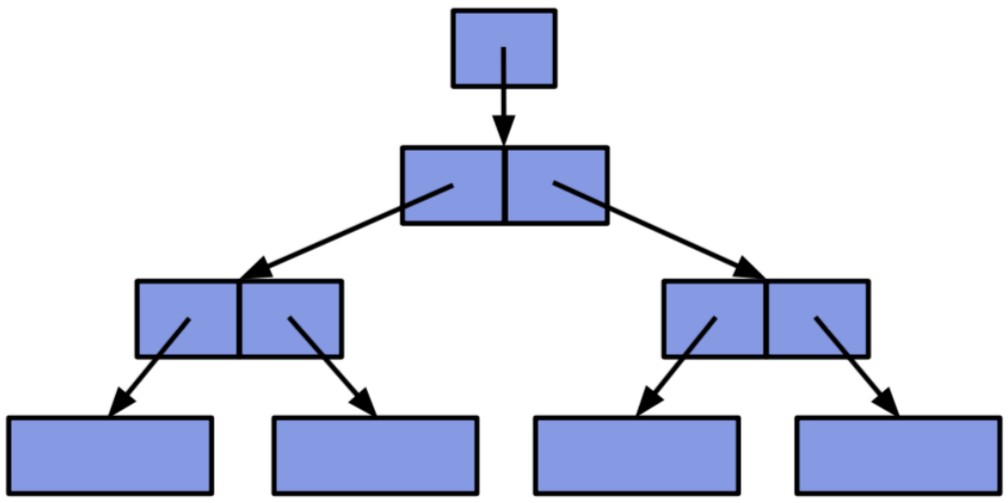
- Can deallocate data virtual addresses

- Translation from data virtual address to device and offset handled by Storage Pool Allocator

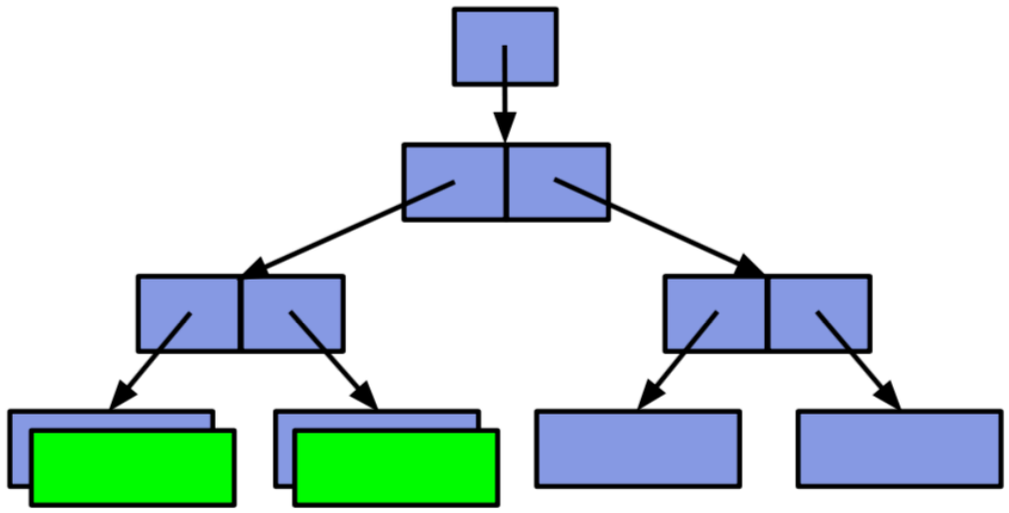
Copy-on-write

Copy-On-Write Transaction Groups (TXG's)

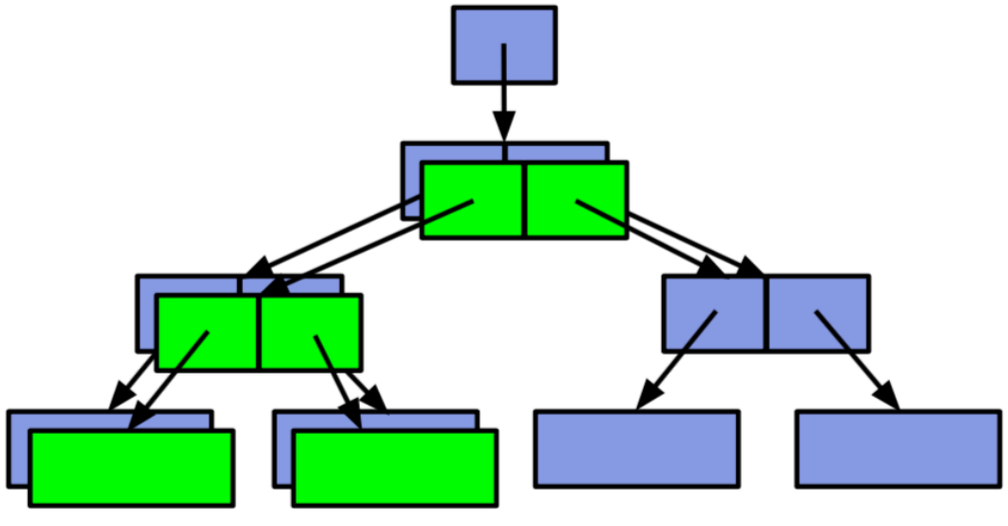
1. Initial block tree



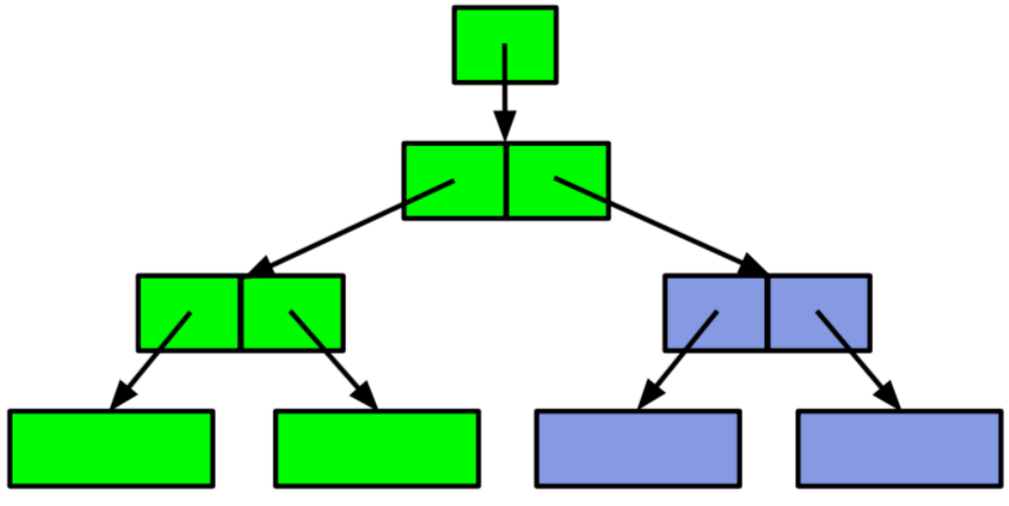
2. COW some blocks



3. COW indirect blocks

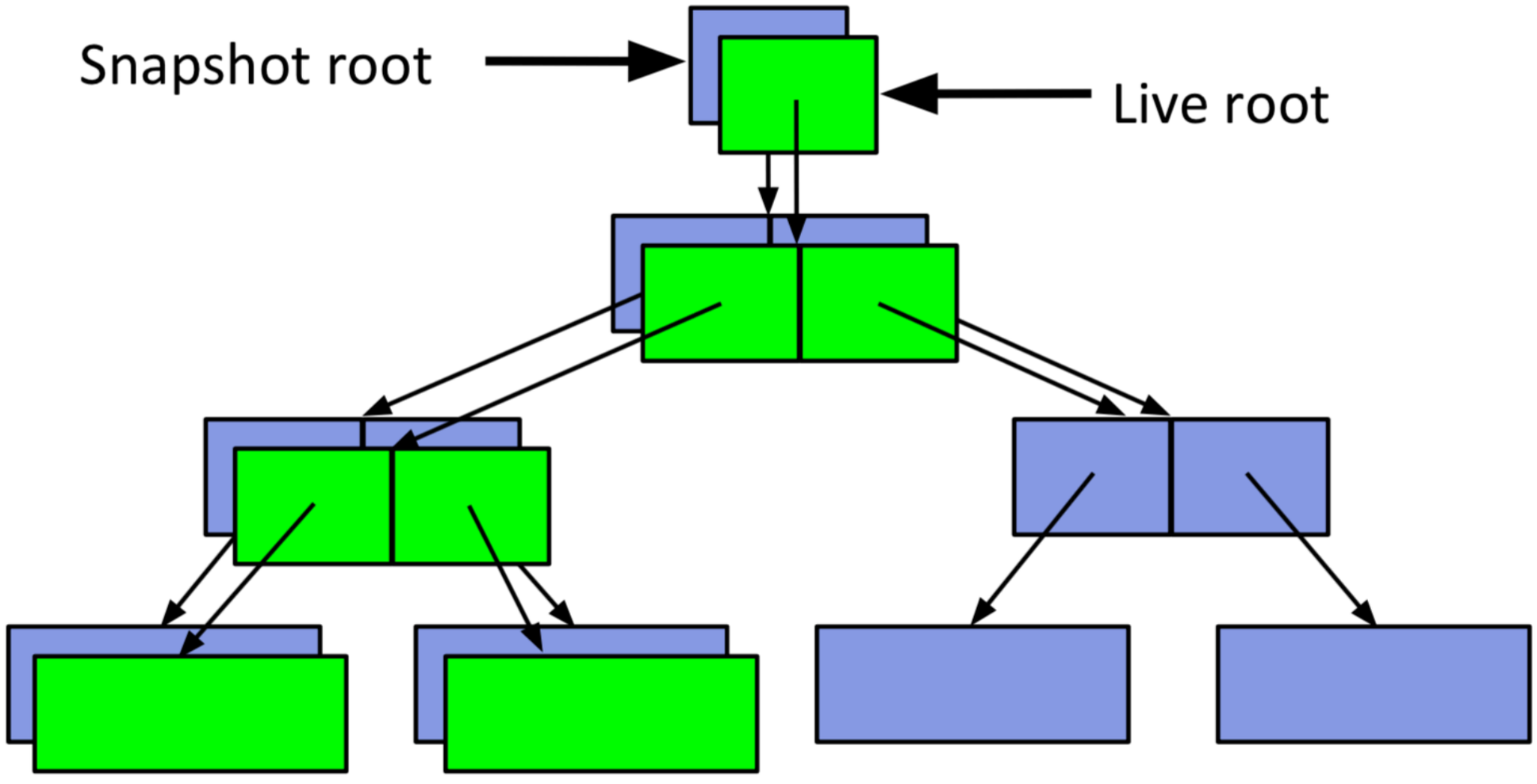


4. Rewrite uberblock (atomic)



Constant-time snapshots

- At end of transaction group, don't free COWed blocks!



I/O Life Cycle: Writes

- Translated to object transactions by the ZFS Posix Layer:
  “Make these 5 changes to these 2 objects”
- Transactions bundled in Data Management Unit into transaction groups that flush when full (>% of system memory) or at regular intervals (30 sec.)
- Blocks making up a transaction group are scheduled and then issued to physical media in the Storage Pool Allocator

I/O Life Cycle: Reads

- Heavy use of caching and prefetching
- If requested blocks are not cached, issues a prioritized I/O that has higher priority than pending writes
- Adaptive Replacement Cache tracks recently (and frequently) used blocks in main memory

Speed

- Copy-on-write design means random writes can be made sequential
- Dynamic striping across all underlying devices eliminates hot spots
- Intelligent resilvering[1] copies only live data

[1] *Resilver*: when an antique mirror gets tarnished or damaged, you make it shiny again by re-silvering it.