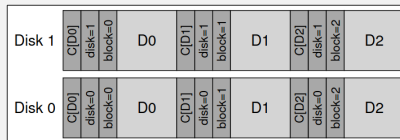


Misdirected Writes

- a *misdirected write* is a write that writes correct data to the wrong location
 - can result from firmware bugs in the drive
- add a *physical ID* to the stored info
 - contains disk, sector of the block



Lost Writes

- a *lost write* occurs when the device reports a completed write that doesn't actually complete
 - e.g. for efficiency, drive may report completion of write once data is in the drive's buffer, before it is actually written to disk
- can't be handled by a checksum on the block written
 - none of the data on disk – the data or the checksum – has been updated so there is no inconsistency
- solutions
 - *write-verify (read-after-write)* – immediately read what has just been written
 - slow (doubles the number of I/O operations per write)
 - include data block checksum in inodes and indirect blocks
 - can detect if one write is lost

Checksum Maintenance

- checksums are read when data is accessed, but most data isn't accessed often
- to prevent the accumulation of *bit rot* impacting the redundant copies needed for error correction, periodically *scrub* the disk by reading and checking all checksums

Overhead of Checksums

- space overhead
 - on disk: 8 byte checksum per 4KB block = 0.19%
 - in memory: must read checksum as well as data block
 - insignificant if checksums are discarded after check
 - something to take into account if checksums are retained (e.g. for protection against memory corruption)
- time overhead
 - CPU: checksums must be computed on both write and read
 - can incorporate checksum computations into copies (piggyback read/write-for-checksum onto read/write-for-copy)
 - I/O
 - design storage of checksums to reduce added I/O
 - scrubbing can be performed at strategic times when the system isn't busy