Reliability and Recovery

CS 537 - Introduction to Operating Systems

System Failures

- All systems fail
- Fatal failures
  - disk bearings fail
  - controllers go bad
  - fire burns down entire system
- Limited failures
  - single block on disk goes bad
  - power goes out

Fatal Failures

- In a fatal failure, all data on system is lost
- To recover data another copy must be kept
  - tape drive
  - floppy drive
  - second hard drive
- Most systems are backed up to tape on a regular basis
  - to save space, only backup files that have changed since the last backup

Limited Failures

- Limited failures may destroy some data but not all
  - single bad block may ruin one file but not all
- Destroyed data may be restored in several ways
  - from backup
  - using error correcting codes (ECC)
  - redo operations that lead to existing system

ECC

- 100's of millions of bits in memory
- 100's of billions of bits on a disk
- Virtually impossible to make a memory or disk without bad bits
- Must find a way to deal with these inevitable bad bits

ECC

- When a given set of bits are stored, an ECC code is calculated
  - this code is stored with the data
- When data is read, the ECC is recalculated and compared with that stored
- If they don't match, there is an error in the data
- These calculations and checks are usually performed by hardware
  - memory controller or disk controller

ECC

- Single error correcting, double error detecting
  - using the ECC code, it is possible to find and correct a single bad bit
  - it is possible to find up to two bad bits and inform the user of the problem
- With more complicated math, you can correct more bits and determine more errors

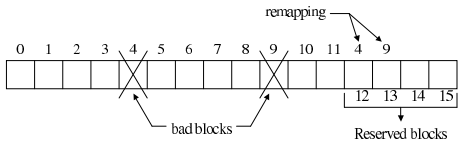
ECC

- do math here

Block Forwarding

- Set aside a number of disk blocks
  - under normal operation, these blocks are not used at all
- If a bad block is detected, the controller will re-map that block to one of the reserved blocks
- All future references to the original block are now forwarded to the new block

Block Forwarding



•All references to blocks 4 and 9 are now forwarded to blocks 12 and 13

Block Forwarding

- This indirection keeps things working
- This indirection can hurt performance
- Disk scheduling algorithms don't work as well any more
  - OS doesn't know about the remapping
  - using the elevator algorithm could now jump all over the disk

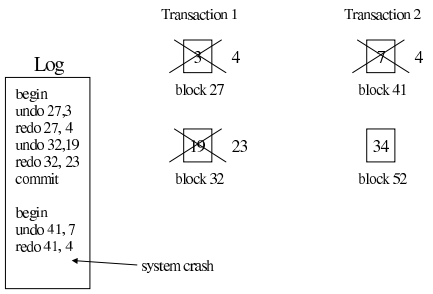
Transaction

- A transaction is a group of operations that are to happen atomically
  - transactions should be synchronized with respect to one another
  - either all of the operation happens or none of it
- This can be difficult to do in the event of a system failure

Logging

- Keep a separate on disk log that tracks all operations
- Mark the beginning of transaction in log
- Mark the end of transaction in log
- On reboot from failure, check the log
  - any transactions that were started and not finished are undone
  - any transactions that were completed are redone
    - this has to be done because of caching in memory

Logging



Logging

- To recover from the above system crash
  - redo transaction 1
    - scan the log and perform the *redo* operations
  - undo the effects of transaction 2
    - scan the log in reverse and perform the *undo* operations
- To make this work, the log should only be written after a transaction has completed

Shadow Blocks

- Never modify a data block directly
- Make a copy of the data block (a *shadow block*) and modify that
- When finished modifying data, make the parent point to the shadow block instead of the original
- Of course, this requires making a copy of the parent to be modified
- This chain continues up to the root
  - when root is written the transaction is committed
  - original blocks are then freed

Shadow Blocks

- In the event of a crash
  - if the transaction did not complete
    - garbage collect the shadow copies
  - if the transaction did complete
    - garbage collect the original copies
- Garbage collection is easy
  - scan the data tree
  - any block not in the tree is garbage

Shadow Blocks

•Modify the data in block 6 from Y to Z