File System Reliability / Crash Consistency

ECE595
Mar 27

Y. Charlie Hu

Roadmap

- Functionality (API)
  - Basic functionality
    - Disk layout
    - File operations (open, read, write, close)
  - Directories
- Performance
  - Disk allocation
  - Buffer cache
    - Interactions with VM
  - File System Interface
  - Disk scheduling
- Reliability
  - FS level
  - Disk level: RAID

File system reliability

- Loss of data in a file system can have catastrophic effect
  - How does it compare to hardware (DRAM) failure?
  - Need to ensure safety against data loss
- Three threats:
  - Accidental or malicious deletion of data → backup
  - Media (disk) failure → disk mirroring (RAID)
  - System crash during file system modifications → consistency

1. Backup

- Copy entire file system onto low-cost media (tape), at regular intervals (e.g. once a day).
  - Implementation – do we need to copy the whole FS?
- In the event of a disk failure, replace disk and restore from backup media
- Amount of loss is limited to modifications occurred since last backup

2. Mirrored Disks

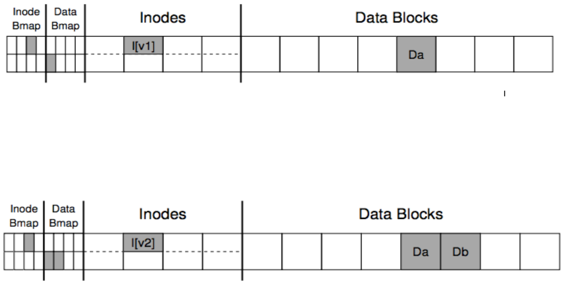
- Multiple copies of the file system are maintained on independent disks
- Disk writes update all redundant disks in parallel
- Used in applications that cannot tolerate any data loss (what applications?)

RAID Disks
(redundant array of independent disks)

- Use multiple parallel disk drives for higher throughput and increased reliability
- e.g. each bit of a data byte is stored on one of 8 disks, a 9th disk stores a *parity bit* for each data byte
- Can recover the data byte if 1 disk fails
- (more next week)

3. Crash Recovery

- After a system crash in the middle of a file system operation, file system metadata may be in an *inconsistent state*
  - Independent of buffer caching


Inode Bmap
Data Bmap
Inodes
Data Blocks
I[v1]
Da
Inode Bmap
Data Bmap
Inodes
Data Blocks
I[v2]
Da
Db