Fast File System (at least in name)

CS333 :: S20

Course Logistics

- Lab 2
  - Where are we at?
- Evening hours?
  - 5:30(ish)-7pm in TCL 312
- Lab 3a,3b: FUSE RefFS

Last Class

- FUSE: file system in user space
- Away at USENIX FAST

This Class

- FAST report
- Fast file system
  - Overview of FFS goals
  - Handout
  - Activity (allocation/placement heuristics)

Recall: Key VFS Data structures

- Inode
  - Persistent information about a single file
  - “Index” node (indirection node?)
- Superblock
  - Persistent information about entire file system
- Allocation structures
  - Free list, bitmap, extent list, etc.

Key **FFS** Data structures

- Inode
  - Persistent information about a single file
  - “Index” node (indirection node?)
- Superblock
  - Persistent information about entire file system
- Allocation structures
  - Inode bitmap, data bitmap

FFS set the stage for FS design

- The FFS Designers:
  - Thought hard about HDD performance
  - Abstracted common file system structures & methods
  - Identified performance bottlenecks
  - Implemented “Common sense” heuristics: use *device awareness* to improve performance
    - Downsides to device awareness?

Problem 1: Dependent Reads

- To read file data, must first read the inode
  - How did the “simple file system” determine placement?

Core issue: data and metadata separation

Problem 2: Small Block Size

- How does **`grep`** work?
- Ideally, what do we want the I/O pattern of our file system to be when running a (recursive) **`grep`**?

Problem 3: Free Space Fragmentation

- Problems with first fit LBA allocation?

Ideas

- Keep related things together
- Cylinder groups (block groups)
- Allocation heuristics:
  - Directory allocation
  - File Allocation
  - File block allocation

Problems

- All heuristics… no guarantees!
- Once you make a decision, you're stuck with it
- Aging: file system performance degradation over time
  - What operations/workloads might cause problems over time?
  - Defragmentation?

HDD Handout

Activity: Allocation & Placement

https://github.com/williams-cs/cs333-class