

The Design and Implementation of a Log-Structured File System

Mendel Rosenblum and John K. Ousterhout
Electrical Engineering and Computer Sciences, Computer
Science Division
University of California
Berkeley, CA 94720

Conference: ACM Transactions on Computer Systems
(TOCS) 10.1 (1992)

Problem Statement

A file system to take into account the following advances:

- Increasing memory sizes
- Writes contribute more to the disk traffic
- More increase in transfer bandwidth as compared to decrease in seek and rotational delay costs

Core idea of LFS

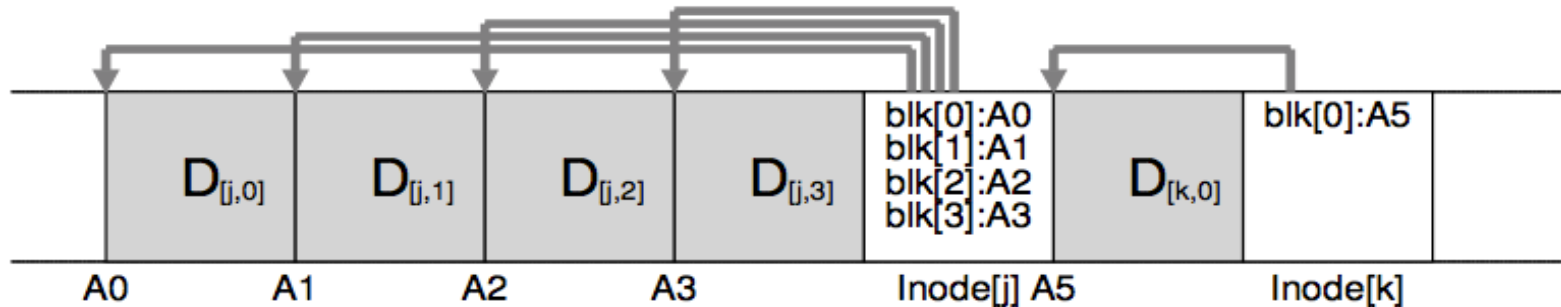
- Sequential writes
- It uses transfer bandwidth better with very less rotational delays.
- Writes are much more efficient

Details

- LFS stores consecutive writes in a memory segment
- When the segment is full, it is written to the disk in one long, sequential write
- LFS never overwrites data on the disk
- Size of segment is determined by transfer rate and rotational delays
- Need to update inode of the file

Details

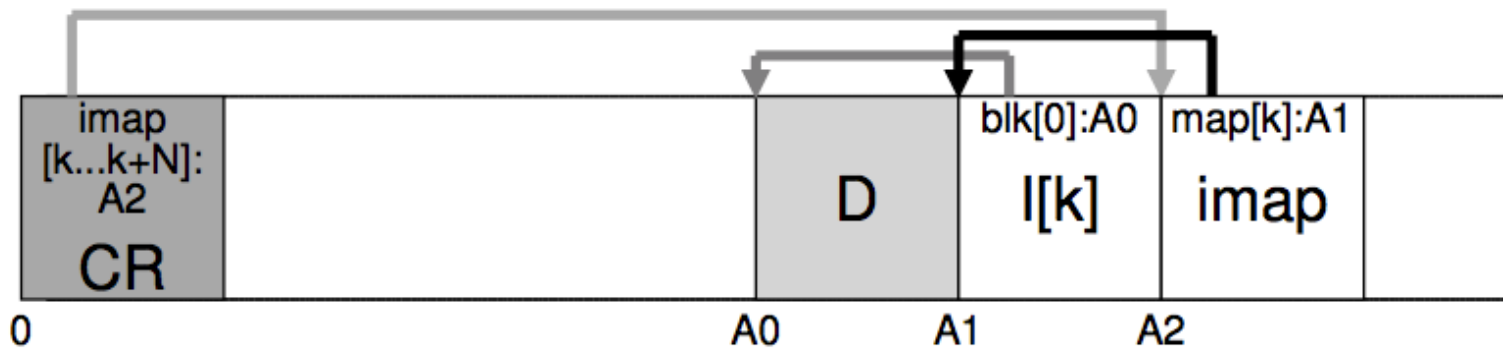
- Added in the same segment as in the following figure: (Source [2])



- Where are the inodes?
- Inode map needs to be updated
- Also stored in the same segment

Details

- Checkpoint region to store recent imap locations as seen in the following figure: (Source [2])



- Older versions of data needs cleaning up
- Use of log and 2 checkpoint regions for crash recovery

Garbage Collection

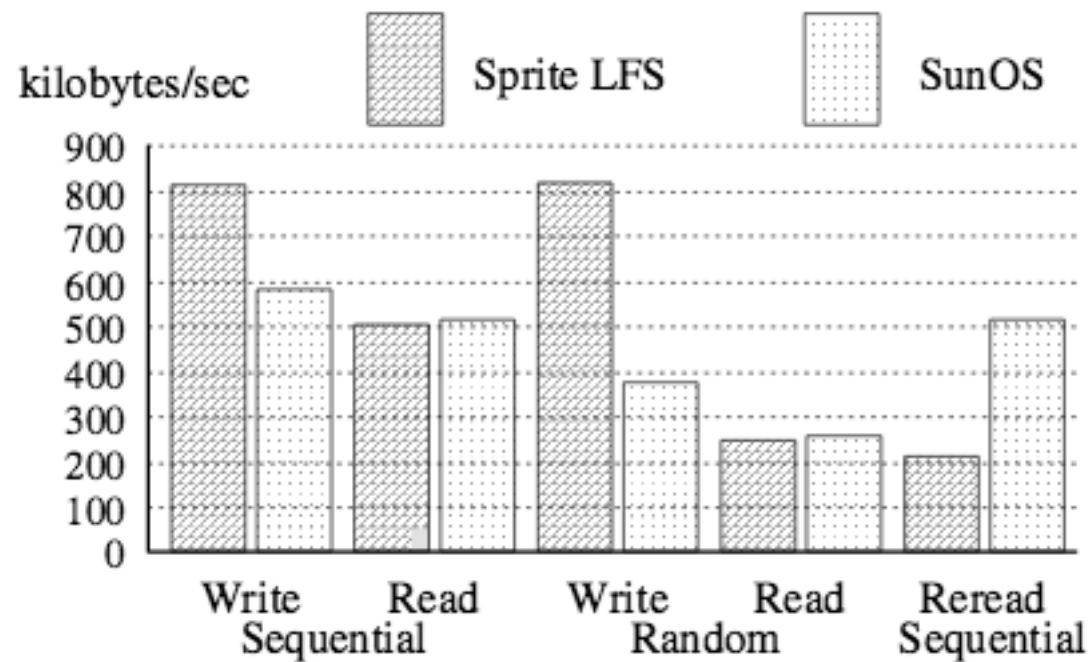
- Only deleting the old data blocks is not enough
- Free holes between allocated spaces
- No contiguous region left to write sequentially
- LFS picks some old segments
- Searches for live data blocks
- Combine these data blocks to form new segments

Garbage Collection

- When is the cleanup done? Periodically
- Determining liveness: LFS stores version number for each block and increases version number in the imap whenever there is an update
- Which blocks to clean? Challenging
- Hot segments where updates are more often
- Don't use these to clean up

Strengths and Weaknesses

- Source [1]



References

1. Rosenblum, Mendel, and John K. Ousterhout. "The design and implementation of a log-structured file system." ACM Transactions on Computer Systems (TOCS) 10.1 (1992): 26-52.
2. "Log-structured File Systems": <http://pages.cs.wisc.edu/~remzi/OSTEP/file-lfs.pdf>