

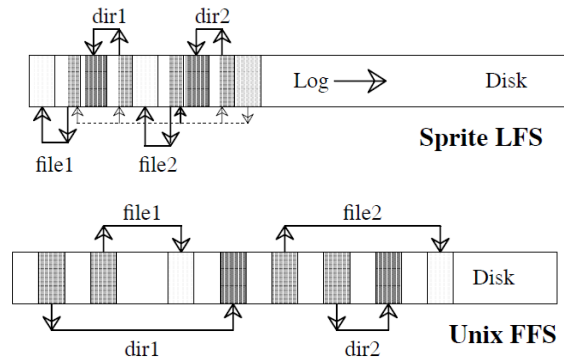
Graduate Operating Systems (Storage & File Systems)





Fall 2020

Paper “Sprite LFS”

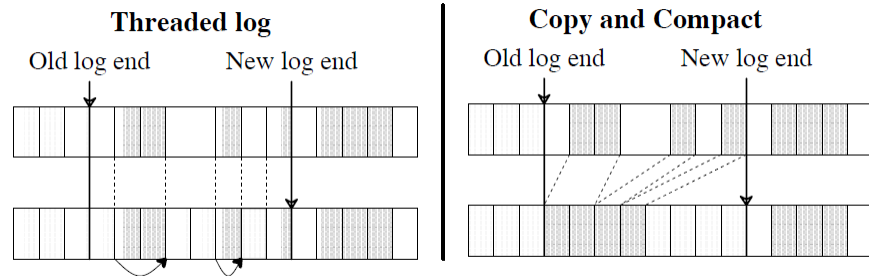
- Does Sprite LFS improve READ or WRITE performance?
- What is the biggest challenge of a log-structured file system?
- Does Sprite LFS focus on large or small files?
- How does Sprite LFS keep reading performance acceptable?




Paper "Sprite LFS"



Block key:  Inode  Directory  Data  Inode map

Paper "Sprite LFS"



Block Key:
 Old data block 
 New data block 
 Previously deleted 

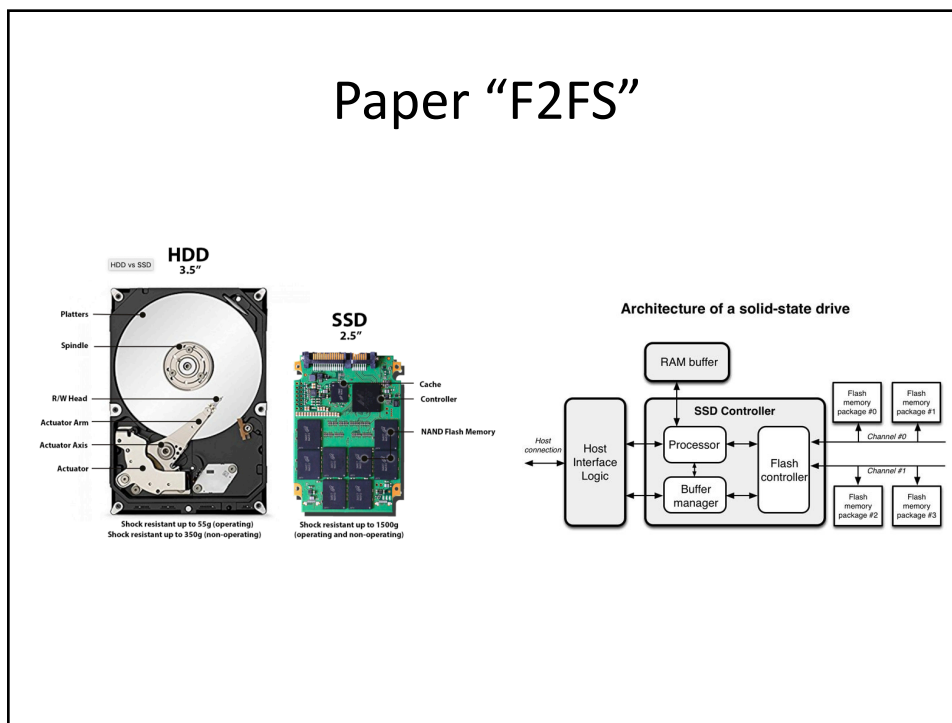
Paper “Sprite LFS”

- Segments, segment cleaning, segment summary block
 - When should the segment cleaner execute?
 - How many segments should it clean at a time?
 - Which segments should be cleaned?
 - How should the live blocks be grouped when they are written out?
- Cost-benefit policies
 - Cold segments should be cleaned at high utilization
 - Hot segments should be cleaned at low utilization
- Checkpoints
- Experimentation; metrics (“write cost”); microbenchmarks; pros/cons; overheads

Summary “Sprite LFS”

- Improve write performance
- Crash recovery
- Concept of segments
- Segment cleaning (garbage collection)
- Pros & cons?

Paper “F2FS”

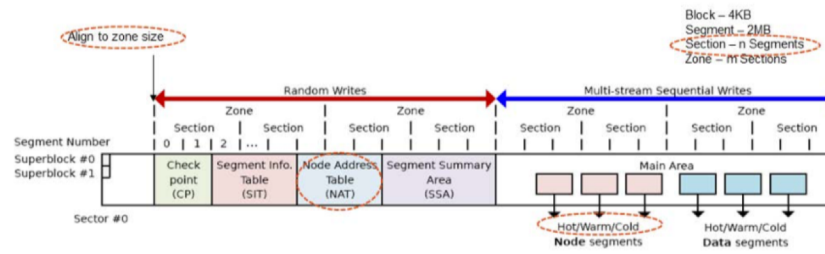


Paper “F2FS”

- NAND flash memory
- Sequential vs. random writes
 - Fragmentation, life time

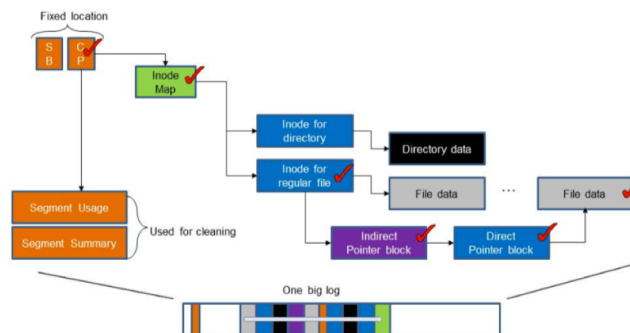
Paper "F2FS"

- Flash-Friendly On-Disk Layout
 - Segment as basic unit
 - Section: consecutive segments
 - Zone: consecutive sections
 - Node: inodes/indices; data: directory/user file data
 - Separation of meta data and "actual" data



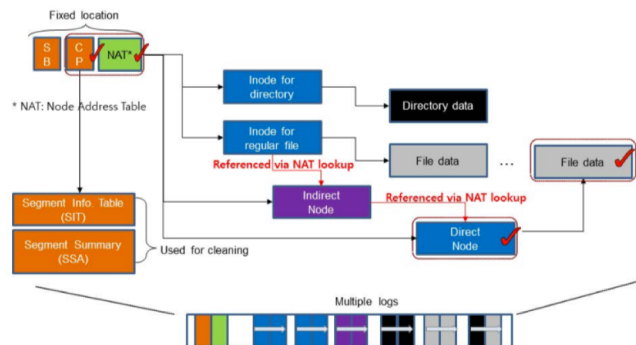
Paper "F2FS"

- Cost-Effective Index Structure
- LFS:



Paper “F2FS”

- Cost-Effective Index Structure
- F2FS:



Paper “F2FS”

- Wandering tree problem
 - LFS: when a file data is updated and written to the end of log, its **direct** pointer changes, its **indirect pointer block** is also updated, and upper index structures (inode, inode map, checkpoint block) are also changed

Paper “F2FS”

- Multi-Head Logging
 - Data temperature classification: hot, warm, cold
 - Six logging segments
- Cleaning
 - Section-level; foreground (need more sections) and background (periodic kernel thread)
 - Greedy for foreground, cost-benefit for background
- Adaptive Logging
 - Switch between normal and threaded logging