Paper summary

In the '90s, the CPU speed increased dramatically, and the process was limited because of disk access. A new disk management technique called log-structured file system is introduced to address this problem, which writes all the data sequentially to improve speed and crash recovery by exploiting temporal locality. Low cleaning overhead is achieved by balancing a trade-off between cost and benefit. The implemented prototype called Sprite LFS is tested on different microbenchmarks and heavy workload and demonstrates excellent results against Berkley FFS except for reading a file sequentially, which is written randomly. Crash recovery in Sprite LFS uses "checkpoint" and "roll-forward." The log-file system avoids I/O being a bottleneck and takes advantage of faster processors.

Strengths

1. Sprite LFS is already in production.

2. Acknowledges the limitation of micro-benchmarks and justifies why they would give a bias opinion for Sprite LFS.

3. Both micro and macro benchmarks are provided with diagrams stats and diagrams. Non-intuitive results are highlighted and elaborated as well.

4. It addresses the potential issues that need to be implemented for the effective use of the system.

5. All the new terms are explained lucidly.

Weaknesses

1. Since crash recovery was a significant contribution, they could have elaborated on why it is more straightforward than Berkeley FFS.

2. It might not work for flash and SSD systems like FFS.

Comments for author

Overall, I enjoyed the paper, especially Section 5. It was easy to follow, and the benchmarks used were much better and elaborated than any of the past papers that we have read.