

Critique Week 7

Andres Calderon - SID:861243796

February 17, 2016

The design and Implementation of a Log-Structured File System (Rosenblum and Ousterhout, 1991)

Disk-bound applications became a bottleneck in comparison with increasing speed on CPUs at that time. The authors present a new technique for disk storage based on a log-structured file system (LFS). As advantages it is able to speed up writes and offer a better support for crash recovery. It also provides indexing capabilities to support reads. The paper focuses on a segment cleaner process to ensure appropriate management of free space. As a final result, they present and evaluate Sprite LFS as a prototype of a log-structured file system.

In the first part of the paper, it stresses the drawbacks of low performance disks, particularly for transfer bandwidth and access time. Also it highlights the importance of small files mostly present in office and engineering environment and how current file systems did not deal with them appropriately (citing too many synchronous writes). So, the work focuses on improving the write performance by buffering changes and then writing them in a single long write operation in order to make efficient use of the bandwidth. However, it will face two issues: how to retrieve information efficiently and how to manage free space on disk.

Accomplish the first issue is relatively simple. Sprite LFS provides index data structures located on the log to allow random-access retrievals. Table 1 summarizes the details and purpose of each of them. The paper focuses more on the second issue. For an efficient free space management they propose the concept of segments as a combination of the threading and copying strategies. In addition, they implement a clear segment cleaning mechanism and policies in a similar fashion as garbage collectors work in programming languages.

Then, the paper focuses on policy details, for example when and how to execute the cleaner. In particular, the paper explores which segments should be clean and how to group the remaining live blocks. The authors state a tradeoff between performance and space utilization and realize that to achieve high performance at low cost the disk has to be forced into a bimodal segment distribution. They reach this goal by implementing a new policy which rates each segment according to its benefit and cost of cleaning.

Finally, the authors test Sprite LFS with a set of micro-benchmarks into an operational implementation of the Sprite network operative system with around 40 real users. Overall, the performance of Sprite LFS is near to one order of magnitude better in comparison with SunOS (based on Unix FFS) for writes and performs relatively good for reads. Personally, I find interesting the related work section where the authors state that LFS shares ideas from quite diverse systems, for example: log-like structures from file systems, garbage collectors from programming languages, and logging schemas, large commits and crash recovery from database systems.