CSCE 410/611 Operating Systems Spring 2023

Homework for Week 14

(Due Date: Check Canvas)

1. (variation of [1]) Describe the modifications to a log-structured file system that would happen when a process requests the creation of a new file /tmp/foo and writes to that file sequentially until the file size reaches 20 KB. (This question is vague. Try the best you can.)

It would create a write buffer that stores all writes in a buffer, and at once writes the whole segment into the memory.The LFS buffer then updates in-memory segment, inodes, checks inode pointer, and then calls garbage collection to clear out any cold inodes in the write buffer.

2. You have a NAND Flash Solid State Drive in your system. Which among the following files systems will give the best performance? You can choose among: UNIX Original File System, UNIX Fast File System, Log-Structured File System. Argue why yours is the best choice for a mixed workload, with a combination of file creations, deletions, random, and sequential accesses. (Feel free to speculate.)

Log-Structured File System would be the best choice for NAND Flash

The use of the write buffer that only stores metadata (inodes & bitmap). It will need to update the checkpoint region of the inode bitmap but uses it for proper RAM accesses.

UNIX Original File System and UNIX Fast File system

Both were designed for disks. The UNIX original system created traffic jams in the writes with 5 IO operations per write being called, and the worst of all the UNIX original file system treated the disk as if it was RAM memory. FFS is optimized to use cylinder groupings, sub directories, symbolic links, and more of a star pattern for sequential writes so the disk head is more likely to find 2 after 1. This did not change the number of IO operations, but changed how long the IO works to find them and improved on original UNIX’s 2%

References

[1] A. Silberschatz, P. Galvin, and G. Gagne, *Applied Operating Systems Concepts*, John Wiley & Sons, Inc., New York, NY, 2000.

[2] Deitel, Deitel, and Choffnes, *Operating Systems*, Pearson / Prentice Hall, 2004. [3] A. S. Tanenbaum, *Modern Operating Systems*, Pearson / Prentice Hall, 2008. [4] L. F. Bic, A. C. Shaw, *Operating Systems Principles*, Prentice Hall 2003. [5] C. Crowley, *Operating Systems, A Design-Oriented Approach*, Irwin 1997. [6] M. Herlihy, N. Shavit, *The Art of Multiprocessor Programming*, Elsevier, 2008

1