Greenlee 560 Summary 6

Elliot Greenlee

March 21, 2017

1 Introduction

This paper details a faster reimplementation of the Unix file system.

2 Problems

What problems does this paper try to solve? Why are such problems important?

The old Unix file system had low system memory throughput. Higher throughput than 512 bytes is needed for image processing and other modern tasks. The free list organization previously became random over time, causing files to be allocated across the free list, requiring random seeks for every file access.

3 Assumptions

What are assumptions made by this paper? Are they verifiable? Are there logical holes in such assumptions?

This paper assumes that increasing the block size is a good method for improving throughput, based on a performance improvement of twofold from a previous block size doubling. Large block sizes have an issue of wasting space because many UNIX file systems have smaller files. The paper writers empirically measured the waste of space present in their current and new systems to compare.

4 Solutions

What are the major solutions of this paper? Do you think the solutions in this paper will work for the problem? Do you think the paper evaluates the solutions in a convincing manner? List two limitations of the solutions proposed in this paper, and outline your method to fix them.

In the new file system, the data is organized so that larger blocks can be accessed in a single disk transaction, which increases system throughput. Increasing the block size actually allows transfers of up to four times as much information per transaction. The new file system solves the issue of smaller files by allowing single file blocks to be fragmented into 2, 4, or 8 addressable fragments at creation. The new file system bases block allocation on hardware capabilities so that the speed of the processor and characteristics of mass storage devices all contribute to the configurability of parameterization.

Proof of success comes from the empirical data proving effectiveness over the long term performance.

4.1 Limitations

There could be an issue with allocating the last disk block as a fragment that might be separated from all the others. If someone is accessing the last block many times that would hurt performance because it requires a large seek separate from the rest of the file. It comes down to a tradeoff between space waste and time.

No deadlock detection is done by the given locking mechanism. This seems untenable, and no explanation is given in the paper. Surely in the future some deadlock system is needed for system preservation.

5 Notes

5.1 Discussion