

*In what ways did the performance characteristics of the hard **disks** of the time influence the design of **FFS**?*

Answer:

Firstly, old File system transfers only 512 bytes per disk transaction and often find next block is not on the same cylinder, which lead to many seeks and limit the file system throughput. Thus, they increase the block size to improve the throughput of FFS.

Secondly, as files were created and removed, free list become random, which increase seeks and access time. Thus, in FFS, some efficient hash algorithms are used to predict the location of the file, so that each time the file is stored, it is not randomly allocated block blocks and the file storage block is limited to a small range, which improves the access speed of the file block.

Thirdly, FFS only allows writing a few megabytes of content in a continuous space, which is not good for large files, but it improves the reading speed of small files, which is still very cost-effective.

What characteristics of modern workload patterns, as described in the paper on "A File is Not a File", are a good match for FFS? What characteristics are a poor match?

Answer:

Good match: A file is not a file; Sequential access is not sequential; Auxiliary files dominate; Frameworks influence I/O

Poor match: Writes are often forced; Renaming is popular

Paper review:

The old file system is very simple, but the simplicity of the design leads to poor reliability and performance. Thus in FFS, many methods like larger block size, fragments and cylinder groups are mentioned to increase reliability and performance. So problems lead to more advanced system.

With the development of applications, how can we deal with the modern workload patterns? In paper A File is Not a File, they list problems, compare them with old file system, and propose many feasible solutions by analyze the I/O behavior of iBench. So Measurement drives file-system design. Applications develop so quickly nowadays, which means system design will develop as well.