1 State of art

We live in an age of information. Hardware and software technology improves every day. There is a need to work with large databases, multi-media applications and to store great amount of data on a memory device. This cause great pressure on performance of storing and retrieving information, viz. input and output(I/O) operations.

Part of an operation system that handles communication with a physical device is called a file system. Because of such increasing requirement on performance of I/O operations, there has been quite wide technological progress in approach to performance issues.

In past years, great many tools and tests (e.g. benchmarks) were developed as means for users and researchers to explore behavior and measure performance of file systems. These usually consist of putting file system under some kind of stress, gathering different measurements in the process and computing overall summary after the test is done. Measured parameters are usually bandwidth, latency and IO operations per second.

The benchmark should be highly configurable, so the researchers can simulate various workloads that would mimic real life scenarios. Stability and reproducibility of tests are also required, if the research is meant to be objective and to have some academic value about file system itself. [3]

Configuration of used benchmark is highly crucial for research to have realistic result, however, large amount of benchmarks are flawed or arbitrary in nature, thus are impossible to configure to provide realistic measurements [4].

The standard workflow is tu run benchmark on a clean instance of operation system without other applications running, to avoid noise, and on clean instance of researched file system.

While this approach brings great results, with correct configuration of the benchmark, it only gives us a general idea about how does file system perform in early stage of it's usage.

However, there is a growing demand from users as well as from developers to explore behavior and performance of a file system, that has been running under some defined conditions for longer period of time, months or even years [2].

Prolonged usage of file system impel it to do more and more optimisations. When there is a lot of free space, there is generally no need for complicated approach. But with more files being deleted, created and expanded, the free space begins to be more and more fragmented. This then leads for files to be spread non contiguously across the physical device.

This is a problem mainly with regard to devices with moving parts, such as widely used hard disk drive. When working with IO operations on this type of device, the fragmentation cause larger seek times, which is a time for magnetic heads to reach desired location, therefore slowing down the file system [1].

Above is a great example of how file system has to be flexible to solve such problems as well as of problems that emerge after running file system for a longer period of time.

The ideal approach for research pointed on effect on age of file system would be to put file system under defined conditions for a few years or months, gathering information in the process.

However, this is apparently impractical or even impossible, because the demand has to be satisfied in matter of weeks to respond to new versions of file systems, which are developed and released quite often.

One of the earliest ideas about how to deal with this problem would be to capture a snapshot of a file system that had already been used for a prolonged time.

This approach of testing on such a snapshot, however, would probably lead to optimising for a very specific instance of old version of given file system.

If researchers want to predict behavior of a new version of file system after prolonged time of usage, they should be able to create a simulation, which would mimic the aging process, but in short period of time.

This can be technically implemented, because the idea is to run the simulation continuously under very heavy workload, which would not have any delays. We believe, that file systems used in real life are not really used continuosly, therefore researchers should be able to compress the time of creating such an aged file system in matter of days.

Unfortunately the demand from users and developers has yet not been met by standard studies. There has been few studies executed in last millenium [2], that shed some light on this topic, but overall, modern research is in hands of regular users, which remain in amateur sphere, lacking professional equipment to execute such research properly.

My general idea is to design a set of heavy workloads, simulating aging process, to run on different types of devices and at least two types of file systems and then to execute a series of performance tests. Results of these tests could be then compared to fresh file systems, showing performance differences between them. Comparing different aged file systems between each other would clearly be an interesting topic of research as well.

Bibliography

[1] G. Nijs, A. Biesheuvel, A. Denissen, and N. Lambert. The Effects of Filesystem Fragmentation. In *Proceedings of Linux Symposium, Volume One*, Ottawa, Ontario, Canada, July 2006.

[2] Sigmetrics Conference. *File System Aging*, Seattle, WA, June 1997.

[3] V. Tarasov, S. Bhanage, E. Zadok, and M. Seltzer. Benchmarking File System Benchmarking: It *IS* Rocket Science. In *Proceedings of HotOS XIII:The 13th USENIX Workshop on Hot Topics in Operating Systems*, Napa, CA, May 2011.

[4] A. Traeger, N. Joukov, C. P. Wright, and E. Zadok. A Nine Year Study of File System and Storage Benchmarking. *ACM Transactions on Storage (TOS)*, 4(2):25–80, May 2008.