

File System Benchmark

Youxu Chen

Pre-conclusion

- Macro Benchmarks

- Aim to simulate real-world workload
- Postmark
- Various compile
- Andrew
- TPC
- SPEC
- SPC
- NetNews
- ...

- Micro Benchmarks

- usually consist of a small number of types of operations and serve to highlight some specific aspect of the file system
- Bonnie and Bonnie++
- Sprite LFS
- System Utilities

- Configurable benchmarks

- Iometer
- Filebench

Source: Traeger A, Zadok E, Joukov N, et al. A nine year study of file system and storage benchmarking[J]. ACM Transactions on Storage (TOS), 2008, 4(2): 5.

Macro benchmarks

Postmark'1997

- Single-thread synthetic benchmark
 - Workload
 - many short-lived, relatively small files
 - Map to real-world applications
 - electronic mail, NetNews, and Web-based commerce transactions as seen by ISP
 - a mix of data- and metadata-intensive operations
 - Flows
 - Create a pool of random text files
 - Transactions(consists of creatation,deletion,read...)
 - Drawback
 - Hard to scale
- FileMark(multi-threads, gettimeofday)**

Compile Benchmarks

- Compiled what?
 - SSH, OS kernel, Am-utils, Emacs, Apache, other packages
- *CPU-intensive*

NetNews'1996

- It is a shell script that performs a small-file workload.

Andrew File System Benchmark'1988

- The Andrew benchmark has the five phases next presented.
 - (1) *MakeDir*: constructs directories in a target subtree identical to the structure of the original subtree.
 - (2) *Copy*: copies all files from source subtree to target subtree.
 - (3) *ScanDir*: performs a stat operation on each file of the target subtree.
 - (4) *ReadAll*: reads every byte of every file in the target subtree once.
 - (5) *Make*: compiles and links all files in the target subtree
- Drawback
 - Hard to scale

TPC'2005

- Transaction Processing Performance Council
- Used in database systems
 - TPC-App, TPC-C, TPC-E, TPC-H
- TPC –C
 - is replaced by TPC-E, which is a data-intensive benchmark, portraying the activity of a wholesale supplier where a population of users executes transactions against a database.
- TPC-H
 - The workload for this benchmark consists of executing ad hoc queries against a database and performing concurrent data modifications

They utilize a database system, which introduce extra complexity.

SPEC'2005

- Standard Performance Evaluation Corporation
- *SFS*
 - To NFS v2 and v3, measure throughput and response time
 - SFS is “unlikely to be extended to support NFSv4,” and that *FileBench* will probably be used instead
- *SDM'1991*
 - simulates a software development environment with a large number of users
 - contains two subbenchmarks, o57.SDET and o61.Kenbus1
 - randomly ordered script with shell, like make, cp, grep, man...
- *SPECweb2005*
 - is used for evaluating the performance of Web servers.
 - GET and Post ops

Other Benchmarks

- NetBench and Dbench'2000
 - Measure the performance of file servers
- Drwabacks
 - NetBench-need 6—150 PC running windows
 - Dbench is close to NetBench

Replaying traces

- Files must be created before accessed
- [SNIA](#) traces

Micro benchmarks

Bonnie and Bonnie++'2000

- The working file size is larger than the amount of memory to ensure not all read requests are satisfied from the page cache
 - 4x memory size recommended
- Drawbacks
 - Does not use a pseudorandom number, so variance is introduced with different Oss and the results may not be comparable
 - the options are not parameterized, only file size is configurable
 - reading and writing one character at-a-time tests the library call throughput more than the file system because the function that Bonnie calls (getc) uses buffering
- Bonnie++
 - Multiple files and new test(create, stat, unlink), but same drawbacks above

Sprite LFS'1992

- *Large-File benchmark*

- (1) creates a 100MB file using sequential writes
- (2) reads the file sequentially
- (3) writes 100MB randomly to the existing file
- (4) reads 100MB randomly from the file
- (5) reads the file sequentially

- *Small-File benchmark*

- (1) creating 10,000 1KB files by creating and opening a file, writing 1KB of data, and closing the file
- (2) reading the files
- (3) deleting the files

Drawbacks

- Fixed file size → hard to scale
- Not pseudorandom
- Cache is not cleaned up

System utilities

- wc, cp, grep, diff, tar, gzip...
- Hard to scale and its version is important with comparable

Configurable benchmarks

Samples

- *Iometer'2001*
 - to emulate the disk or network I/O load with synthetic IO loads
- *Buttress'2004*
 - to issue I/O requests with a high accuracy, even when high throughputs are requested
- *FileBench'2005*
 - Macro-NFS mail(~Postmark), fileserver(~SPEC SFS), database server, web server and web proxy
 - Micro-more and flexible
- *Fstress'2002*
 - For NFS v3 and requests are sent at a steady rate