

Filebench

Spencer Shepler

Eric Kustarz

Andrew Wilson

[Richard McDougall]

Filebench Discussion

- Filebench motivation
- Filebench description
- Preparation for inclusion in OpenSolaris
- Issues
- What next?

Testing filesystem performance

- Dd
- Tar
- mkfile
- Bonnie
- Iozone
- And on and on...
 - fsstress, ffsb, fsrandom, mongo, iometer

Why invest in a File System Perf Framework?

- Need complete test coverage for file level applications
 - Current coverage is mostly micro benchmarks:
 - bonnie, iozone, mongo
 - Coverage was very limited (less than 10% of important application cases covered)
 - Current approach is to use benchmark full application suites: e.g. Oracle using TPC-C: expensive, labor intensive
 - Up to 100 different benchmarks are required to accurately report on filesystem performance today
- For NFS use, SPECsfs is limited to NFS Version 3
 - NFSv3 workload only represents “home directory servers”
 - upcoming addition of CIFS will be limited in workload

Model based methodology study

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Filebench Architecture

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Model Allows Complex/Important Scaling Curves

- For example:
 - Throughput/latency vs. working set size
 - Throughput/latency vs. # of users
 - CPU efficiency vs. throughput
 - Caching efficiency vs. working set size/memsize

Flow States: Open Ended Flow

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Characterize and Simulate via Cascades of Workload Flows:

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Flow States: Synchronized Flow

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Examples of Per-flow Operations

- Types
 - Read
 - Write
 - Create
 - Delete
 - Append
 - Getattr
 - Setattr
 - Readdir
 - Semaphore block/post
 - Rate limit
 - Throughput limit
- Attributes
 - Sync_data
 - Sync_metadata
 - IO Size
 - IO Pattern, probabilities
 - Working set size
 - Etc.

Simple Random I/O Workload Description

```
define file name=bigfile0,path=$dir,size=$filesize,prealloc,reuse,paralloc

define process name=rand-read,instances=1
{
  thread name=rand-thread,memsize=5m,instances=$nthreads
  {
    flowop read name=rand-read1,filename=bigfile0,iosize=$iosize,random
    flowop eventlimit name=rand-rate
  }
}
```



Files and Filesets

- Files: a definition of a single file
 - Will soon be folded into filesets
- Filesets: a definition of a set of files
 - A fractal tree of files
 - A fileset has a depth and size, width of directories is computed from these
 - Can also have a depth of 1 to make one large directory
 - Can have uniform sizes, depths, widths or configured as a [gamma] distribution
 - Filesets that mimic file servers typically use gamma distribution for size and depth

NFS Client testing: POSIX level workload + NFS Server

QuickTime™ and a
BMP decompressor
are needed to see this picture.



Running a single Filebench workload

Example varmail run:

```
filebench> load varmail
```

```
Varmail personality successfully loaded
```

```
Usage: set $dir=<dir>
```

```
set $filesize=<size> defaults to 16384
```

```
set $nfiles=<value> defaults to 1000
```

```
set $dirwidth=<value> defaults to 20
```

```
set $nthreads=<value> defaults to 1
```

```
set $meaniosize=<value> defaults to 16384
```

```
run <runtime>
```

```
filebench> set $dir=/tmp
```

```
filebench> run 10
```

```
Fileset mailset: 1000 files, avg dir = 20, avg depth = 2.3, mbytes=15
```

```
Preallocated fileset mailset in 1 seconds
```

```
Starting 1 filereader instances
```

```
Starting 1 filereaderthread threads
```

```
Running for 10 seconds...
```

```
IO Summary: 21272 iops 2126.0 iops/s, (1063/1063 r/w) 32.1mb/s, 338us cpu/op, 0.3ms latency
```

The steps behind the “run” command

QuickTime™ and a
BMP decompressor
are needed to see this picture.



Listing available workloads

```
filebench> load
bringover
copyfiles
createfiles
deletefiles
filemicro_create
filemicro_createfiles
filemicro_createrand
filemicro_delete
filemicro_rread
filemicro_rwrite
filemicro_rwritedsync
filemicro_rwritefsync
filemicro_seqread
filemicro_seqwrite
filemicro_seqwriterand
filemicro_writefsync
fileserver
mongo
multistreamread
multistreamreaddirect
multistreamwrite
multistreamwritedirect
oltp
randomread
randomrw
randomwrite
singlestreamread
singlestreamreaddirect
singlestreamwrite
singlestreamwritedirect
tpcso
varmail
webproxy
webserver
```

Database Emulation Overview

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Database Emulation Process Tree

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Filebench pre-defined workloads

- “File Macro”
 - Small database
 - Large database
 - Multi-threaded web server
 - Multi-threaded proxy server
 - Home directory server
 - NFS mail server
 - DB Mail server
 - Video server
- “File Micro”
 - Sequential read/write
 - Multistream read/write
 - Allocating writes
 - Reallocating writes
 - Random read/write
 - MT random read/write
 - File create/delete
 - File meta-data ops
 - I/O types: O_DSYNC, etc.
 - Directory size scaling

Preparation for OpenSolaris

- Code cleanup
 - cstyle and lint clean
 - Remove unused code
 - Lots of additional comments
 - Full 64bit version for amd64
 - Additional error path handling
 - Linux cleanup
- Versioning of filebench and workloads
- filebench command now the perl “wrapper”
- go_filebench is the “real” executable
- go_filebench now offers command completion

Filebench in the wild

- NFS protocol verification
 - <http://nasconf.com/pres05/kustarz.pdf>
- Local filesystem improvement
 - http://blogs.sun.com/erickustarz/entry/vdev_cache_improvements_to_help
- SATA/IO layer analysis
 - http://blogs.sun.com/erickustarz/entry/ncq_performance_analysis
- (SCSI) protocol analysis
 - Non-volatile cache flushing with varmail workload

Filebench issues

- Filebench can not deal with “no work to do”
 - Work around is to increase “nfiles”
- No method to tell workloads “run to completion”
 - This is for varmail/postmark type workloads
 - Potential solution may be to set “runtime” to 0
- Linux / OSX / *BSD support
 - Mostly works
 - Need additional use and verification
- Ease of use
 - .prof syntax is very picky
 - Error messages should provide pointers to solutions
- Integration with multi-client framework

Future: NFS plugin

QuickTime™ and a
BMP decompressor
are needed to see this picture.

Can filebench replace SFS?

- Things to solve before it is viable
 - Multi-client support
 - Plugins for NFSv3, NFSv4, CIFS
 - Verification of filebench code
 - Cross platform support/verification
- SFS workloads become filebench workloads

Documentation / Discussion

- <http://sourceforge.net/projects/filebench/>
- <http://opensolaris.org/os/community/performance/>
- <http://www.solarisinternals.com/wiki/index.php/FileBench>
- http://www.solarisinternals.com/wiki/index.php/Filebench_for_Programmers
- http://www.solarisinternals.com/wiki/index.php/FileBench_Workload_Language

Filebench

Spencer Shepler
(blogs.sun.com/shepler)

Eric Kustarz
(blogs.sun.com/erickustarz)