

The New and Improved

FileBench

File System Benchmarking Framework

Eric Kustarz

Spencer Shepler

Andrew Wilson

Presented on 2/27/08 by:

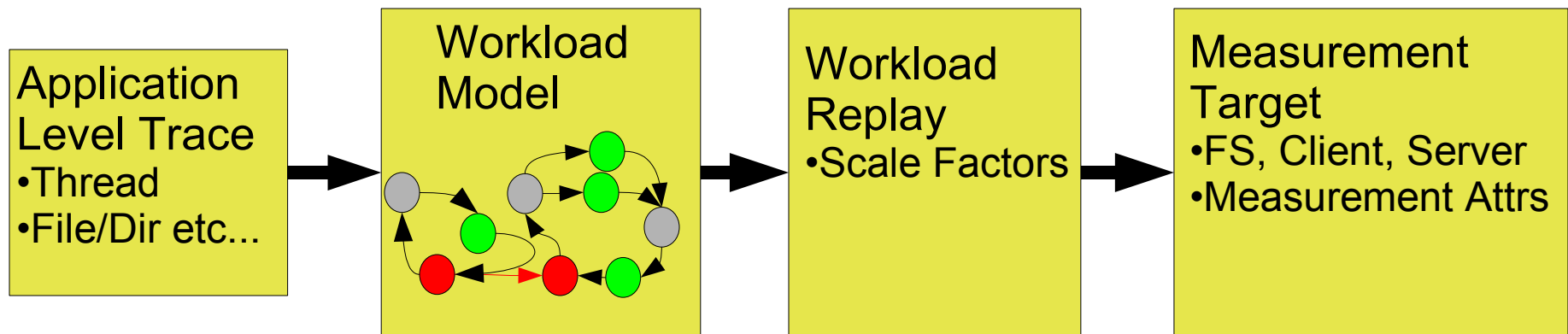
Andrew Wilson

Senior Software Engineer

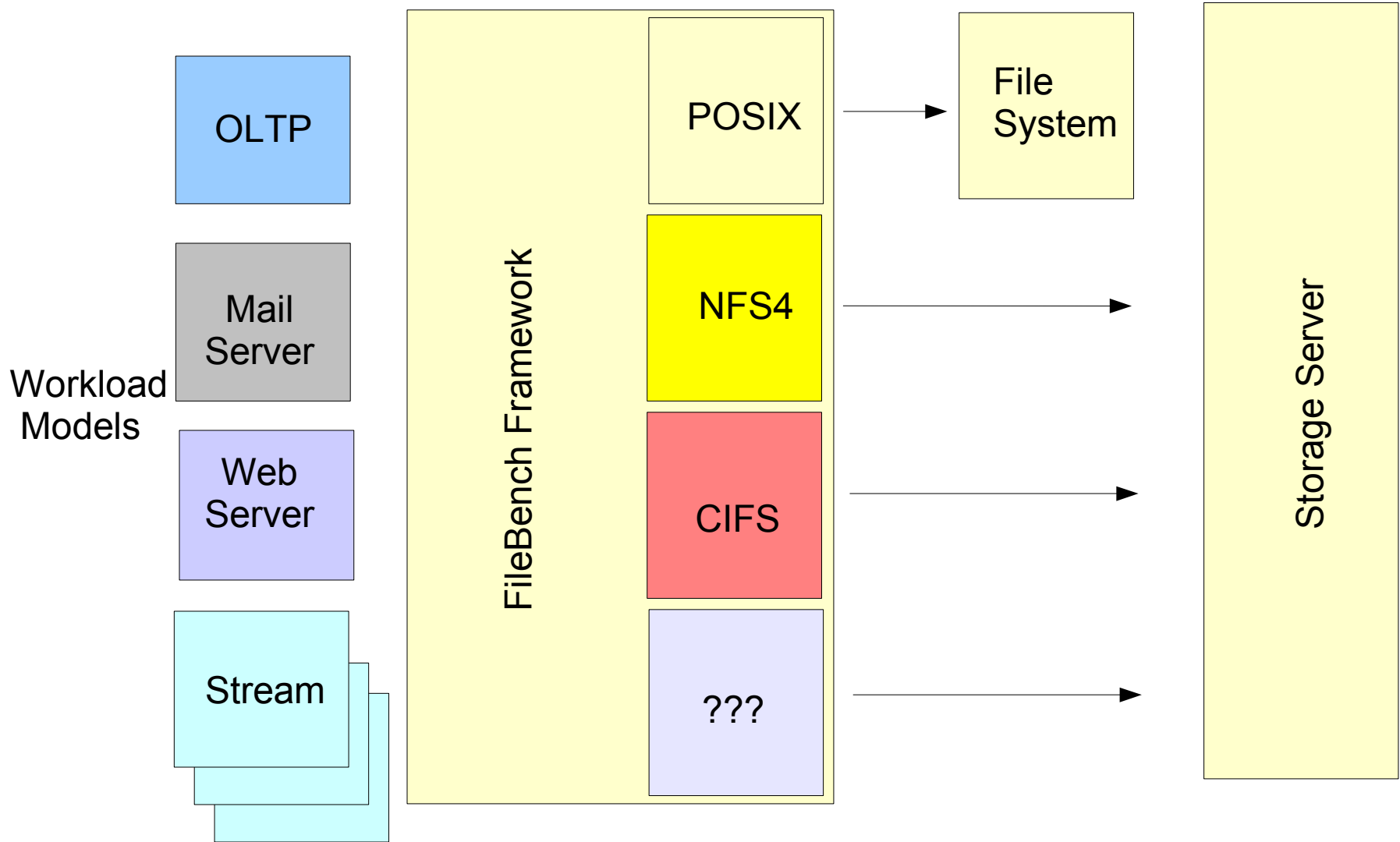
Sun Microsystems

Motivation for FileBench

- Need complete test coverage for file level applications
 - Current coverage is mostly micro benchmarks: bonnie, iohome ...
 - Existing Macro benchmarks only cover a few important application cases or use full application suites which is expensive (e.g. TPC-C).
 - Up to 100 different benchmarks are required to accurately report on filesystem performance today
- What is needed is a configurable, model based workload framework:



FileBench Architecture



FileBench Features

- Distribution includes many pre-defined workloads
 - Highly configurable at runtime through workload defined *Variables*
 - Explicit support for *Threads* and *Processes*
 - Groups files into *Filesets* which are a fractal tree of files
 - Behavior defined by lists of per-flow operations called *Flowops*
 - All of the above accept *Attributes* to customize behavior
-
- | | |
|---|--|
| <ul style="list-style-type: none"> • Example Per-flow Operations: <ul style="list-style-type: none"> – Read – Write – Create – Delete – Append – Semaphore block/post – Rate limit | <ul style="list-style-type: none"> • Example Attributes <ul style="list-style-type: none"> – Sync_data – Sync_metadata – IO Size – IO Pattern, probabilities – Working set size – Etc. |
|---|--|

Simple Random I/O Workload

Workload Model File: randomread.f:

```
set $dir=/tmp
set $nthreads=1
set $iosize=8k
set $filesize=1m

define file name=largefile1,path=$dir,size=$filesize,prealloc,reuse,paralloc

define process name=rand-read,instances=1
{
  thread name=rand-thread,memsize=5m,instances=$nthreads
  {
    flowop read name=rand-read1,filename=largefile1,iosize=$iosize,random
    flowop eventlimit name=rand-rate
  }
}
```

Results:

```
4599: 83.637: Per-Operation Breakdown
rand-rate          0ops/s    0.0mb/s    0.0ms/op    0us/op-cpu
rand-read1        329ops/s   0.6mb/s    3.0ms/op    48us/op-cpu

4599: 83.637:
IO Summary:      19891 ops 329.2 ops/s, (329/0 r/w)  0.6mb/s,  3108us cpu/op,  3.0ms latency
```

Recent Modifications

- Code cleanup
 - cstyle and lint clean
 - Remove unused code
 - Lots of additional comments
 - Full 64bit version for amd64
 - Additional error path handling
 - Linux cleanup
- Integrated into OpenSolaris and included with SDXE
- Sources at SourceForge.net and OpenSolaris.org
- Versioning of filebench and workloads
- filebench command now the perl “wrapper”
- go_filebench is the “real” executable
- go_filebench now offers command completion

FileBench Features in development

- Random Variables
- Composite Flowops
- NFS / CIFS plug-ins
- Multi-client Framework

FileBench is Open Source!

Community Contributions are Encouraged

FileBench Information

- <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
- http://www.solarisinternals.com/wiki/index.php/FileBench_Workload_Modeling_Guide

Eric Kustarz (blogs.sun.com/erickustarz)

Spencer Shepler (blogs.sun.com/shepler)

Andrew Wilson (Drew.Wilson@sun.com)