FileBench

A Prototype Model Based Workload for File Systems

Work In Progress Report – 4/1/2004

Richard McDougall

Glenn Colaco

Sun Microsystems

Sun microsystems

We make the net work.

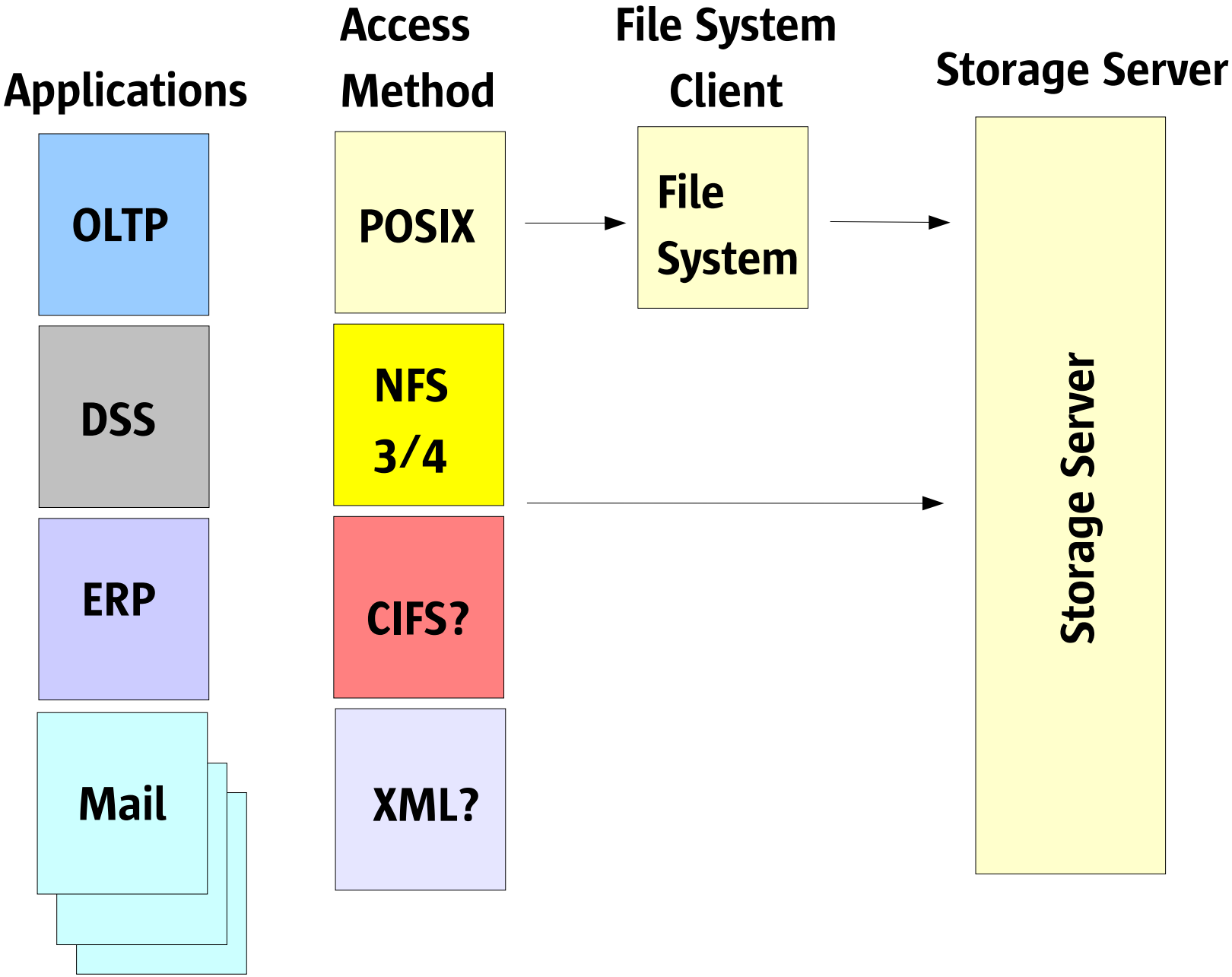
Benchmarks?

- For Vendors
  - Product characterization
  - Product design goaling
  - Benchmarketing
- For Customers
  - Purchasing Guide
  - Configuration characterization/tuning/verification

Requirements for file-level benchmarking

- Represent Apps rather than I/Os
- Trace-derived synthesis
- Thread-level representation
- Inter-thread dependency/sync.
- Forward Path
- Extensible to new protocols
- Modular to include test of client: process/thread model, cpu efficiency etc...
- Pre-structuring/aging of file sets
- Scalable
  - Throughput, #Users
  - #Files/Directories
  - Working set size
  - #Clients
  - Client resources (mem/cpu)

What do we want to characterize?



Characterization Strategies

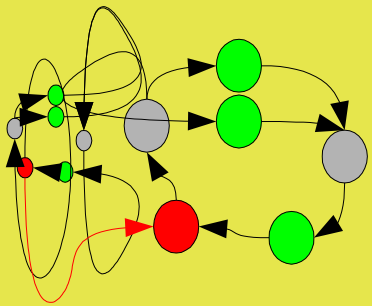
- I/O Microbenchmarking
  - Pros: Easy to run
  - Cons: Small test coverage, Hard to correlate to real apps
- Trace Capture/Replay
  - I/O Trace, NFS Trace, Application Trace
  - Pros: Accurate reconstruction of real application I/O mix
  - Cons: Large traces, difficult to reconstruct I/O dependencies
- Model Based
  - Distillation of trace into representive model
  - Probability based, Simulation based
  - Pros: Easy to run, Scalable in multiple dimensions
  - Cons: Care required to ensure accurate real-world representation

Model based methodology study

Application Level Trace

- Thread
- File/Dir
- Attrs etc...

Workload Model



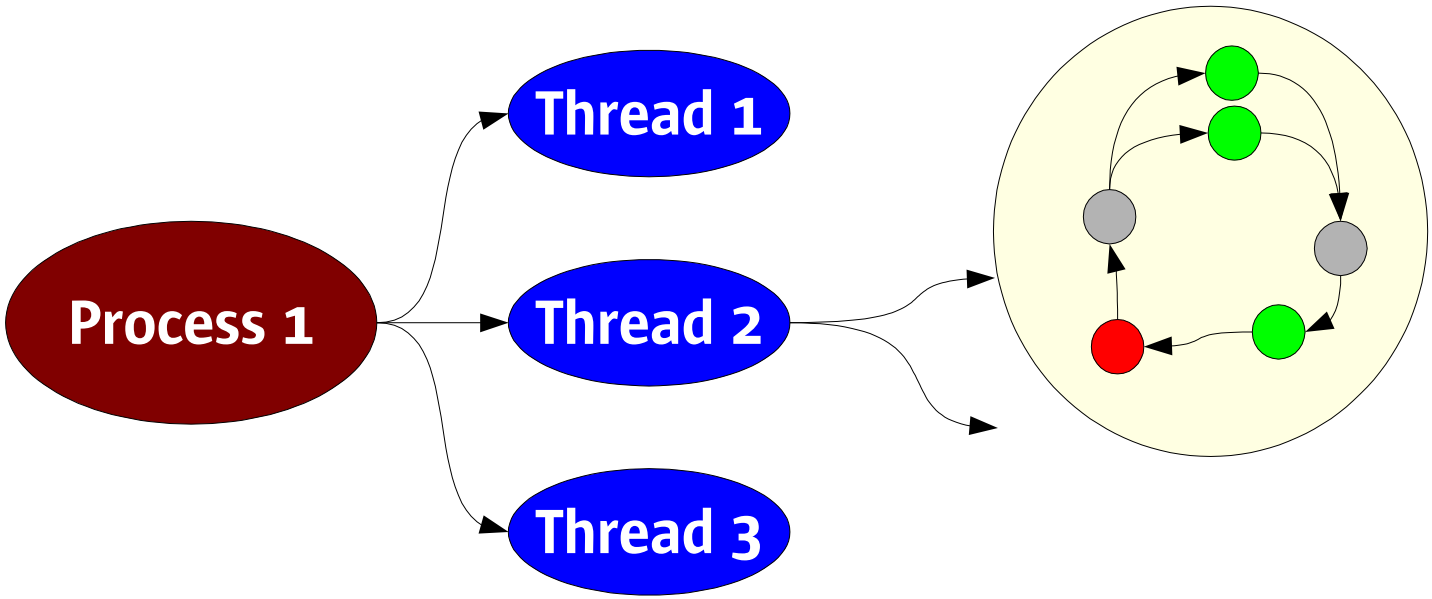
Workload Replay

- Scale Factors

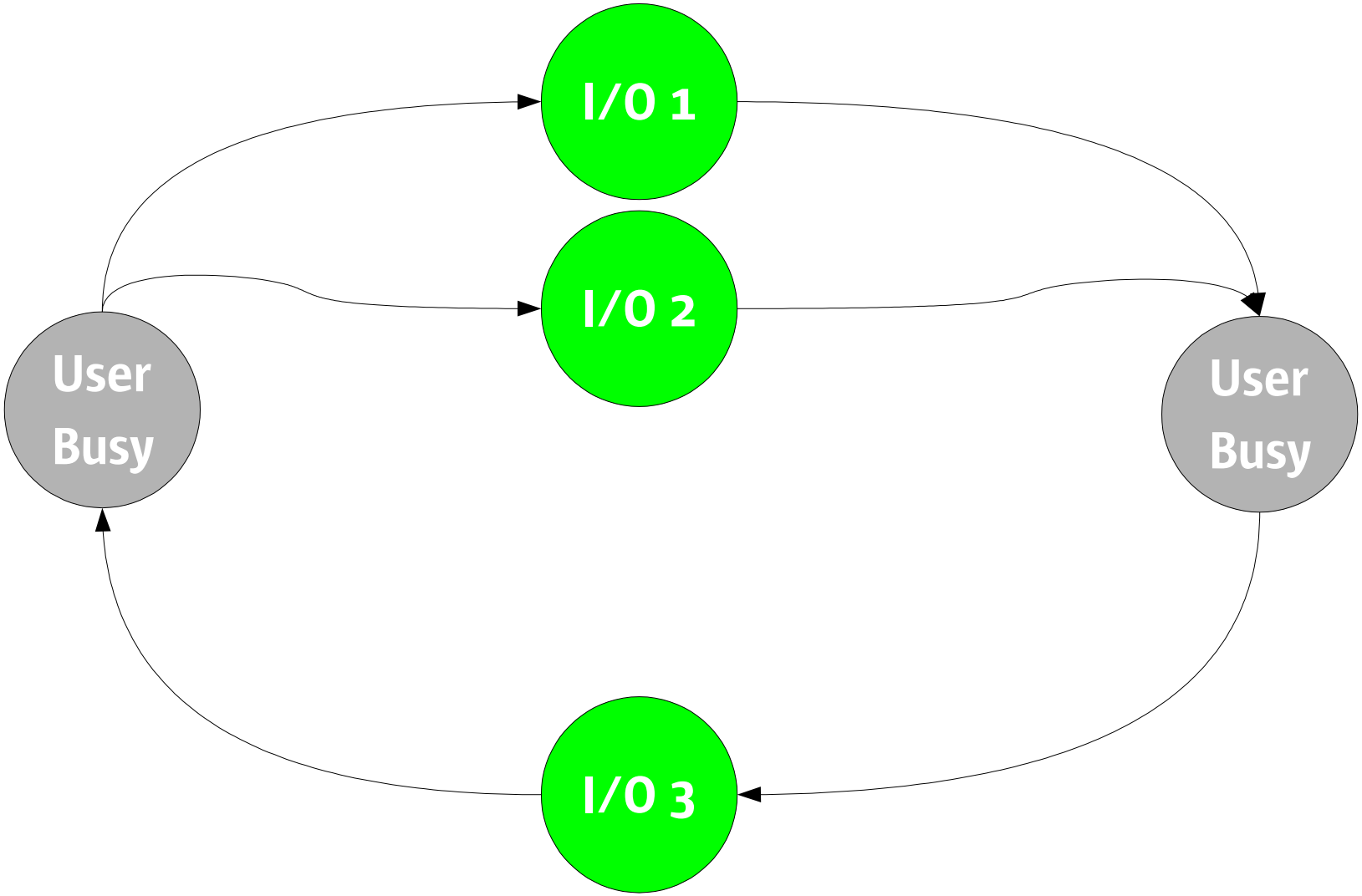
Measurement Target

- FS, Client, Server etc
- Measurement Attrs

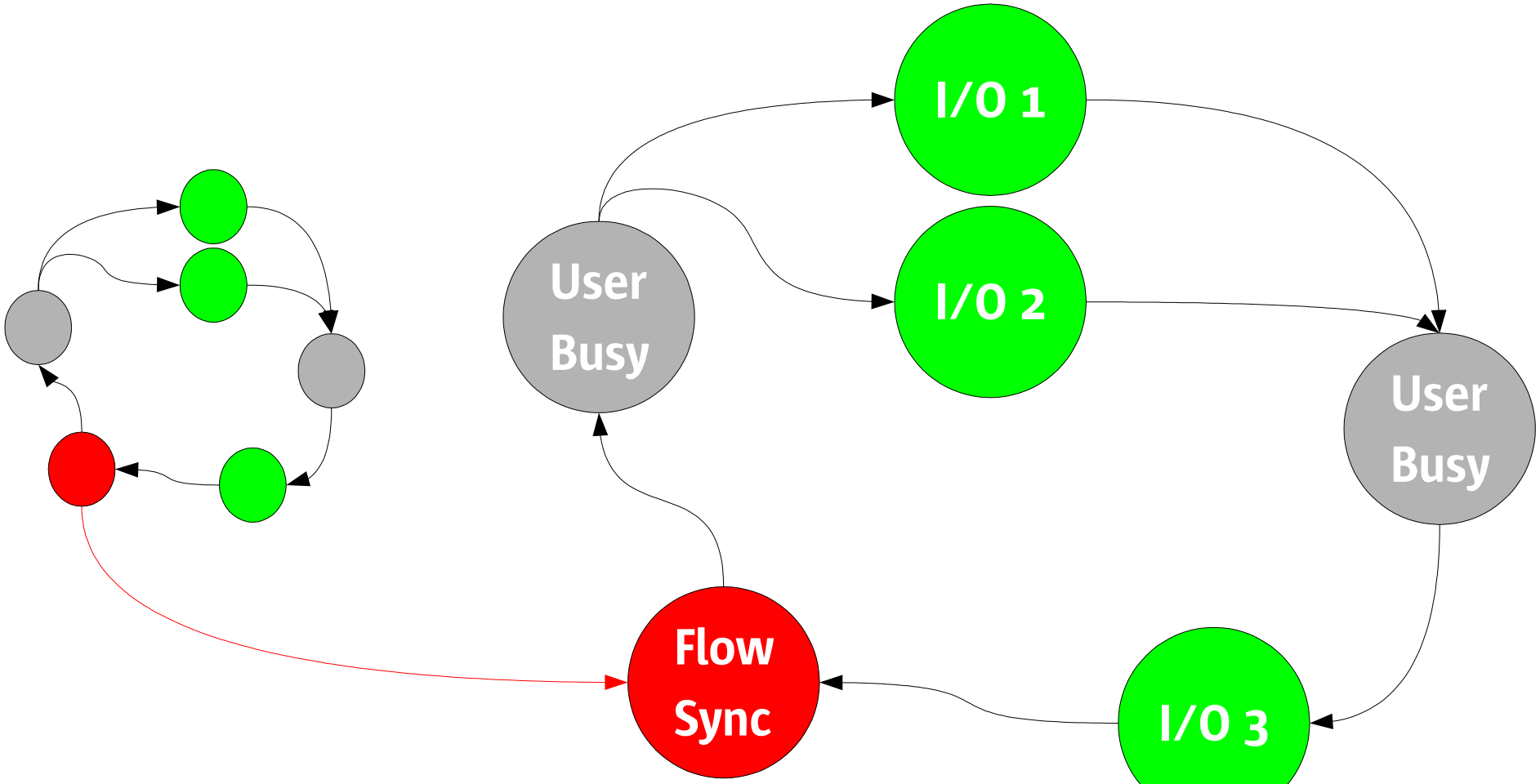
Characterize and Simulate via Cascades of Workload Flows:



Flow States: Open Ended Flow



Flow States: Synchronized Flow



Flow blocks until completion of other flow

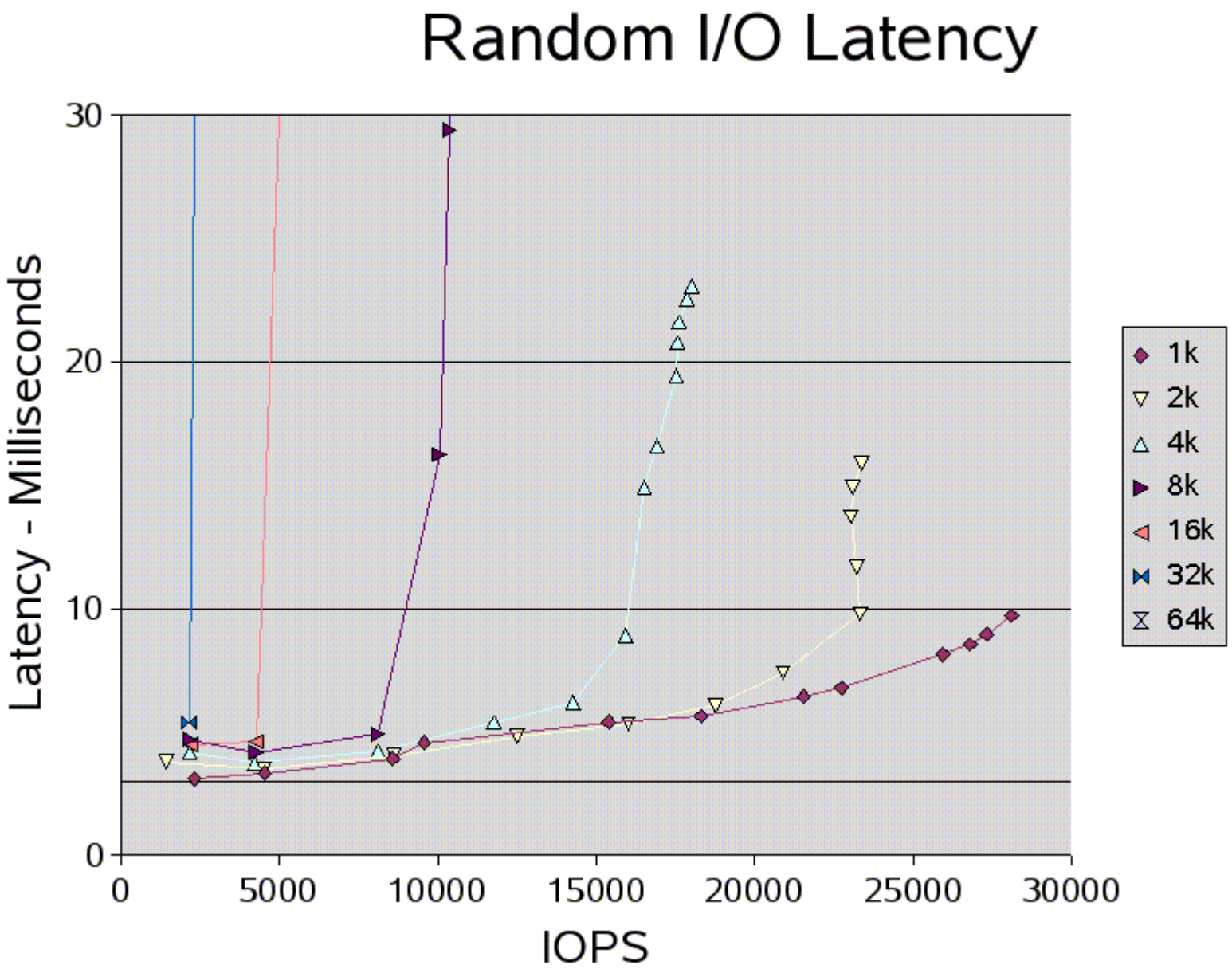
Examples of Per-flow Operations

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

Simple Random I/O Program

```
#!./filebench -f

define file bigfile path=/tpcc.ntap.new/fs_log2/mybigfile,size=10g,prealloc,reuse
define process random-read instances=100

define thread random-thread procname=random-read,memsize=10m,instances=1
{
    define flowop random-read type=read,filename=bigfile,
                        random,dsync,iosize=$io_size
    define flowop random-rate type=flowrate
}

create files
set $sleeptime=120

foreach $io_size in 512,1k,2k,4k,8k,16k,32k,64k,128k,256k,512k,1m
{
    foreach $rate in 200,400,600,800,1000,2000,3000,4000,5000,
                6000,7000,8000,10000,12000,14000,16000,18000,20000,22000
    {
        create processes
        stats clear
        sleep $sleeptime
        stats snap
        log $io_size,$rate,$stats.iorate,$stats.iobandwidth,
            $stats.iolatency,$stats.iocpu,$stats.iocpusys
    shutdown processes
    }
}
```

Random I/O – NFS V3



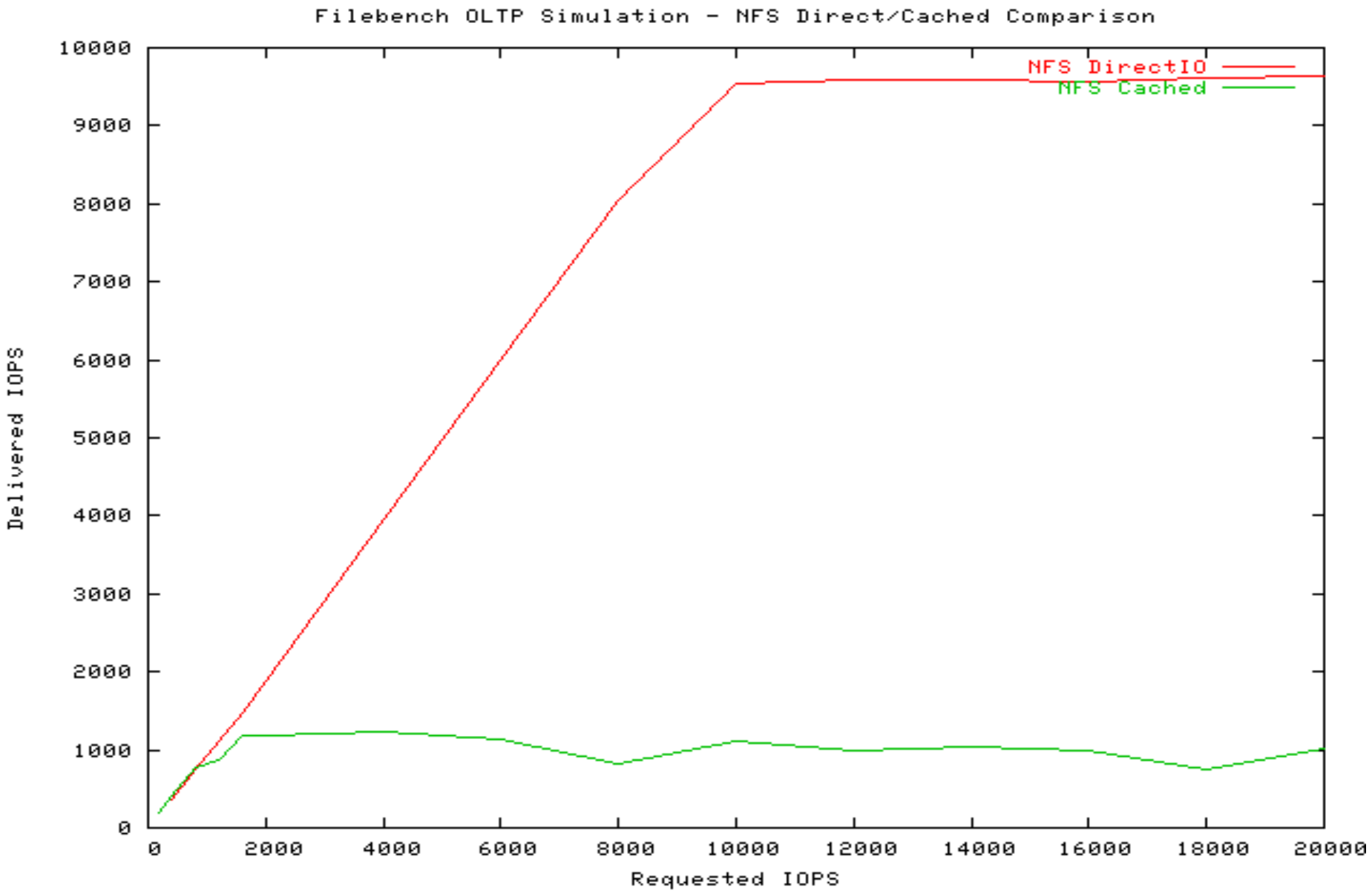
Simplified OLTP Database Program

```
define file log path=/bigfs/log.dbf,size=100m,prealloc,reuse
define file datafile path=/bigfs/datafile.dbf,size=1g,prealloc,reuse
create files

define process logwr
define process dbwr instances=10
define process shadow instances=20

define thread logwr procname=logwr,memsize=10m
{
  define flowop log-aiowrite inherit=aiowrite,filename=log iosize=1m,workingset=1m,dsync,iters=10
  define flowop log-aiowait inherit=aiowait
  define flowop log-block inherit=semblock,value=40,highwater=100
  define flowop log-delay inherit=delay,value=1
}
define thread dbwr procname=dbwr,memsize=10m
{
  define flowop dbaiowrite-a
      inherit=aiowrite,filename=datafile,iosize=8192,workingset=1g,random,dsync,iters=10
  define flowop dbwr-aiowait inherit=aiowait
  define flowop dbwr-hog inherit=hog,value=10000
  define flowop dbwr-block inherit=semblock,value=10,highwater=1000
}
define thread shadow procname=shadow,memsize=10m
{
  define flowop shadowread-a
      inherit=read,filename=datafile,iosize=2048,workingset=10m,random,dsync
  define flowop shadow-post-log inherit=sempost,value=1,target=log-block,blocking
}
create processes
sleep 30
```

OLTP Program – Sample Run

```
106297: 109.515: Flowop totals:
106297: 109.515: flowop shadow-post-log     3022 ops,    100.7, ops/s   0.0mb/s  172.052ms/op    0.019s-cpu          6us/op-cpu
106297: 109.516: flowop shadowread-a        3042 ops,    101.4, ops/s   0.2mb/s    8.340ms/op    0.072s-cpu         23us/op-cpu
106297: 109.516: flowop dbwr-block             0 ops,      0.0, ops/s   0.0mb/s    0.000ms/op    0.000s-cpu          0us/op-cpu
106297: 109.517: flowop dbwr-hog              10 ops,      0.3, ops/s   0.0mb/s    0.385ms/op    0.003s-cpu        301us/op-cpu
106297: 109.517: flowop dbwr-aiowait          10 ops,      0.3, ops/s   0.0mb/s  824.917ms/op    0.003s-cpu        300us/op-cpu
106297: 109.517: flowop dbaiowrite-a         100 ops,      3.3, ops/s   0.0mb/s    2.768ms/op    0.077s-cpu        771us/op-cpu
106297: 109.518: flowop log-delay             16 ops,      0.5, ops/s   0.0mb/s 1003.555ms/op    0.000s-cpu         13us/op-cpu
106297: 109.518: flowop log-block             16 ops,      0.5, ops/s   0.0mb/s    0.094ms/op    0.000s-cpu         13us/op-cpu
106297: 109.519: flowop log-aiowait           16 ops,      0.5, ops/s   0.0mb/s  765.619ms/op    0.002s-cpu        110us/op-cpu
106297: 109.519: flowop log-aiowrite         170 ops,      5.7, ops/s   5.7mb/s    0.223ms/op    0.014s-cpu         81us/op-cpu
106297: 109.520: Total   3312 iops 110.3 iops/s, 101.3r/s,   9.0w/s   5.9mb/s,    4.012ms/op
```

NFS OLTP – IOPS Scaling



Model Allows Complex/Important Scaling Curves

- e.g.
  - Thoughput/Latency vs. Working set size
  - Thoughput/Latency vs. #users
  - CPU Efficiency vs. Thoughput
  - Caching efficiency vs. Workingset size/Memsize

Workload Discussion

	File Access						
Workload	File Size	# files	#Streams	Sharing	I/O Mix	Seek Mode	Access type mmap/posix
Web Server	Small	Large	Large	Low	<5%	90% Random Read/10% Sequential Write	Both
Small DB	Large	Small	~100	High	50r/50w, 1% large sequential	99% Random	POSIX
Large DB	Large	Small	~1000	High	50r/50w, 1% large sequential	99% Random	POSIX
DB Mail Server	Large	Small	>1000	High	?		
NFS Mail Server	Moderate	Moderate	>10k	Low	?	Sequential	POSIX
HPTC	Huge	Small	Small	Low	50r/50w	Sequential	POSIX
SW Development	Small	Large	>1000	Low	5r/5w/90a	Sequential	POSIX
Video Streaming							

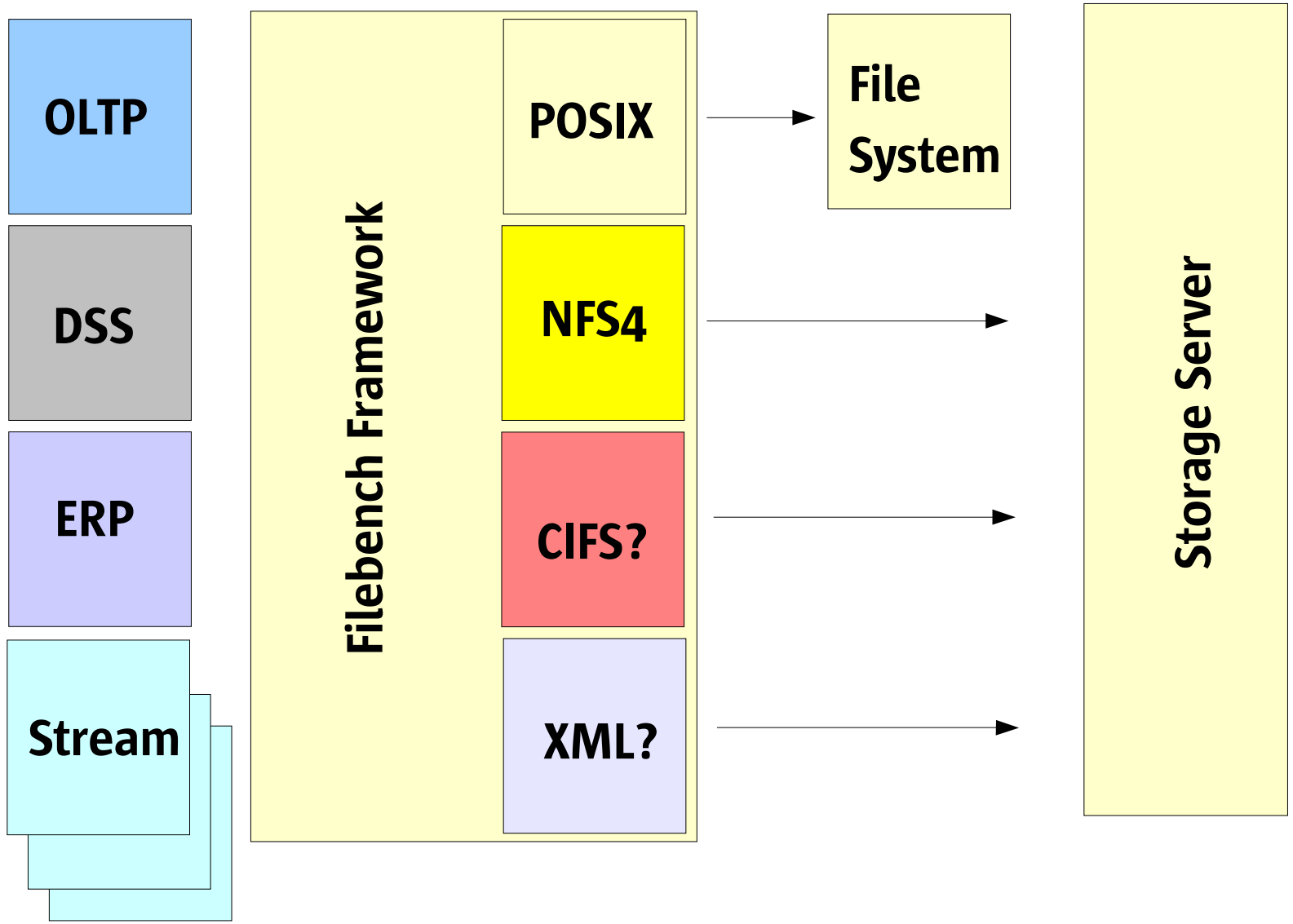
	I/O Characteristics					
Workload	App/IO CPU Content	Typical IOPS	Data Set Size	Working Set Size	Typical I/O Size	Typical Bandwidth
Web Server	99/1	<1000 per cient			<64k	<1MB/s
Small DB	90/10	~1000	1-10GB	50.00%	Random 2-8k, 128k sequential	~10MB/s
Large DB	80/20	>10000	10GB-1TB	30.00%	Random 2-8k, 128k sequential	50MB/s
DB Mail Server	90/10?				Small?	?
NFS Mail Server	90/10?	Low			Large reads, small writes	1-10MB/s
HPTC	80/20?	~1000?			~1MB	>100MBs Client, 1GB/s Server
SW Development	95/5?	~1000			~32k	~100mb/s

Example Composite

NAS Filer Workload	IOPS	BW	Weight
OLTP-Small	5123	40	30
OLTP-Large	2532	21	10
ERP	4928	25	10
Web Serving	3241	3.5	5
Data Warehouse	75	75	5
HPTC – Single Stream	89	89	10
HPTC – Multi Stream	120	120	5
Mail-DB	2132	2	5
Mail-NFS	781	50	5
SW Development	4123	10	10
Video Streaming	120	120	5

Score: 3028 IOPS @ 48MB/s

Filebench Achitecture



Call to action

- Define and standardize trace methodology
  - Application (Thread-level)
  - Wire-based (NFS, SMB)
  - Block-based (SCSI)
- Join the Interest group for fs-workloads
  - (Send email)

Comments?

r@sun.com

Sun Microsystems

Sun microsystems
We make the net work.