NFS BENCHMARKING USING NFSREPLAY

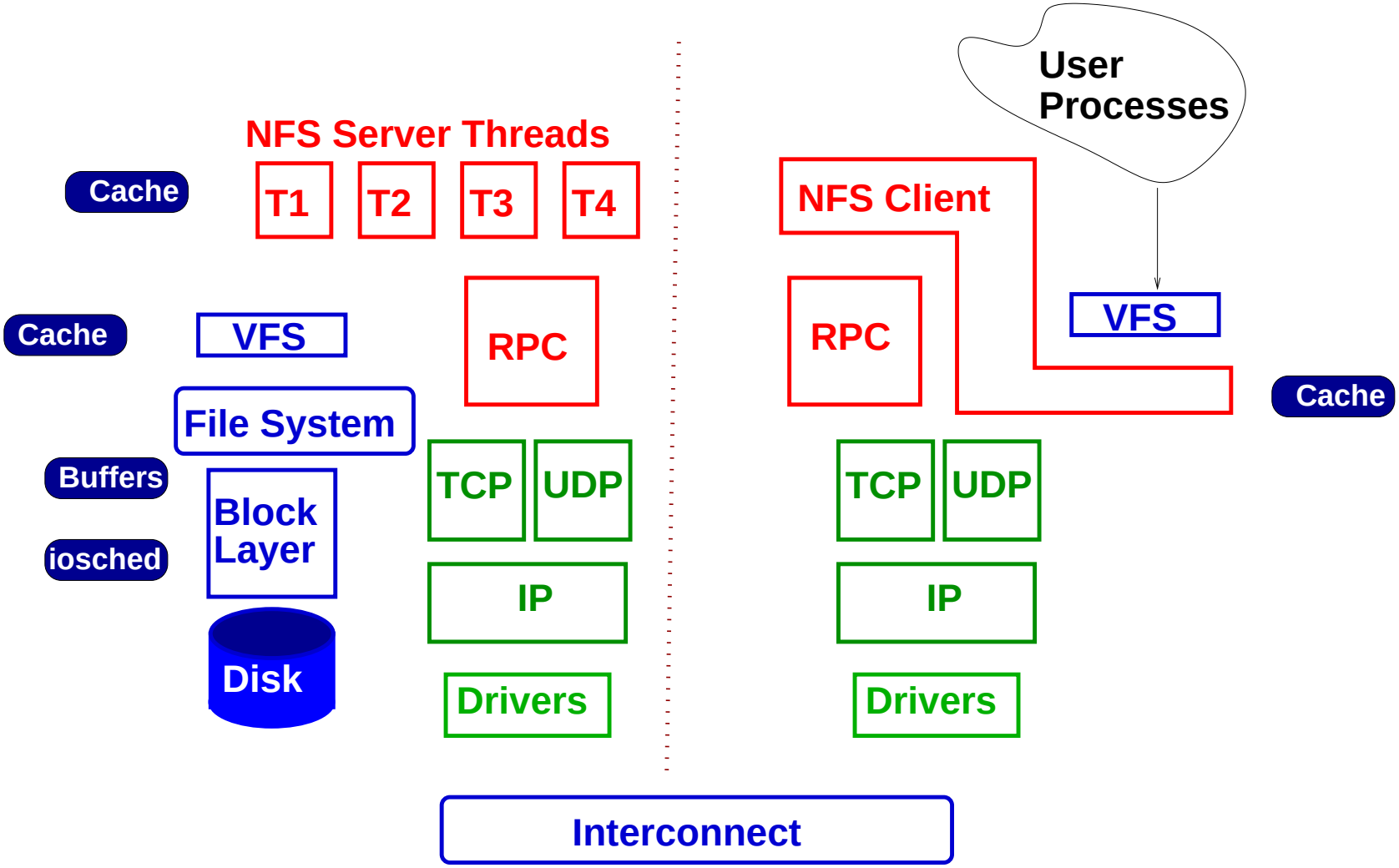
Shehjar Tikoo

Supported by UNSW and HP through the Gelato Federation and National ICT Australia

Overview

- Motivation
- Issues in NFS traffic replay
- nfsreplay
- Measurements

NFS Stack

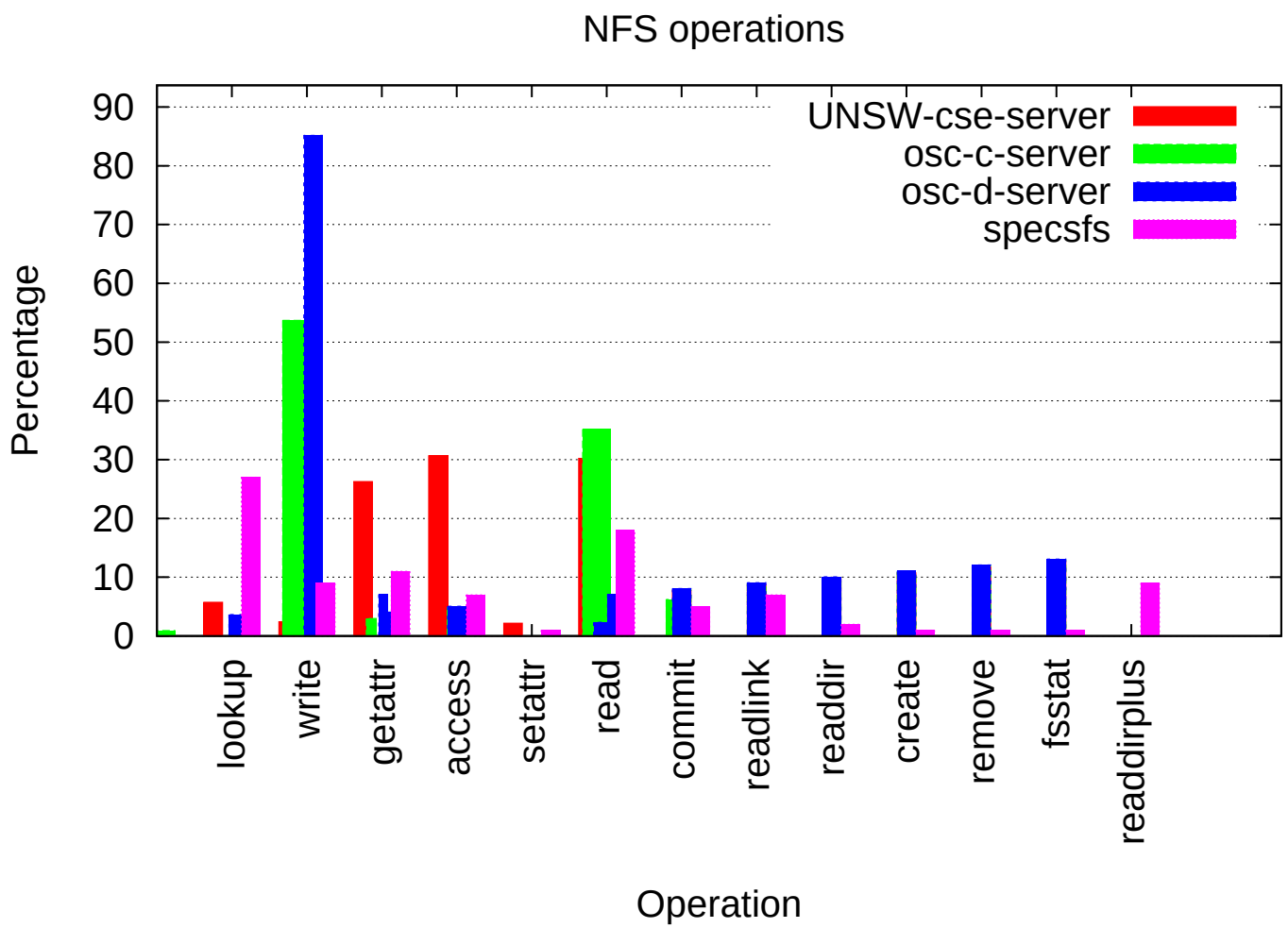

User Processes
NFS Server Threads
T1
T2
T3
T4
Cache
NFS Client
VFS
Cache
RPC
RPC
Cache
File System
Buffers
TCP
UDP
TCP
UDP
Block Layer
iosched
IP
IP
Disk
Drivers
Drivers
Interconnect

NFS Traffic Patterns

① Access pattern
- ➜ Percentage of requests of each type
- ➜ Buffers, caches, NFS read/write size
- ➜ Application pattern

② Timing pattern
- ➜ Inter-request times
- ➜ Response times
- ➜ Capacities, bandwidth

Percentage Distribution of Request Types



Motivation for nfsreplay

- ➜ Exploit maximum realism in network traffic
- ➜ Capture traces from network
- ➜ Replay the trace

Issues in NFS traffic replay

① Trace anonymization

② Trace processing

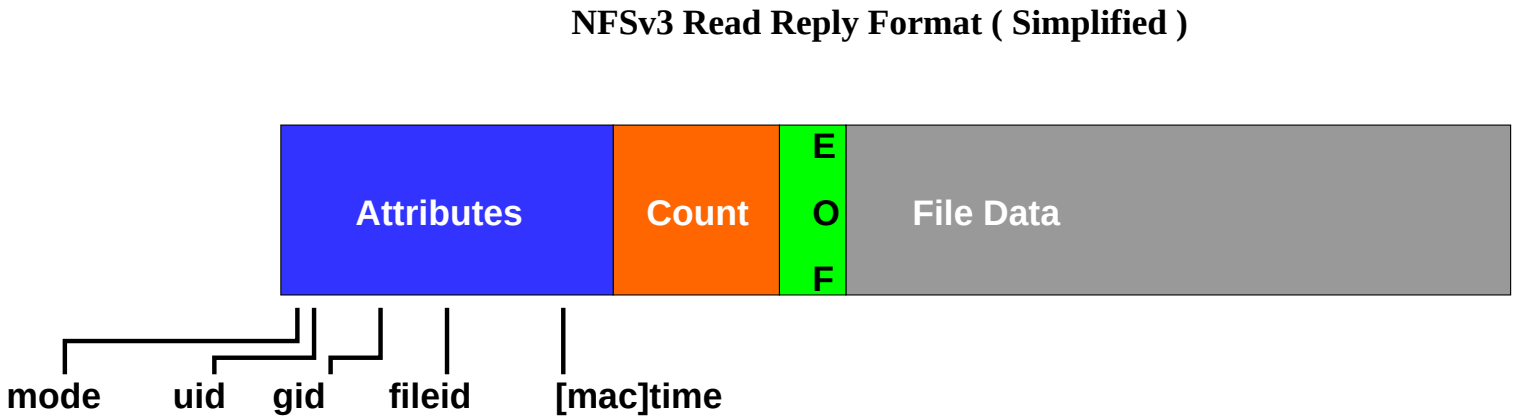
③ Filesystem hierarchy replication

④ Replay

TRACE ANONYMIZATION

- Traces contain file and user info/data - Must clean before releasing trace to public
- Solution: Anonymize
- IPs, uids, gid, file contents, file/dir names.
- Developed as an extension to TShark
  `www.gelato.unsw.edu.au/IA64wiki/NFSTrafficAnonymizer`

TRACE ANONYMIZATION EXAMPLE



- Anonymize NFSv3 Read Reply
- File contents get zeroed out

Nature of NFS Requests

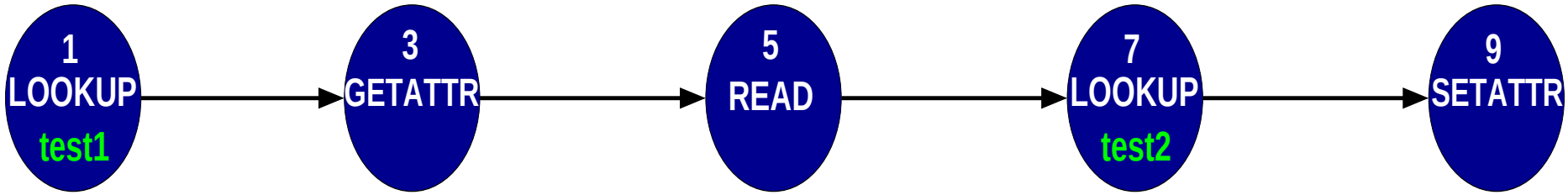
① Request order
   ➜ in time

② Request dependencies
   ➜ on file/dir handles

EXAMPLE TRACE

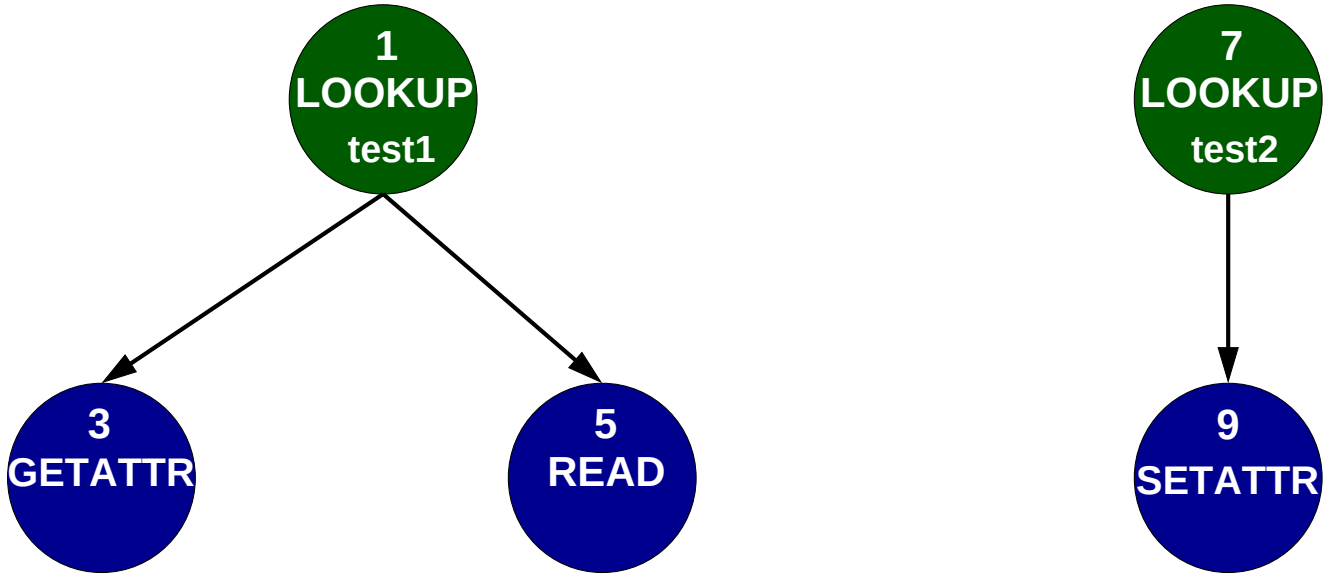
```
1 10.13.1.55 -> 10.13.0.10 NFS V3 LOOKUP Call, DH:0x24060004/test1
2 10.13.0.10 -> 10.13.1.55 NFS V3 LOOKUP Reply (Call In 7), FH:0x5e080908
3 10.13.1.55 -> 10.13.0.10 NFS V3 GETATTR Call, FH:0x5e080908
4 10.13.0.10 -> 10.13.1.55 NFS V3 GETATTR Reply (Call In 9) Regular File mode:0777
5 10.13.1.55 -> 10.13.0.10 NFS V3 READ Call, FH:0x5e080908 Offset:0 Len:1024
6 10.13.0.10 -> 10.13.1.55 NFS V3 READ Reply, (Call In 9) Len:1024
7 10.13.1.55 -> 10.13.0.10 NFS V3 LOOKUP Call, DH:0x24060004/test2
8 10.13.0.10 -> 10.13.1.55 NFS V3 LOOKUP Reply (Call In 7), FH:0x7e080908
9 10.13.1.55 -> 10.13.0.10 NFS V3 SETATTR Call, FH:0x7e080908
10 10.13.0.10 -> 10.13.1.55 NFS V3 SETATTR Reply (Call In 9)
```

Request Order



Simply, the order in which requests have been captured.

Request Dependency



① Based on filehandles

② Skip non-replayable requests

③ Towards low-overhead replay

Trace Processing

- ➜ Order and dependency needed for sustained load
- ➜ CPU Intensive!
- ➜ Established in a pre-replay phase

REPLICATING FS HIERARCHY

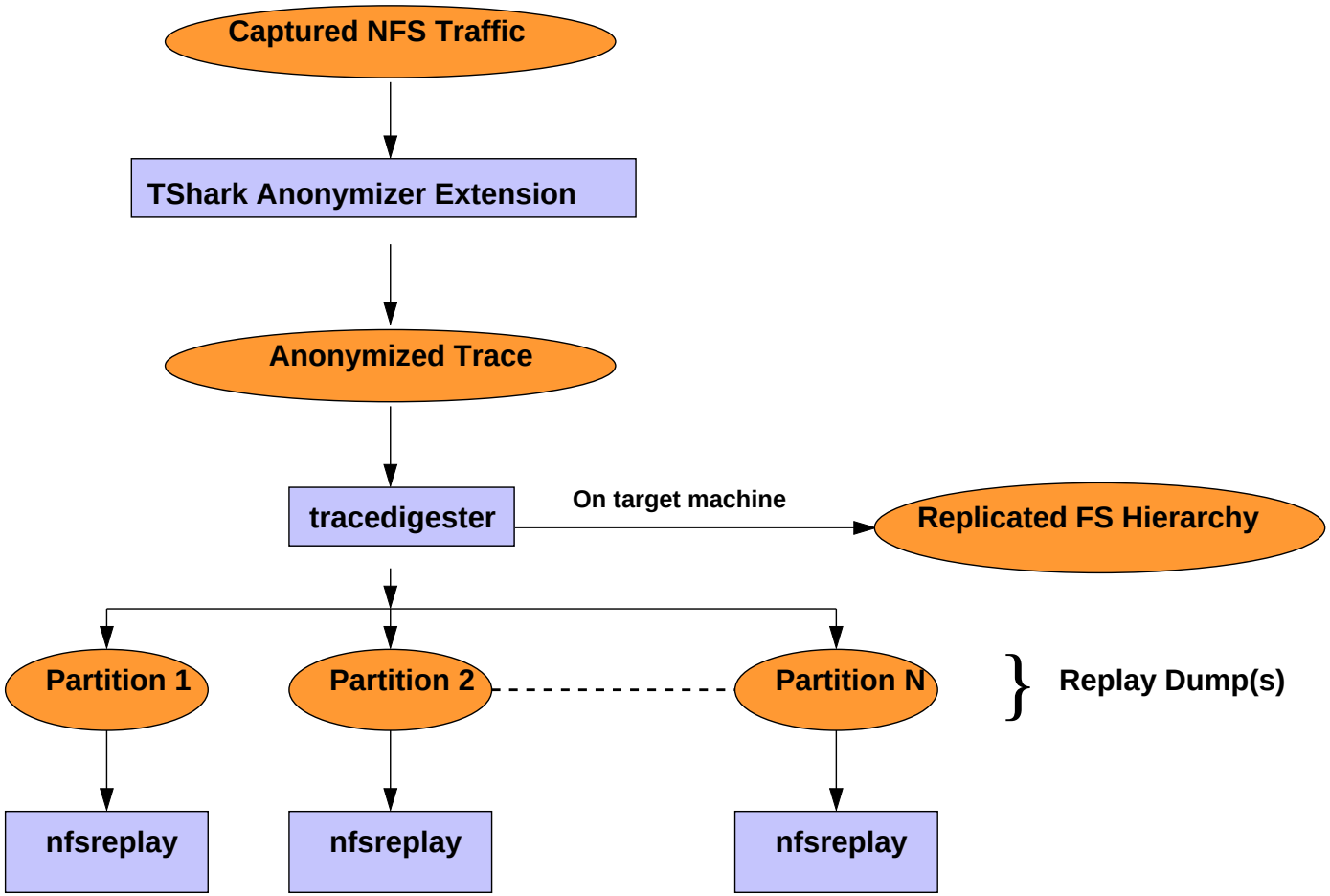
① Trace access subset of the exported FS hierarchy

② Target server must export a close replica

③ BUT with anonymized file and directory names

④ Create the accessed hierarchy using `tracedigester`

NFSREPLAY TOOLCHAIN

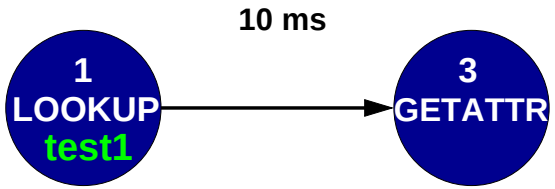


NFSREPLAY

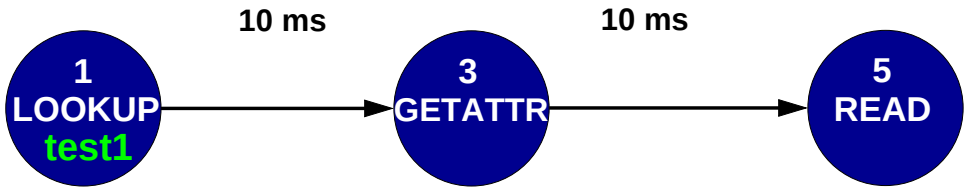
- Replay trace in replay dump
- Avoids client NFS stack overhead
- Types of workload scaling
  1. Scaled replay
  2. Pipelined replay

SCALED REPLAY

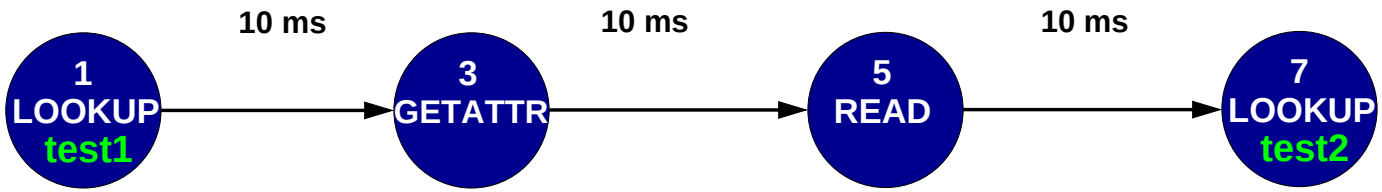
SCALED REPLAY



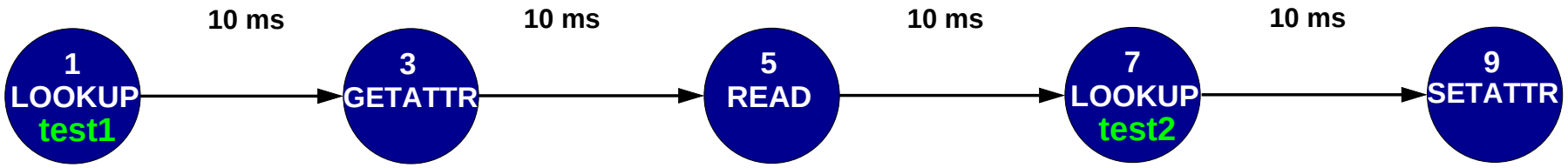
Scaled Replay



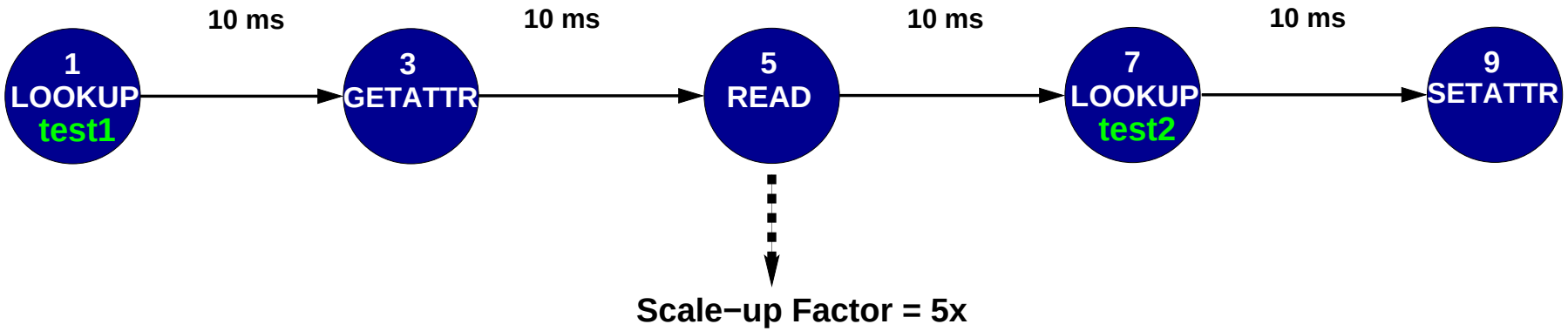
SCALED REPLAY



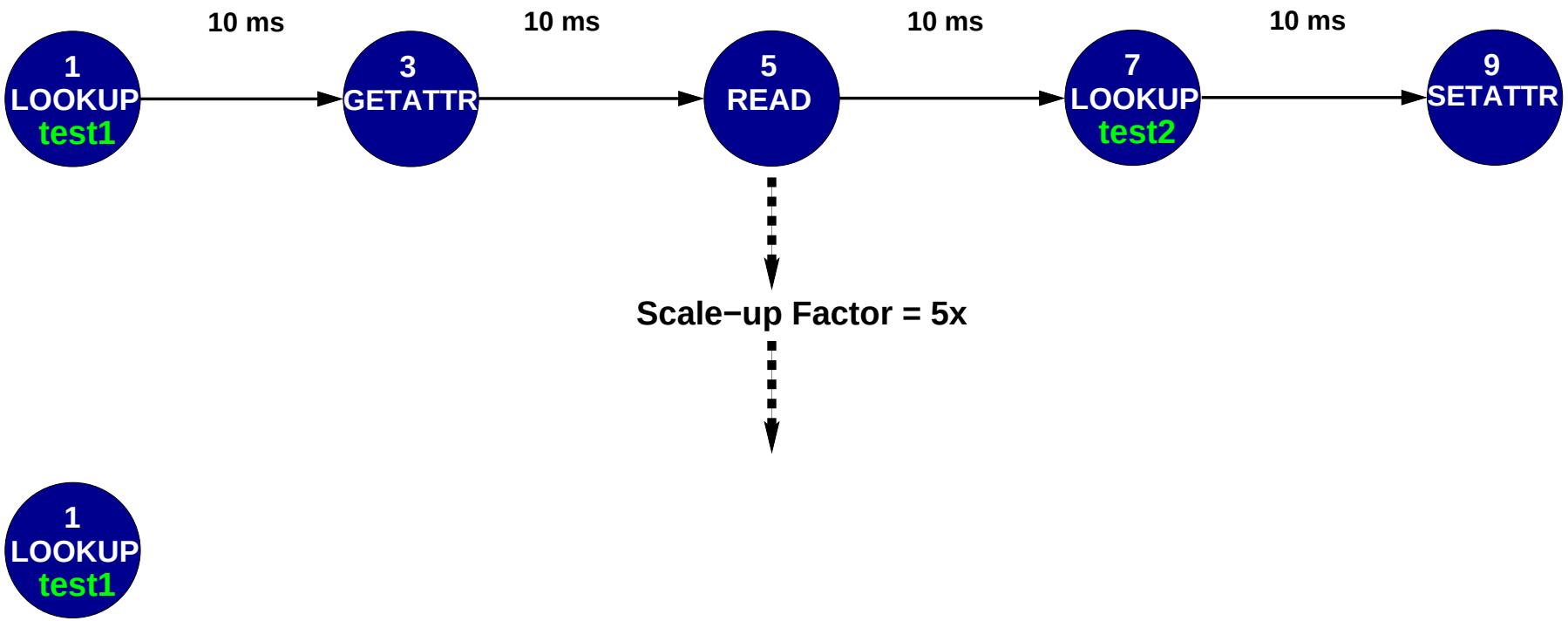
SCALED REPLAY



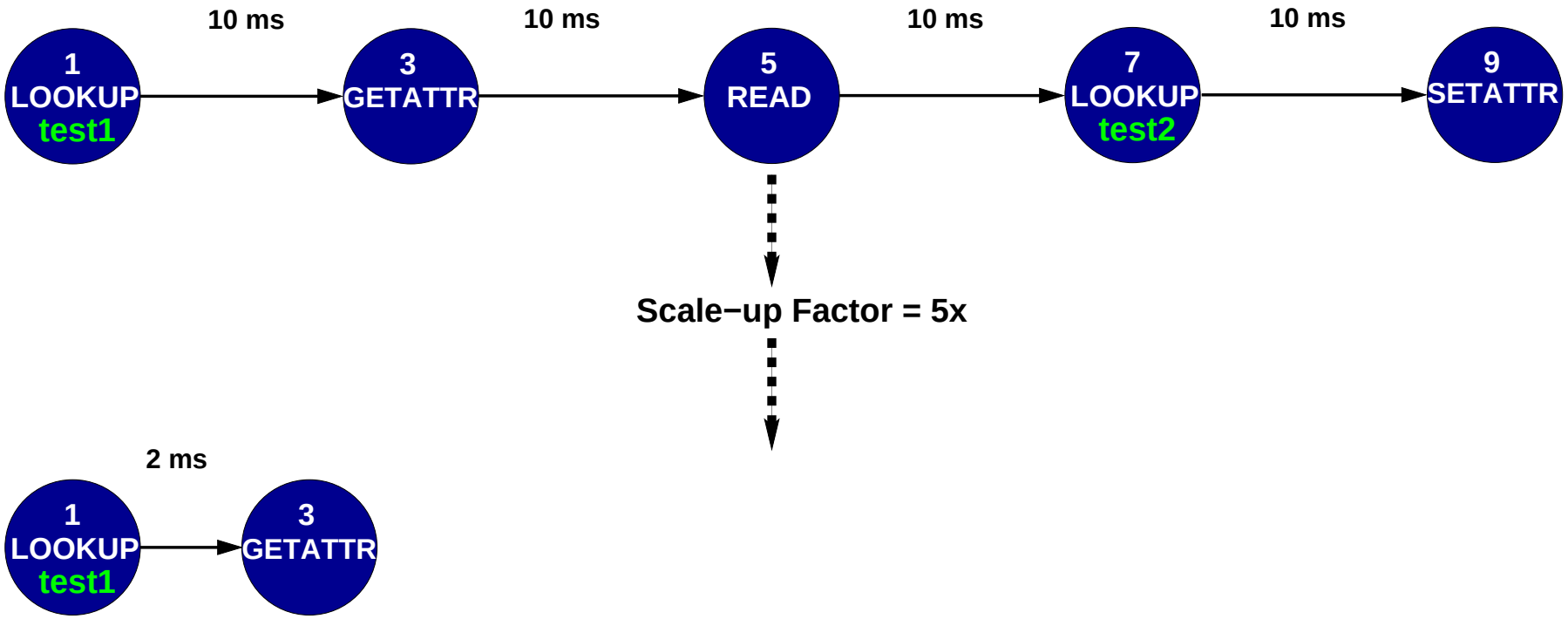
Scaled Replay



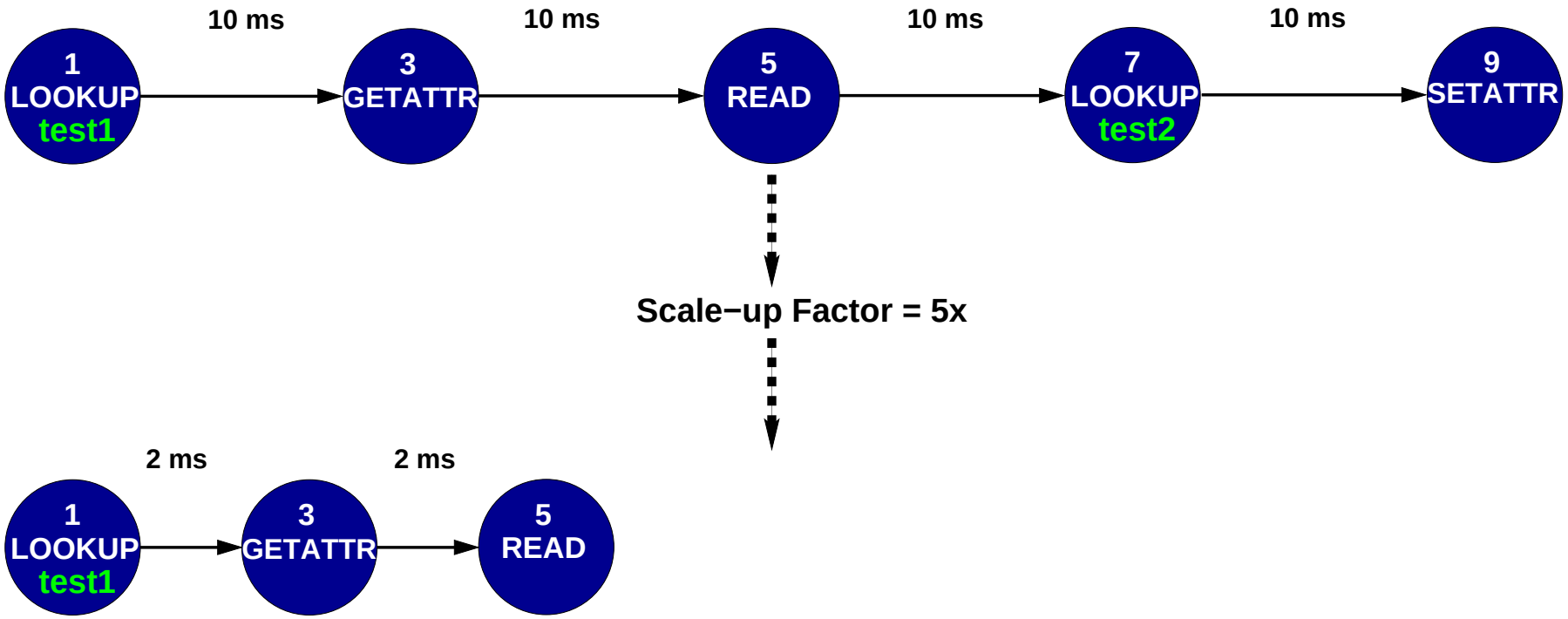
Scaled Replay



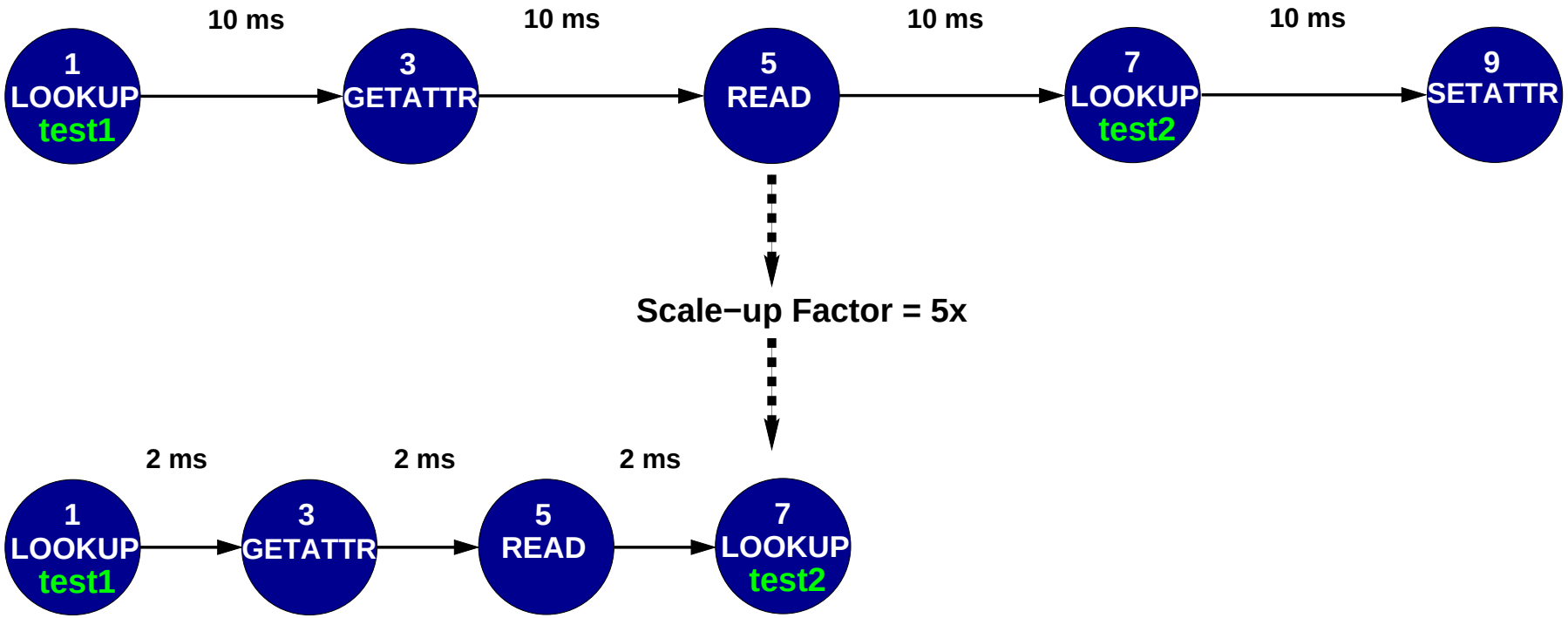
SCALED REPLAY



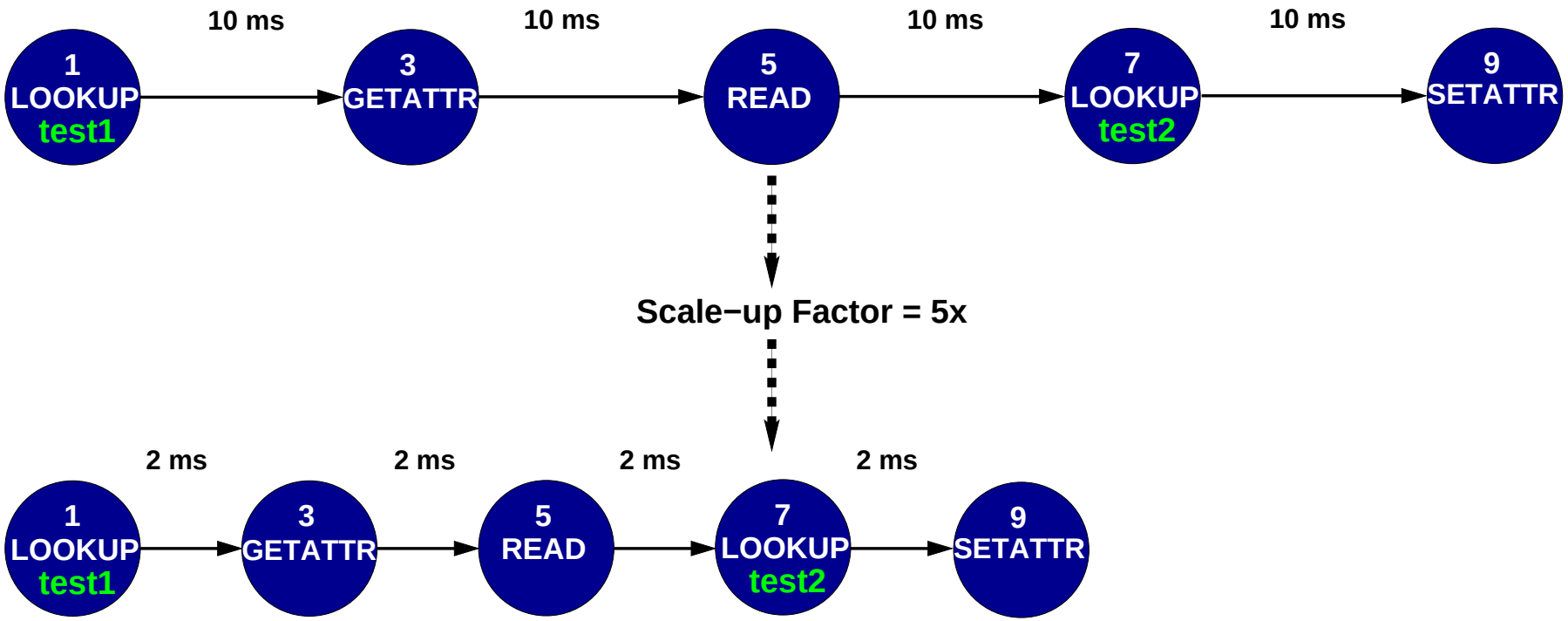
SCALED REPLAY



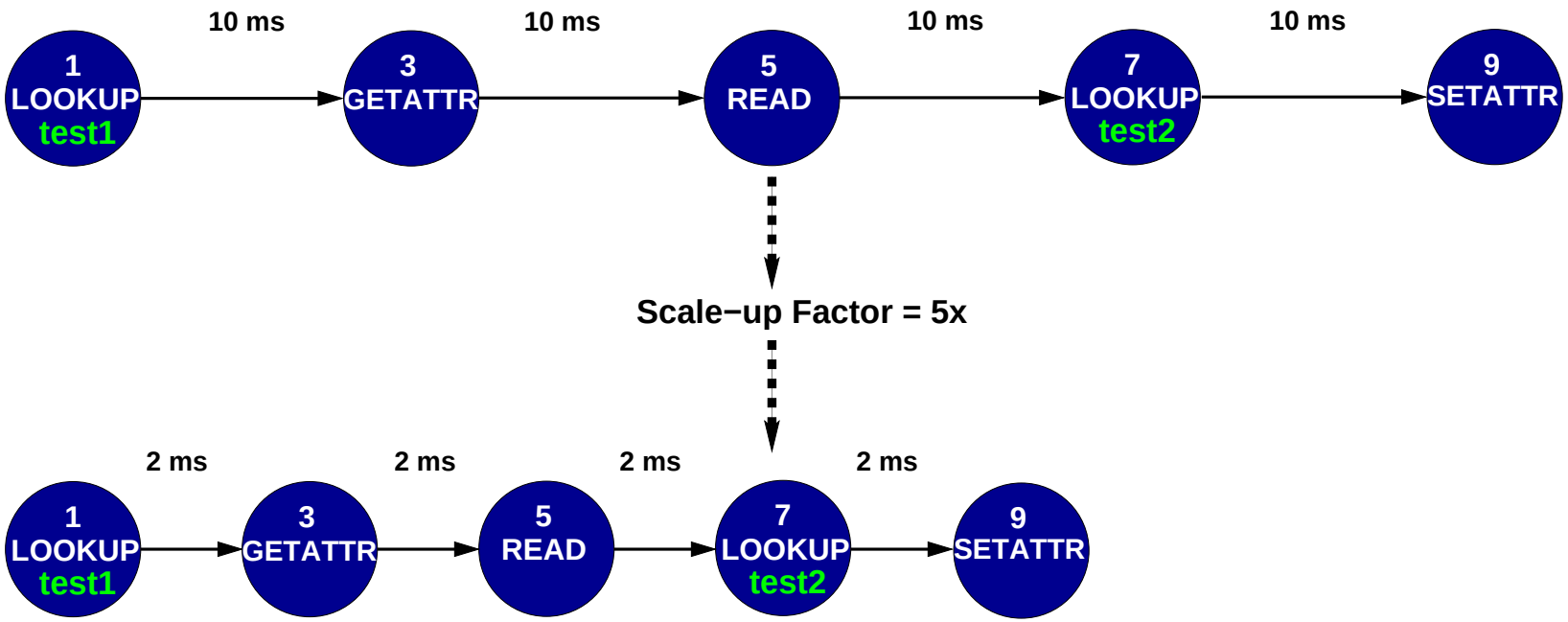
SCALED REPLAY



Scaled Replay

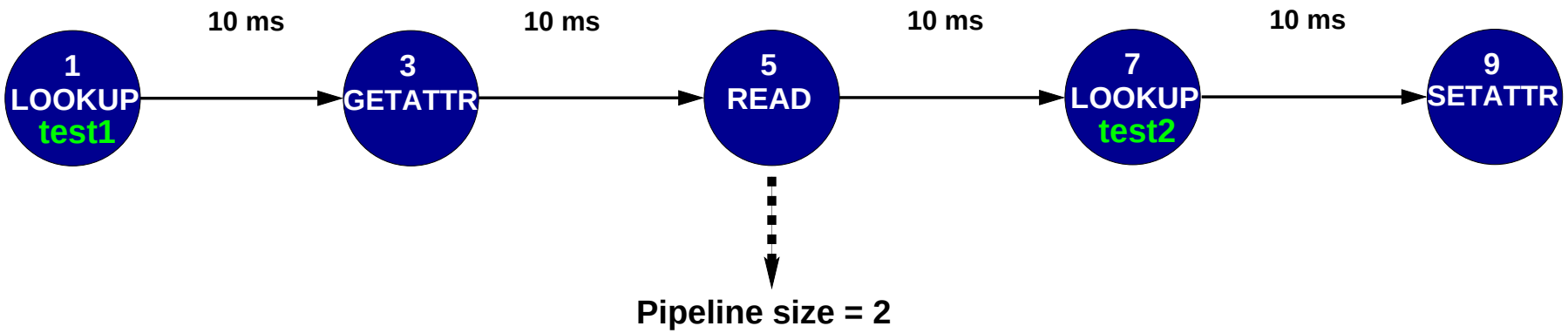


Scaled Replay

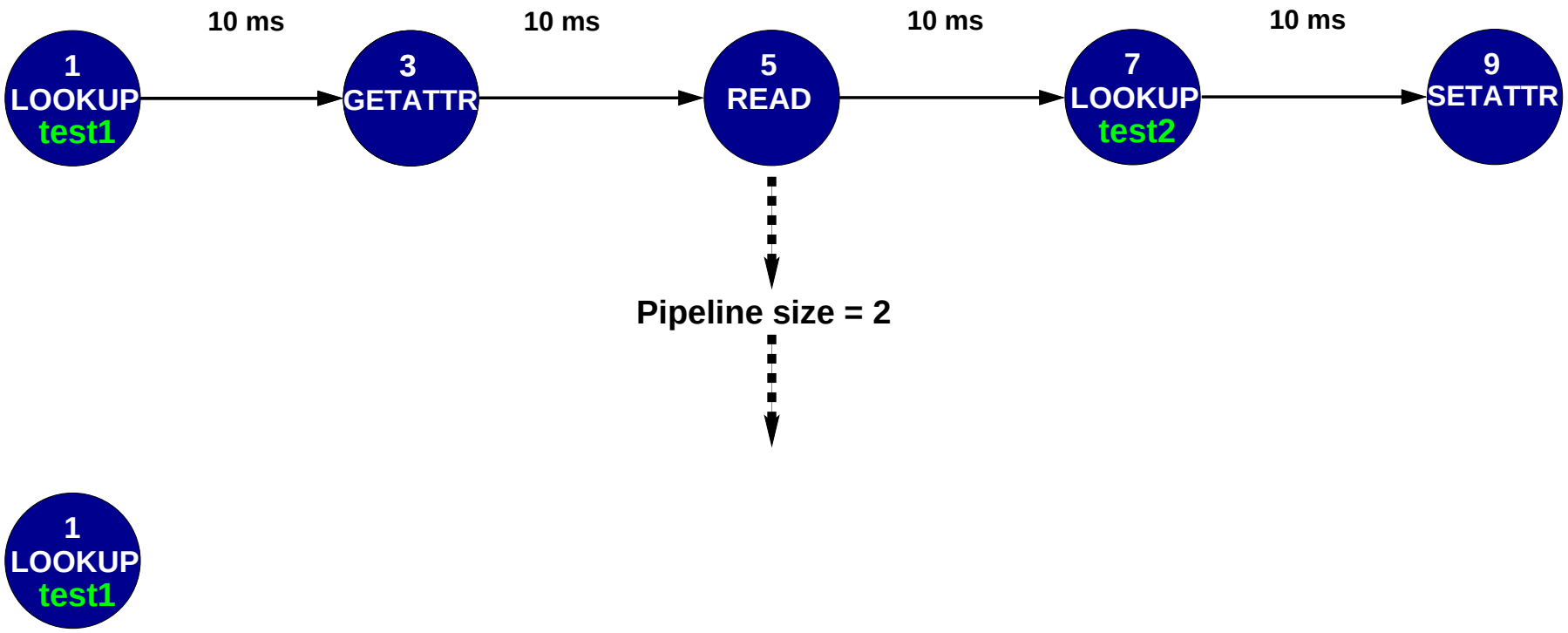


- Catch: Rate still limited by server response time
- `nfsreplay --tscale <scalefactor>`

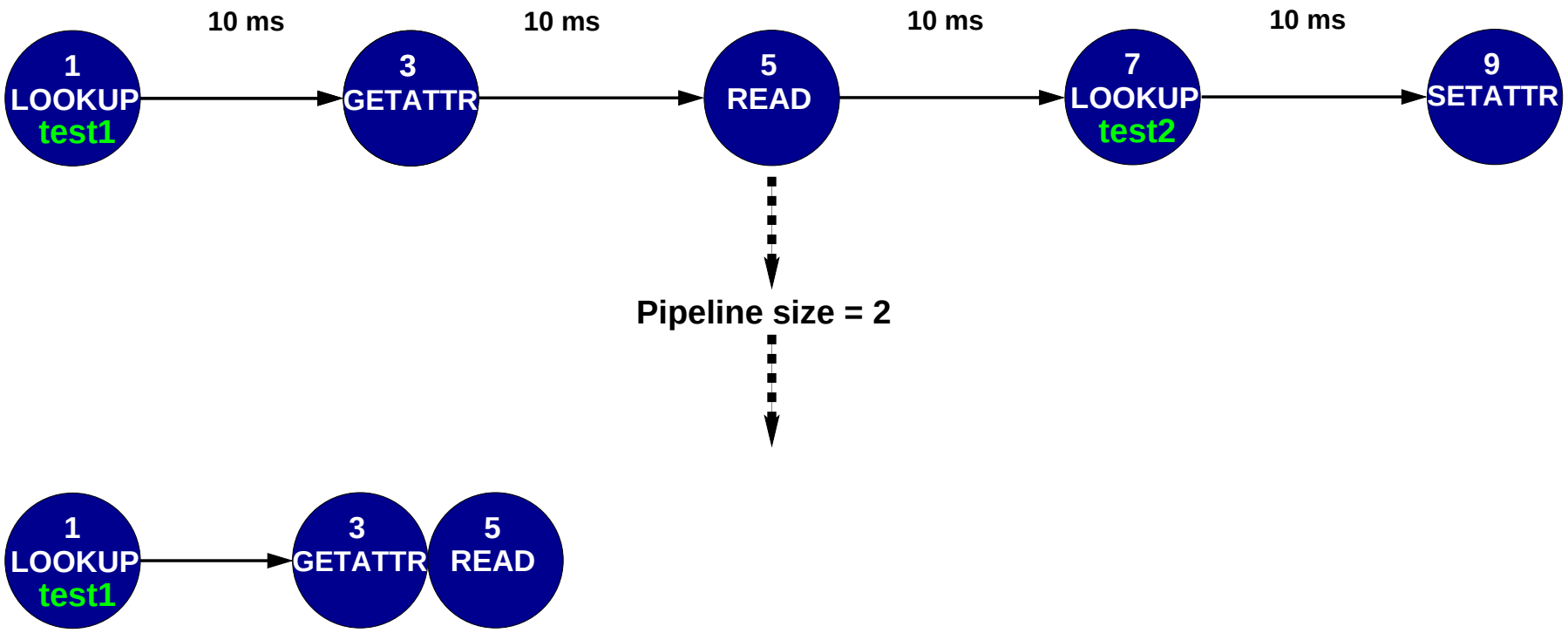
Pipelined Replay



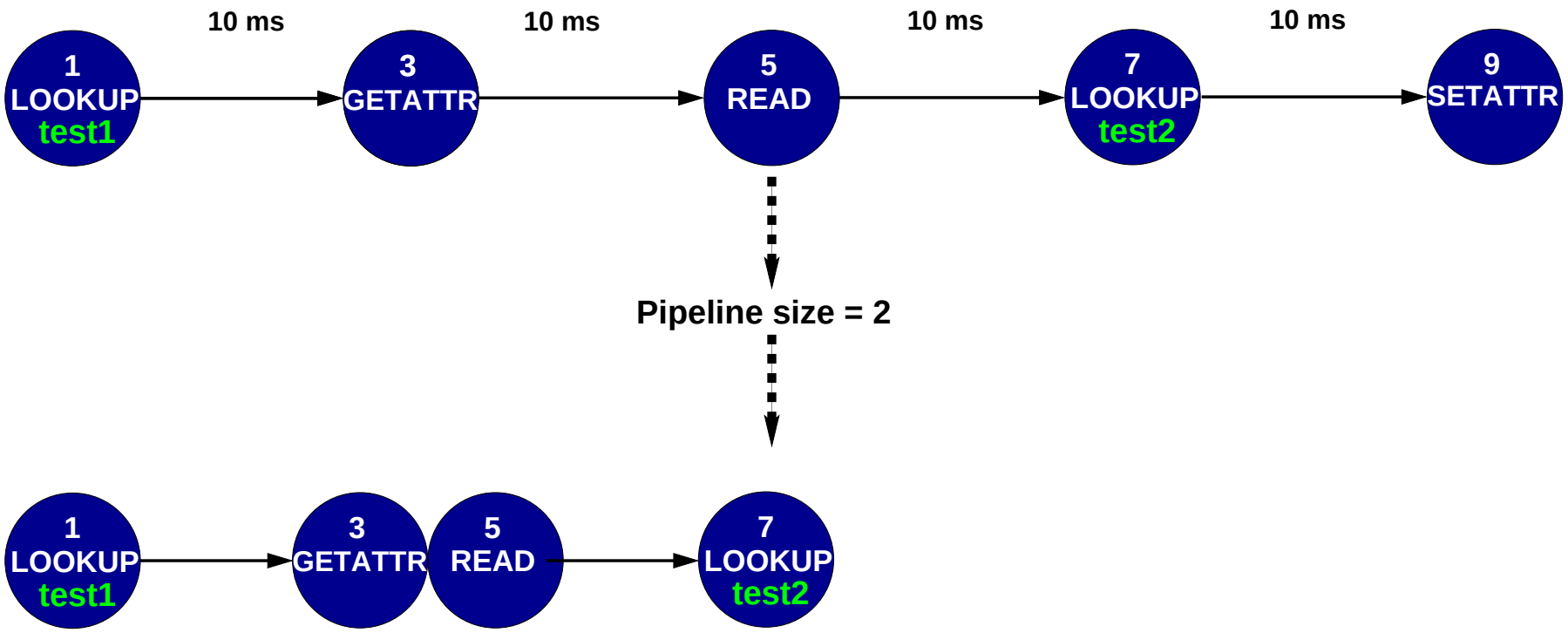
Pipelined Replay



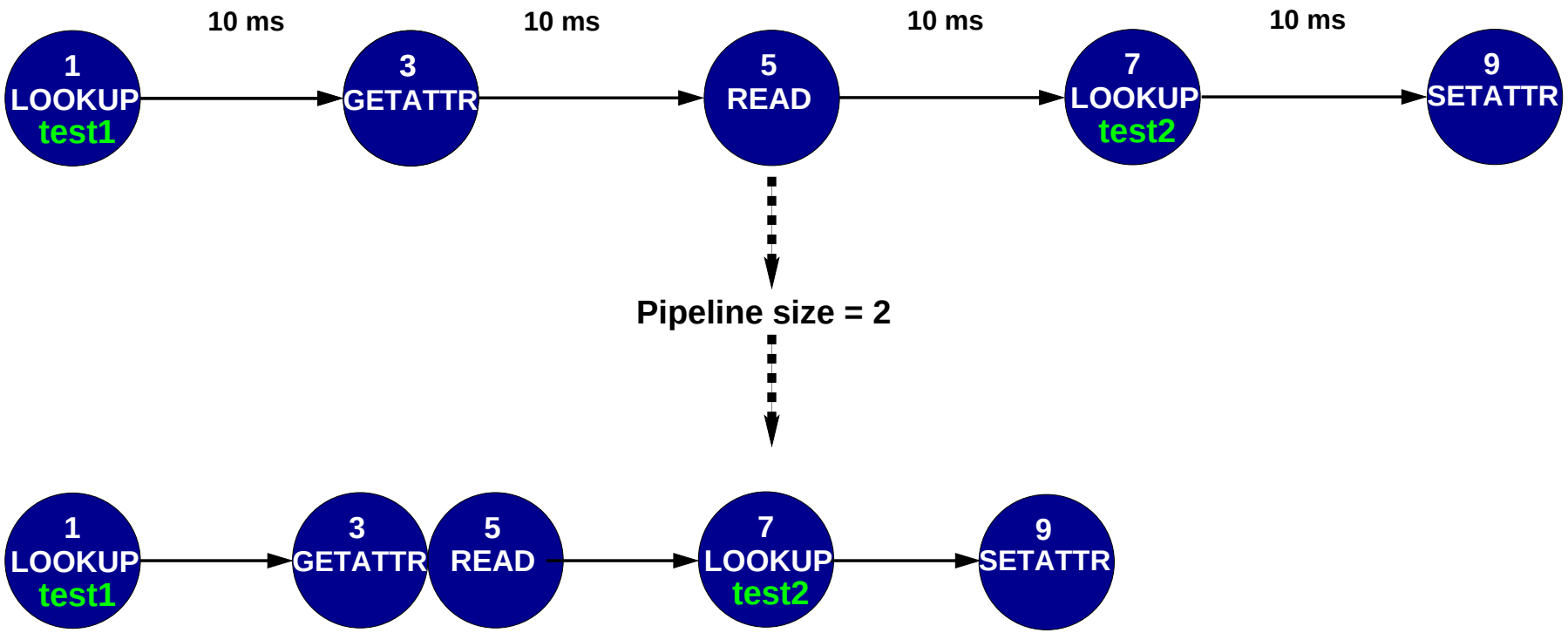
Pipelined Replay



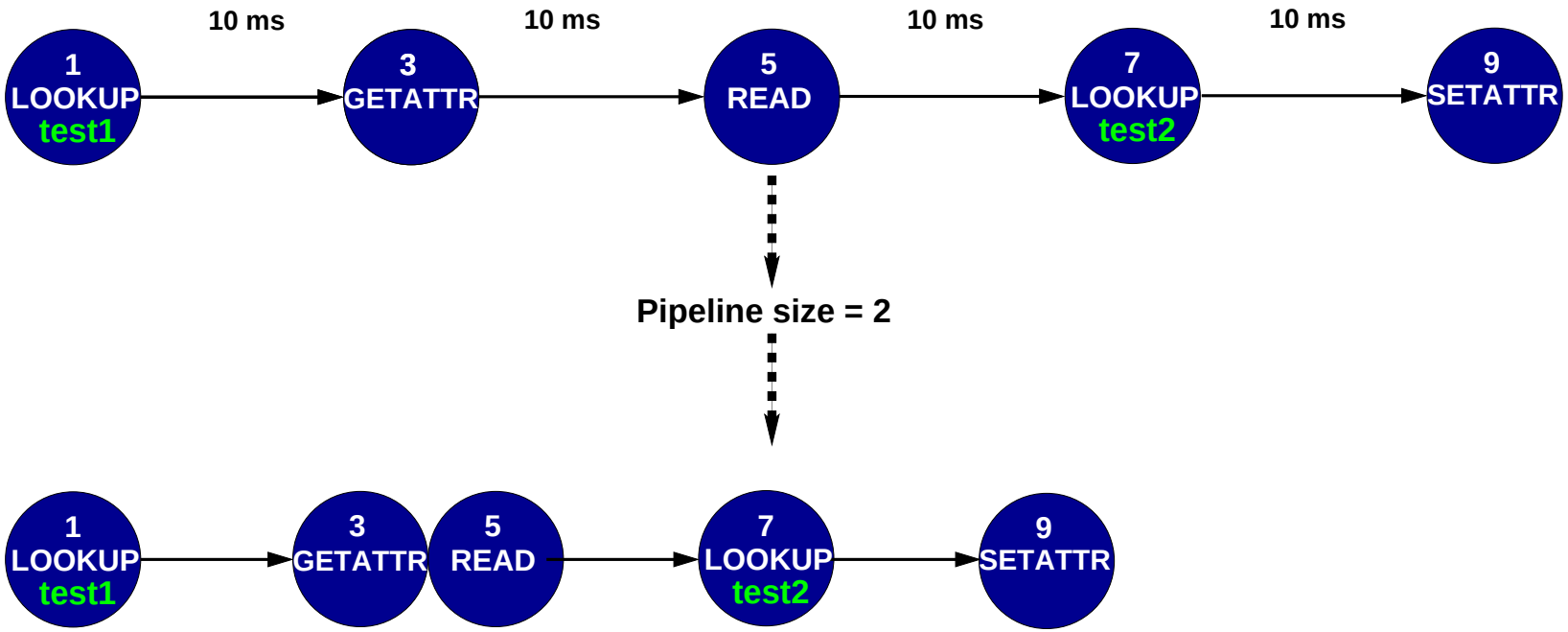
Pipelined Replay



Pipelined Replay



Pipelined Replay



- ➜ Keep server queues filled
- ➜ Catch: Must respect dependencies

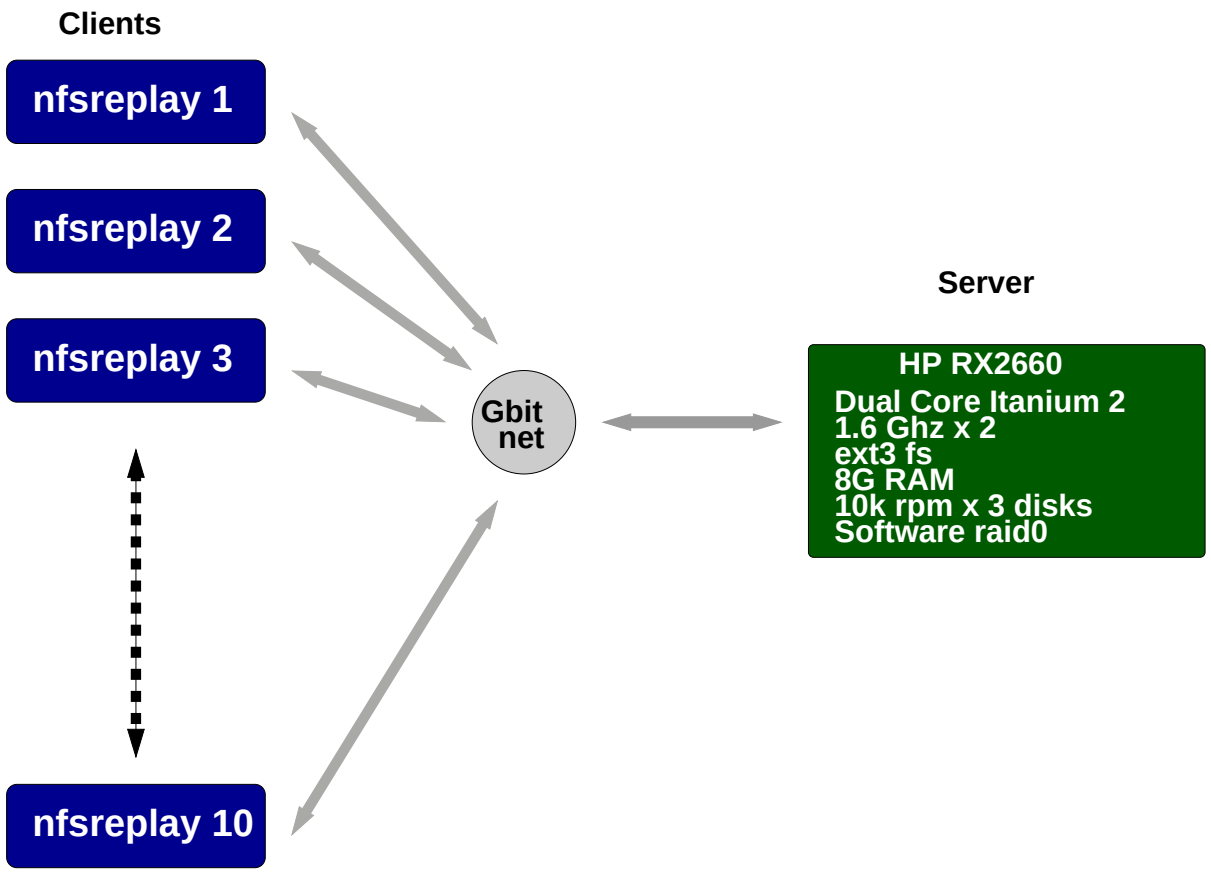
ASYNCHRONOUS RPC LIBRARY

- Glibc - Not flexible enough
- We needed;
  1. Non-blocking calls
  2. Callbacks-based reply notification
- `http://gelato.unsw.edu.au/IA64wiki/AsyncRPC`

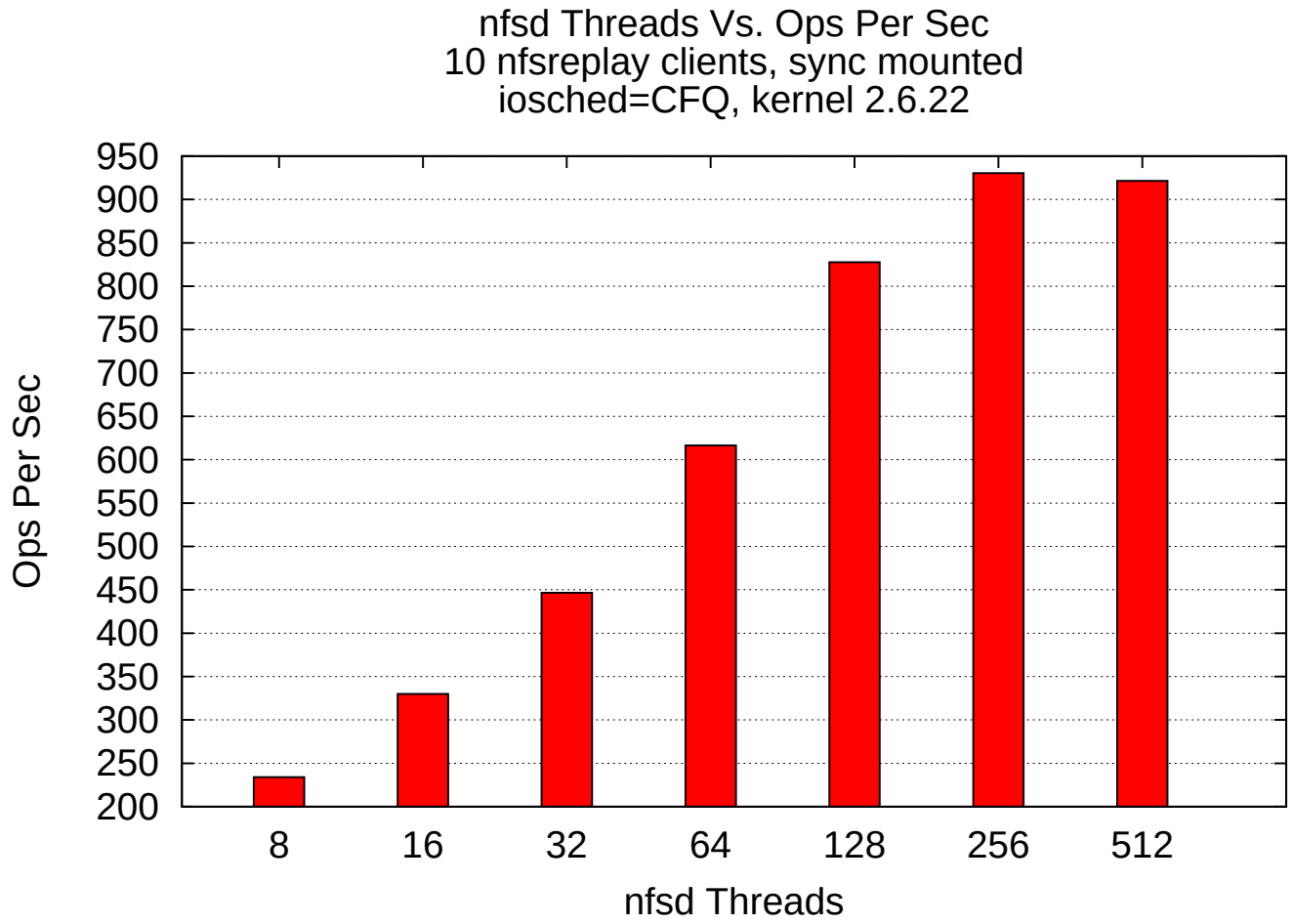
LIBNFSCLIENT

- Userspace NFS client ops library
- Hand NFS messages to per-request functions
- Performs XDR translation
- `http://gelato.unsw.edu.au/IA64wiki/libnfsclient`
- Aync RPC lib for async NFS requests

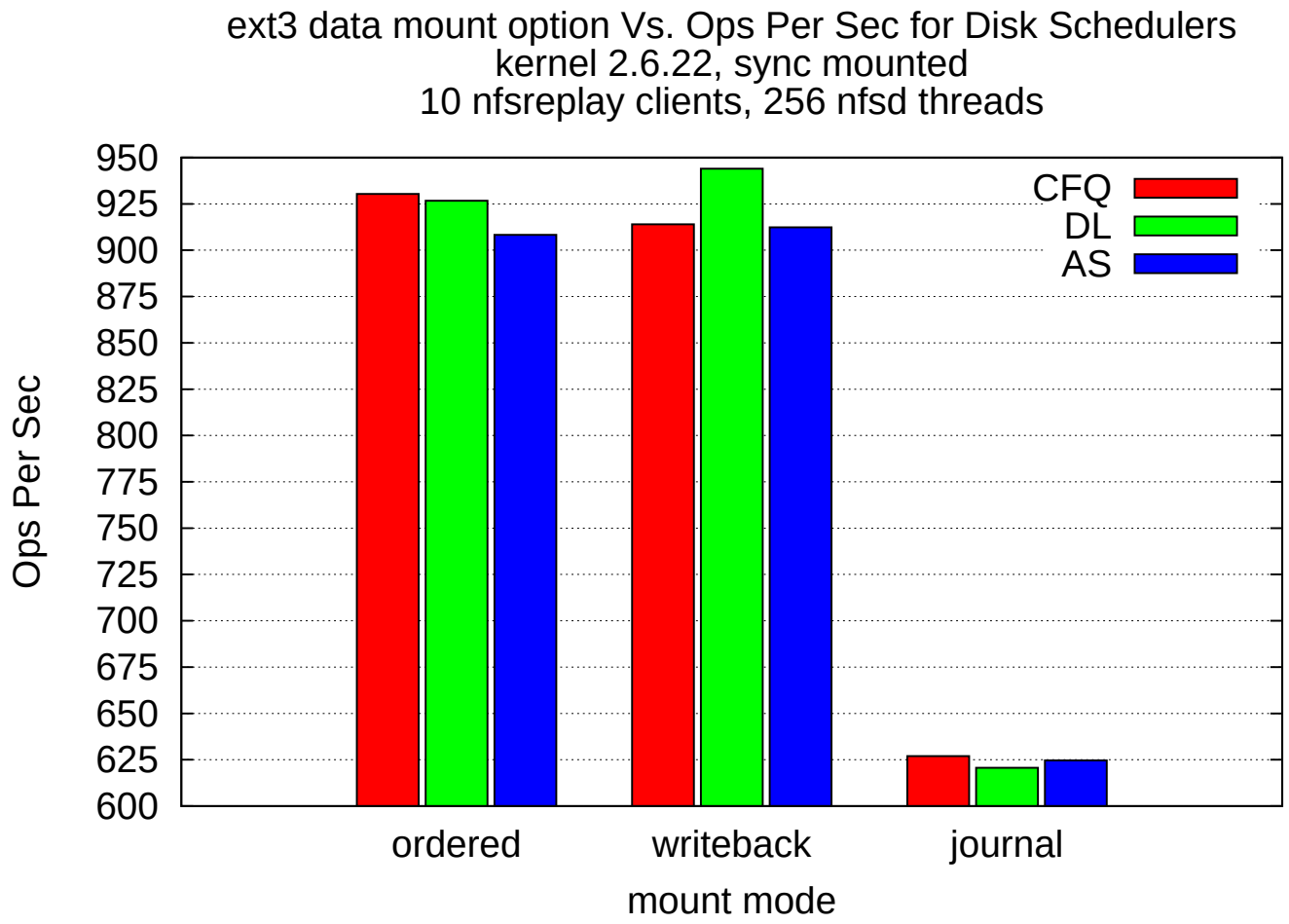
Topology


Clients
nfsreplay 1
nfsreplay 2
nfsreplay 3
nfsreplay 10
Gbit net
Server
HP RX2660
Dual Core Itanium 2
1.6 Ghz x 2
ext3 fs
8G RAM
10k rpm x 3 disks
Software raid0

MEASUREMENTS



MEASUREMENTS



CONCLUSIONS

① Largest share of server time: Storage

② If storage is unoptimal- Dont expect much win from
   - ➜ VM,
   - ➜ Network tuning

③ CFQ performs better than Deadline, AS

THANKS!

URLS:

① NFS Benchmarking project
`http://gelato.unsw.edu.au/IA64wiki/NFSBenchmarking`

② nfsreplay
`http://nfsreplay.sourceforge.net/`

③ nfsreplay Tech. Report
`http://gelato.unsw.edu.au/IA64wiki/nfsreplayTR`

REFERENCES

① *TBBT: Scalable and Accurate Trace Replay for File Server Evaluation*
Ningning Zhu, Jiawu Chen, Tzi-cker Chiueh

② *NFS Version 3 Protocol Specification*
Callaghan et. al.

③ *NFS Tracing by Passive Network Monitoring*
Matthew Blaze

④ *New NFS Tracing Tools and Techniques for System Analysis*
Daniel Ellard and Margo Seltzer

⑤ *RFC 1831 - RPC: Remote Procedure Call Protocol Specification Version 2*
R. Srinivasan

⑥ *RFC 1832 - XDR: External Data Representation Standard*
R. Srinivasan

APPENDICES

Traffic Capture

- NFS over TCP
- TCP is byte-stream oriented - Implications for RPC packet capture
- Need to capture full packets
- Popular capture tools:
  - `tcpdump -s 0`
  - `ttshark -s 65536`
- Details at:
  `www.gelato.unsw.edu.au/IA64wiki/RPCOverTCPCapture`

Replay Dumps

① Result of **the** pre-processing
② Binary format for storing NFS request/replies
③ Contains ordering and dependency info
④ AKA. `rdumps`
⑤ Generated using `tracedigester`
⑥ Used as input to nfsreplay

Trace Partitioning

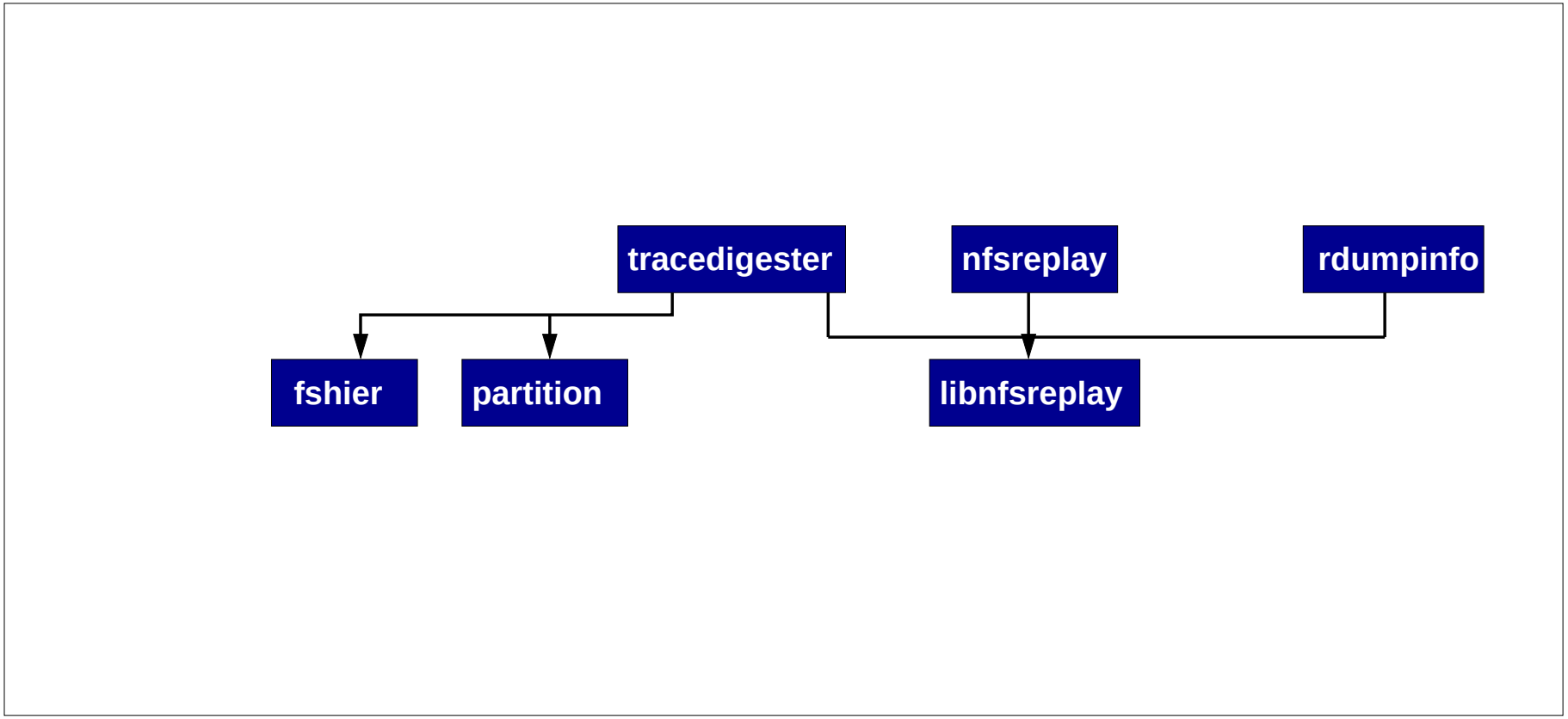
① Trace contains all requests

② Need per-client rdumps?

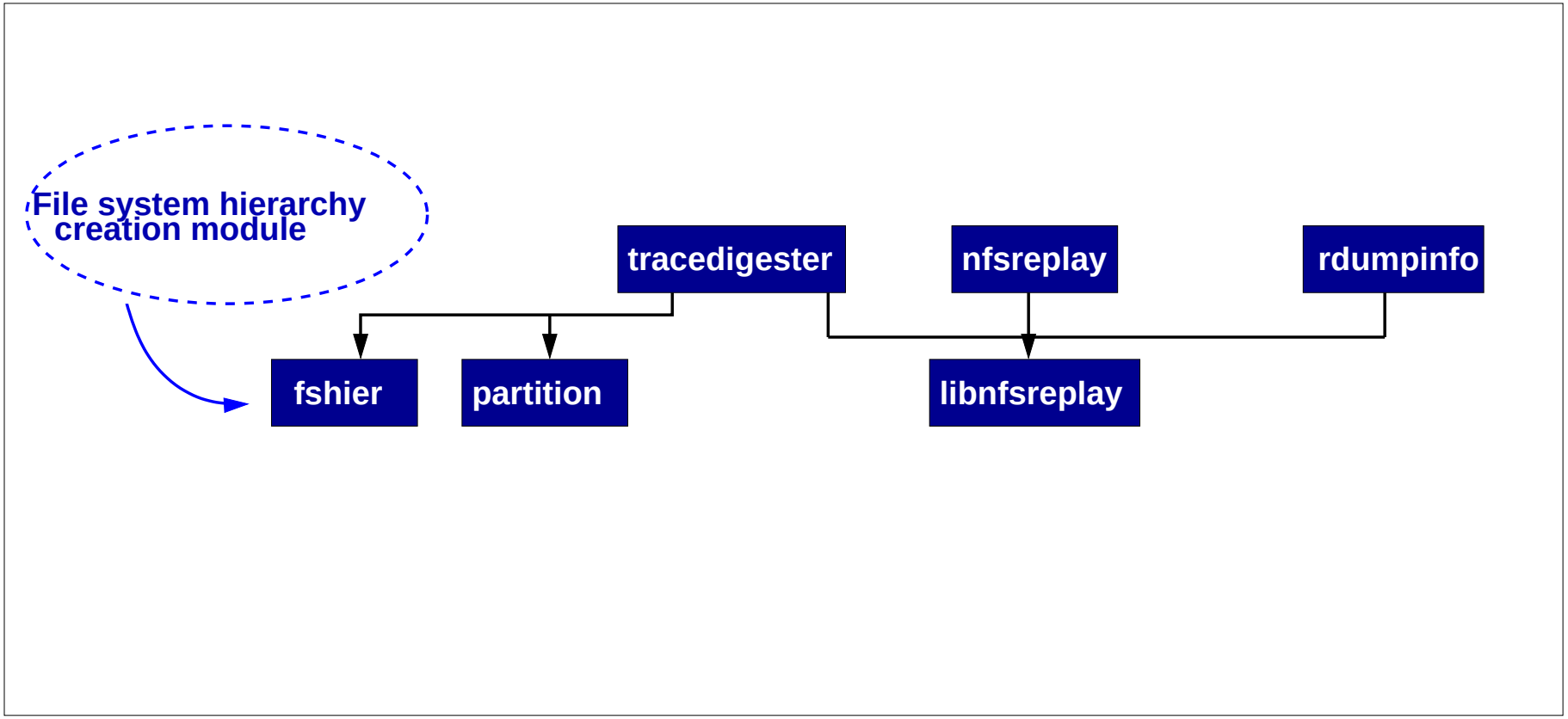
③ Partition on source addresses

④ Input to multiple nfsreplay instances

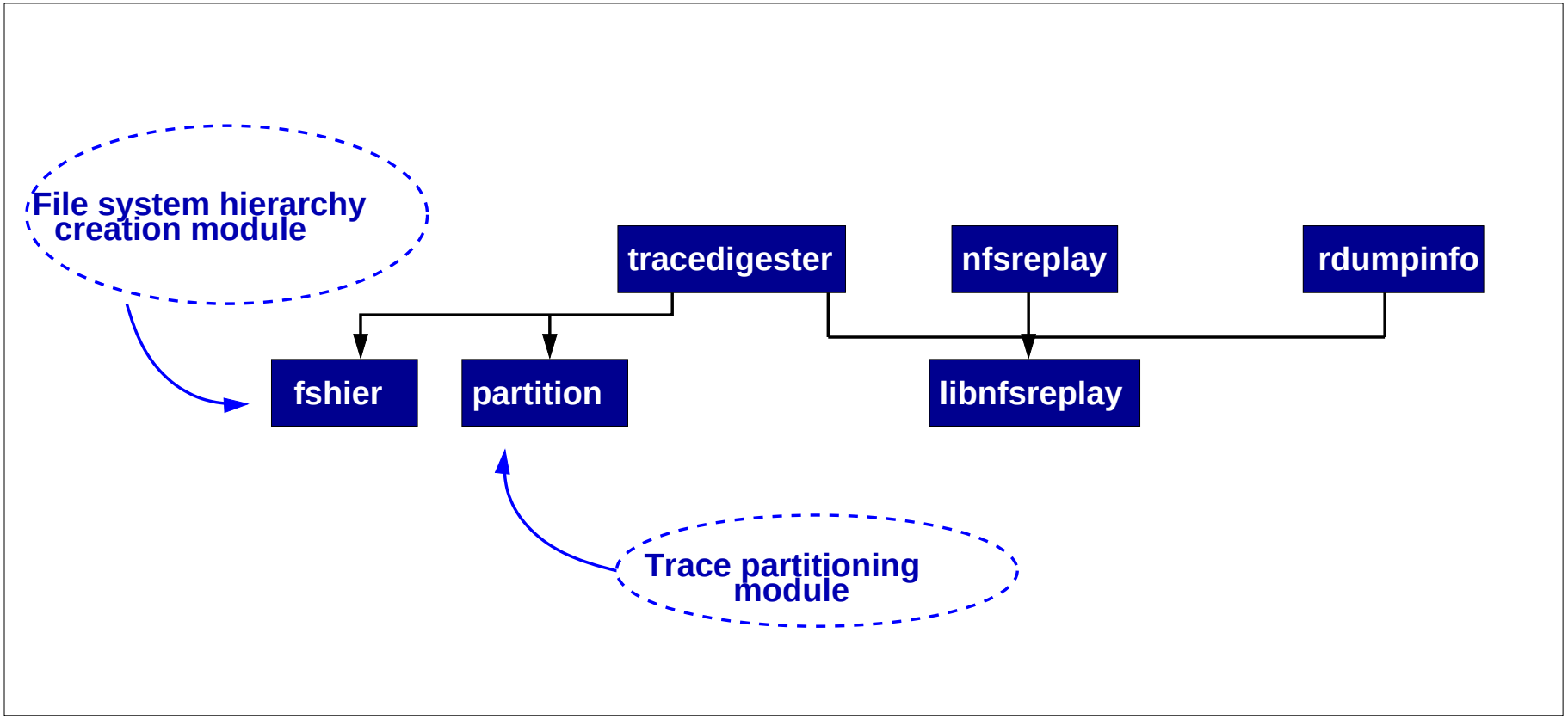
NFSREPLAY COMPONENTS



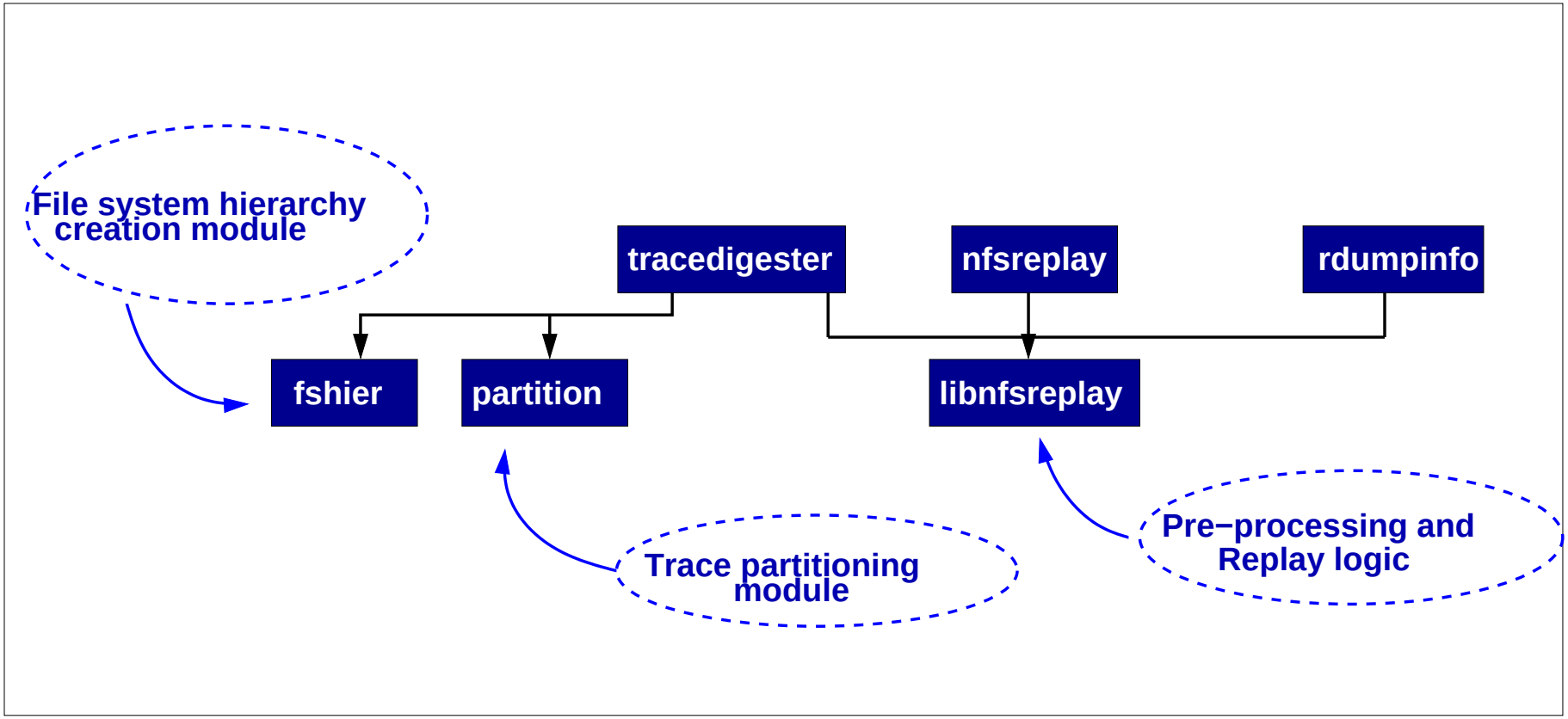
NFSREPLAY COMPONENTS



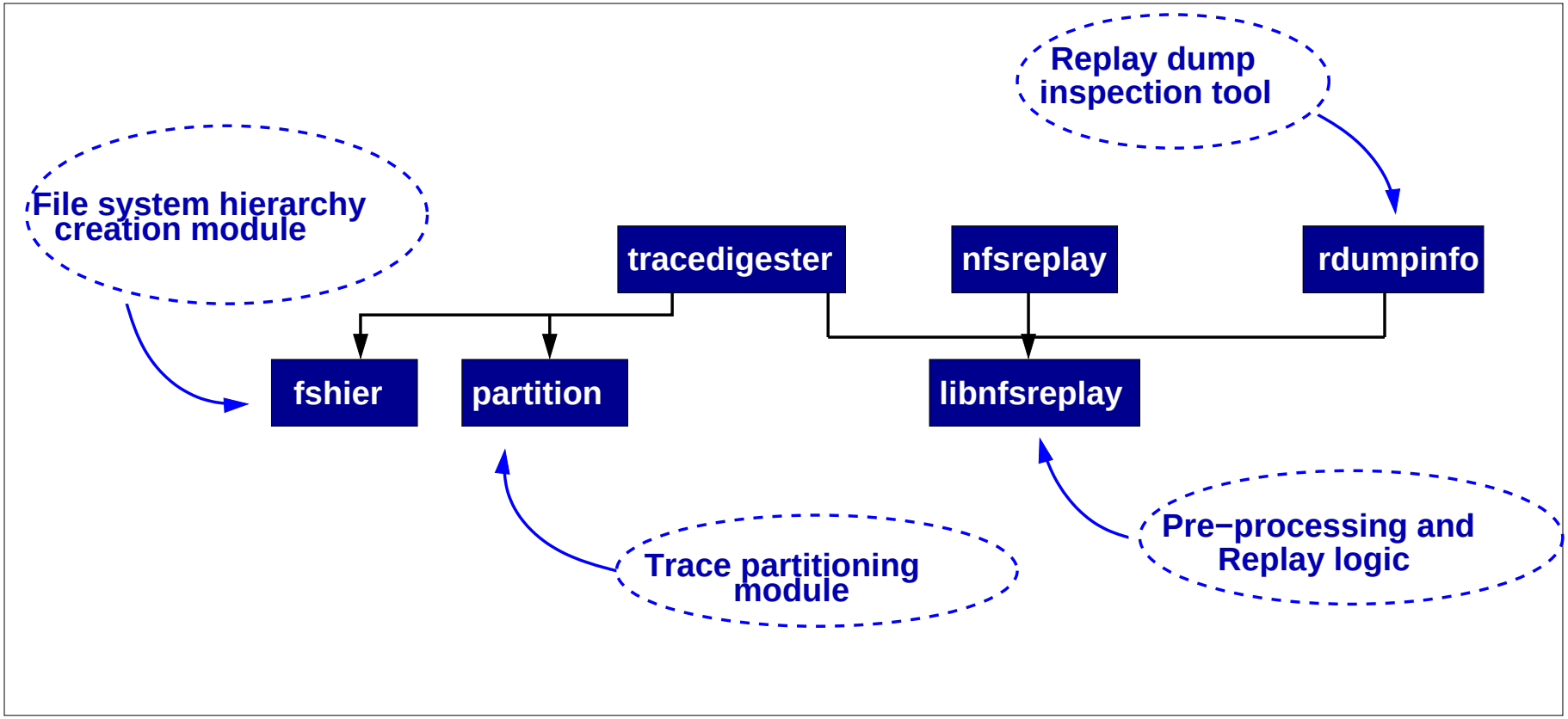
NFSREPLAY COMPONENTS



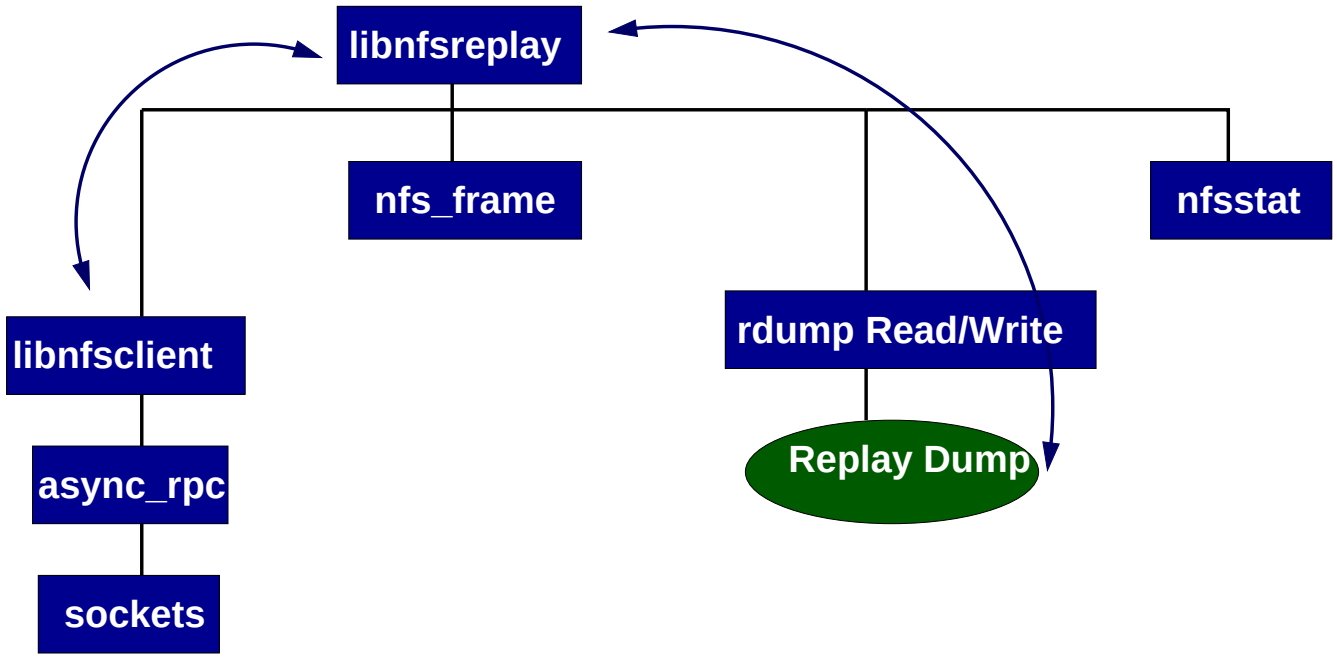
NFSREPLAY COMPONENTS



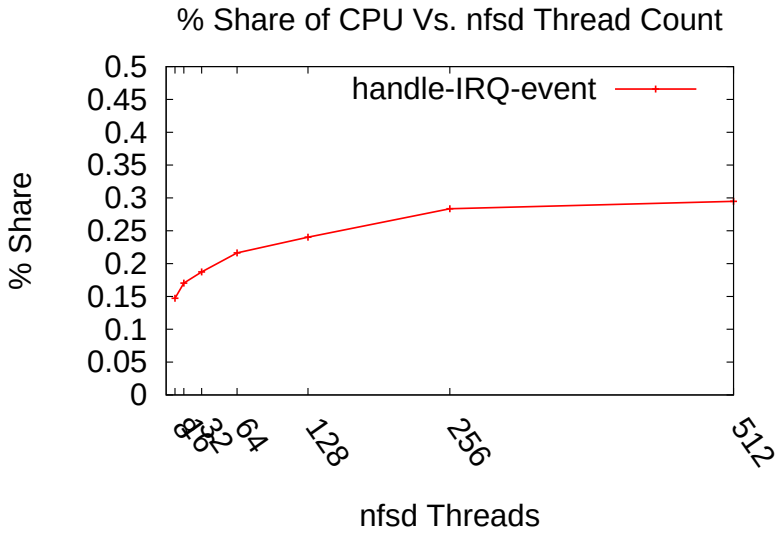
NFSREPLAY COMPONENTS

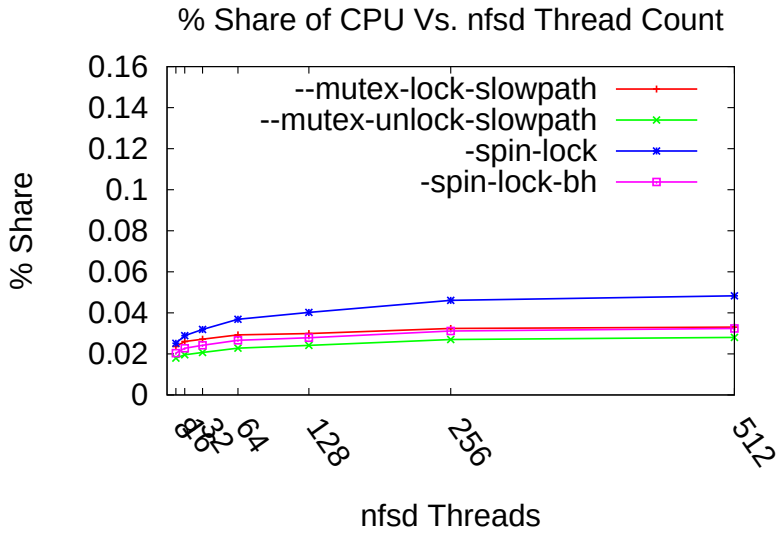


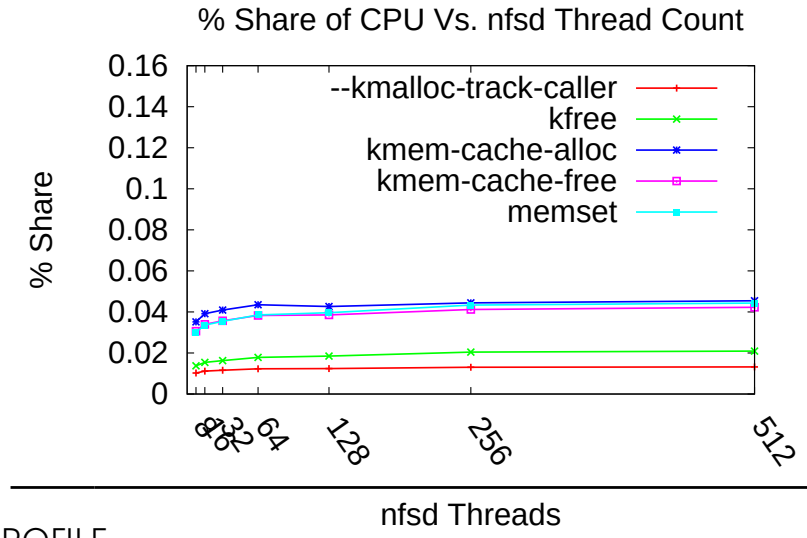
NFSREPLAY COMPONENTS

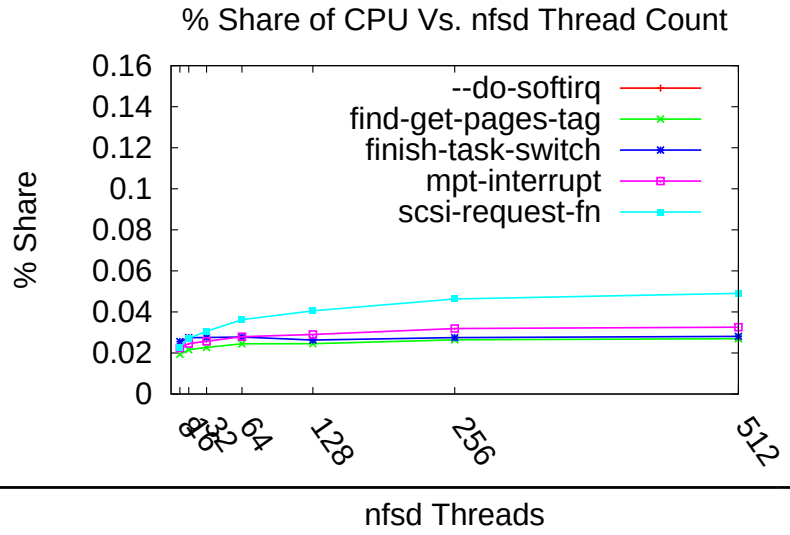


KERNEL PROFILE

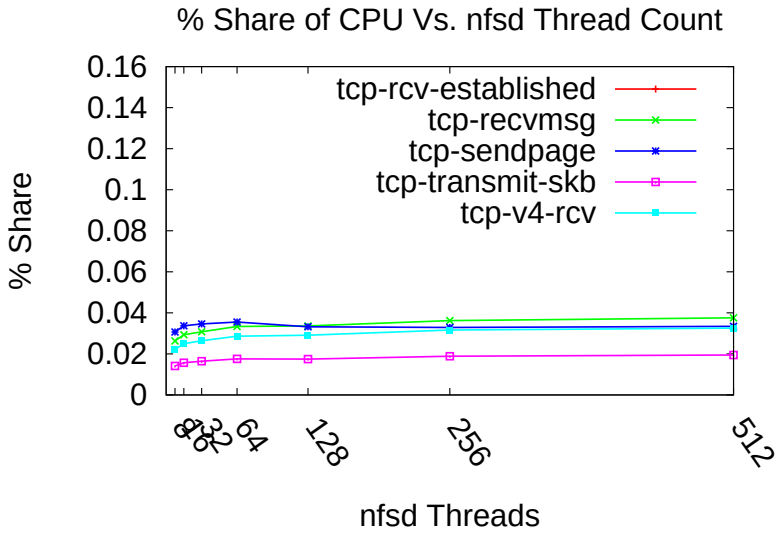


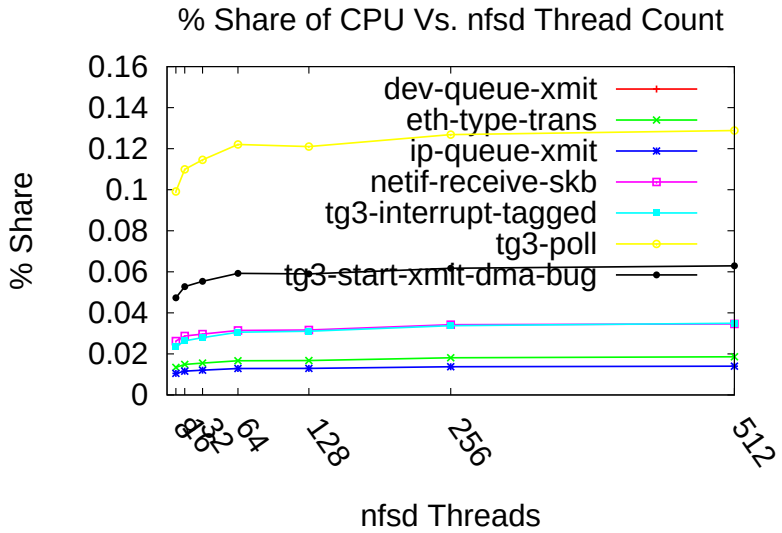


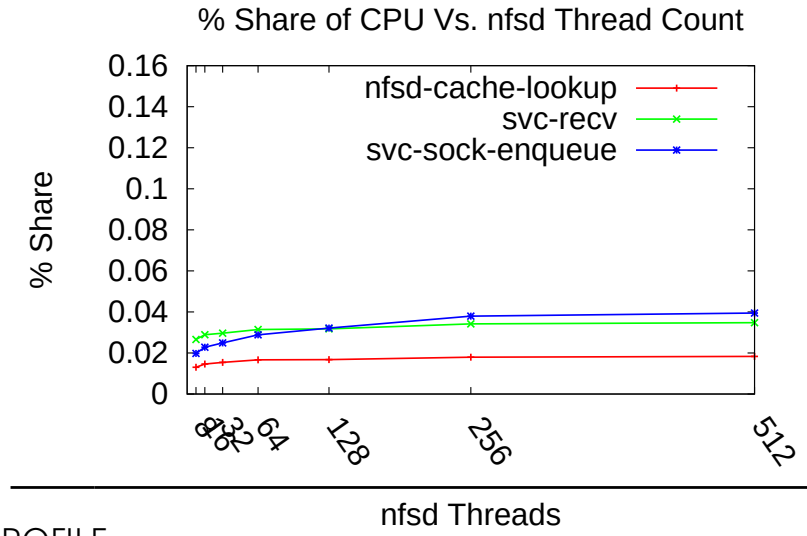




KERNEL PROFILE







Trace Anonymization
Original NFSv3 Read reply hexdump

```
00000140 00 00 00 02 00 00 00 00 00 00 03 e8 00 00 00 00 |................|
00000150 00 00 4f f5 00 00 00 00 00 00 50 00 00 00 00 00 |..O.......P.....|
00000160 00 00 00 00 00 00 00 00 00 00 03 02 00 00 00 00 |................|
00000170 00 00 00 14 46 08 7d 70 00 00 00 00 45 b7 e2 6a |....F..p....Ej|
00000180 00 00 00 00 46 08 66 00 00 00 00 00 00 00 04 00 |....F.f.........|
00000190 00 00 00 00 00 00 04 00 0a 09 09 4c 69 6e 75 78 |............Linux|
000001a0 20 6b 65 72 6e 65 6c 20 63 6f 64 69 6e 67 20 73 | kernel coding s|
000001b0 74 79 6c 65 0a 0a 54 68 69 73 20 69 73 20 61 20 |tyle..This is a |
000001c0 73 68 6f 72 74 20 64 6f 63 75 6d 65 6e 74 20 64 |short document d|
000001d0 65 73 63 72 69 62 69 6e 67 20 74 68 65 20 70 72 |escribing the pr|
000001e0 65 66 65 72 72 65 64 20 63 6f 64 69 6e 67 20 73 |eferred coding s|
000001f0 74 79 6c 65 20 66 6f 72 20 74 68 65 0a 6c 69 6e |tyle for the.lin|
00000200 75 78 20 6b 65 72 6e 65 6c 2e 20 20 43 6f 64 69 |ux kernel.  Codi|

00000210 6e 67 20 73 74 79 6c 65 20 69 73 20 76 65 72 79 |ng style is very|
```

Trace Anonymization
Anonymized NFSv3 Read reply hexdump

```
00000000 d4 c3 b2 a1 02 00 04 00 00 00 00 00 00 00 00 00 |................|
00000010 ff ff 00 00 01 00 00 00 3f 7c 08 46 58 15 0d 00 |........?|.FX...|
00000020 44 00 00 00 44 00 00 00 00 00 00 01 00 00 00 02 |D...D...........|
00000030 00 00 00 00 55 fc 8d 31 00 00 00 03 00 00 00 06 |....U.1.........|
00000040 00 00 00 1c 01 00 00 00 00 00 00 00 00 00 00 00 |................|
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
00000060 00 00 00 00 00 00 00 00 00 00 04 00 3f 7c 08 46 |............?|.F|
00000070 07 1a 0d 00 7c 00 00 00 7c 00 00 00 00 00 00 02 |....|...|.......|
00000080 00 00 00 01 00 00 00 01 55 fc 8d 31 00 00 00 03 |........U.1....|
00000090 00 00 00 06 00 00 00 00 00 00 00 01 00 00 00 01 |................|
000000a0 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 |................|
000000b0 00 00 00 00 00 00 4f f5 00 00 00 00 00 00 50 00 |......O......P.|
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 02 |................|
000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
* <----- Denotes contiguous zeroes
000000f0 00 00 04 00 00 00 00 00 |........|
```