RALPH BÖHME / SAMBA TEAM

SerNet

SAMBA FILESERVER PERFORMANCE

HOW I LEARNED TO LOVE PERF . . . AND SYSTEMTAP

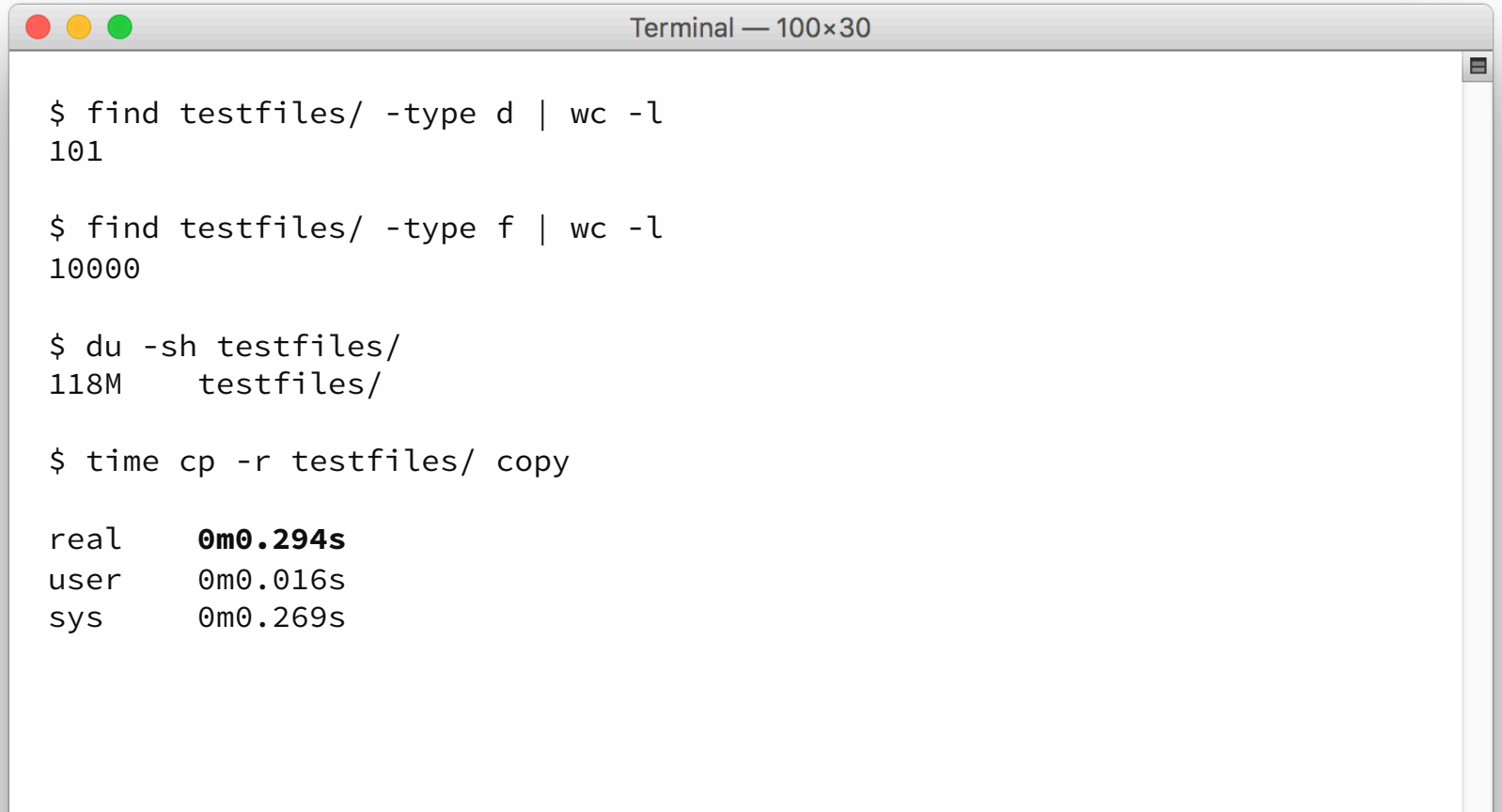
AGENDA

SerNet

1. Introduction: understanding Samba fileserver performance
    1.1. Case study: cp 10k 10 KB files
2. Performance metrics: tools / tracing frameworks
    2.1. /proc based tools (ps, pidstat, top, vmstat, iostat, …)
    2.2. Systemtap
    2.3. perf
3. Demo: perf
4. Recent performance improvements in smbd

1.1. CASE STUDY

- Small files copy: 10,000 files, each 10 KB
- Time for local copy is under one second at nearly 100% CPU:



1.1. CASE STUDY

SerNet

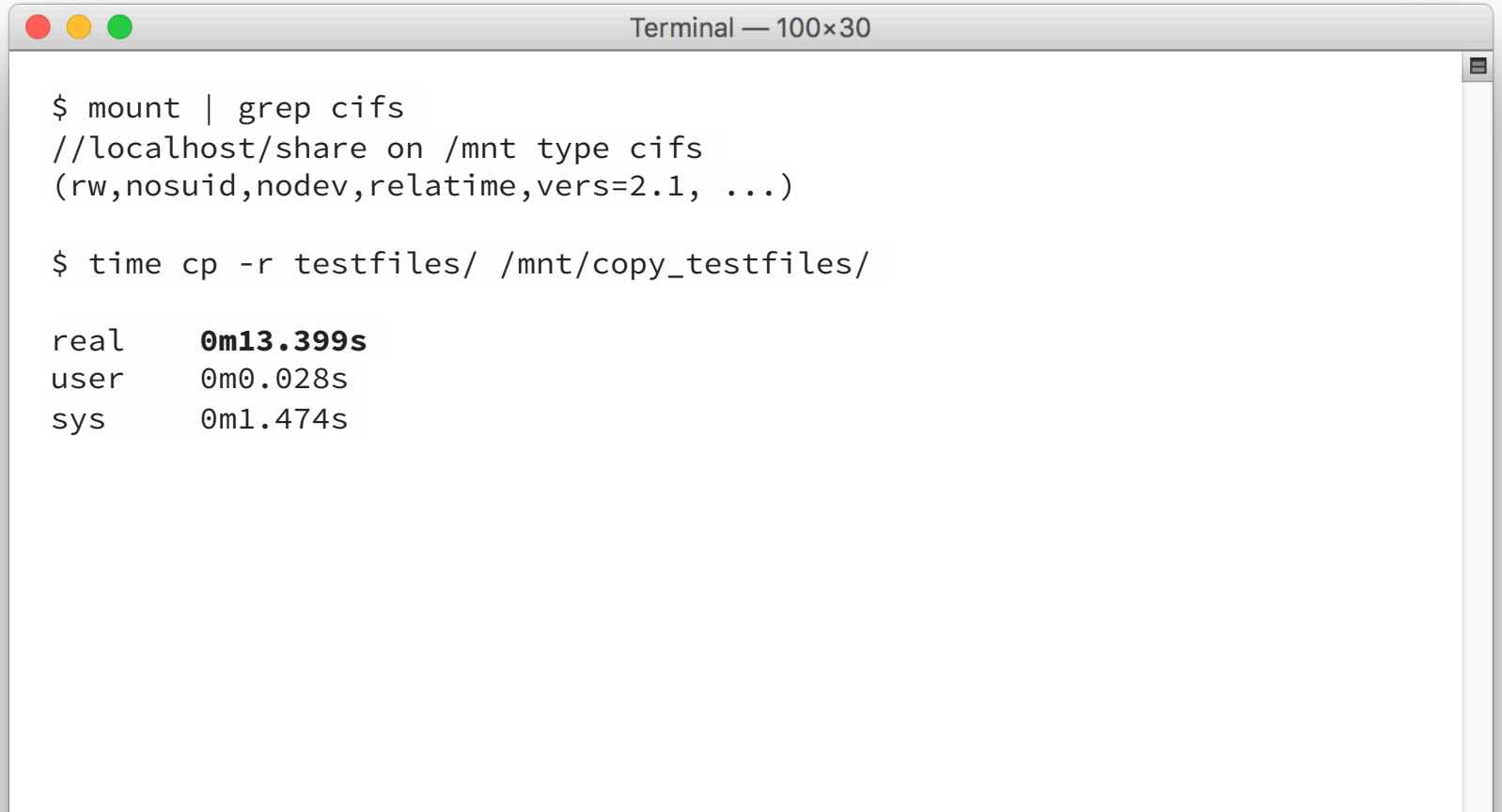
- Copy to Linux kernel client mount (SMB2)
- Start copy, wait…. and wait…
- pidstat shows smbd using more then 50% CPU:

Terminal — 100×30

```
$ pidstat -p 20182 1
Linux 4.10.12-200.fc25.x86_64+debug (kazak)     04/30/2017      _x86_64_

03:46:13 PM   UID       PID    %usr %system  %guest    %CPU   CPU  Command
03:46:14 PM  1000     20182   22.00   38.00    0.00   60.00     2  smbd
03:46:15 PM  1000     20182   23.00   37.00    0.00   60.00     2  smbd
03:46:16 PM  1000     20182   25.00   36.00    0.00   61.00     2  smbd
03:46:17 PM  1000     20182   22.00   39.00    0.00   61.00     2  smbd
03:46:18 PM  1000     20182   26.00   32.00    0.00   58.00     2  smbd
03:46:19 PM  1000     20182   28.00   30.00    0.00   58.00     3  smbd
03:46:20 PM  1000     20182   23.00   33.00    0.00   56.00     3  smbd
03:46:21 PM  1000     20182   21.00   29.00    0.00   50.00     3  smbd
03:46:22 PM  1000     20182   18.00   32.00    0.00   50.00     1  smbd
^C
Average:     1000     20182   23.11   34.00    0.00   57.11     -  smbd
```

1.1. CASE STUDY

- takes 13 seconds, so roughly 50x slower:



1.1. CASE STUDY

1.1. CASE STUDY

Some background info on the system under test:

- Idle system, no other noticeable CPU consumers
- Linux system in Virtualbox, 4 cores
- SSD disk
- mount over loopback network device

1.1. CASE STUDY

- What the hell is smbd is doing:

  1. Processing data (and so using the CPU)

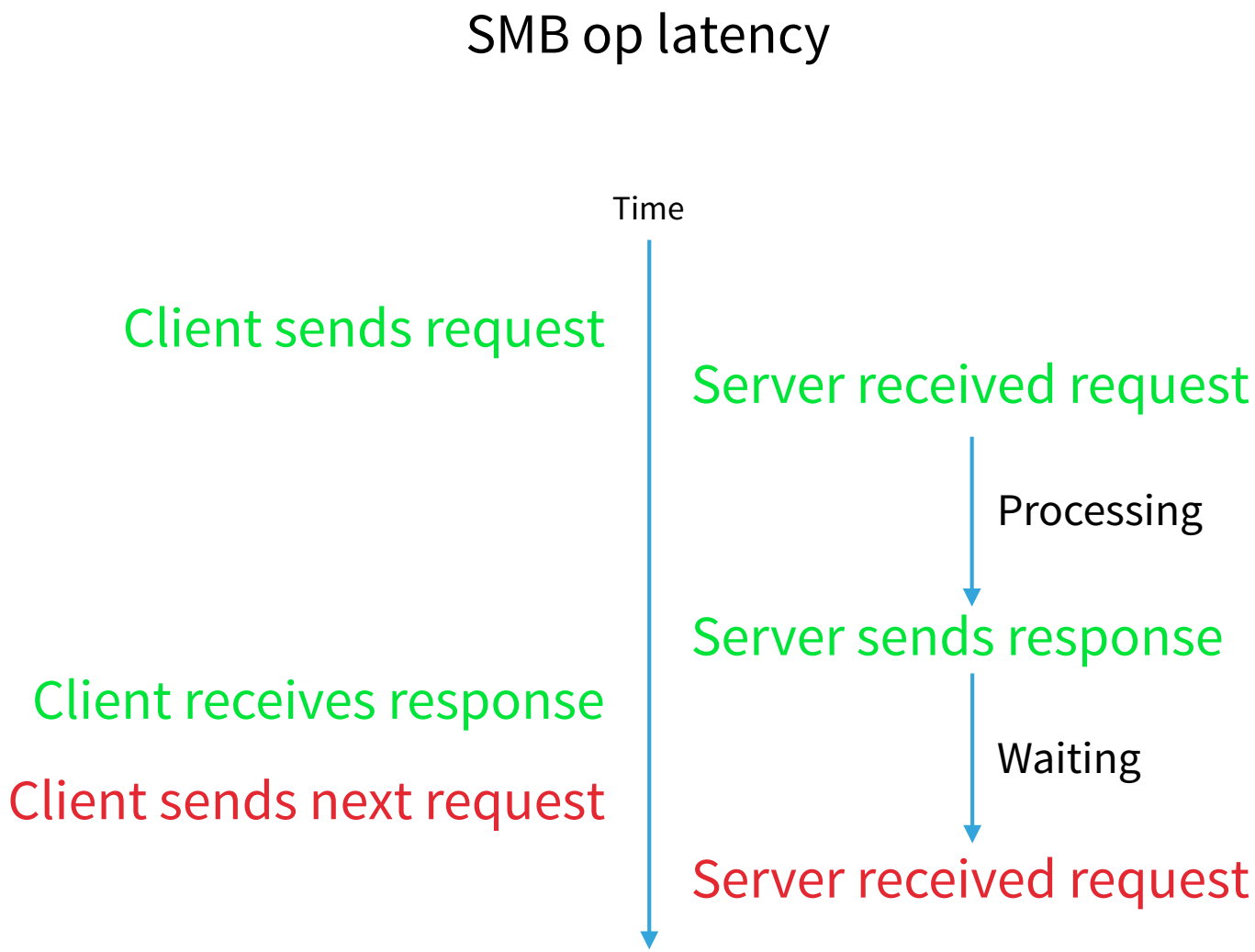
  2. Waiting for data (disk or network)

- And it turns out in this case we're actually 50% waiting

- 50% of 13s means 6.5 s. Good heavens!

- SMB(2): request / response protocol and for many operations the client needs the result before proceeding (eg opening a file handle) and can't progress ⇒ latency

1.1. CASE STUDY



1.1. CASE STUDY

- The client *could* send other requests, but it doesn't
- So this explains half of the desaster
- Oh, and remember: I'm interested in per client throughput, not system throughput – for system throughput just add more clients (and possibly more CPUs)
- Is there another way of inferring wait time without interpolating from top/pidstat?
- Let's look at the performance metrics tools available on Linux (this is 2017, not 2007)

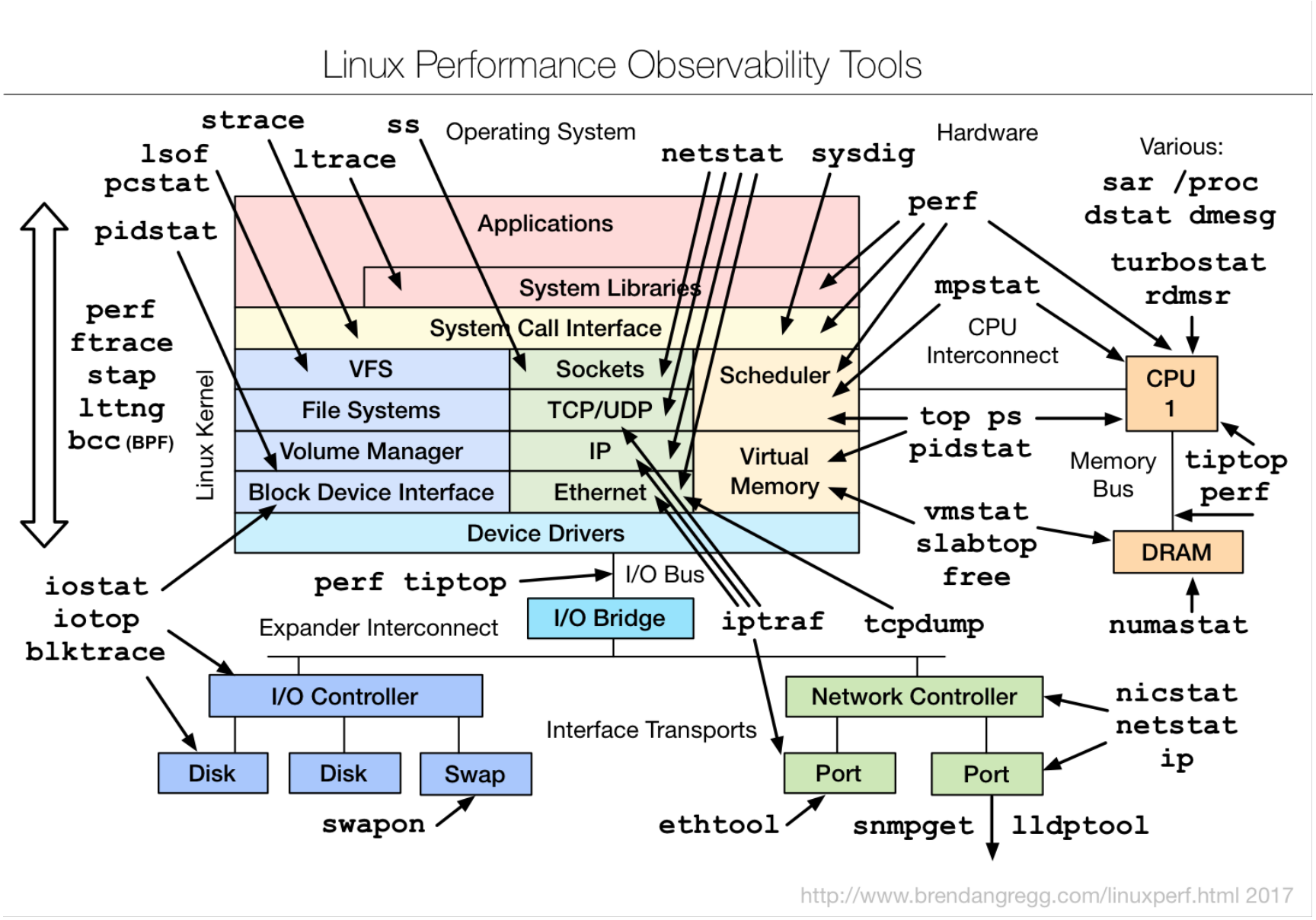
SerNet

It's Tool Time!

LINUX PERFORMANCE OBSERVABILITY TOOLS

2. TOOLS & KERNEL FRAMEWORKS

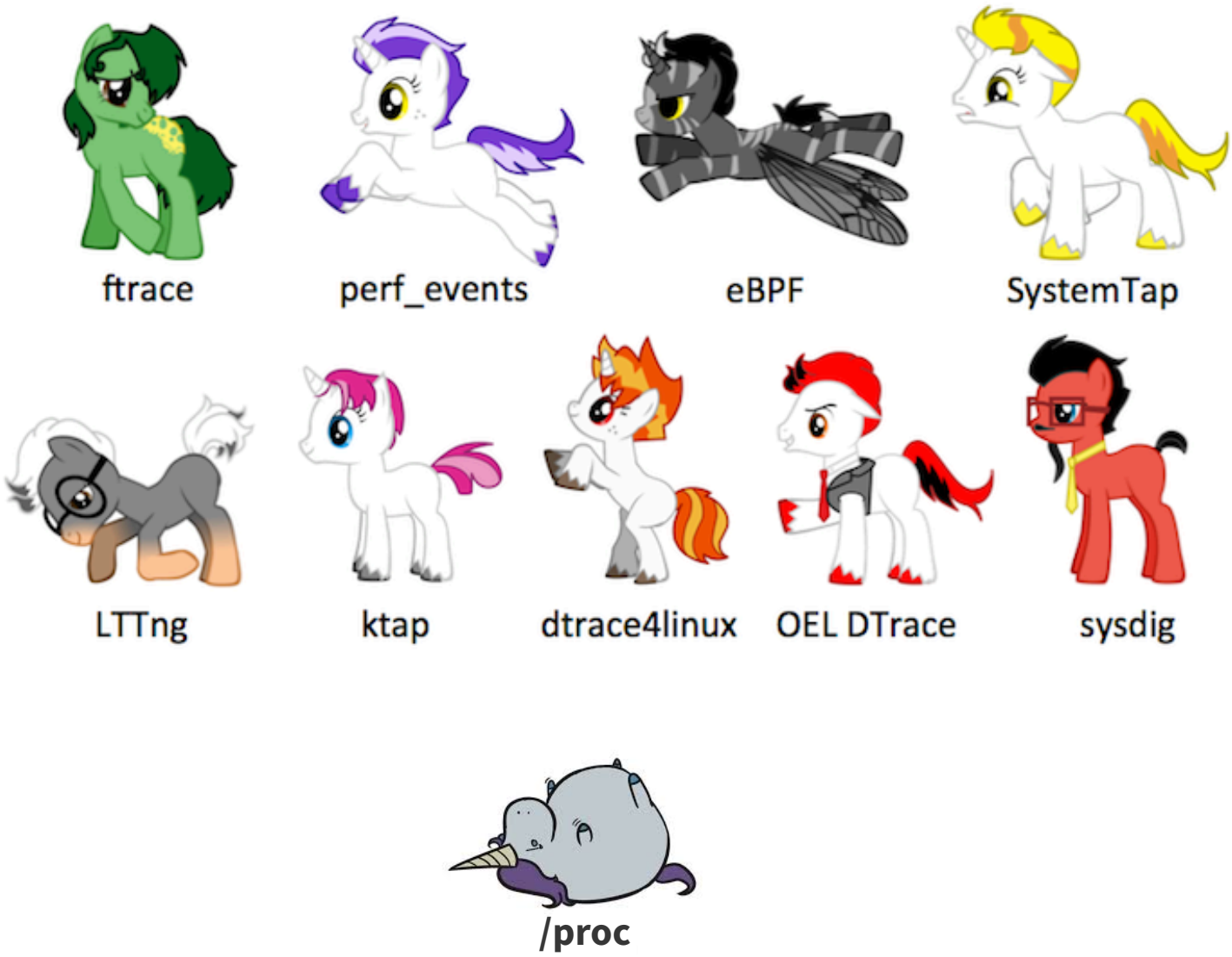
SerNet



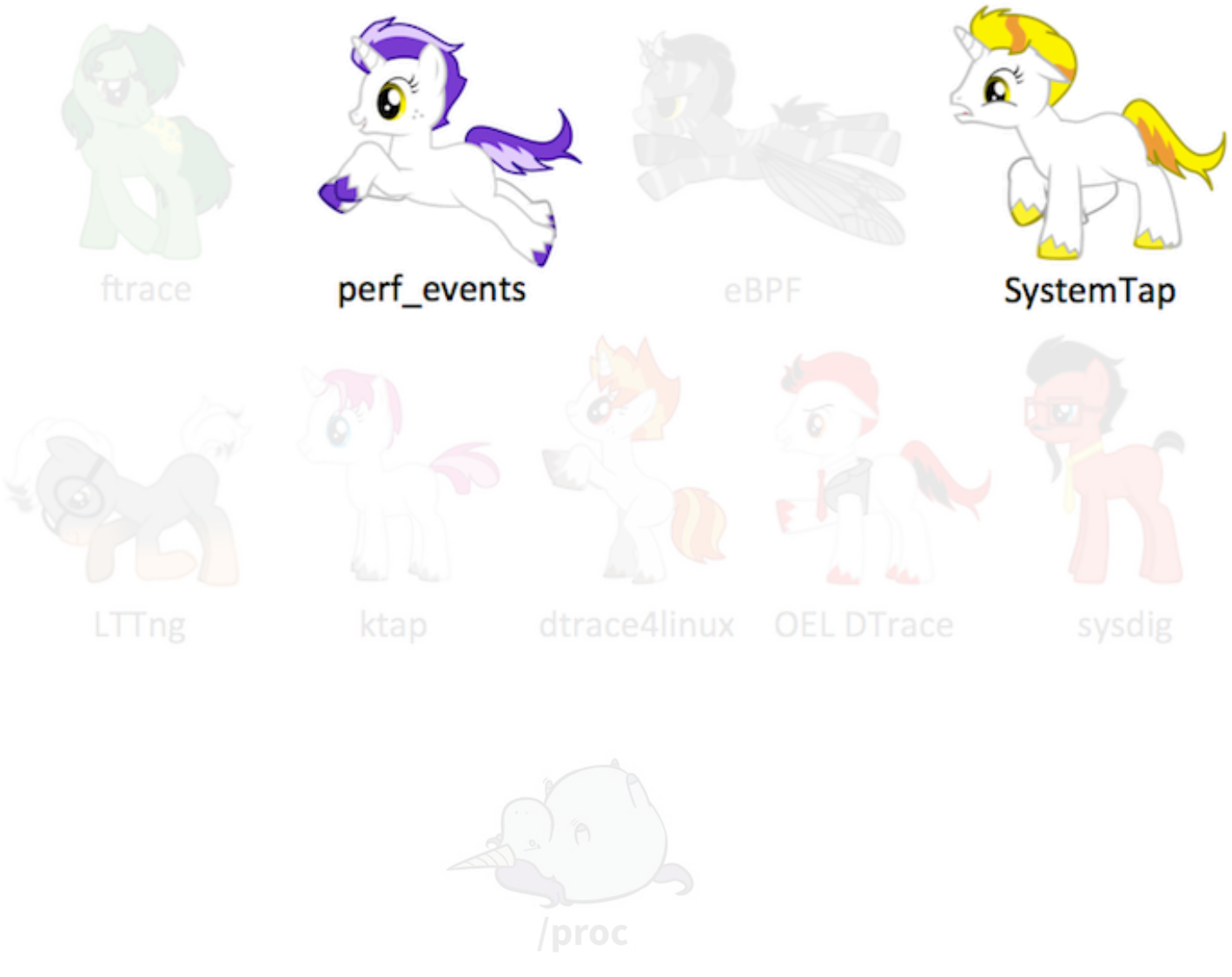
2. KERNEL FRAMEWORKS: OLD

/proc

2. KERNEL FRAMEWORKS: ALL



2. KERNEL FRAMEWORKS: NEW



2.1. /PROC BASED TOOLS

- /proc based tools, lets try one
- Waiting for network or disk can be traced with strace, example:

2.1. /PROC BASED TOOLS

- Quite nice, but copy time goes up the roof: 150 s
- Even though tracing is limited: pread/pwrite, readv/writev and epoll_wait
- iow: strace sucks (hey Jeremy!), but at least it's documented in man strace(1): BUGS: A traced process runs slowly.

Terminal — 100×30

```
$ sudo strace -cf -e pread64,pwrite64,epoll_wait,readv,writev -p 27376
strace: Process 27376 attached
strace: Process 29649 attached
^Cstrace: Process 27376 detached
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 41.21    1.210760          23     51837           epoll_wait
 32.24    0.947304          11     83600           readv
 20.17    0.592665          14     41835           writev
  6.38    0.187556          19     10000           pwrite64
------ ----------- ----------- --------- --------- ----------------
100.00    2.938285                187272           total
```

2.1. /PROC BASED TOOLS

- Anything else?
- Requirement: near zero overhead tracer, at least for syscalls, but more (user-space functions) is better
- The answer (at least on Linux):

2.2. SYSTEMTAP

- „SystemTap provides a simple command line interface and scripting language for writing instrumentation for a live running kernel *plus* user-space applications.“
- „The essential idea behind … systemtap … is to name *events*, and to give them *handlers*. Whenever a specified event occurs, the Linux kernel runs the handler.“
- *You* write the handlers in the Systemtap script language (C like, but safe)
- Can trace everything and profile data can be aggregated any way you like
- Requires kernel-debuginfo for diving into the kernel, but (hopefully) not needed for instrumenting user-space with static-probes

2.2. SYSTEMTAP

So two things:

1. Trace points / events / probes:
   - Trace points are no-ops when not enabled, low overhead when enabled
   - Many existing trace points in the kernel (man stapprobes)
   - For user-space applications add „static probes“ in the source code
   - With DWARF debug symbols it’s even possible to use source lines as trace points
2. Scripts in the Systemtap script language
   - Scripts provide handlers for events
   - Scripts are translated to C and then compiled to create a kernel module
   - When the module is loaded, it activates all the requested events

2.2. SYSTEMTAP

So lets apply this to smbd and add static probes.

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
commit dc82855eedf84fb963cf71ddf2e2f5e9343d6426
Author:     Ralph Boehme <slow@samba.org>
AuthorDate: Sun Apr 23 11:53:47 2017 +0200
Commit:     Ralph Boehme <slow@samba.org>
CommitDate: Fri Apr 28 11:34:30 2017 +0200

    s3/smbd: add SMB2 request probe points

---
 source3/smbd/smb2_server.c | 9 +++++++++
 1 file changed, 9 insertions(+)

diff --git a/source3/smbd/smb2_server.c b/source3/smbd/smb2_server.c
index cf5be6a..e00d1a4 100644
--- a/source3/smbd/smb2_server.c
+++ b/source3/smbd/smb2_server.c
@@ -32,6 +32,8 @@
 #include "auth.h"
 #include "lib/crypto/sha512.h"

+#include <sys/sdt.h>
+
 static void smbd_smb2_connection_handler(struct tevent_context *ev,
                                          struct tevent_fd *fde,
                                          uint16_t flags,
```

2.2. SYSTEMTAP

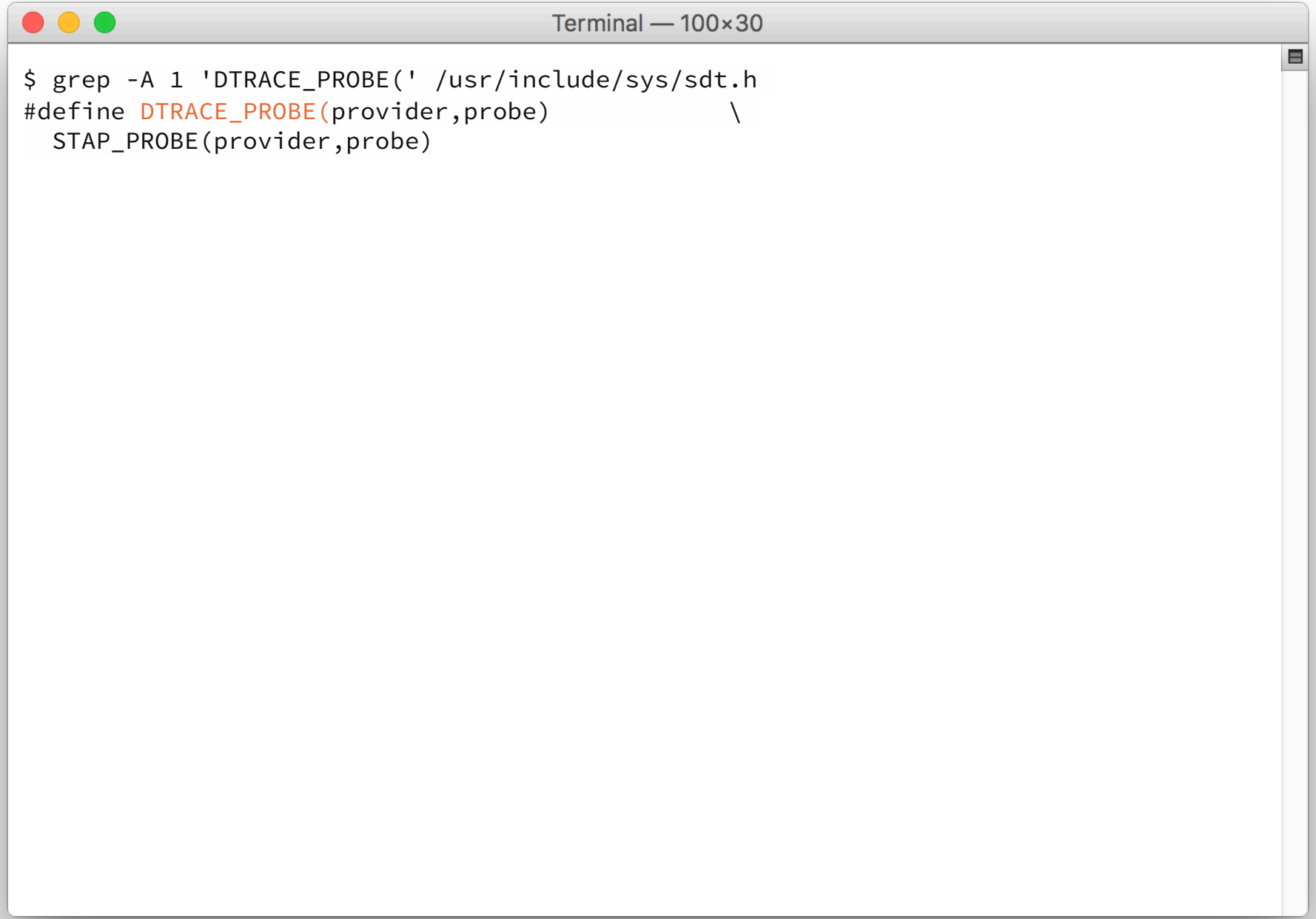
SerNet

Terminal — 100×30

```
@@ -2294,6 +2296,7 @@ NTSTATUS smbd_smb2_request_dispatch(struct smbd_smb2_request *req)
                }
        }

+       DTRACE_PROBE2(smb2, request_start, mid, opcode);
        /*
         * Check if the client provided a valid session id,
         * if so smbd_smb2_request_check_session() calls
@@ -2998,6 +3001,8 @@ NTSTATUS smbd_smb2_request_done_ex(struct smbd_smb2_request *req,
        struct iovec *outbody_v;
        struct iovec *outdyn_v;
        uint32_t next_command_ofs;
+       uint64_t mid;
+       int16_t opcode;

        DEBUG(10,("smbd_smb2_request_done_ex: "
                  "idx[%d] status[%s] body[%u] dyn[%s:%u] at %s\n",
@@ -3018,6 +3023,10 @@ NTSTATUS smbd_smb2_request_done_ex(struct smbd_smb2_request *req,
        outbody_v = SMBD_SMB2_OUT_BODY_IOV(req);
        outdyn_v = SMBD_SMB2_OUT_DYN_IOV(req);

+       mid = BVAL(outhdr, SMB2_HDR_MESSAGE_ID);
+       opcode = SVAL(outhdr, SMB2_HDR_OPCODE);
+       DTRACE_PROBE2(smb2, request_end, mid, opcode);
+
        next_command_ofs = IVAL(outhdr, SMB2_HDR_NEXT_COMMAND);
        SIVAL(outhdr, SMB2_HDR_STATUS, NT_STATUS_V(status));
```

2.2. SYSTEMTAP

SerNet



2.2. SYSTEMTAP

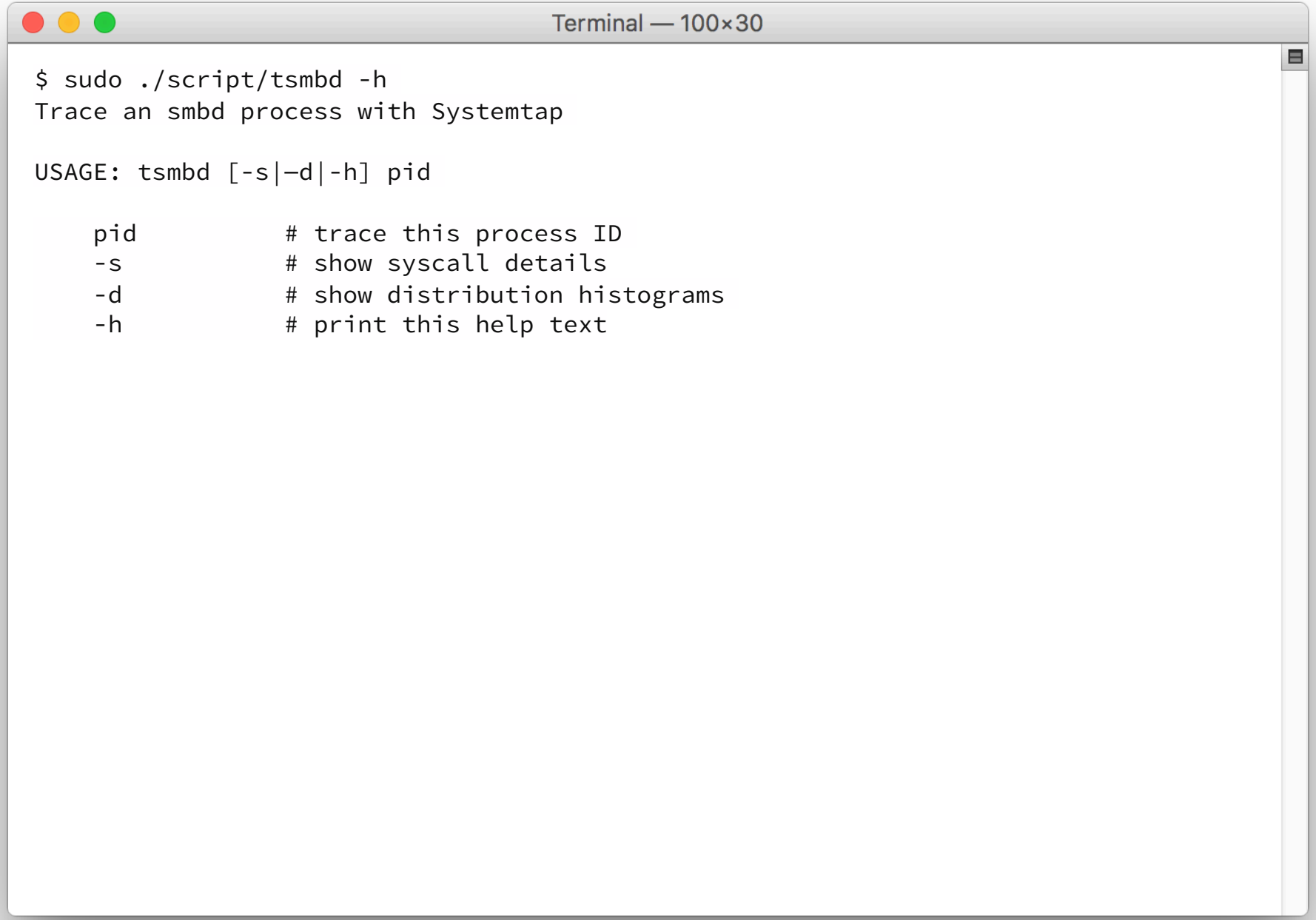
While at it, let's also add instrumentation for these:

1. tevent (epoll_wait() in the epoll backend, could use syscall tracing instead)
2. sending / receiving data from the network
3. syscalls (so we don't have to use bloody strace)
4. smbd ⟺ ctdb communication latency

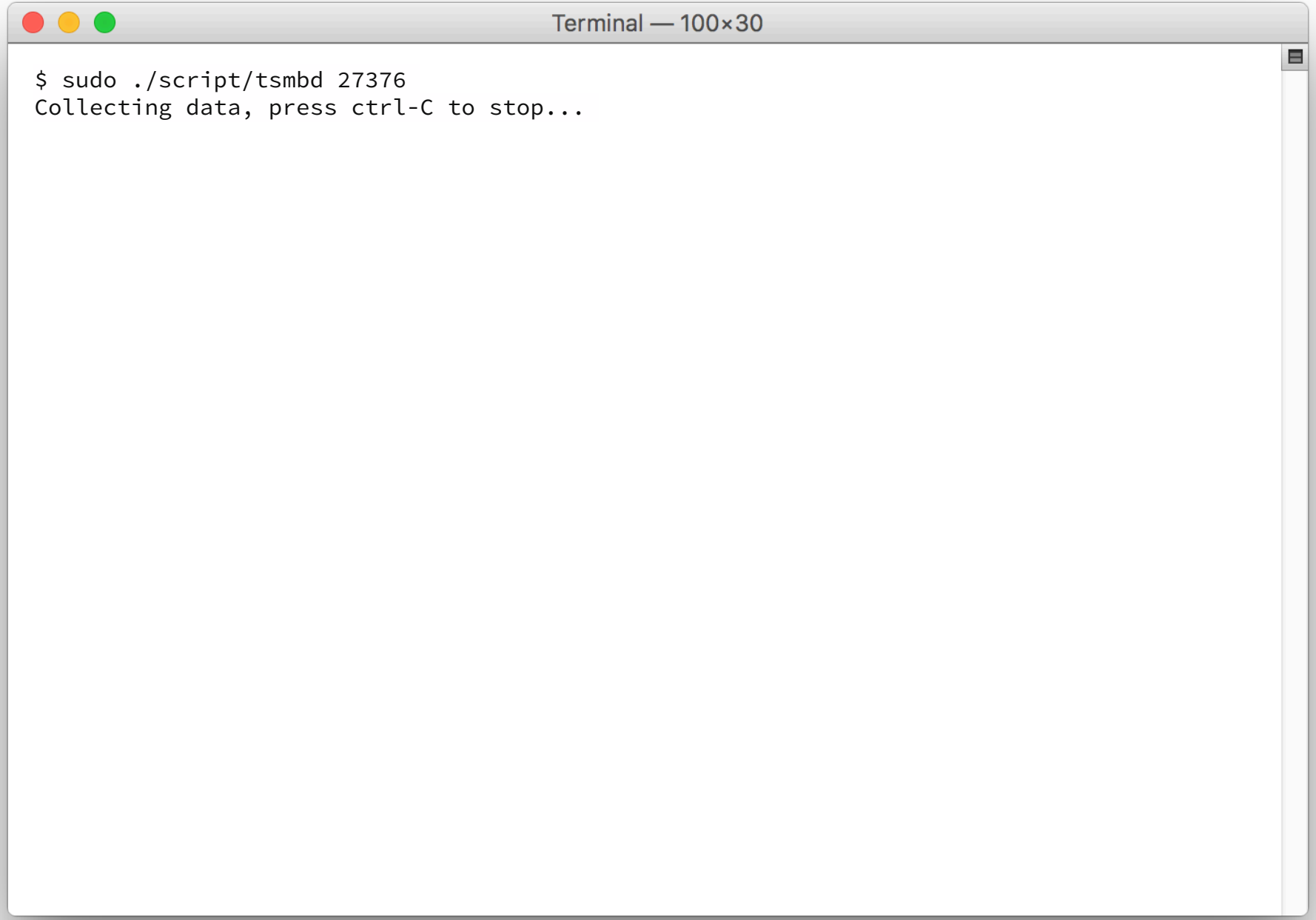
Let me introduce you to `tsmbd`:

2.2. SYSTEMTAP

SerNet



2.2. SYSTEMTAP



2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                              17147 ms

Time waiting for events:               9302 ms
Time receiving SMB2 packets:            224 ms
Time parsing SMB2 packets:               92 ms
Time running SMB2 requests:            5744 ms
Time sending SMB2 packets:             1508 ms
Time waiting for ctdb:                    0 ms
=========================================
Time sum:                             16871 ms

Time in syscalls:                      2307 ms
Time in READ disk IO:                     0 ms
Time in WRITE disk IO:                  167 ms

Number of tevent events:              51214
Number of SMB2 requests:              41212
Number of ctdb requests:                  0

SMB2 Requests                    Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                  10201       311821            30            15           603
SMB2_OP_CREATE                   10507      3137339           298            80          6985
SMB2_OP_CLOSE                    10403      1189089           114            23          2931
SMB2_OP_SETINFO                    101         3717            36            29           199
SMB2_OP_WRITE                    10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                            17147 ms

Time waiting for events:             9302 ms
Time receiving SMB2 packets:          224 ms
Time parsing SMB2 packets:             92 ms
Time running SMB2 requests:          5744 ms
Time sending SMB2 packets:           1508 ms
Time waiting for ctdb:                  0 ms
=========================================
Time sum:                           16871 ms

Time in syscalls:                    2307 ms
Time in READ disk IO:                   0 ms
Time in WRITE disk IO:                167 ms

Number of tevent events:            51214
Number of SMB2 requests:            41212
Number of ctdb requests:                0

SMB2 Requests                   Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                 10201       311821            30            15           603
SMB2_OP_CREATE                  10507      3137339           298            80          6985
SMB2_OP_CLOSE                   10403      1189089           114            23          2931
SMB2_OP_SETINFO                   101         3717            36            29           199
SMB2_OP_WRITE                   10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                              17147 ms

Time waiting for events:               9302 ms
Time receiving SMB2 packets:            224 ms
Time parsing SMB2 packets:               92 ms
Time running SMB2 requests:            5744 ms
Time sending SMB2 packets:             1508 ms
Time waiting for ctdb:                    0 ms
=========================================
Time sum:                             16871 ms

Time in syscalls:                      2307 ms
Time in READ disk IO:                     0 ms
Time in WRITE disk IO:                  167 ms

Number of tevent events:              51214
Number of SMB2 requests:              41212
Number of ctdb requests:                  0

SMB2 Requests                    Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                  10201       311821            30            15           603
SMB2_OP_CREATE                   10507      3137339           298            80          6985
SMB2_OP_CLOSE                    10403      1189089           114            23          2931
SMB2_OP_SETINFO                    101         3717            36            29           199
SMB2_OP_WRITE                    10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                          17147 ms

Time waiting for events:           9302 ms
Time receiving SMB2 packets:        224 ms
Time parsing SMB2 packets:           92 ms
Time running SMB2 requests:        5744 ms
Time sending SMB2 packets:         1508 ms
Time waiting for ctdb:                0 ms
========================================
Time sum:                         16871 ms

Time in syscalls:                  2307 ms
Time in READ disk IO:                 0 ms
Time in WRITE disk IO:              167 ms

Number of tevent events:          51214
Number of SMB2 requests:          41212
Number of ctdb requests:              0

SMB2 Requests                  Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                10201       311821            30            15           603
SMB2_OP_CREATE                 10507      3137339           298            80          6985
SMB2_OP_CLOSE                  10403      1189089           114            23          2931
SMB2_OP_SETINFO                  101         3717            36            29           199
SMB2_OP_WRITE                  10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                              17147 ms

Time waiting for events:               9302 ms
Time receiving SMB2 packets:            224 ms
Time parsing SMB2 packets:               92 ms
Time running SMB2 requests:            5744 ms
Time sending SMB2 packets:             1508 ms
Time waiting for ctdb:                    0 ms
=========================================
Time sum:                             16871 ms

Time in syscalls:                      2307 ms
Time in READ disk IO:                     0 ms
Time in WRITE disk IO:                  167 ms

Number of tevent events:              51214
Number of SMB2 requests:              41212
Number of ctdb requests:                  0

SMB2 Requests                    Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                  10201       311821            30            15           603
SMB2_OP_CREATE                   10507      3137339           298            80          6985
SMB2_OP_CLOSE                    10403      1189089           114            23          2931
SMB2_OP_SETINFO                    101         3717            36            29           199
SMB2_OP_WRITE                    10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                              17147 ms

Time waiting for events:               9302 ms
Time receiving SMB2 packets:            224 ms
Time parsing SMB2 packets:               92 ms
Time running SMB2 requests:            5744 ms
Time sending SMB2 packets:             1508 ms
Time waiting for ctdb:                    0 ms
=========================================
Time sum:                             16871 ms

Time in syscalls:                      2307 ms
Time in READ disk IO:                     0 ms
Time in WRITE disk IO:                  167 ms

Number of tevent events:              51214
Number of SMB2 requests:              41212
Number of ctdb requests:                  0

SMB2 Requests                    Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                  10201       311821            30            15           603
SMB2_OP_CREATE                   10507      3137339           298            80          6985
SMB2_OP_CLOSE                    10403      1189089           114            23          2931
SMB2_OP_SETINFO                    101         3717            36            29           199
SMB2_OP_WRITE                    10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                              17147 ms

Time waiting for events:               9302 ms
Time receiving SMB2 packets:            224 ms
Time parsing SMB2 packets:               92 ms
Time running SMB2 requests:            5744 ms
Time sending SMB2 packets:             1508 ms
Time waiting for ctdb:                    0 ms
=========================================
Time sum:                             16871 ms

Time in syscalls:                      2307 ms
Time in READ disk IO:                     0 ms
Time in WRITE disk IO:                  167 ms

Number of tevent events:              51214
Number of SMB2 requests:              41212
Number of ctdb requests:                  0

SMB2 Requests                    Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                  10201       311821            30            15           603
SMB2_OP_CREATE                   10507      3137339           298            80          6985
SMB2_OP_CLOSE                    10403      1189089           114            23          2931
SMB2_OP_SETINFO                    101         3717            36            29           199
SMB2_OP_WRITE                    10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                            17147 ms

Time waiting for events:             9302 ms
Time receiving SMB2 packets:          224 ms
Time parsing SMB2 packets:             92 ms
Time running SMB2 requests:          5744 ms
Time sending SMB2 packets:           1508 ms
Time waiting for ctdb:                  0 ms
=========================================
Time sum:                           16871 ms

Time in syscalls:                    2307 ms
Time in READ disk IO:                   0 ms
Time in WRITE disk IO:                167 ms

Number of tevent events:            51214
Number of SMB2 requests:            41212
Number of ctdb requests:                0

SMB2 Requests                   Count     Total us        Avg us        Min us        Max us
SMB2_OP_GETINFO                 10201       311821            30            15           603
SMB2_OP_CREATE                  10507      3137339           298            80          6985
SMB2_OP_CLOSE                   10403      1189089           114            23          2931
SMB2_OP_SETINFO                   101         3717            36            29           199
SMB2_OP_WRITE                   10000      1081732           108            52          3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
$ sudo ./script/tsmbd 27376
Collecting data, press ctrl-C to stop... ^C

Ran for:                            17147 ms

Time waiting for events:             9302 ms
Time receiving SMB2 packets:          224 ms
Time parsing SMB2 packets:             92 ms
Time running SMB2 requests:          5744 ms
Time sending SMB2 packets:           1508 ms
Time waiting for ctdb:                  0 ms
=========================================
Time sum:                           16871 ms

Time in syscalls:                    2307 ms
Time in READ disk IO:                   0 ms
Time in WRITE disk IO:                167 ms

Number of tevent events:            51214
Number of SMB2 requests:            41212
Number of ctdb requests:                0

SMB2 Requests                  Count      Total us          Avg us          Min us          Max us
SMB2_OP_GETINFO                10201        311821              30              15             603
SMB2_OP_CREATE                 10507       3137339             298              80            6985
SMB2_OP_CLOSE                  10403       1189089             114              23            2931
SMB2_OP_SETINFO                  101          3717              36              29             199
SMB2_OP_WRITE                  10000       1081732             108              52            3614
```

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

…continued…

System call	Count	Total us
geteuid	16	7
getegid	15	5
setgroups	5	10
setresgid	7	8
setresuid	7	12
stat	215373	256907
getcwd	50989	26160
open	20205	312274
fstat	50206	45133
getdents	20410	175707
close	20207	15967
lstat	61208	52244
read	42021	58799
fcntl	470178	408604
getxattr	62719	201745
mkdir	101	4191
setxattr	10101	136778
getgroups	2	1
sendmsg	30202	298217
listxattr	201	528
chdir	40400	38939
fgetxattr	10000	10658
clock_gettime	20001	17726
pwrite	10014	159469
write	10000	40025
utimensat	10000	47363

2.2. SYSTEMTAP

SerNet

Terminal — 100×30

```
…
SMB2_OP_CREATE distribution (microseconds)
value |------------------------------------------------- count
   16 |                                                    0
   32 |                                                    0
   64 |@                                                 289
  128 |@@@@                                              846
  256 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 9315
  512 |                                                   48
 1024 |                                                    7
 2048 |                                                    0
 4096 |                                                    1
 8192 |                                                    2
16384 |                                                    0
32768 |                                                    0
…
WRITE IO distribution (microseconds)
value |------------------------------------------------- count
    2 |                                                    0
    4 |                                                    0
    8 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 5913
   16 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@                  3909
   32 |@                                                 149
   64 |                                                   27
  128 |                                                    0
  256 |                                                    0
  512 |                                                    1
 1024 |                                                    0
 2048 |                                                    1
 4096 |                                                    0
 8192 |                                                    0
…
```

2.2. SYSTEMTAP

Systemtap summary:

- Low overhead, not completely free, but cheap
  - not traced: 13 s
  - traced: 14 s
- It would be easy to enhance tsmbd to trace all smbd (as an option)
- But this may turn out to be not so cheap anymore :-)

2.2. SYSTEMTAP

2.2. SYSTEMTAP

Because:

1. 20% network IO and SMB2 packet processing

2. 40% waiting for requests

3. 40% CPU usage for SMB2 fileserver engine

SerNet

LINUX PERF

perf_events

2.3. PERF

- Remember 50% CPU over 13 seconds
- compared to ~100% over 0.3 seconds for local copy
- So a tremendous computation overhead, smbd is doing a *lot*
- But what is it doing?
- tsmbd told us 30% CPU is in the kernel servicing syscalls
- the rest is user-space CPU time (20%)
- So let’s eliminate syscalls and optimise our code
- But where? We need profiling info

2.3. PERF

- Linux profilers:
  1. GNU gprof: requires special compilation
  2. Valgrind Callgrind: slooooooooooooooow
  3. oprofile
  4. …or…

2.3. PERF

- `perf`: a kernel subsystem(s) and a user-space tool
- it can instrument CPU performance counters, tracepoints, kprobes, and uprobes (go read up on it on the web, I'm not going to explain that here)
- Put it differently: like Systemtap without the scripts
- It aggregates metrics and dumps them in a file:
  # perf record [-p pid]
- You then use the user-space tool perf to print the data
  # perf report[-p pid]
- Can sample the stack of processes, including kernel stack
  # perf record **-g**

2.3. PERF

- It has an awesome interface to display the stack profile info:
  # perf report -g …

Example:

2.3. PERF

SerNet

Terminal — 100×30

```
Samples: 2K of event 'cpu-clock', Event count (approx.): 685500000
  Children      Self  Command  Shared Object                Symbol
+   52.48%     0.15%  smbd     libtevent.so                 [.] epoll_event_loop
+   52.44%     0.00%  smbd     libsmbd-base-samba4.so       [.] smbd_process
+   52.44%     0.00%  smbd     smbd                         [.] smbd_accept_connection
+   52.44%     0.22%  smbd     libtevent.so                 [.] epoll_event_loop_once
+   52.44%     0.18%  smbd     libtevent.so                 [.] std_event_loop_once
+   52.44%     0.22%  smbd     libtevent.so                 [.] _tevent_loop_once
+   52.44%     0.00%  smbd     libtevent.so                 [.] tevent_common_loop_wait
+   52.44%     0.00%  smbd     libtevent.so                 [.] std_event_loop_wait
+   52.44%     0.00%  smbd     libtevent.so                 [.] _tevent_loop_wait
+   52.44%     0.00%  smbd     smbd                         [.] smbd_parent_loop
+   52.44%     0.00%  smbd     smbd                         [.] main
+   52.44%     0.00%  smbd     libc-2.24.so                 [.] __libc_start_main
+   52.44%     0.00%  smbd     smbd                         [.] _start
+   49.12%     0.66%  smbd     [kernel.vmlinux]             [k] entry_SYSCALL_64_fastpath
+   33.95%     0.00%  smbd     libsmbd-base-samba4.so       [.] smbd_smb2_connection_handler
+   33.84%     0.26%  smbd     libsmbd-base-samba4.so       [.] smbd_smb2_io_handler
+   32.79%     0.15%  smbd     libsmbd-base-samba4.so       [.] smbd_smb2_request_dispatch
+   20.75%     0.00%  smbd     libsmbd-base-samba4.so       [.] smbd_smb2_request_process_create
+   18.23%     0.07%  smbd     libsmbd-base-samba4.so       [.] smbd_smb2_create_send
+   12.84%     0.07%  smbd     [kernel.vmlinux]             [k] do_readv_writev
+   12.80%     0.00%  smbd     libsmbd-base-samba4.so       [.] create_file_default
+   12.80%     0.00%  smbd     libsmbd-base-samba4.so       [.] vfswrap_create_file
+   12.80%     0.00%  smbd     libsmbd-base-samba4.so       [.] smb_vfs_call_create_file
+   12.76%     0.04%  smbd     libsmbd-base-samba4.so       [.] create_file_unixpath
+   12.47%     0.00%  smbd     [kernel.vmlinux]             [k] sock_sendmsg
+   12.33%     0.07%  smbd     [kernel.vmlinux]             [k] do_iter_readv_writev
+   11.82%    11.82%  smbd     [kernel.vmlinux]             [k] _raw_spin_unlock_irqrestore
+   11.71%     0.00%  smbd     [kernel.vmlinux]             [k] do_writev
+   11.71%     0.00%  smbd     [kernel.vmlinux]             [k] sys_writev
+   11.67%     0.00%  smbd     [kernel.vmlinux]             [k] vfs_writev
+   11.56%     0.00%  smbd     [kernel.vmlinux]             [k] sock_write_iter
```

2.3. PERF

- There's a nice visualization for call-stack profiles: Flamegraphs

2.3. PERF

- Try them out, they are a good starting point
- More on Flamegraphs hopefully in the next talk from Douglas and later by Andrew

PERF DEMO

SerNet

SMBD PERFORMANCE IMPROVEMENTS

3. SMBD PERFORMANCE IMPROVEMENTS

SerNet

1. Clustered Samba: directory enumeration
2. Name mangling: new option „mangled names = illegal“
3. Make use of struct smb_filename plumbing in the 4.5 VFS
4. GPFS VFS module improvements
5. Messaging

3. DIRECTORY ENUMERATION

SerNet

Clustered Samba: directory enumeration improvements

- enumerating a directory requires asking locking.tdb for every directory entry (we want the file write time from a possible open file record)
- on a cluster this goes via ctdb and takes some time
- smbd sits idle waiting for the answer
- the fix: add an async dwrap_fetch_send/recv API
- uses async communication with ctdb at the low level
- smbd can now pipeline the requests and continue processing the directory
- at the end it collects the results
- Time for enumerating directory with 50k file goes from 10 s to 3 s

3. NAME MANGLING = ILLEGAL

SerNet

- man smb.conf: „ mangled names = … controls whether non-DOS names under UNIX should be mapped to DOS-compatible names …“
- any name longer then 8.3 is a non-DOS name
- this means that with SMB2+ we’re still providing mangled names in FIND responses, Windows server doesn't
- calculating the mangled name mapping is awfully expensive
- new option setting: „mangled names = illegal“
- only add mapping for filenames with illegal NTFS characters that still really require mangling

3. STRUCT SMB_FILENAME PLUMBING

SerNet

- as part of the CREATE call we allocate a struct smb_filename
- for existing files „struct smb_filename->SMB_STRUCT_STAT st“ will already contain valid stat info
- in Samba 4.4 many VFS functions weren’t passed the smb_filename but a char * with just the path
- so sometimes VFS modules and parts of smbd had to stat again
- thanks to Jeremy struct smb_filename is now plumbed through the VFS
- so I could remove a few stats in a few places

3. GPFS VFS MODULE / CREATION_DATE

- stat functions in vfs_gpfs were all calling into GPFS to update the file creation time
- stat may be called multiple times from different contexts, every time we call down into GPFS
- but at the top level smbd uses a different VFS call (SMB_VFS_GET_DOS_ATTRIBUTES) to request file creation time from the VFS
- so after adding the creation time hook to the GPFS SMB_VFS_GET_DOS_ATTRIBUTES implementation, remove the calls into GPFS from the stat hooks

3. GPFS VFS MODULE / NFSV4 ACL PARMS

SerNet

- Initialize smb.conf NFSv4 ACL parameters at tree connect (from Volker)
- Without this we call out to smb.conf parsing code for every ACL related VFS function in VFS GPFS which is bad, bad, bad

3. MESSAGING

- for every message smbd processes send to one another we go through the full machinery of connection setup, send message and connection teardown
- but there are connections that could be reused: file change notifications from smbd processes to notifyd
- solution: keep idle connections for 1 s
- this helps a lot for the small file copy workload

3. RESULT

Small file copy throughput:

- before: 136 files / s
- after: 151 files / s
- ~10% improvement just by optimising existing code, no new code

4. LINKS

https://git.samba.org/?p=slow/samba.git;a=shortlog;h=refs/heads/perf

THANK YOU!

QUESTIONS?

Ralph Böhme <slow@samba.org>