Introduction to Networking and Systems Measurements

Advanced Measurements

Dr Andrew W. Moore

andrew.moore@cl.cam.ac.uk

Dr Noa Zilberman

noa.zilberman@cl.cam.ac.uk

perf

- So far we discussed *performance*
- What about *events?*
- Perf is a Linux profiler tool
- Allows to instrument CPU performance counters, tracepoints and probes (kernel, user)

perf

- list – find events
- stat – count events
- record – write event data to a file
- report – browse summary
- script – event dump for post processing

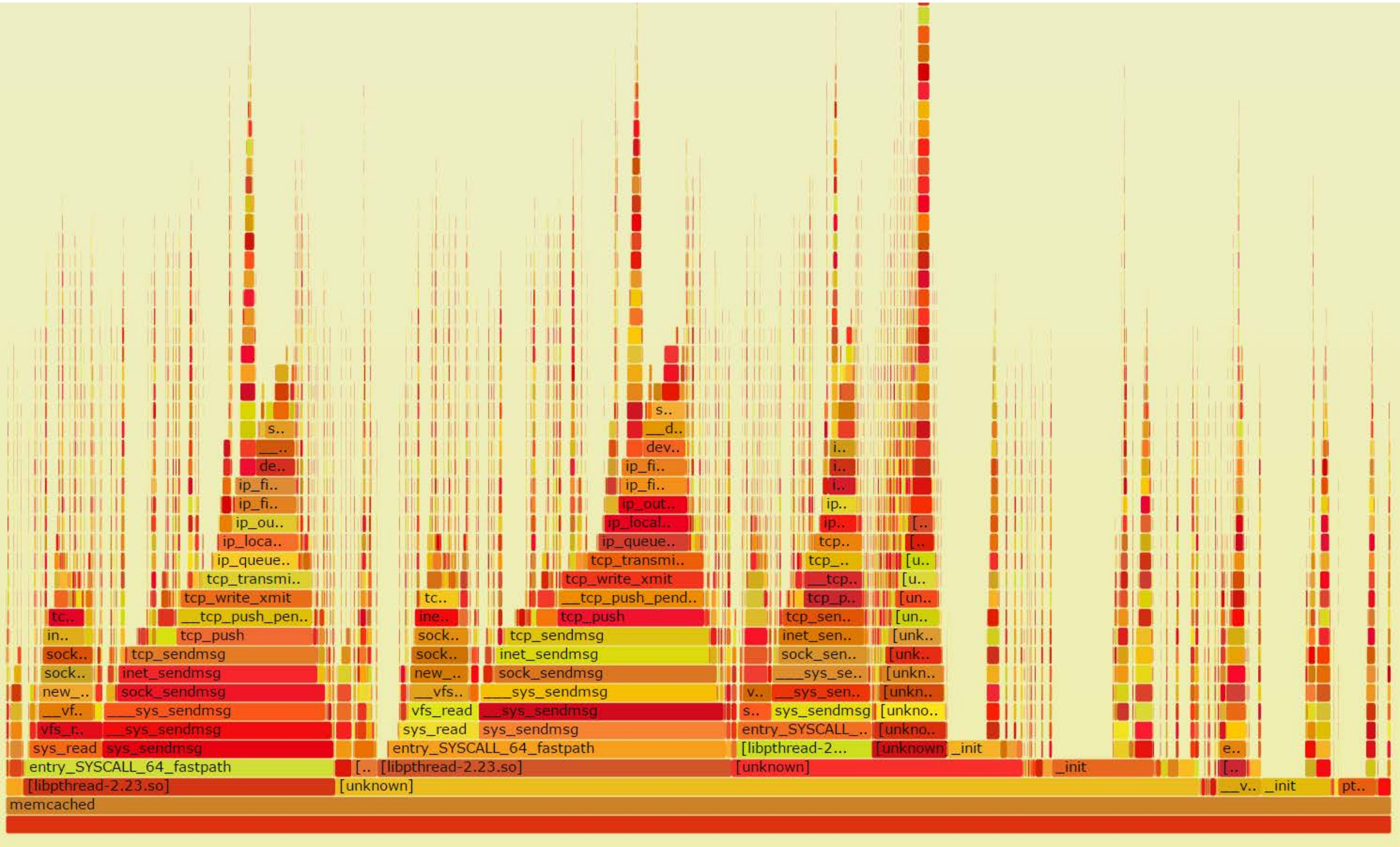
Perf - example

```
:~/.ssh$ perf stat ps
  PID TTY          TIME CMD
 8747 pts/2    00:00:00 bash
11667 pts/2    00:00:00 perf
11670 pts/2    00:00:00 ps

 Performance counter stats for 'ps':

         12.745507   task-clock (msec)         # 0.929 CPUs utilized
                 4   context-switches          # 0.314 K/sec
                 0   cpu-migrations            # 0.000 K/sec
               140   page-faults               # 0.011 M/sec
        32,322,489   cycles                    # 2.536 GHz                 (40.80%)
   <not supported>   stalled-cycles-frontend
   <not supported>   stalled-cycles-backend
        27,644,922   instructions              # 0.86  insns per cycle     (68.86%)
         5,133,583   branches                  # 402.776 M/sec             (68.92%)
           157,503   branch-misses             # 3.07% of all branches     (94.06%)

       0.013726555 seconds time elapsed
```

the tool **scales** the count based on total time enabled vs time running

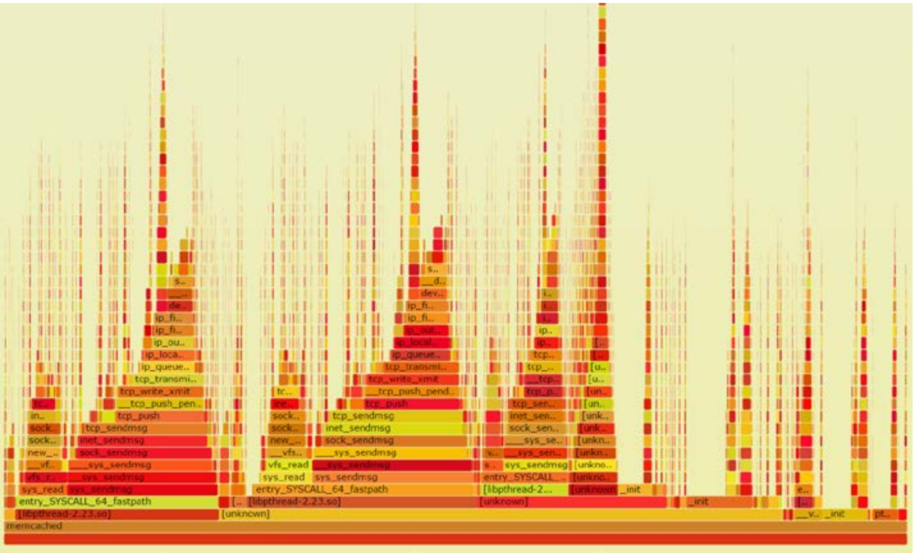
Flame Graphs

- Parsing traces is like finding a needle in a haystack
- Flame graphs - Visualise the outputs of profiling tools
  - E.g., using perf, dtrace
- Easy to understand
- Open source
  - https://github.com/brendangregg/FlameGraph
  - Brendan Gregg has several other useful performance-related tools


memcached
[libpthread-2.23.so]
[unknown]
entry_SYSCALL_64_fastpath
[libpthread-2.23.so]
[unknown]
_init
__v..
pt..
sys_read
sys_sendmsg
vfs_read
__sys_sendmsg
___sys_sendmsg
sock_sendmsg
inet_sendmsg
tcp_sendmsg
tcp_push
__tcp_push_pen..
__tcp_push_pend..
tcp_write_xmit
tcp_transmi..
ip_queue..
ip_loca..
ip_local..
ip_ou..
ip_out..
ip_fi..
de..
dev..
s..
__d..
entry_SYSCALL_..
[libpthread-2...
[unknown]
sys_sendmsg
___sys_sen..
___sys_se..
sock_sen..
inet_sen..
tcp_sen..
tcp_p..
__tcp..
tcp_..
tcp..
ip..
i..
[unkno..
[unkn..
[unk..
[un..
[u..
[..
vfs_r..
__vf..
new_..
sock..
in..
tc..
ine..
__vfs..
e..

Flame Graphs

- Width is relative to “how much running on the CPU”
- Top-down shows ancestry
- Not good for idles – so don’t try to use for profiling network events!
- Different types of flame graphs
  - ➢ E.g. CPU, memory, differential

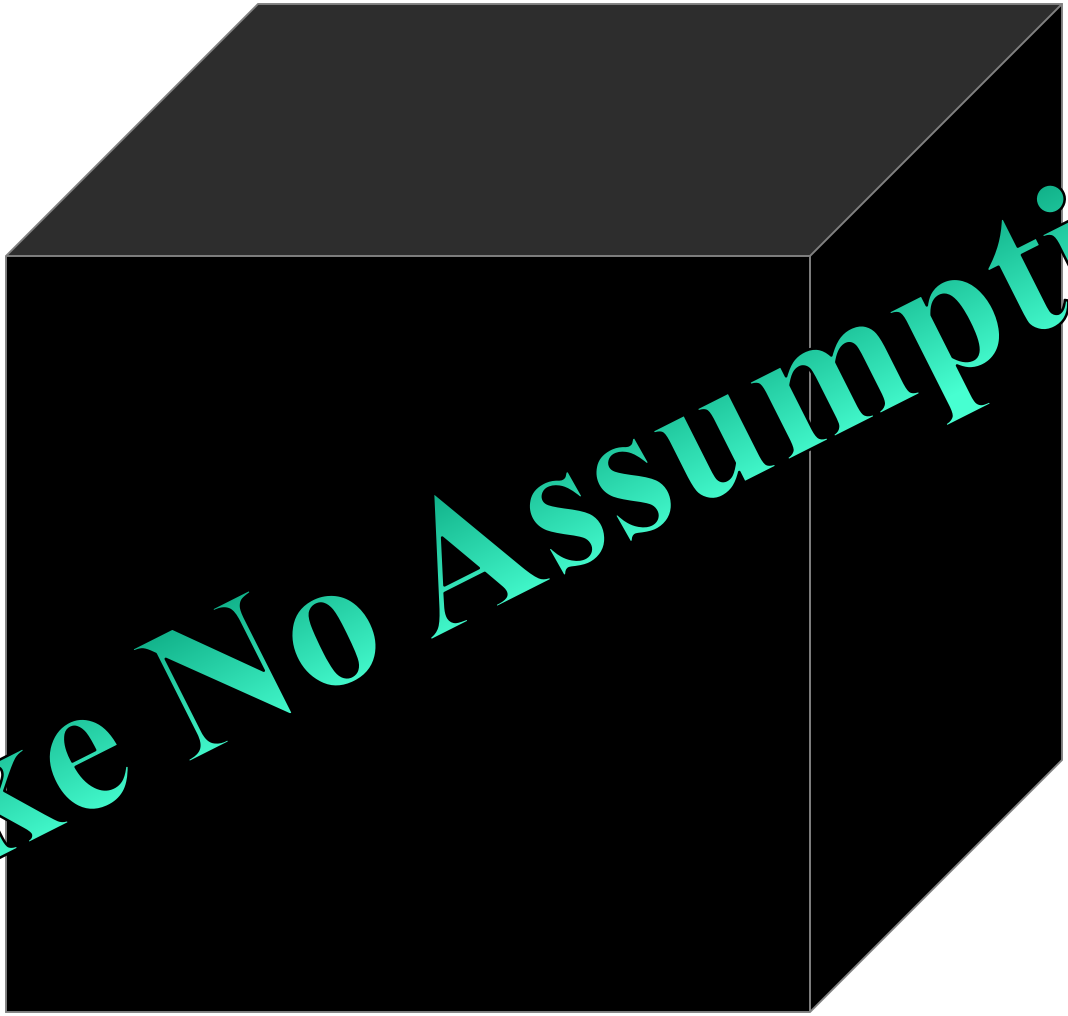


Introduction to Networking and Systems Measurements

Device and System Characterization

Dr Andrew W. Moore
andrew.moore@cl.cam.ac.uk

Dr Noa Zilberman
noa.zilberman@cl.cam.ac.uk



Make No Assumptions

What is the goal?

- Functional validation?
- Performance testing?
- Characterization?
- Comparison?
- Detecting problems?
- Finding the bottlenecks?

Different goals ⇒ different setup + experiments

What is the goal?

- Functional validation, e.g.,:
  - Can we send traffic from port A to port B?
- Performance testing, e.g.,:
  - What is the throughput of sending traffic from port A to port B?

Vantage Points

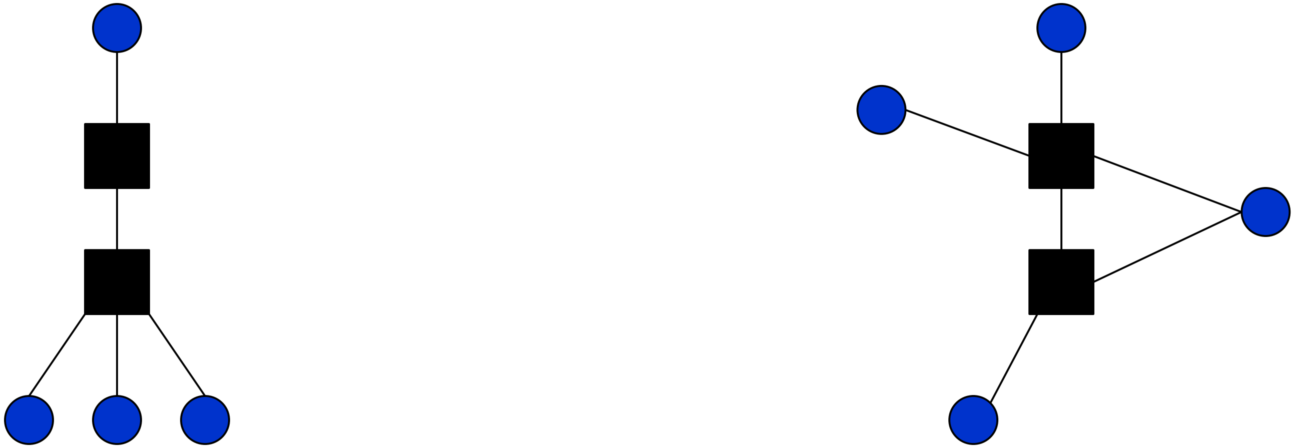
- Characterisation is limited by vantage points
- Single vantage point:
  - Round trip measurements, topology measurements

  OR

  - Passive measurements
- Two vantage points:
  - One way latency measurements, bandwidth measurements + everything a single vantage point can do
- Three vantage points?

Vantage Points

- <Number> of vantage points is not sufficient
- <Location> of vantage points is important



Vantage Points

- Is your vantage point static?
- Mobile vantage points: Mobile phones, laptops
  - Sometimes good if you seek to increase coverage
- But also (for example):
  - IP addresses reallocation
  - Virtual machines reallocation

What is the workload?

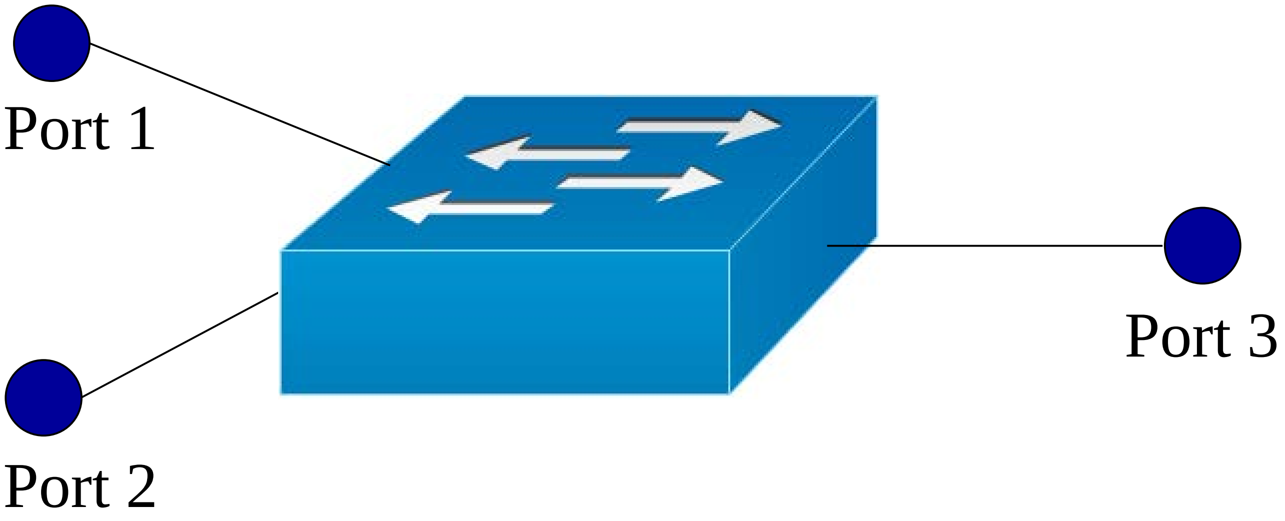
- Synthetically generated, e.g.,
  - 128B IPv4 Packets
- Protocol level, e.g.,
  - TCP flows
- Application level, e.g.,
  - Key-value store application

What is the workload?

- Everything matters!
- Packet size distribution
- Traffic rate
  - E.g., Average rate, peak rate,
- Traffic shape
  - E.g. bursts
- Payload
  - Some payloads are more likely to cause errors than others
- Protocol
- ….

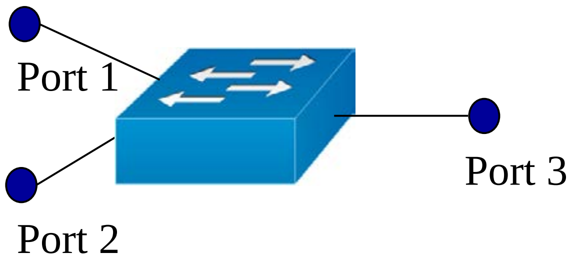
Example

- What can we learn about the internals of a switch using latency measurements and 3 vantage points?
- Assuming a sterile environment



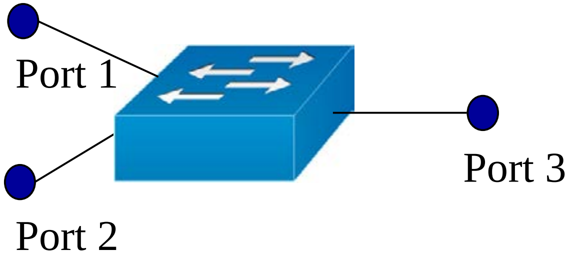
Example

- What is the basic latency of the switch?
  - Send packets from port 1 to port 2, measure the latency
- Is the switch design symmetric?
  - Send packets from port 2 to port 1, measure the latency
- Is the switch design identical for all ports?
  - Send packets from port X to port Y, measure the latency for all combinations



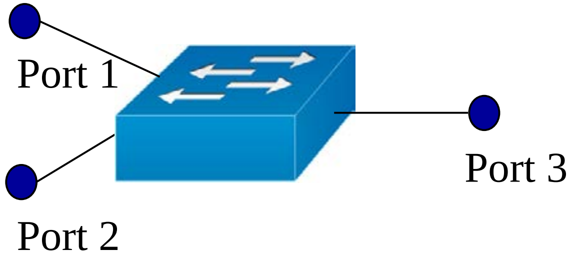
Example

- What type of switch is it?
  - Send packets of various sizes from port 1 to port 2, measure the latency
  - A cut-through switch will have the same latency for all packet sizes, a store-and-forward switch will have a higher latency for bigger packet sizes
- Is the switch sensitive to throughput?
  - Send packets at full line rate from port 1 to port 2, measure the latency
  - Do the results change over time?



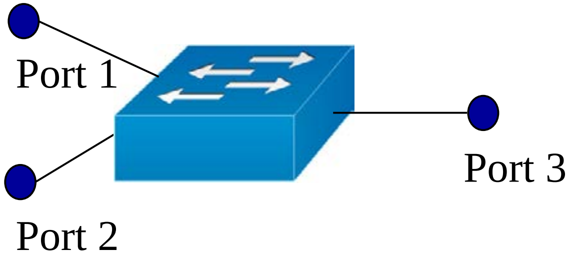
Example

- What can learn about the output queueing and output scheduling of the switch?
  - Send packets at port 1 to port 3, measure the latency And at the same time
  - Send packets at port 2 to port 3, measure the latency
  - Vary the packet rate and discover more….



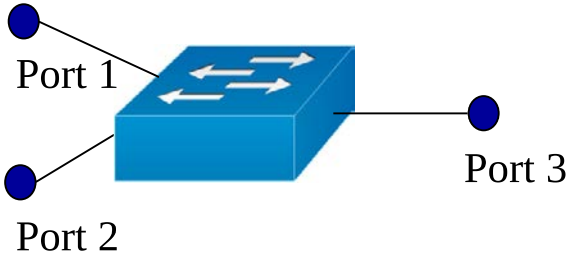
Example

- What can learn about the input queueing and input scheduling of the switch?
  - ➢ Send packets at port 1 to port 3, measure the latency
    And at the same time
  - ➢ Send packets at port 2 to port 4
  - ➢ Vary the packet rate and discover more….
  - ➢ Why is sending from port 2 to port 1 a bad idea?



Example

- What can we learn about the internals of a switch using latency measurements and 3 vantage points?
- A lot!
- This was just a small subset



Example 2

- Mellanox Spectrum vs Broadcom Tomahawk
  - Tolly report, 2016 http://www.mellanox.com/related-docs/products/tolly-report-performance-evaluation-2016-march.pdf
- Bandwidth distribution, 3→1 scenario
  - Source ports 25,26,27, Destination port 31 **33**% BW from each port, on both devices
  - Source ports 24,25,26, Destination port 31 **33**% BW from each port, on Spectrum **25**% from ports 25,26, **50**% from port 24 on Tomahawk
- What does it mean?

Synchronization

- Recall Lecture 3
- Synchronization of time between multiple machines
  - E.g., allow one-way latency measurements
- Synchronization of measurements
  - Can you trigger multiple vantage points to start an experiment at once?
    - E.g. what happens if you measure congestion effects without triggering at once?

Tools Selection

- When to use hardware tools? When to use software tools?
- You don’t always have omniscient control over resources
  - You may not even have permissions for some basic tools
- What can you do?
  - Similar tools using different protocols
  - Write your own tools
  - Redesign your experiment

So lets start measuring!

- Wait!
- What is your goal?
- What do you know about your experimentation environment?
- Have you collected metadata?
- Are you aware of any limitations to the environment / tests / DUT / usage / …?
- Is your experiment reproducible?

Advice

- Getting measurements right is *HARD*
- More isn’t necessarily better.
- Prefer:
    - Better methodology
    - Detailed measurements
    - Reproducibility
    - Understanding the results
    - Become an expert of your work