Introduction to Networking and Systems Measurements

Measurement Pitfalls

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

Common Measurement Pitfalls

- What are the hidden assumptions?
- What did you not notice (in the system, setup, ….)?
- What can your tool do?
- Vantage points
- Repeatability pitfalls
- Performance pitfalls
- Reading the results

Hidden Assumptions - Examples

- The path from A to B is the same (reverse) as the path from B to A
- There is no packet reordering
- Device throughput is the same for all packet sizes
- Test packets will experience the same effects as application's traffic
- The effect of DNS lookup is negligible
- The measurement tool has negligible overhead
- Previous work was correct

Another take:

8 fallacies of Distributed Systems

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

System and Setup
Did you notice that….

- There are other jobs running on the same core
- ICMP traffic is throttled by the OS
- CPU frequency scaling is enabled
- The CPU that you are using is not connected directly to the NIC
- Kernel version has been updated overnight
- The 2x40G NIC uses PCIe Gen 3 x8 (~60Gbps)
- There is a new Errata…

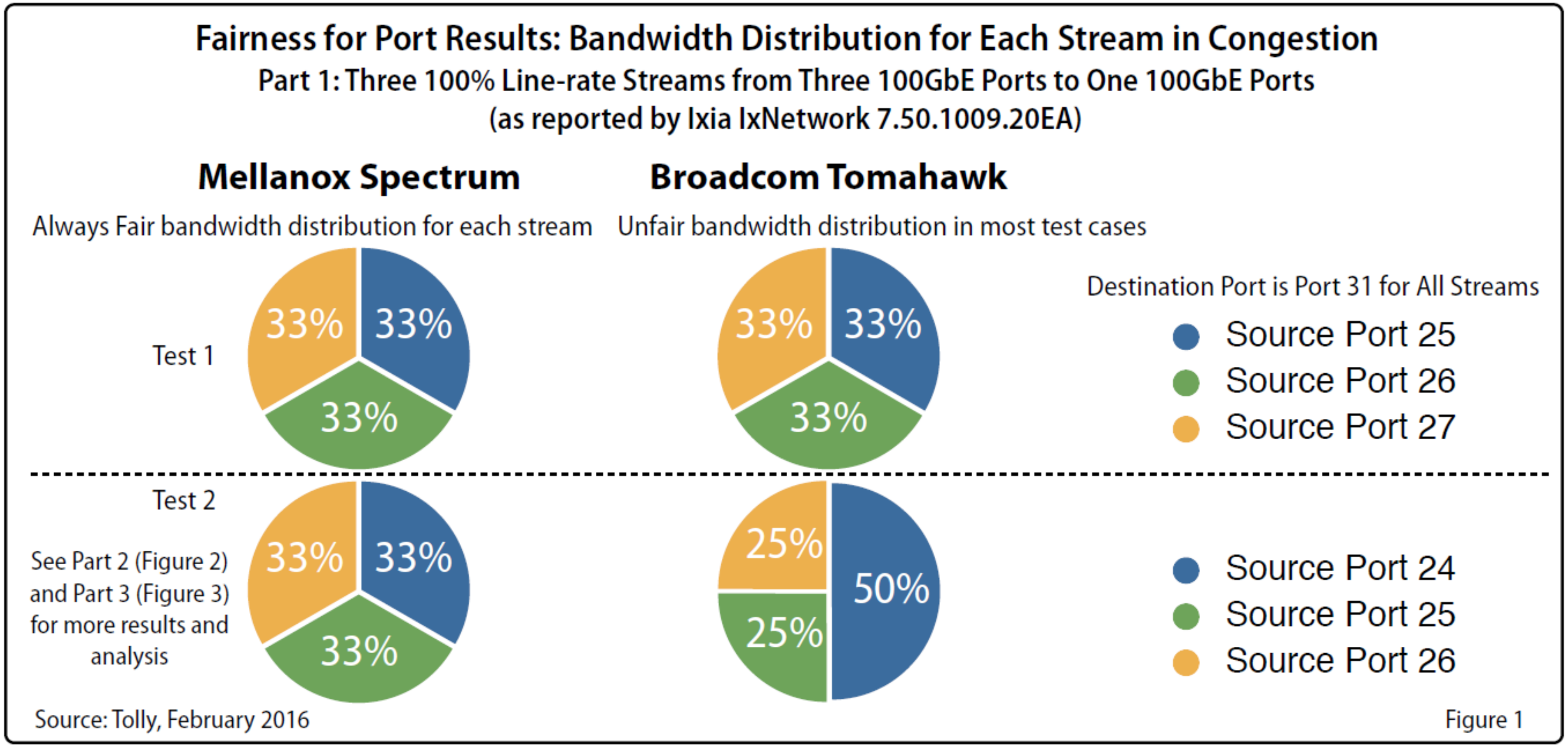
What can your tool do? - Examples

- An oldish SSD can write at 450MB/s
  - Don't try to write data captured at 10Gbps
- The latency for reading CPU timestamp is ~tens of cycles
  - Don't try to use it to measure cache access time
- DAG resolution is 4ns
  - Don't try to measure the propagation delay through 1m fibre
- OSNT can only capture at low rate
  - Don't try to measure latency of 10Gbps flows

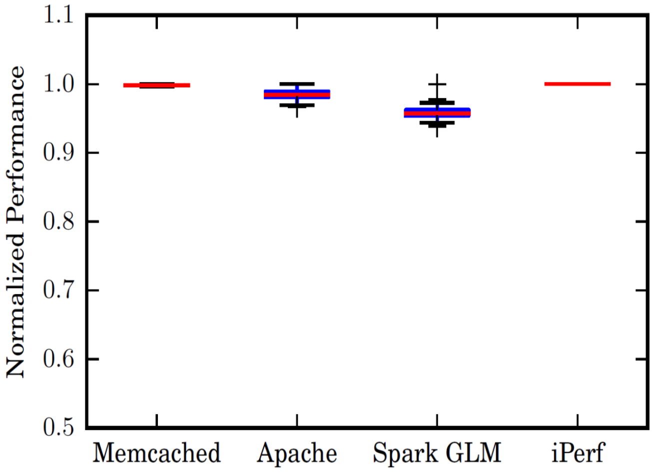
Vantage Points: Example 2 (Lecture 5)

- 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?

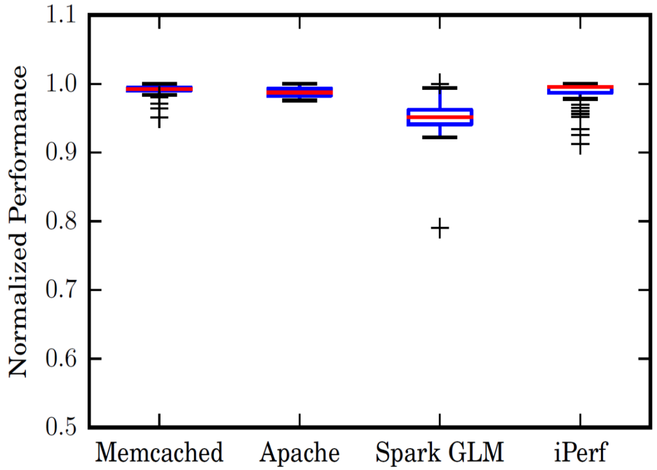
Vantage Points: Example 2
Or: What is wrong with Broadcom Tomahawk?



Repeatability Pitfalls - Examples

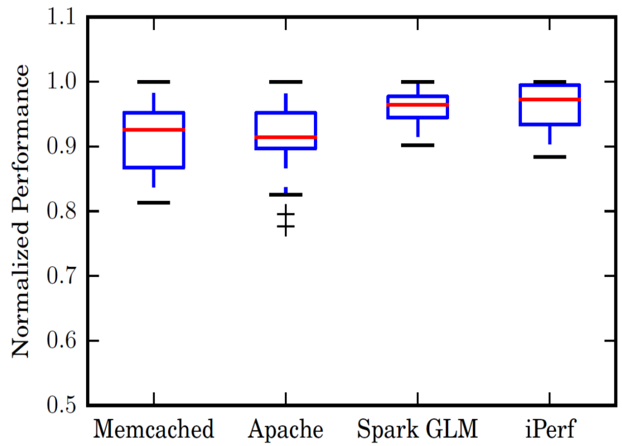


Running on bare metal (local)



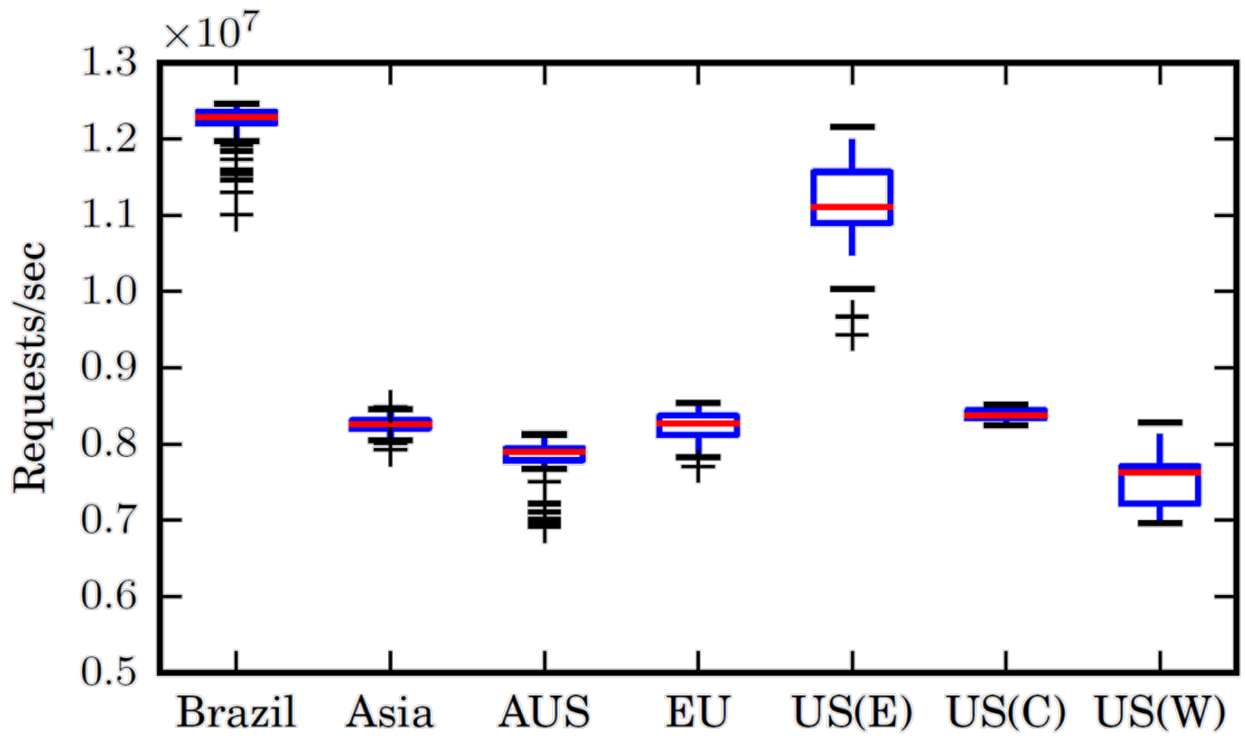
Running on a VM (local)

Increased performance variance



Running in the cloud

Repeatability Pitfalls - Examples



Apache Webserver - Running in the cloud
38% difference in median performance

Latency Pitfalls - Examples

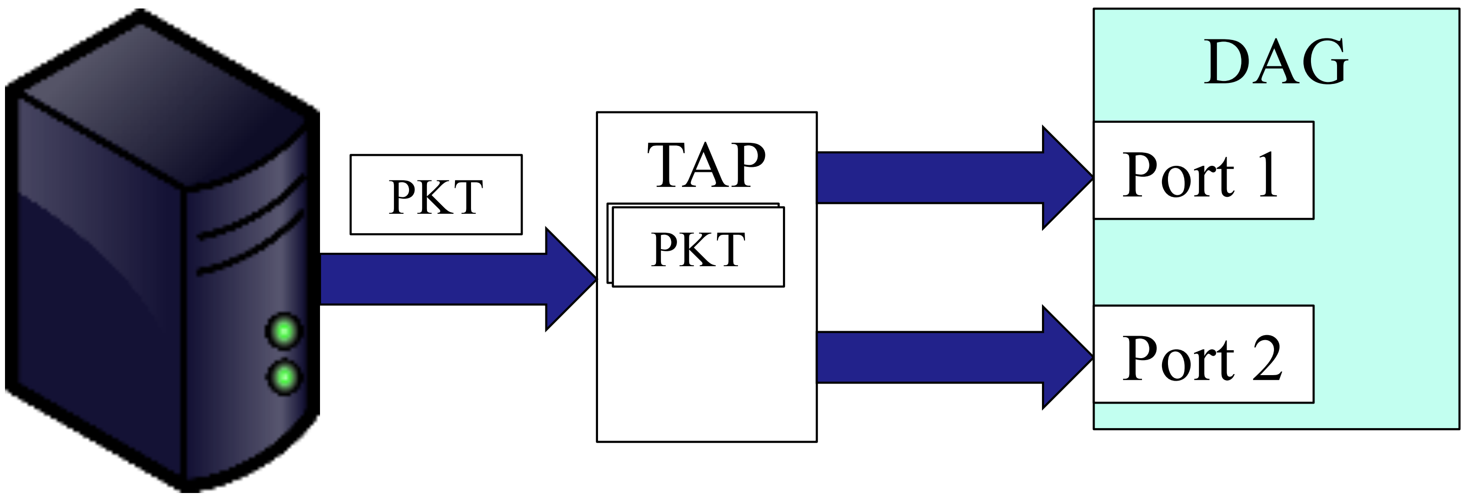
- What is the definition of “latency”?
  - Propagation delay? Inter packet gap? Round trip time? Flow completion time?
- How was the latency measured?
  - Start of packet to start of packet? Start of packet to end of packet?
  - Single packet? Packet-pair? Packet-train?
- Where was the timestamp taken?
  - …and how did it affect the measurement?
- Resolution, precision and accuracy…

Bandwidth Pitfalls - Examples

- What is the definition of “bandwidth”?
  - Link capacity? Average throughput? Peak throughput?
- Controllability
  - Packet size? Protocol? QoS?
- What was the status of the network?
- Net neutrality?
- Did you pass through the bottlenecks?
- Resolution, precision and accuracy…

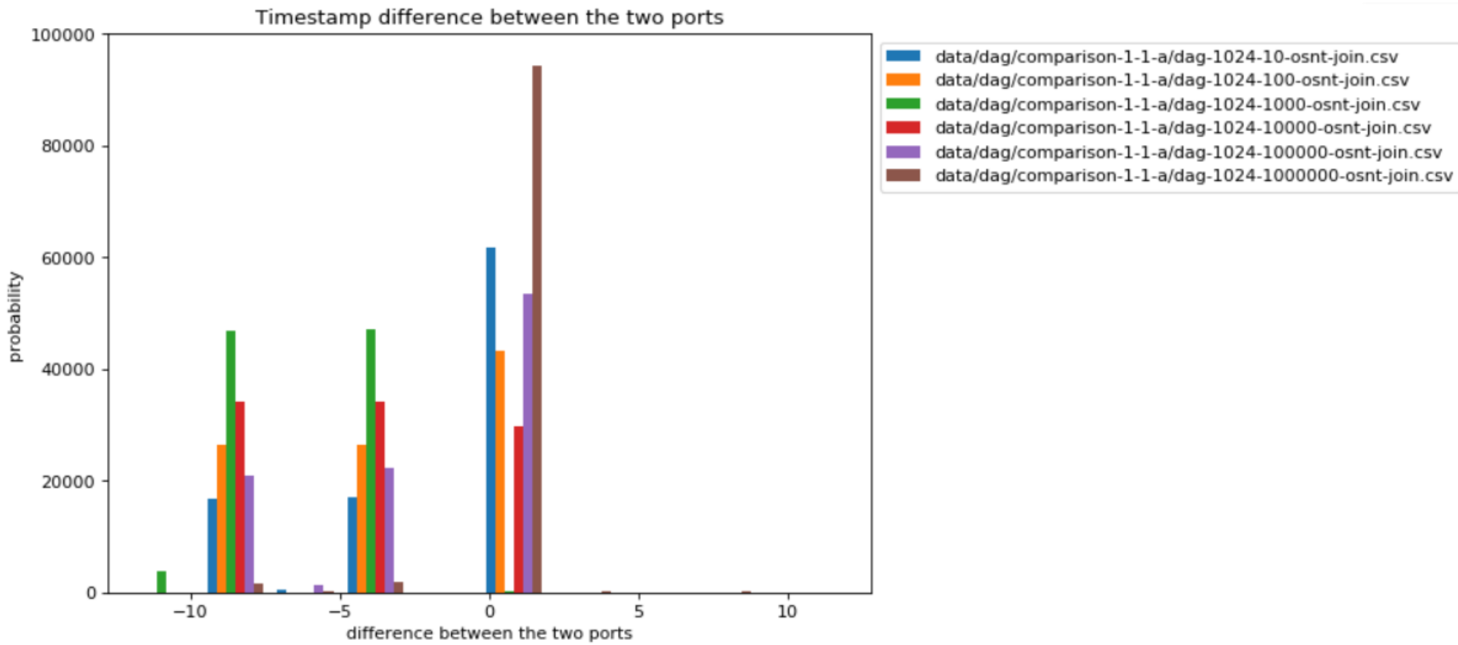
Example: Timestamp difference between ports

- Recall Lab 2, experiment 2.1 b
- Measuring the timestamp difference between 2 ports:



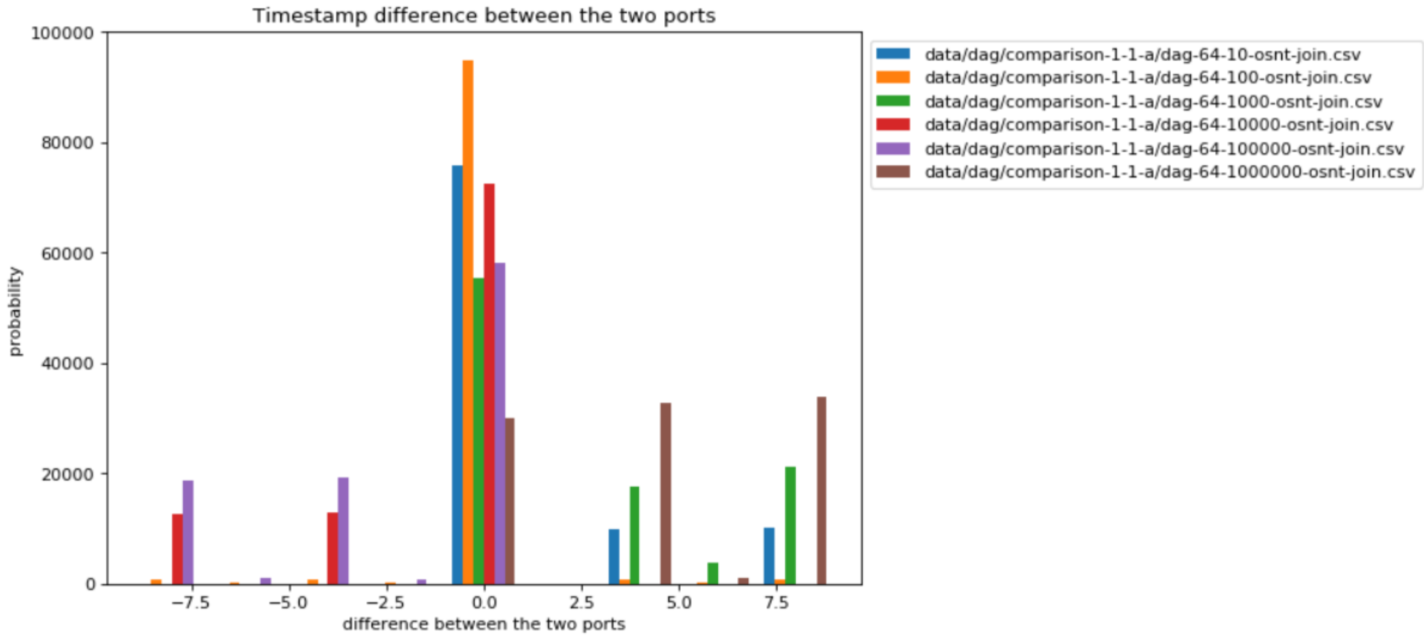
Example: Timestamp difference between ports

- 100,000 packets, **1024B**
- Different Inter Packet Gaps (IPG)



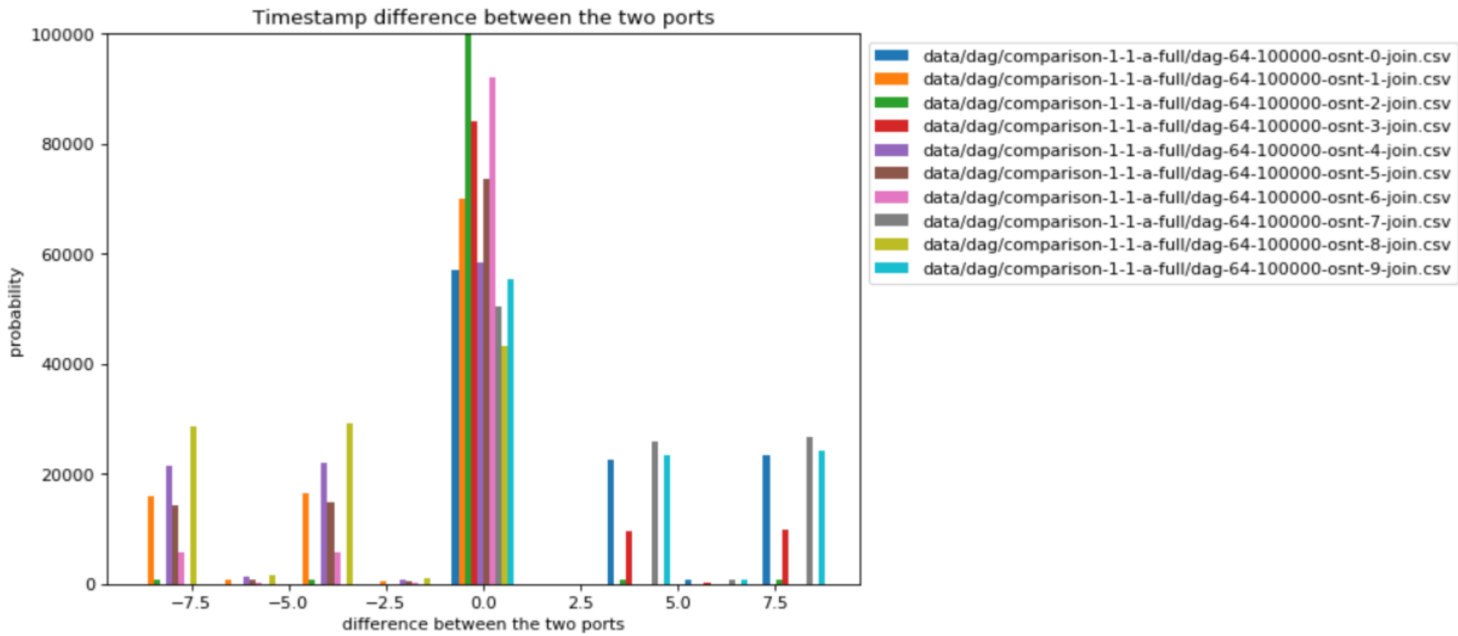
Example: Timestamp difference between ports

- 100,000 packets, **64B**
- Different Inter Packet Gaps (IPG)



Example: Timestamp difference between ports

- 100,000 packets, **64B**, running 10 times
- Same Inter Packet Gap (IPG)



Example: Switch Throughput

- The reported iperf result for a NetFPGA reference switch is 9.4Gbps
- User complaint: I see only 8.9Gbps and packet drop in the switch

```
Connecting to host 10.0.0.13, port 5201
[  4] local 10.0.0.12 port 54764 connected to 10.0.0.13 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  1.02 GBytes  8.76 Gbits/sec   74    313 KBytes
[  4]   1.00-2.00   sec  1.03 GBytes  8.86 Gbits/sec   34    198 KBytes
[  4]   2.00-3.00   sec  1.03 GBytes  8.87 Gbits/sec   34    281 KBytes
[  4]   3.00-4.00   sec  1.04 GBytes  8.92 Gbits/sec   34    238 KBytes
[  4]   4.00-5.00   sec  1.04 GBytes  8.93 Gbits/sec   32    208 KBytes
[  4]   5.00-6.00   sec  1.04 GBytes  8.92 Gbits/sec   29    187 KBytes
[  4]   6.00-7.00   sec  1.04 GBytes  8.95 Gbits/sec   27    365 KBytes
[  4]   7.00-8.00   sec  1.04 GBytes  8.94 Gbits/sec   28    233 KBytes
[  4]   8.00-9.00   sec  1.03 GBytes  8.88 Gbits/sec   30    420 KBytes
[  4]   9.00-10.00  sec  1.04 GBytes  8.96 Gbits/sec   33    423 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  10.4 GBytes  8.90 Gbits/sec  355             sender
[  4]   0.00-10.00  sec  10.4 GBytes  8.90 Gbits/sec                  receiver
```

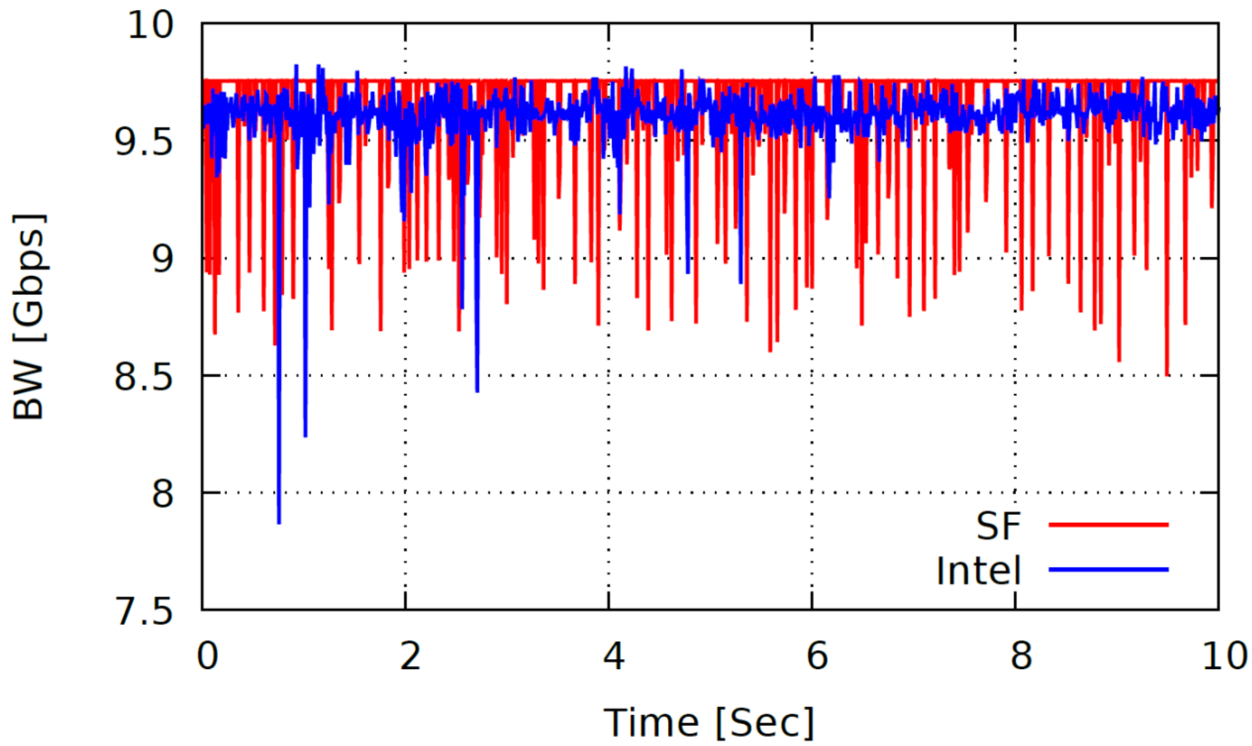
Example: Switch Throughput

- Debug: Have you tried changing rx-usec?
- User: no more packet drop in the switch!
- …but bandwidth is down to 7.5Gbps…

- New insight: NIC used on reference setup (Solarflare) is different than the NIC used by user (Intel)
- (skipping a few steps forward)

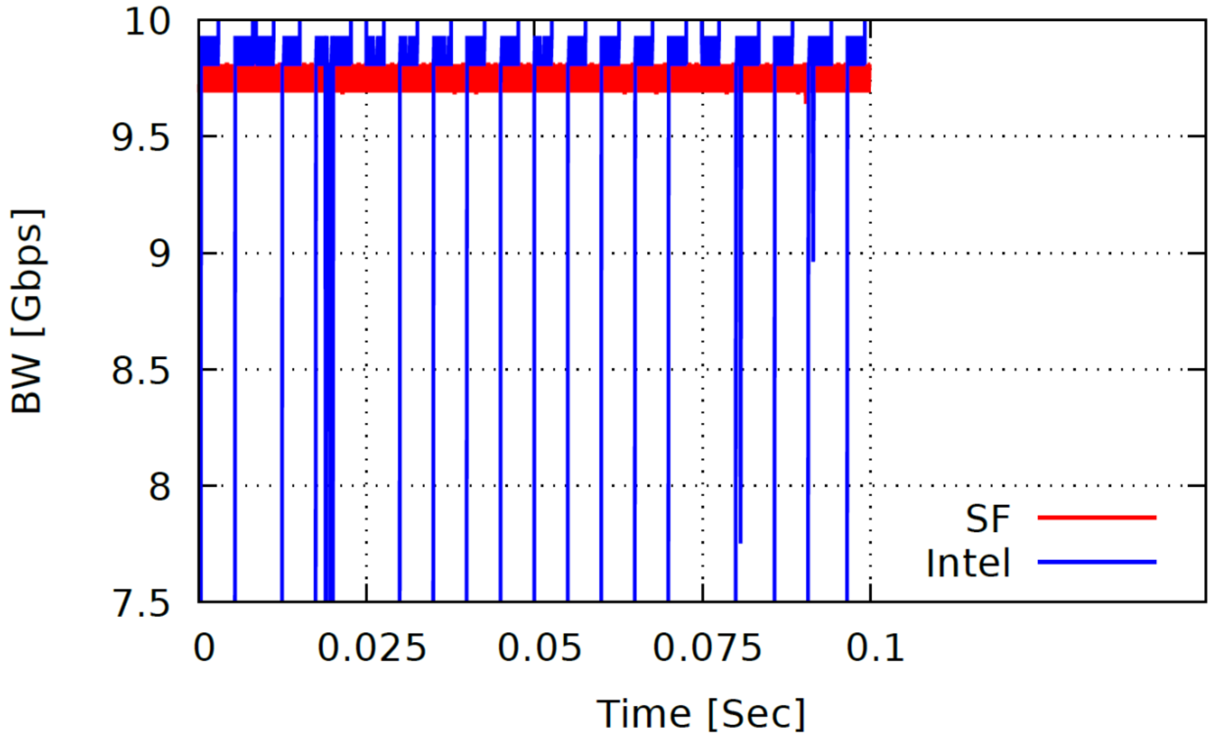
Example: Switch Throughput

- Switch throughput over time (10ms sampling resolution)



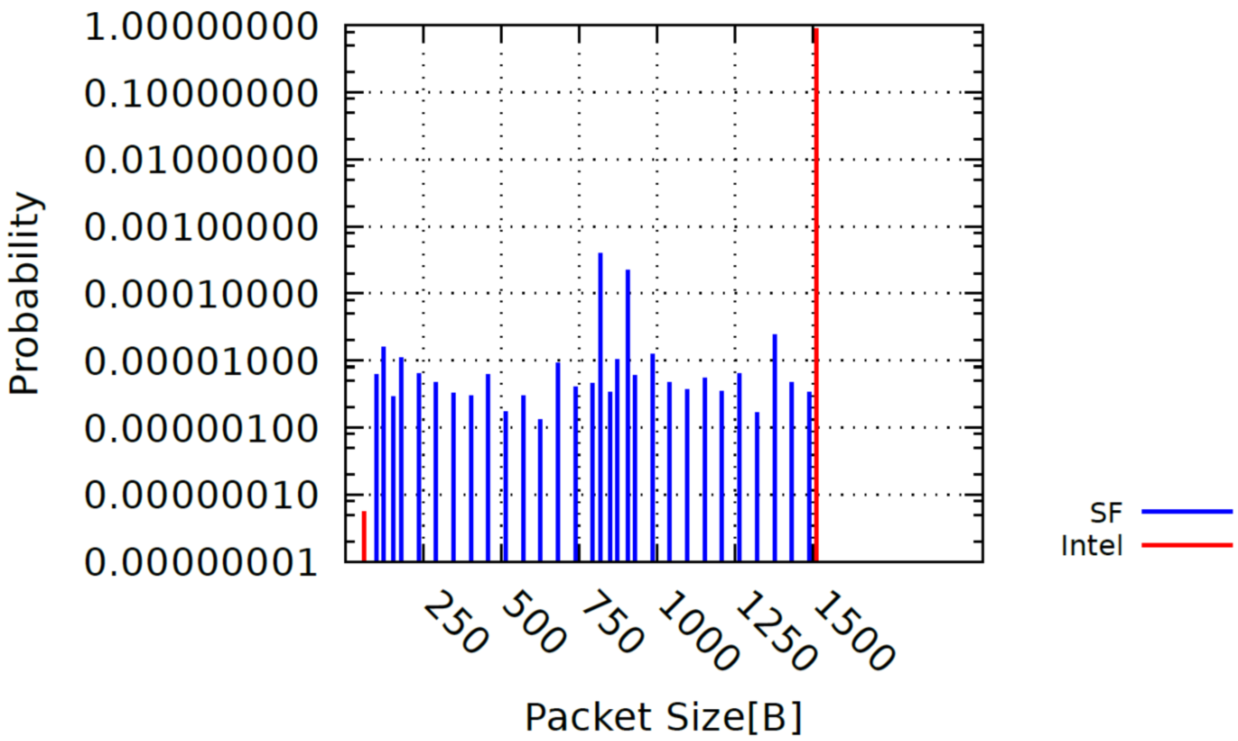
Example: Switch Throughput

- Switch throughput over time (100µs sampling resolution)



Example: Switch Throughput

- What else is different?



Example: TSC Access

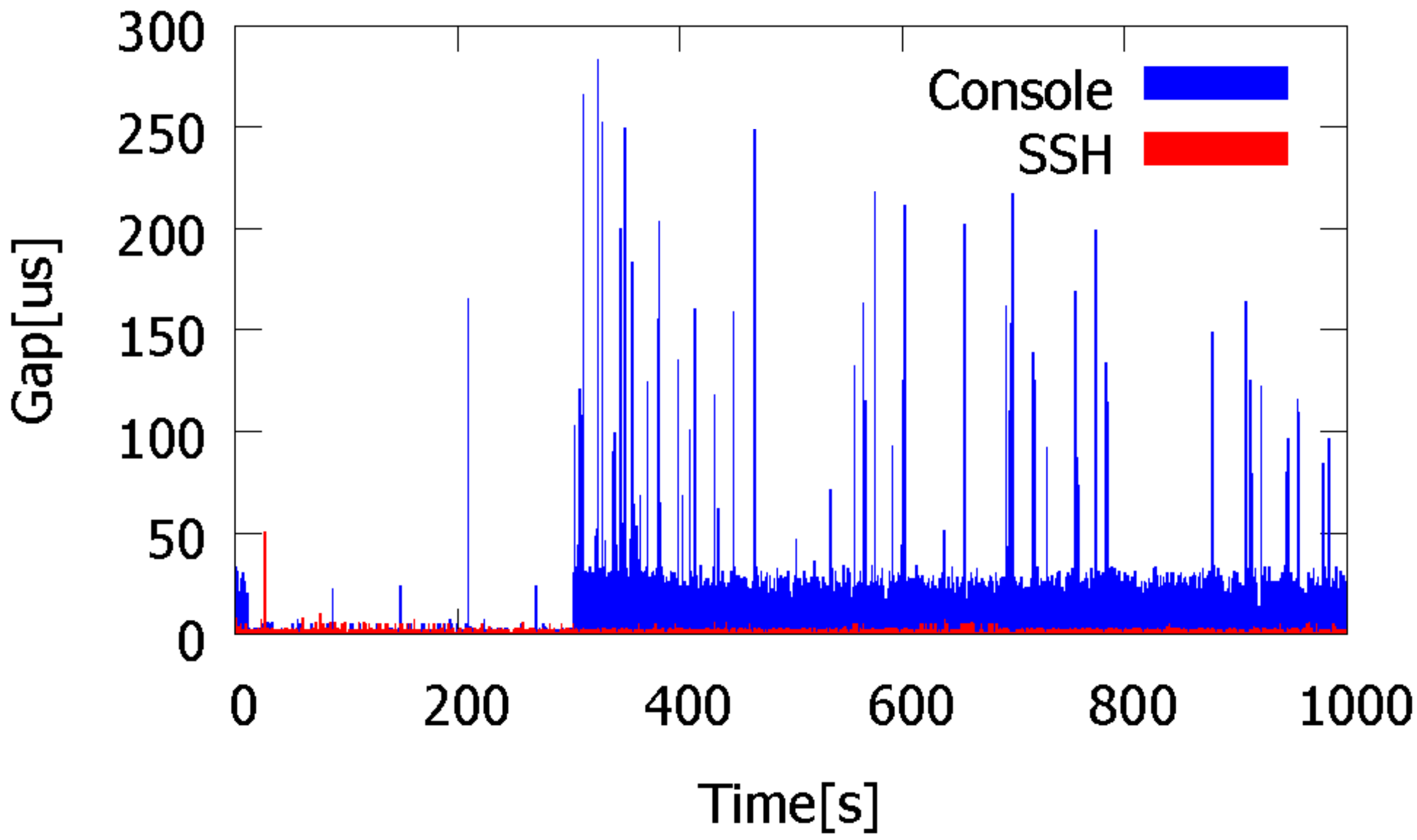
- Goals:
  - Evaluate the accuracy & precision of time-taking using CPU time stamp counter (TSC)
- Methodology:
  - Read TSC twice
  - Measure the time-gap between the two consecutive reads
- Results:
  - Min/Median/99.9%: 9ns/10ns/11ns

Example: TSC Access

```
while (!done)
{
    //Read TSC twice, one immedately after the other
    do_rdtscp(tsc, cpu);
    do_rdtscp(tsc2,cpu2);
    //If the gap between the two reads is above a
        certain threshold, save it
    if ((tsc2 - tsc > threshold) && (cpu == cpu2))
       buffer[samples++] = tsc2-tsc;
}
```

Example: TSC Access

What happens over time?



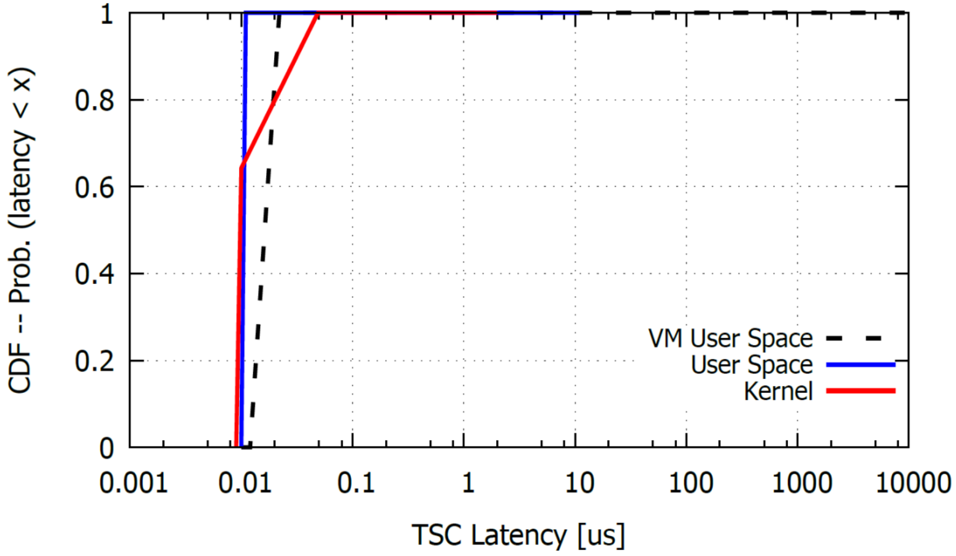
Example: TSC Access

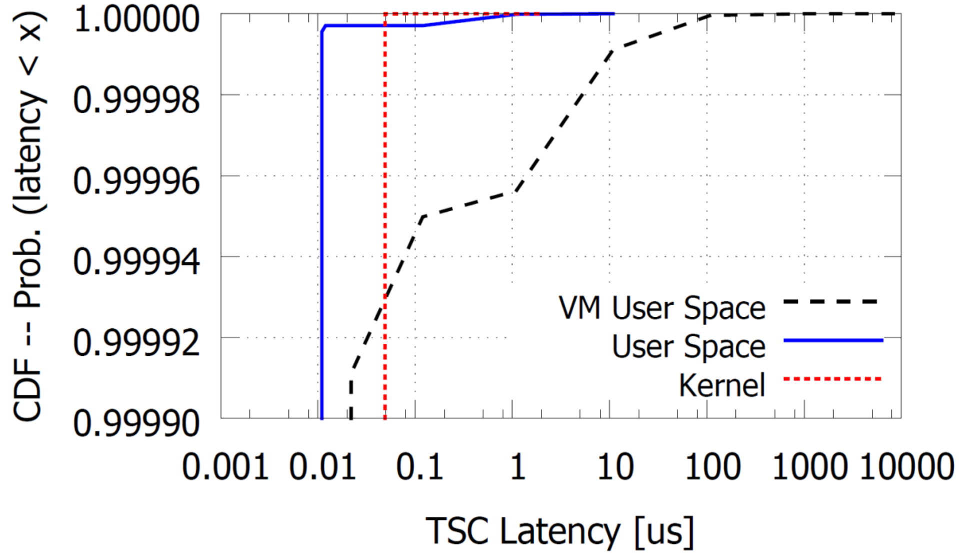
- Source data:

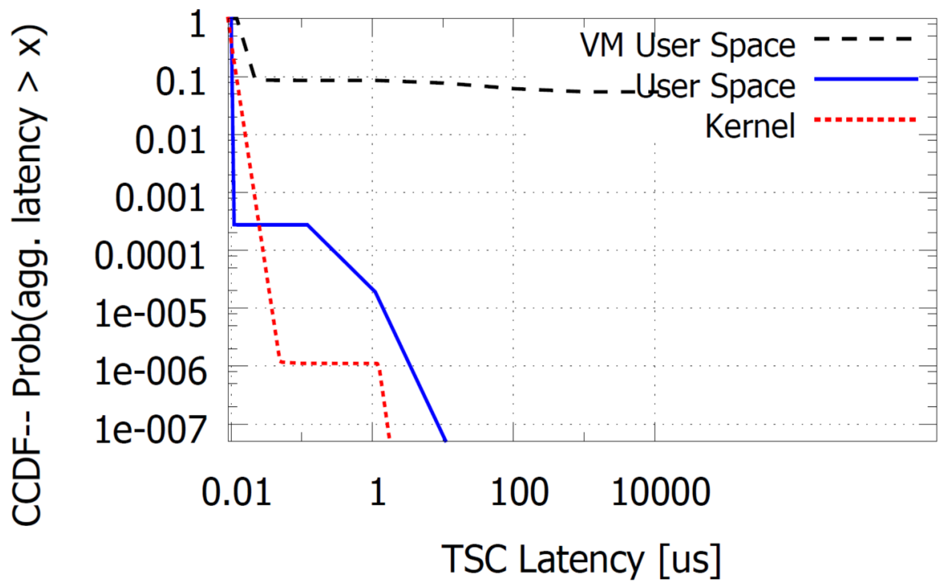
X ≤	User space Events
10	91428291492
11	404700
12	268521
22	268291
120	267465
1097	10768
10869	1

X ≤	Kernel Events
9	11117819727
10	3973891503
49	287
53	201
98	90
1155	86
1184	85
1241	77
1982	1

Example: TSC Access







```c
while (!done) {
    //Read TSC twice, one immedately after the other
    do_rdtscp(tsc, cpu);
    do_rdtscp(tsc2,cpu2);
    //If the gap between the two reads is above a threshold, save it
    if ((tsc2 - tsc > threshold) && (cpu == cpu2))
       buffer[samples++] = tsc2-tsc; }
```

Listing 1.1. Reading and Comparing TSC - Code 1.

```c
while (!done) {
    //Read TSC once
    do_rdtscp(tsc, cpu);
    //If the gap between the current and the previous reads is above a
        threshold, save it
    if ((tsc - last > threshold) && (cpu == lastcpu))
       buffer[samples++] = tsc-last;
    last = tsc;
    lastcpu = cpu; }
```

Listing 1.2. Reading and Comparing TSC - Code 2.

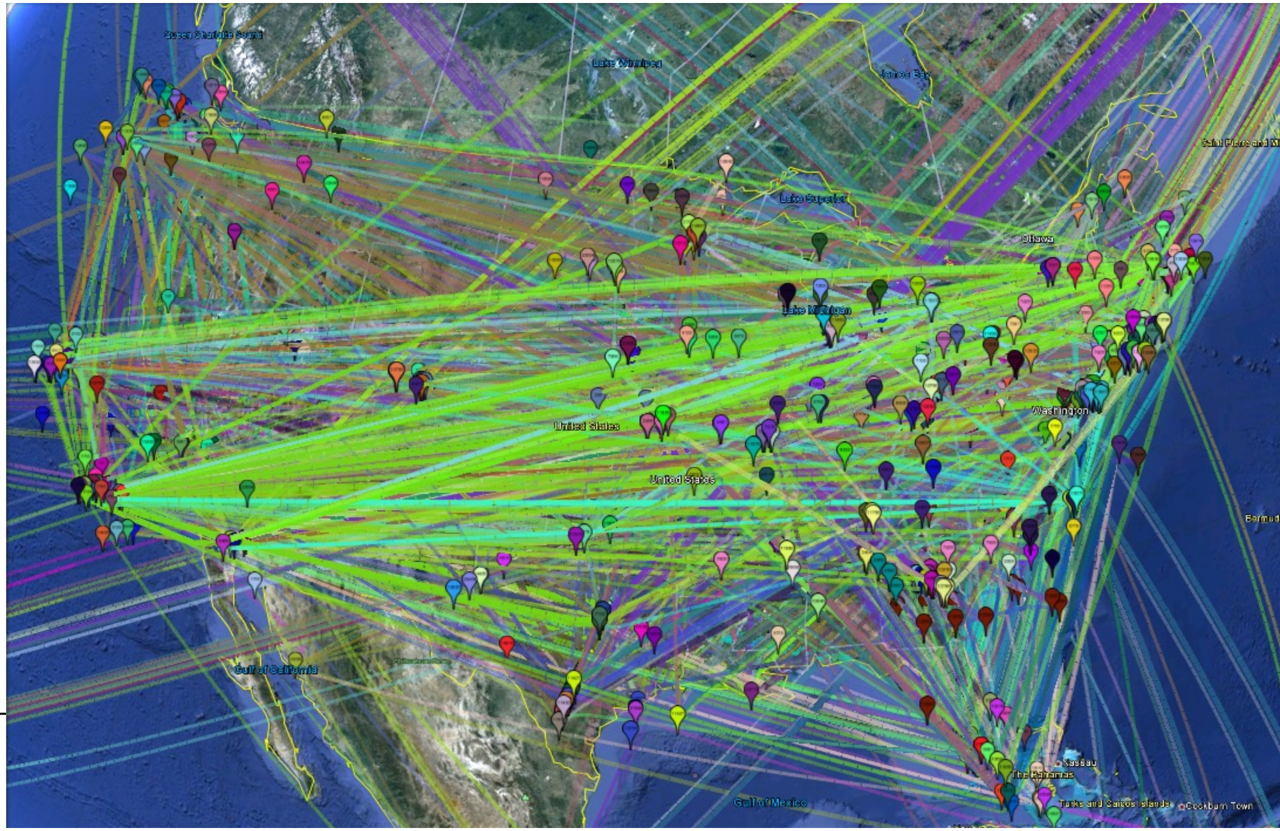
Example: Topology Measurements

- Goal:
  - Build a map of network connectivity that assigns IP addresses to locations
- Method:
  - Simple option: name resolution
    - 4.69.166.1 ⇒ ae-119-3505.edge4.London1.Level3.net
    - But many times information is missing, not indicative or is inaccurate
  - Better option: use geolocation services
    - Most services claim to be over 99% accurate

Example: Topology Measurements

Building a map of the network:

- Measurements for connectivity
- Geolocation databases for location



Example: Topology Measurements
What is your ground truth?

- Geolocation databases are over 99% accurate

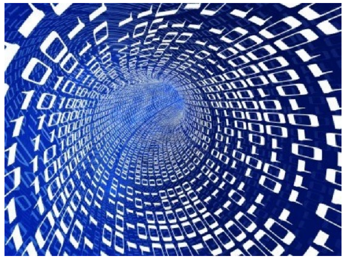
Verizon/MCI/UUNET (ASN 703)
10-nodes PoP

	Maxmind	IP2Location	IPligence	HostIP	Netacuity	Spotter	Geobytes
Maxmind	0	5	6	158	24	666	216
IP2Location	5	0	11	382	171	836	48
IPligence	6	11	0	452	148	821	433
HostIP	158	382	452	0	264	890	528
Netacuity	24	171	148	264	0	392	418
Spotter	666	836	821	890	392	0	1053
Geobytes	216	48	433	528	418	1053	0

Heatmap – Median distance between databases (2011)

Validation

- Measurements need to be validated
- Don’t make assertions!
- Use ground truth (where available)
- Compare different tools and methodologies
- Do the results make sense?
  - ➢ RTT can’t be faster than traveling at the speed of light…
- Have I mentioned validation?



Lab report 2

Reproduce Experiments 1,2,3,4,6a

from

“Where has my time gone?”
Zilberman *et al.* 2017 PAM

Provide your own analysis as appropriate.
(these instructions may be refined)

3 Latency Results

	Experiment	Minimum	Median	99.9th	Tail	Observation Period
1a	TSC - Kernel Cold Start	7ns	7ns	7ns	11ns	1 Hour
1b	TSC - Kernel	9ns	9ns	9ns	6.9μs	1 Hour
1c	TSC - From User Space	9ns	10ns	11ns	49μs	1 Hour
1d	TSC - From VM User Space	12ns	12ns	13ns	64ms	1 Hour
2a	User Space + OS (same core)	2μs	2μs	2μs	68μs	10M messages
2b	User Space + OS (other core)	4μs	5μs	5μs	31μs	10M messages
3a	Interconnect (64B)	552ns	572ns	592ns	608ns	1M Transactions
3b	Interconnect (1536B)	976ns	988ns	1020ns	1028ns	1M Transactions
4	Host	3.9μs	4.5μs	21μs	45μs	1M Packets
5	Kernel Bypass	895ns	946ns	1096ns	5.4μs	1M Packets
6a	Client-Server (UDP)	7μs	9μs	107μs	203μs	1M Packets
6b	Client-Server (Memcached)	10μs	13μs	240μs	20.3ms	1M Queries
7a	NIC - X10 (64B)	804ns	834ns	834ns	10μs	100K Packets
7b	NIC - SFN8522 (64B)	960ns	985ns	1047ns	3.3μs	100K Packets
8a	Switch - ExaLINK50 (64B)	0^{α}	2.7ns $^{\alpha}$	17.7ns $^{\alpha}$	17.7ns $^{\alpha}$	1000 Packets
8b	Switch - ExaLINK50 (1514B)	0^{α}	2.7ns $^{\alpha}$	17.7ns $^{\alpha}$	17.7ns $^{\alpha}$	1000 Packets
8c	Switch - 7124FX (64B)	512ns	534ns	550ns	557ns	1000 Packets
8d	Switch - 7124FX (1514B)	512ns	535ns	557ns	557ns	1000 Packets

Table 1. Summary of Latency Results. Entries marked $^{\alpha}$ return results that are within DAG measurement error-range

Final Report - Recommendations

- Include all figures within the report
  - Use proper scale, adapt the template if need be
- Make sure that your environment does not affect the results
- Do not make assertions
  - Support your claims through experimentations
- Discuss your results in depth:
  - Compare and contrast results gained through different vantage points, using different tools, on different platforms etc
  - Provide side-by-side comparisons
  - Use the questions in the handouts as guiding examples
- Use the right terminology (accuracy, precision, resolution)
- Correct typos and grammar mistakes
- Follow the instructions in the handout

Course Summary

- This course has covered measurements tools and measurement techniques
- But also “why out most basic assumptions are wrong”, “graphs lie”, “what you don’t know about your system”, …
- Remember:
  - Constant vigilance
  - Look at the data, best-practice, think.
- These ideas apply to

  all types of measurements

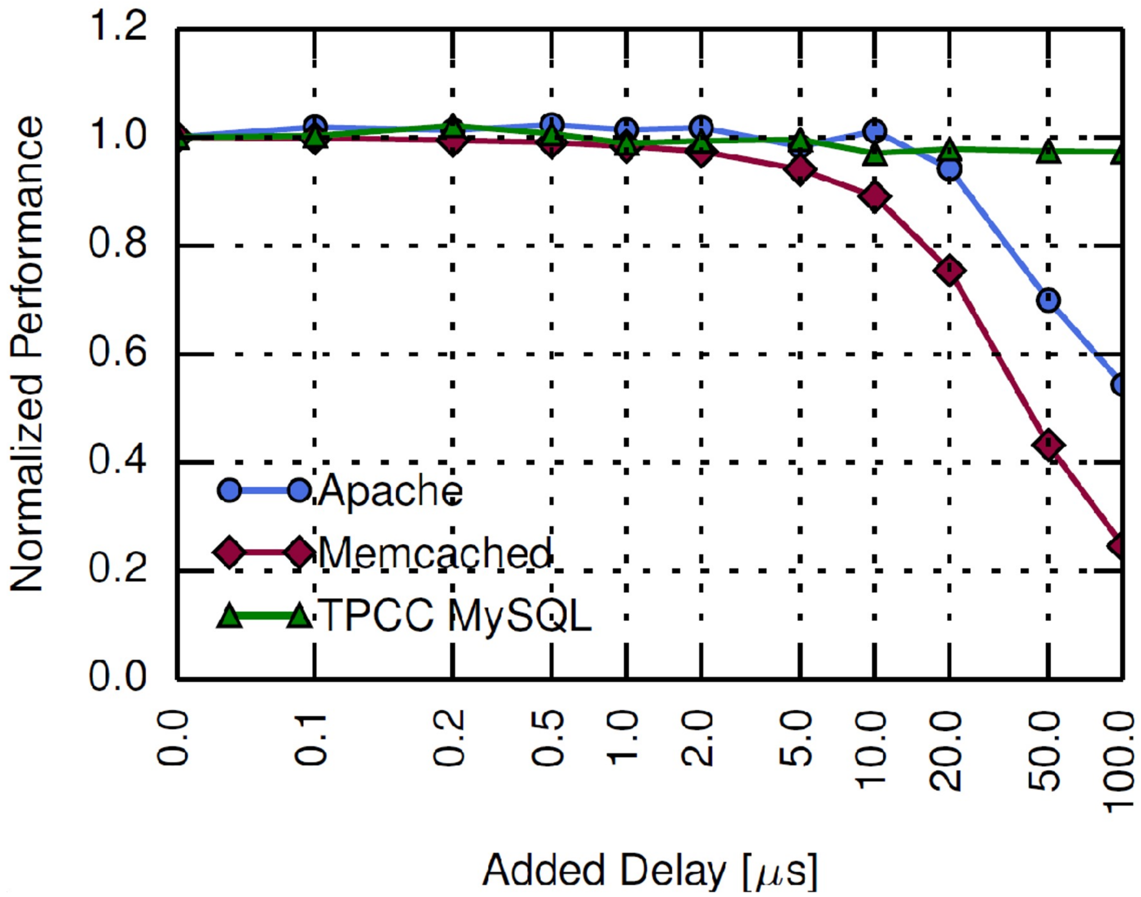
Appendix
(slides from PAM2017 presentation)

UNIVERSITY OF CAMBRIDGE

Where Has My Time Gone?

Noa Zilberman, Matthew Grosvenor, Diana Andreea Popescu, Neelakandan Manihatty-Bojan, Gianni Antichi, Marcin Wójcik, Andrew W. Moore

Latency matters



It’s Time For Low Latency + Low Variance!

2011

It’s Time for Low Latency

Stephen M. Rumble, Diego Ongaro, Ryan Stutsman,
Mendel Rosenblum, and John K. Ousterhout
Stanford University

:ommunity has ignored network
ı the past, speed-of-light delays
and unoptimized network hard-
)μs round-trip times impossible.
w years datacenters will be de-
ʼ Ethernet. Without the burden
the datacenter campus and net-
rnet devices, it will be up to us
this benefit through to applica-
researchers must lead the charge
s to push the boundaries of low-
nunicat

	1983	2011	Improved
CPU Speed	1x10Mhz	4x3GHz	$>$ 1,000x
Memory Size	$\leq$ 2MB	8GB	$\geq$ 4,000x
Disk Capacity	$\leq$ 30MB	2TB	$>$ 60,000x
Net Bwidth	3Mbps	10Gbps	$>$ 3,000x
RTT	2.54ms	80μs	32x

Table 1: Network latency has improved far more slowly over the last three decades than other performance metrics for commodity computers. The V Distributed System [5] achieved round-trip RPC times of 2.54ms. Today, a pair of modern Linux servers require 80μs for 16-byte RPCs over TCP with 10Gb Ethernet.

Component	Delay	Round-Trip
Network Switch	10-30μs	100-300μs
Network Interface Card	2.5-32μs	10-128μs
OS Network Stack	15μs	60μs
Speed of Light (in Fiber)	5ns/m	0.6-1.2μs

Table 2: Factors that contribute to latency in TCP datacenter communication. “Delay” indicates the cost of a single traversal of the component, and “Round-Trip” indicates the total impact on round-trip time. Messages typically traverse 5 switches in each direction in a large datacenter network and must pass through the OS stack 4 times.

2016

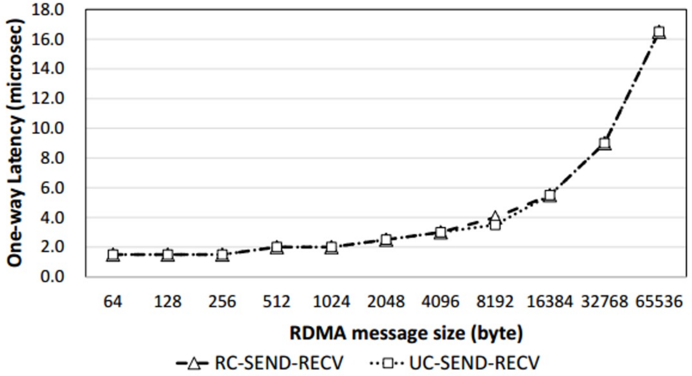


Fig. 4: Median one-way latency of RoCE RC and UC transport types.

"Exploring Low-latency Interconnect for Scaling Out Software Routers“, Ma, Kim, and Moon

“Design Guidelines for High Performance RDMA Systems” Kalia, Kaminsky and Andersen

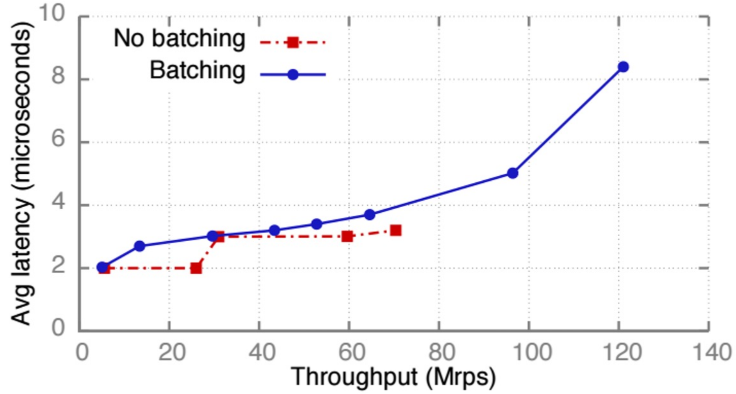


Figure 8: Impact of response batching on `Spec-S0` latency

It's Time For Low Latency + Low Variance!

2011

It's Time for Low Latency

Stephen M. Rumble, Diego Ongaro, Ryan Stutsman,
Mendel Rosenblum, and John K. Ousterhout
Stanford University

:ommunity has ignored network
ı the past, speed-of-light delays
and unoptimized network hard-
)μs round-trip times impossible.
w years datacenters will be de-
ı Ethernet. Without the burden
the datacenter campus and net-
rnet devices, it will be up to us
this benefit through to applica-
researchers must lead the charge
s to push the boundaries of low-
nunicat

	1983	2011	Improved
CPU Speed	1x10Mhz	4x3GHz	$>$ 1,000x
Memory Size	$\leq$ 2MB	8GB	$\geq$ 4,000x
Disk Capacity	$\leq$ 30MB	2TB	$>$ 60,000x
Net Bwidth	3Mbps	10Gbps	$>$ 3,000x
RTT	2.54ms	80μs	32x

Table 1: Network latency has improved far more slowly over the last three decades than other performance metrics for commodity computers. The V Distributed System [5] achieved round-trip RPC times of 2.54ms. Today, a pair of modern Linux servers require 80μs for 16-byte RPCs over TCP with 10Gb Ethernet.

Component	Delay	Round-Trip
Network Switch	10-30μs	100-300μs
Network Interface Card	2.5-32μs	10-128μs
OS Network Stack	15μs	60μs
Speed of Light (in Fiber)	5ns/m	0.6-1.2μs

Table 2: Factors that contribute to latency in TCP datacenter

2016

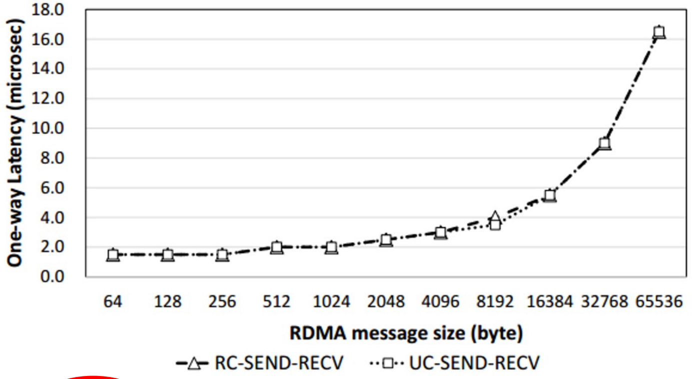


Fig. 4: Median one-way latency of RoCE RC and UC transport types

"Exploring Low-latency Interconnect for Scaling Out Software Routers", Ma, Kim, and Moon

"Design Guidelines for High Performance RDMA Systems" Kalia, Kaminsky and Andersen

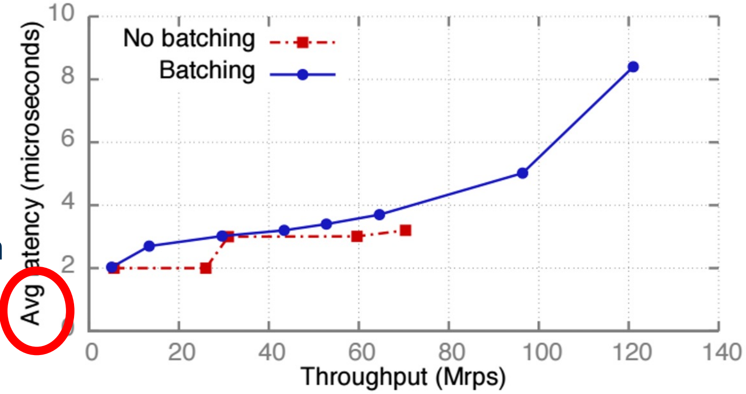


Figure 8: Impact of response batching on `Spec-S0` latency

It Usually Works

The Unavoidable Latency

I will talk about:

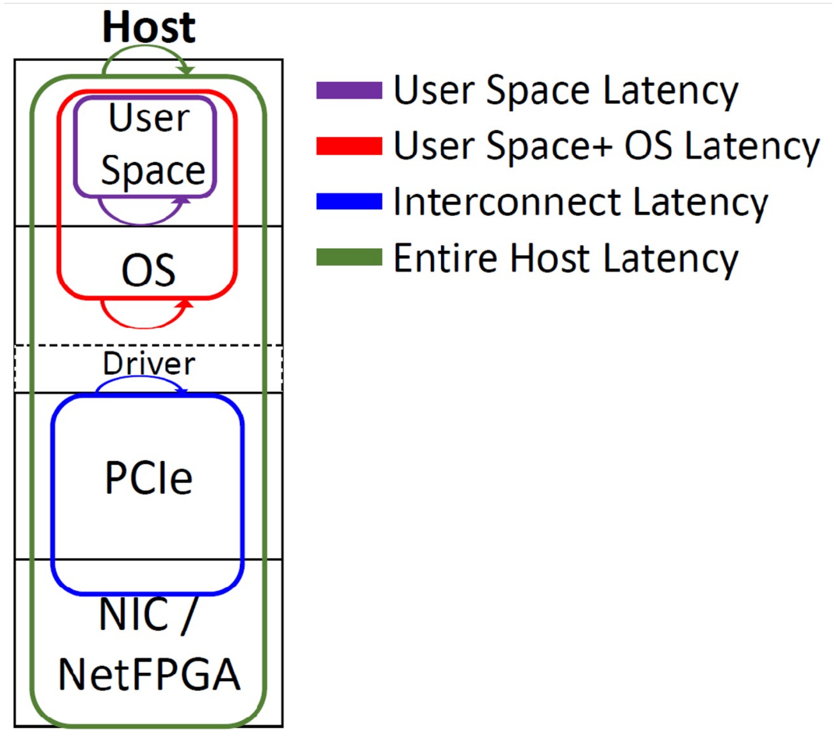
- The **essential** latency contributions
- **Commodity** hardware, standard coding
- Set a (reproducible) **baseline** for research

I will not talk about:

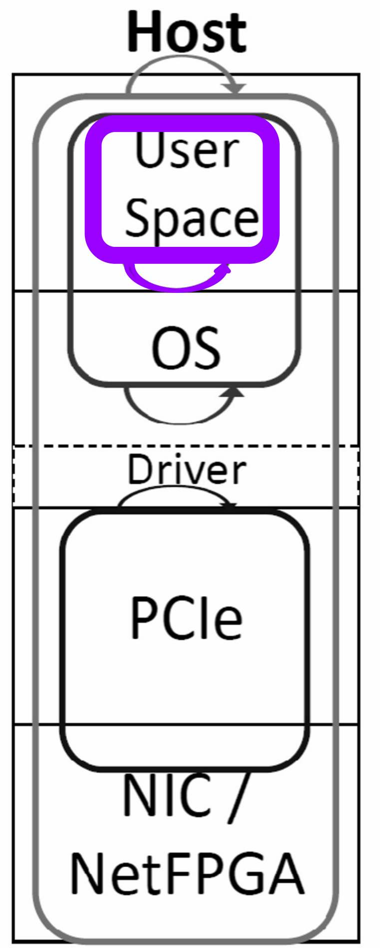
- TCP, DCTCP, MPTCP etc.
- Congestion, Buffer bloat, In-cast, etc.

Methodology

- Breakdown overall system latency into **component contributions**
- Use decompositional analysis to report component distributions e.g. min, median, 99.9th, max

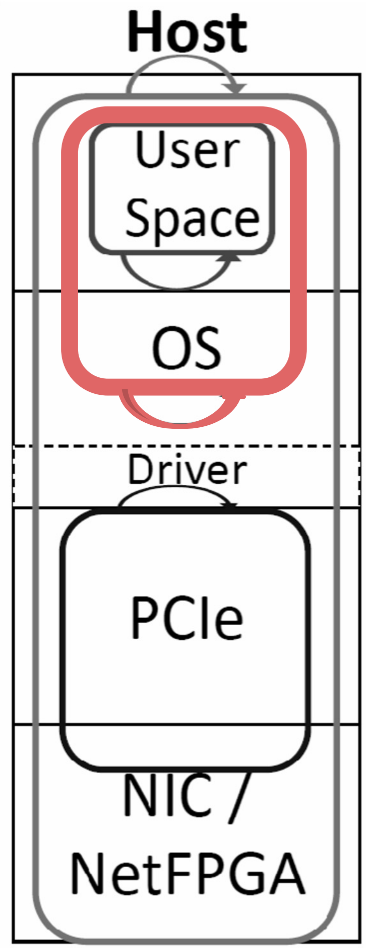


Time Stamp Counter (TSC) Measurement Accuracy



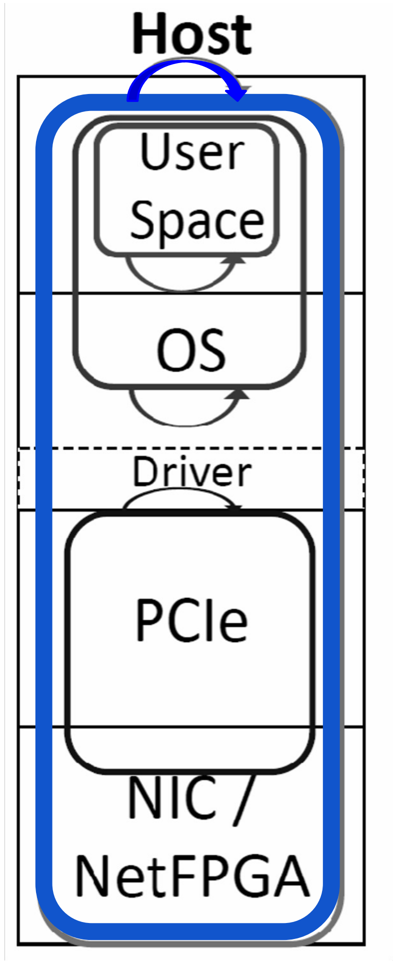
Experiment	Minimum	Median	99.9th	Tail
TSC - From User Space	9 ns	10 ns	11 ns	**49 μs**
TSC - Kernel	9 ns	9 ns	9 ns	**6.9 μs**
TSC - Kernel Early Boot	7 ns	7 ns	7 ns	**11 ns**
TSC - From VM User Space	12 ns	12 ns	13 ns	**64 ms**

A Breakdown of Basic Latency Components: User space + OS



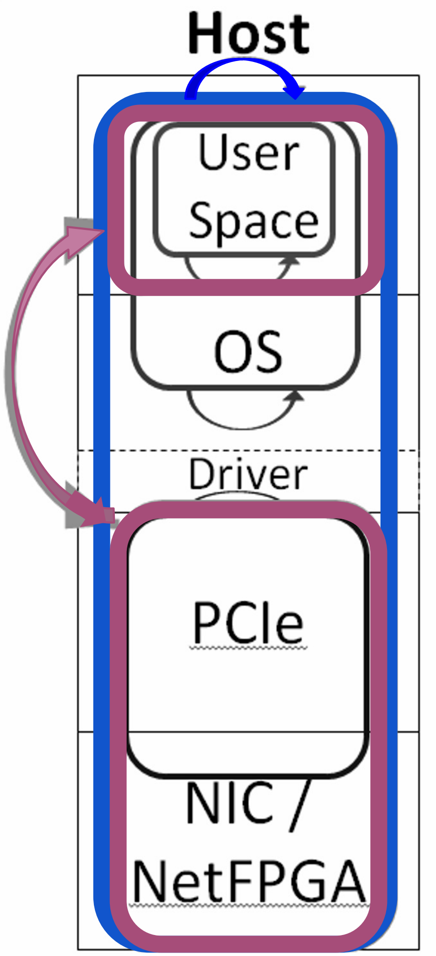
Experiment	Minimum	Median	99.9th	Tail
User space + OS (same core)	2 µs	2 µs	2 µs	68 µs
User space + OS (other core)	4 µs	5 µs	5 µs	31 µs

A Breakdown of Basic Latency Components: Host



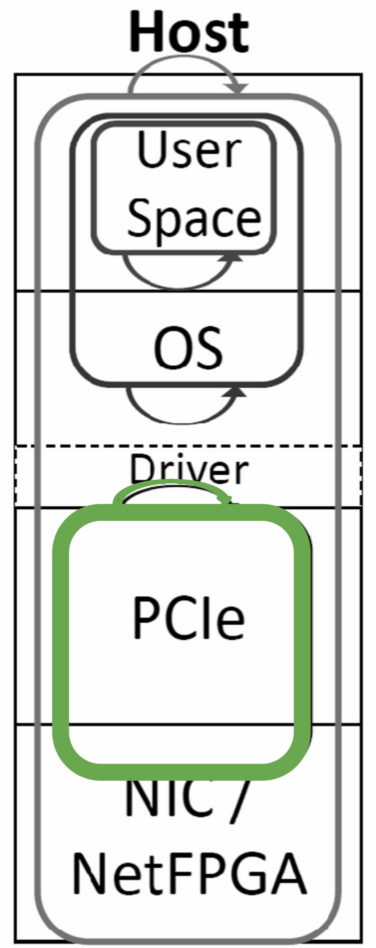
Experiment	Minimum	Median	99.9th	Tail
Host	3.9 μs	4.5 μs	21 μs	45 μs

A Breakdown of Basic Latency Components: Host



Experiment	Minimum	Median	99.9th	Tail
Host	3.9 µs	4.5 µs	21 µs	45 µs
Kernel Bypass	0.89 µs	0.94 µs	1.1 µs	5.4 µs

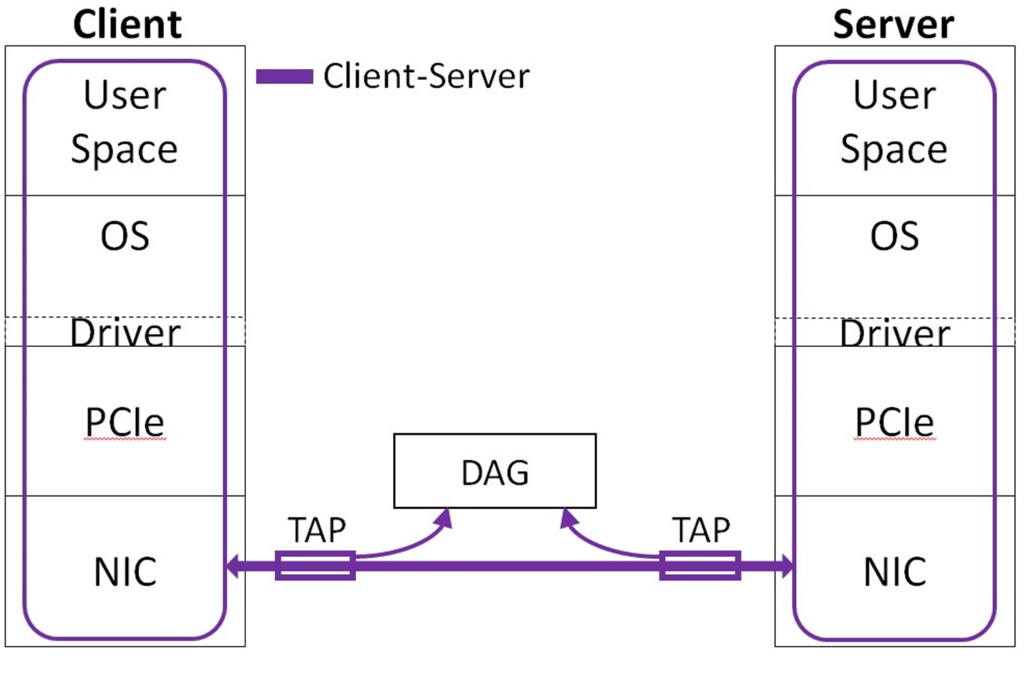
A Breakdown of Basic Latency Components: PCIe



Experiment	Minimum	Median	99.9th	Tail
Host	3.9 µs	4.5 µs	21 µs	45 µs
Kernel Bypass	0.89 µs	0.94 µs	1.1 µs	5.4 µs

Experiment	Minimum	Median	99.9th	Tail
PCIe Interconnect (64B)	**0.55 µs**	0.57 µs	0.59 µs	**0.6 µs**
PCIe Interconnect (1536B)	**0.97 µs**	0.98 µs	1.02 µs	**1.03 µs**

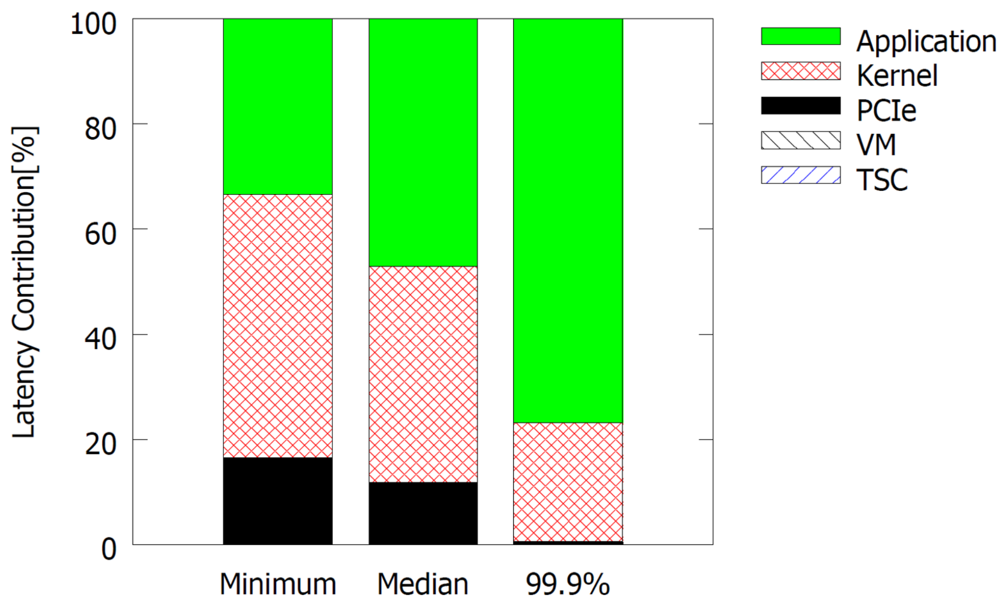
A Breakdown of Basic Latency Components: Client-Server



Experiment	Minimum	Median	99.9th	Tail
Client-Server (UDP)	7 µs	9 µs	107 µs	**203 µs**
Client-Server (Memcached)	10 µs	13 µs	240 µs	**20.3 ms**

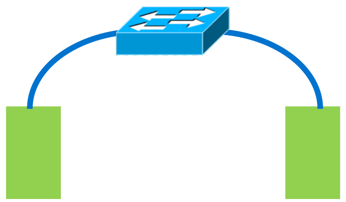
Summary end host results

- All experiments bounded by application variance (TSC)
- PCIe latency variance is low



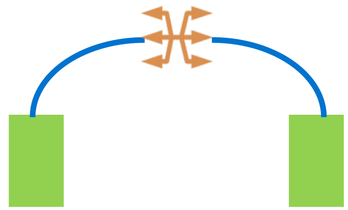
Basic Latency Components of the Network

Single Rack



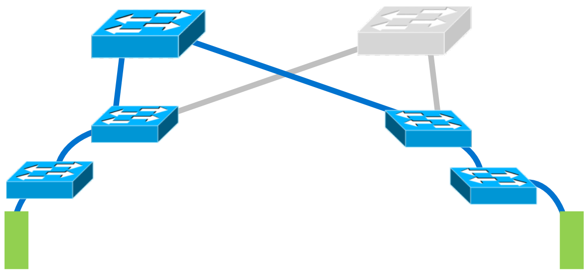
Cut Through ToR, 2m fibres, 10GE

HFT



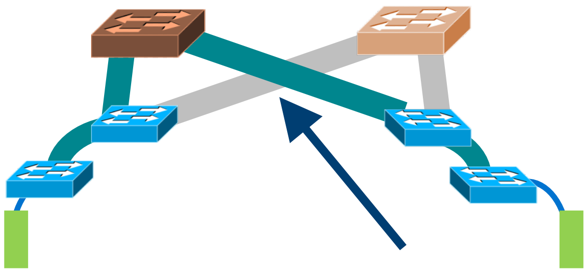
L1 *Switching*, 1.5m copper, 10GE

Fat tree

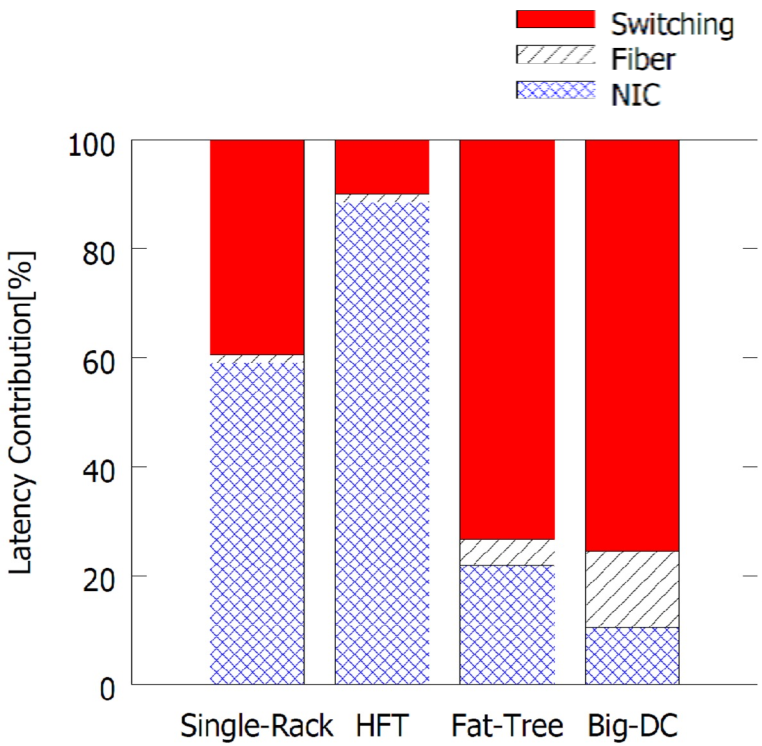


Cut Through, 2m, 5m & 10m fibres, 10GE

Big DC Fat tree



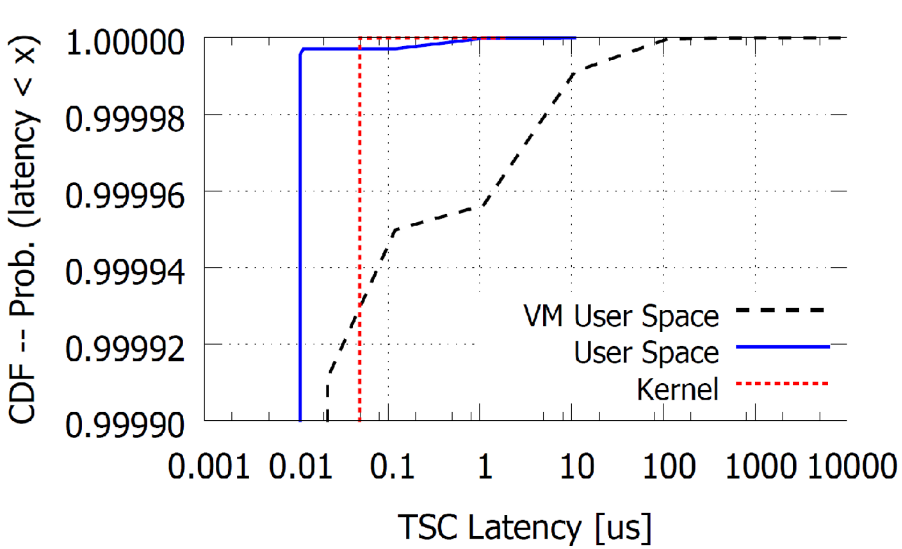
Store-forward spine, 100m /100GE fibres

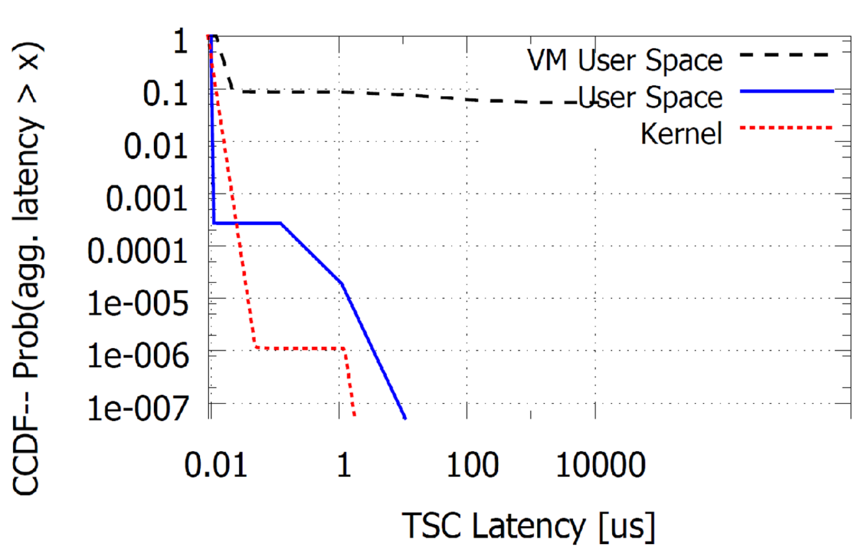


Based on median results

Tail Latency

- The long latency tail can take much more than you think
- For VM, events of 1ms or longer take almost 5% of the time.





The Good, The Bad and The Ugly

Good (≤1μs)

- Simple kernel and user space operations
- PCIe
- Single through-switch latency (no queueing)

Bad (1μs-100 μs)

- Sending packets over user space+OS
- Host latency and client-server latency
- Multi-stage network topology
  - RTT over 100m fibers

Ugly (>100μs)

- The far end of the tail, i.e. the “variance”
- Mostly in user space and within a VM.

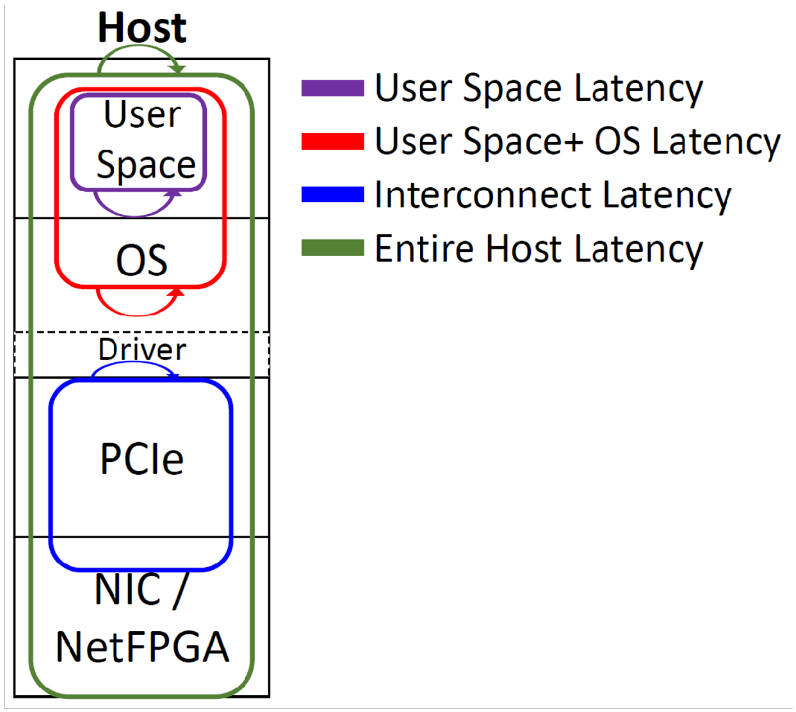
Summary

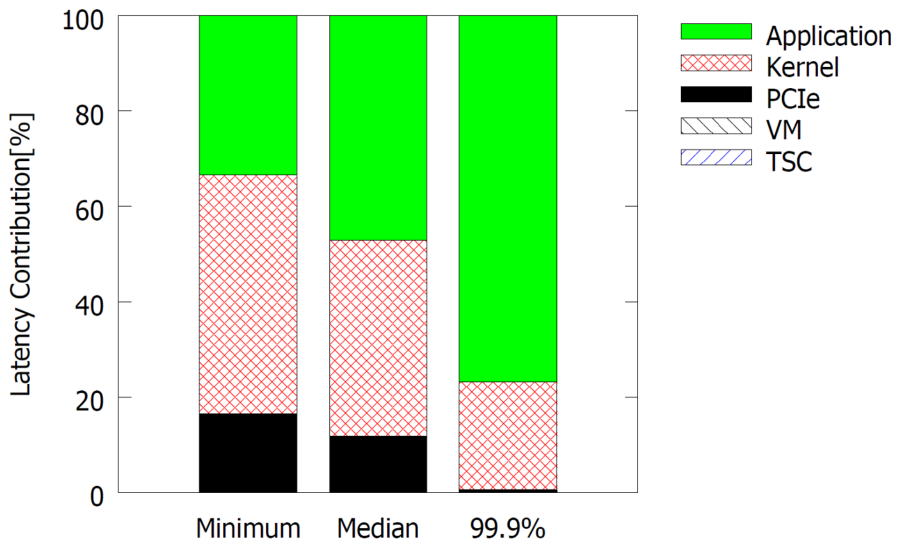
- End host latency has significantly improved over half a decade, but
  - Apps and VMs are the source of significant variance
- Fibre length matters
- “Variance” is no longer negligible
  - Fewer opportunities to improve end-host networking & interconnect

- The toolkit and reproduction environment are available!

  www.lowlatencylab.org/data/pam2017

Questions?


Host
User Space
OS
Driver
PCIe
NIC / NetFPGA
User Space Latency
User Space+ OS Latency
Interconnect Latency
Entire Host Latency


Latency Contribution[%]
100
80
60
40
20
0
Minimum
Median
99.9%
Application
Kernel
PCIe
VM
TSC

Acknowledgements

UNIVERSITY OF CAMBRIDGE

The Leverhulme Trust

EPSRC
Pioneering research and skills

SOLARFLARE®

EXABLAZE

Improving Tail Latency

- Running in real time
- Pinning to a core
- Inhibiting interrupts
- Coding practices

```
while (!done)
{
    //Read TSC once
    do_rdtscp(tsc, cpu);
    //If the gap between the current and the previous
         TSC value is above a certain threshold, save
         it
    if ((tsc - last > threshold) && (cpu == lastcpu))
       buffer[samples++] = tsc-last;
    last = tsc;
    lastcpu = cpu;
}
```

65% Min latency
50% Max latency

```
while (!done)
{
    //Read TSC twice, one immedately after the other
    do_rdtscp(tsc, cpu);
    do_rdtscp(tsc2,cpu2);
    //If the gap between the two reads is above a
         certain threshold, save it
    if ((tsc2 - tsc > threshold) && (cpu == cpu2))
       buffer[samples++] = tsc2-tsc;
}
```

Application Only

- What we do: Read TSC
- Min: 9ns
- Median: 10ns
- 99.9%: 11ns
- Max: 10’s to 100’s of μs
- 50-100 events/second > 1 μs