UNIVERSITY OF CAMBRIDGE

Where Has My Time Gone?

Noa Zilberman, Matthew Grosvenor, Neelakandan Manihatty-Bojan, Diana Andreea Popescu, Gianni Antichi, Salvator Galea, Andrew Moore, Robert Watson, Marcin Wojcik

It's Time For Low Latency

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
the past, speed-of-light delays
and unoptimized network hard-
μs round-trip times impossible.
w years datacenters will be de-
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-
nunicati

	1983	2011	Improved
CPU Speed	1x10Mhz	4x3GHz	> 1,000x
Memory Size	≤ 2MB	8GB	≥ 4,000x
Disk Capacity	≤ 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

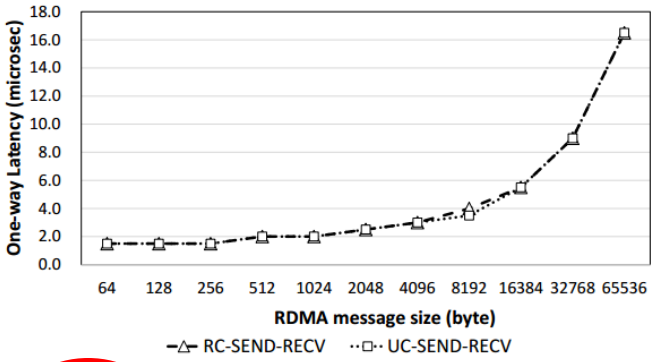


Fig. 4: Median one-way latency of RoCE RC and UC transport types

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

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

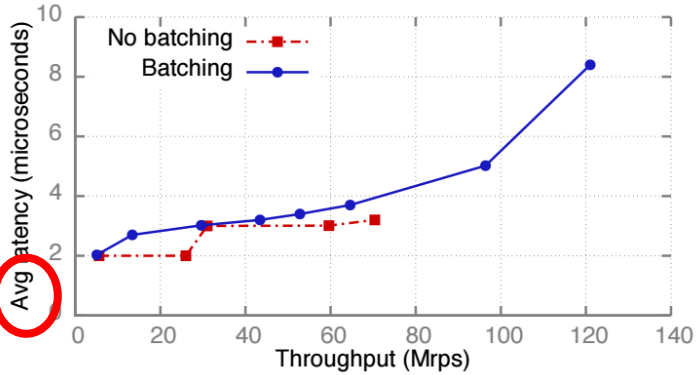
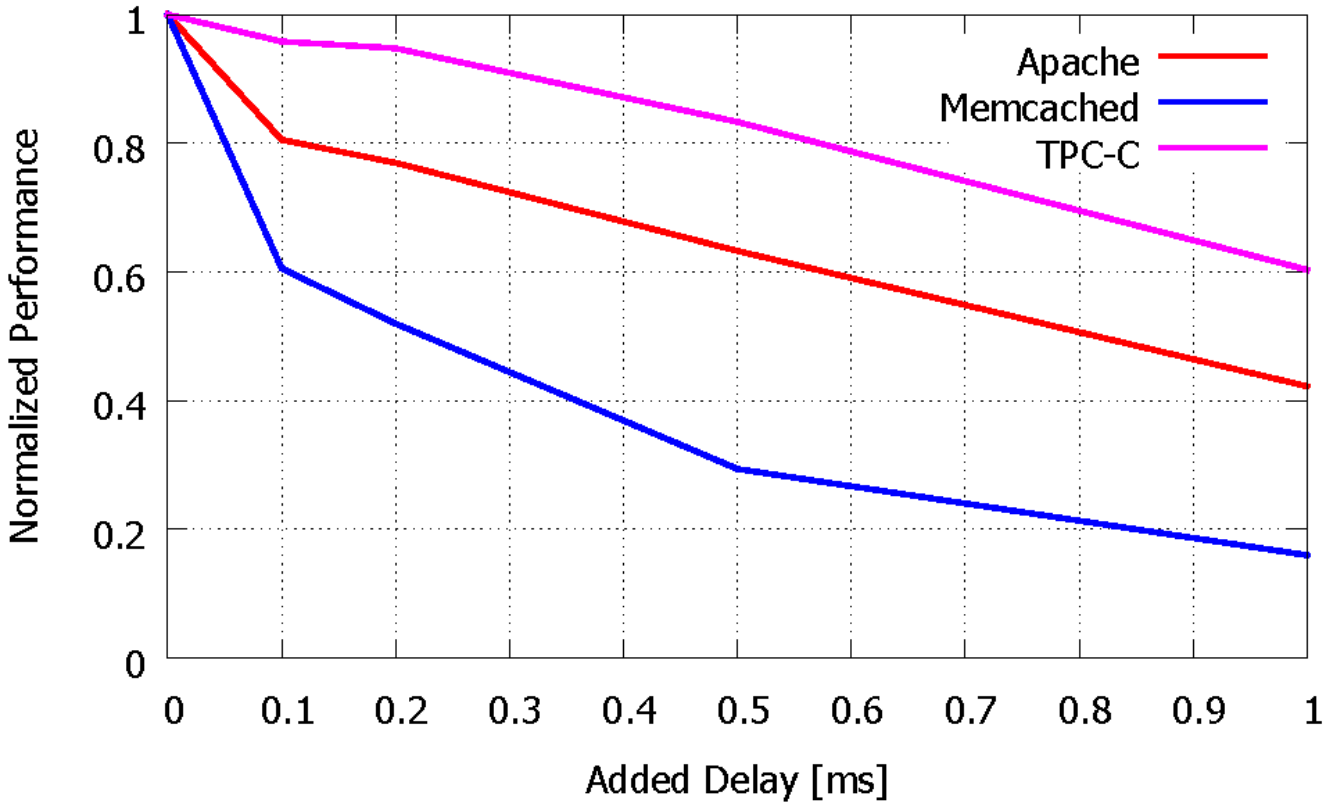


Figure 8: Impact of response batching on `Spec-S0` latency

It Usually Works

Why Should We Care About Latency?

Latency effect on Data centre applications:



DISCLAIMER

I will not talk about:

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

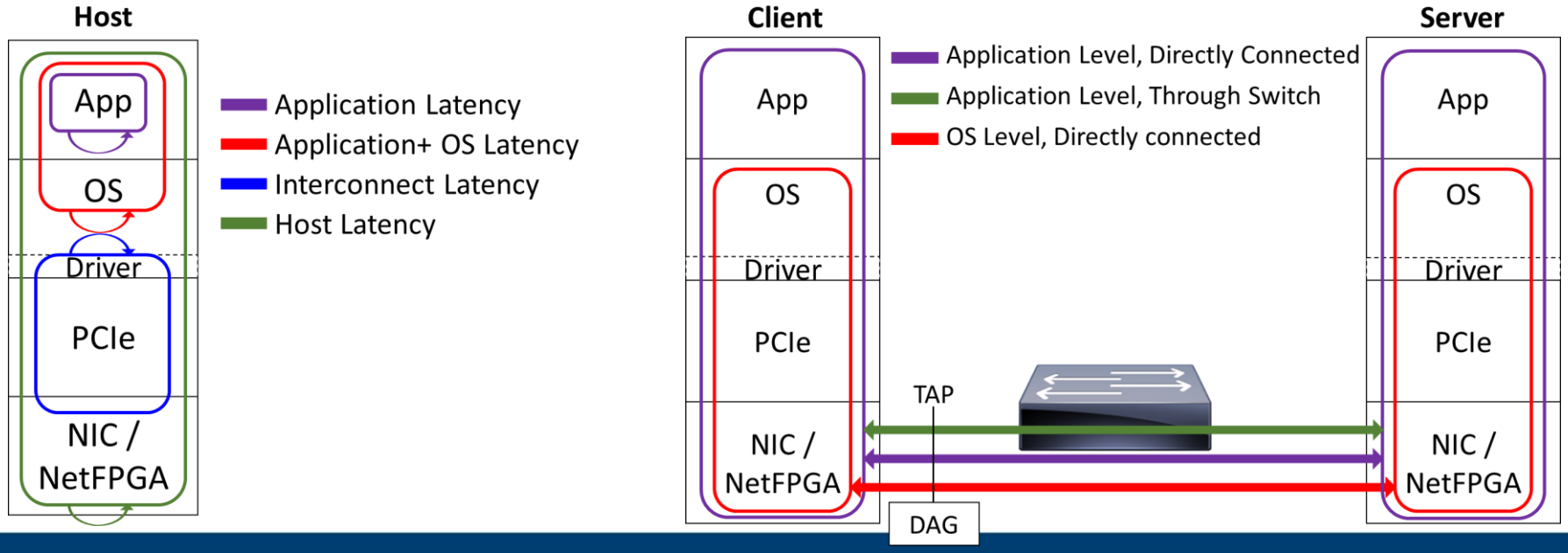
I will talk about:

- The unavoidable latency contributions
- Commodity hardware, standard coding

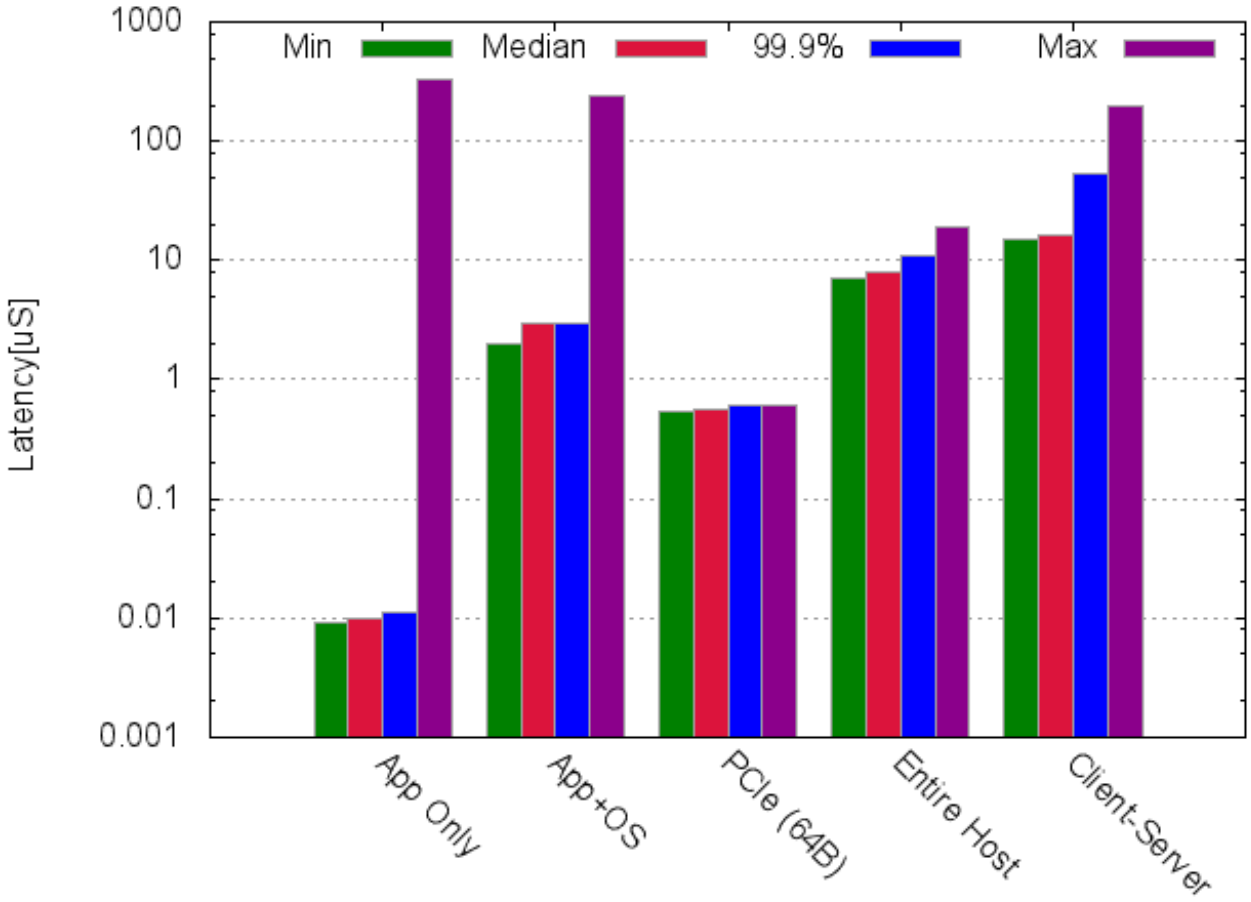
Ongoing work

Setup

- CPU: Intel Xeon E5-2637 v4, 3.5GHz
- Motherboard: SuperMicro X10-DRG-Q
- OS: Ubuntu server 14.04LTS, kernel version 4.4.0-28-generic,
- NICs: Intel X710-DA2, Solarflare SFN6122F, ExaNIC X4
- NetFPGA-SUME

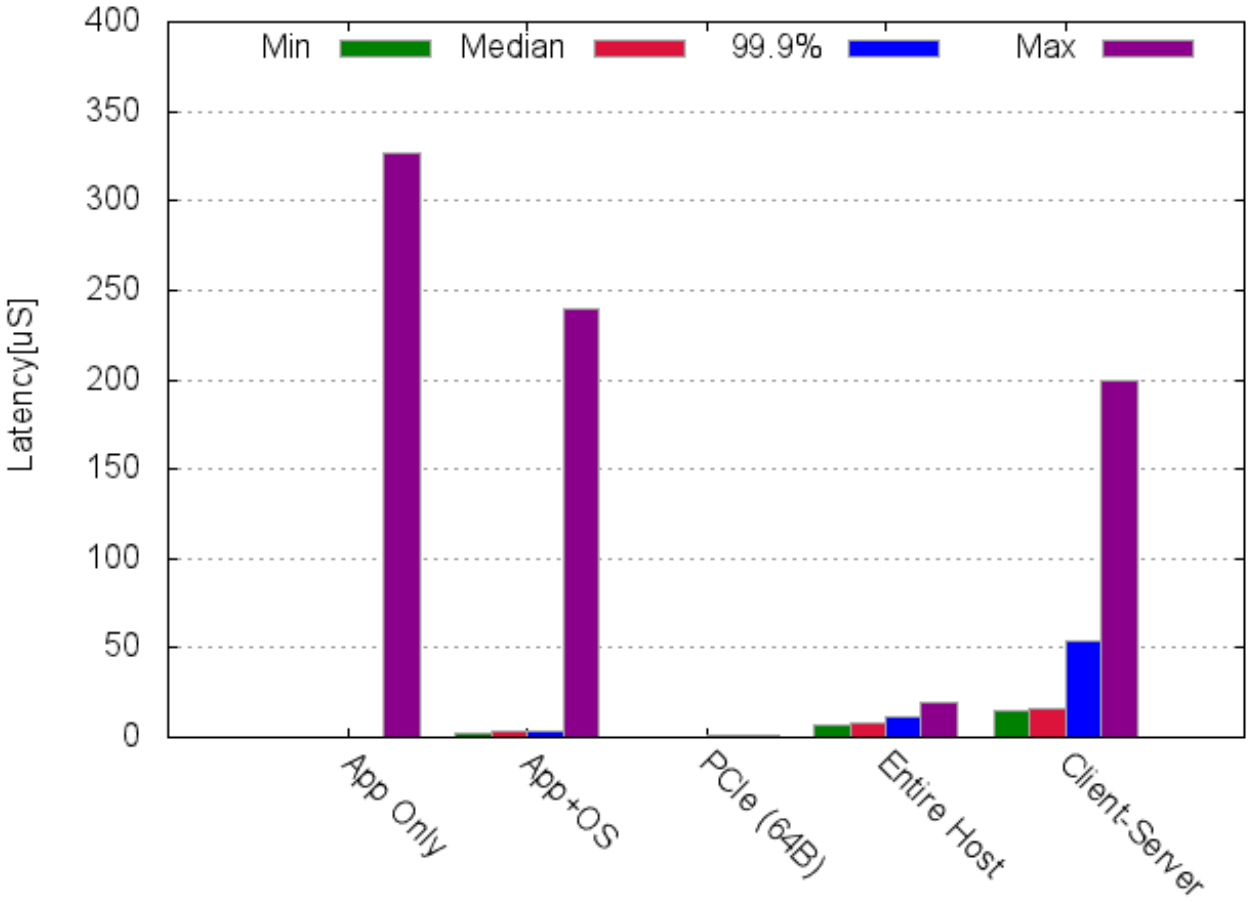


End Host Latency



Note The Log Scale!

End Host Latency



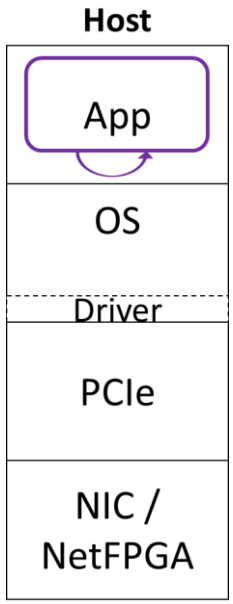
Same graph, linear scale

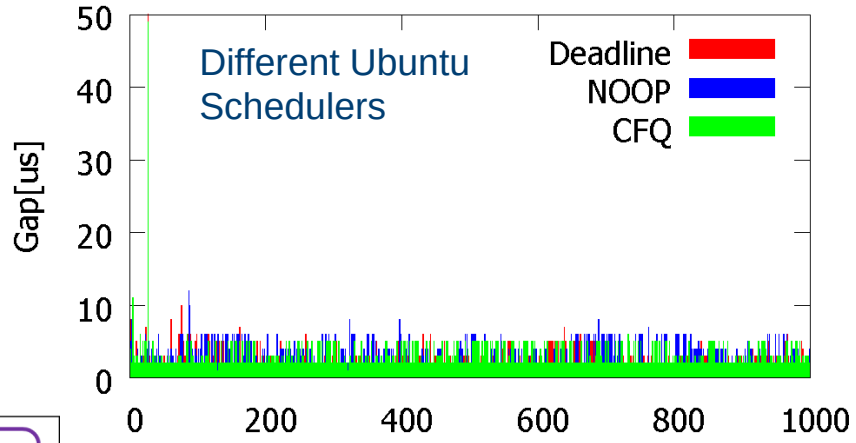
A single MAX event is x1000-x100000 the 99.9%.

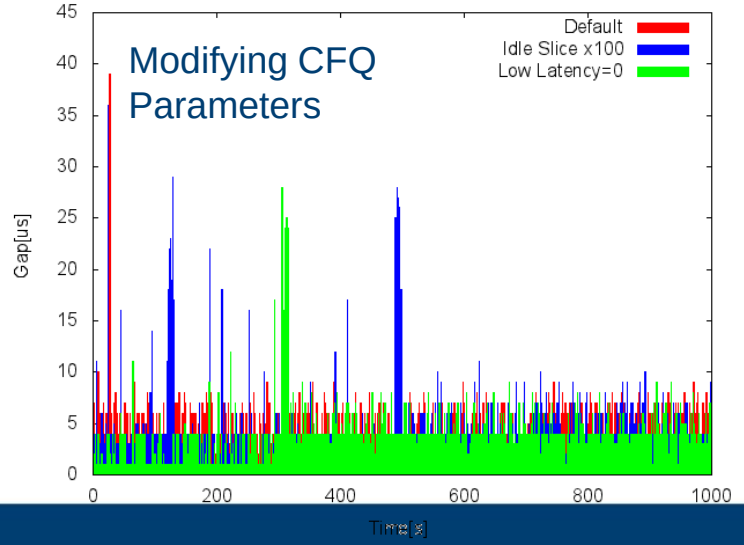
Application Only

- What we do: Read TSC
- Min: 9ns
- Median: 10ns
- 99.9%: 11ns
- Max: 10's to 100's of us

- 50-100 events/second > 1us

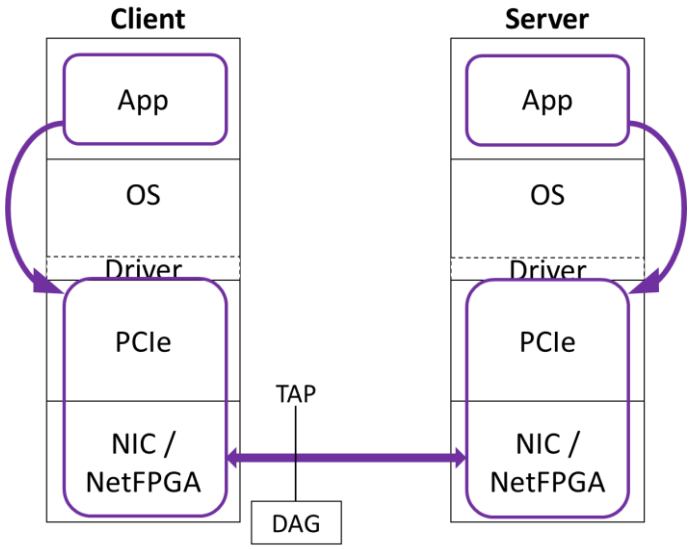




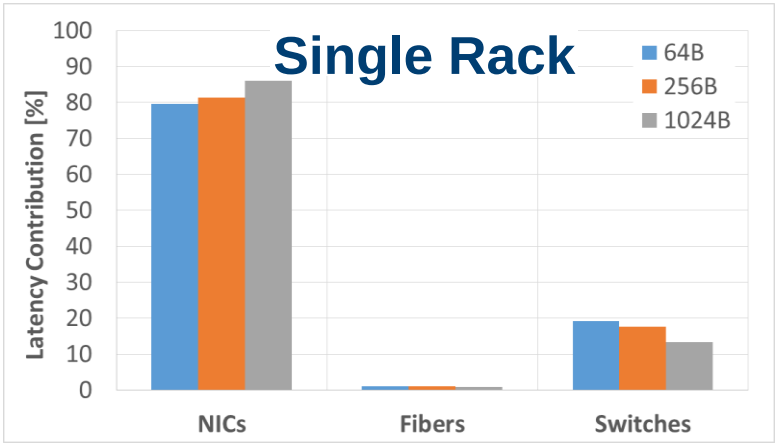


Kernel Bypass

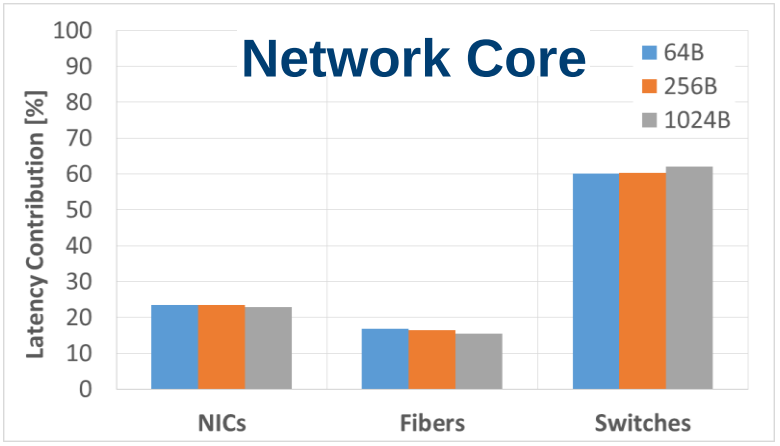
- Question: Is the OS really the problem?
- Using Exablaze's Exasock, Memcached memtier benchmark
- The good news:
  - Min: -55% (9us → 5us)
  - Median: -46% (13us → 7us)
  - 99%: : -56% (150us →66us)
- The bad news:
  - Max: No difference or worse (ms → ms)
    - But server side is 8us max (measured by DAG)
    - So what is your app doing?



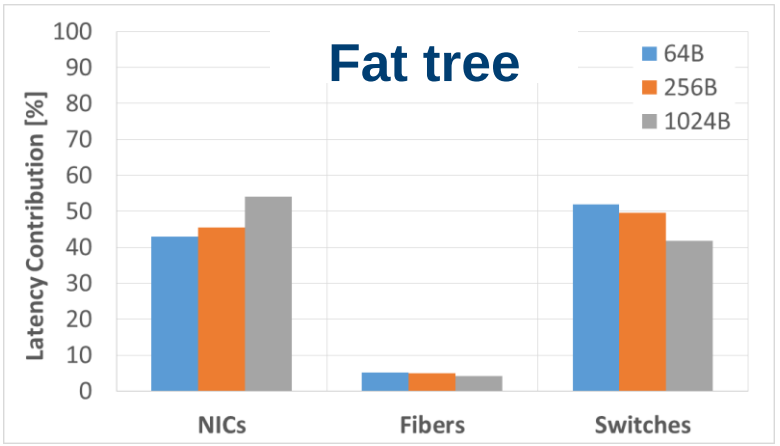
Client-Server Latency Contribution - Different Scenarios



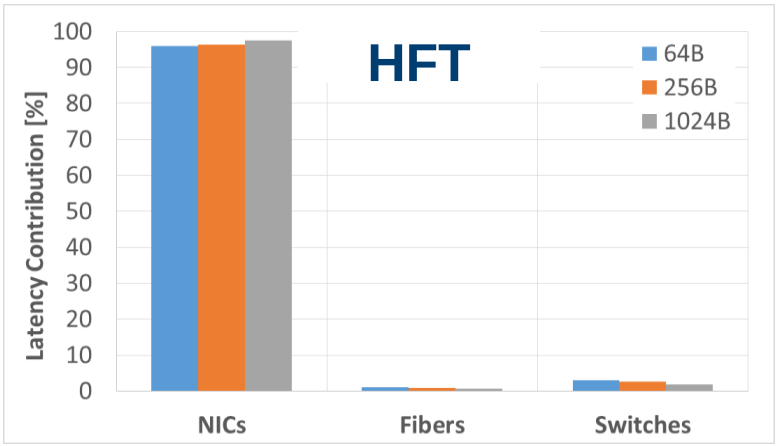
CT ToR, 2m fibres, 10GE



Store-forward, 100m fibres, 100GE



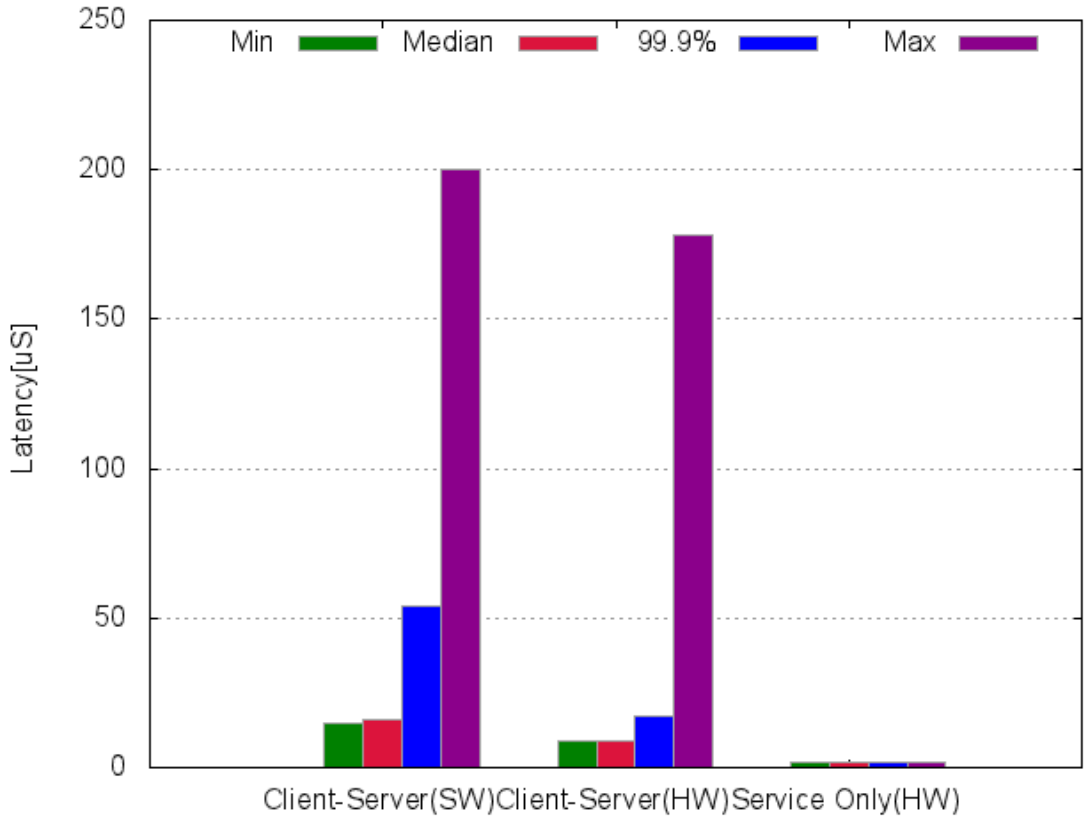
Cut Through, 2m, 5m & 10m fibres, 10GE



L1 Switching, 1.5m copper, 10GE

Services In Hardware

- Moving services to hardware reduces latency (old news)
- Moving services to hardware reduces jitter (good news)

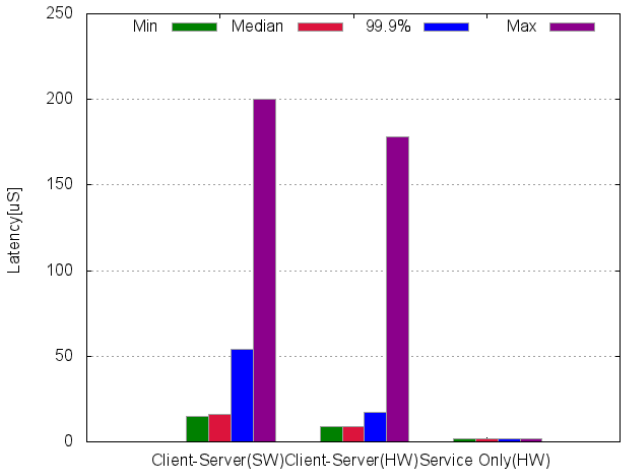


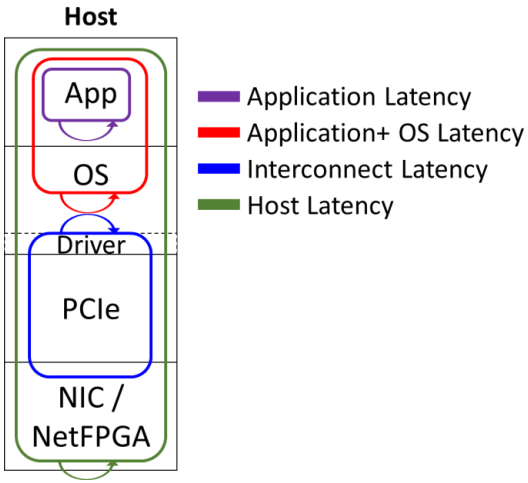
Instrumentation

- Lots of tools and instrumentation
- But they do not interoperate…
  - Poor debug ability
- We need instrumentation:
  - cross-layers
  - cross-fields (compute/network/storage)

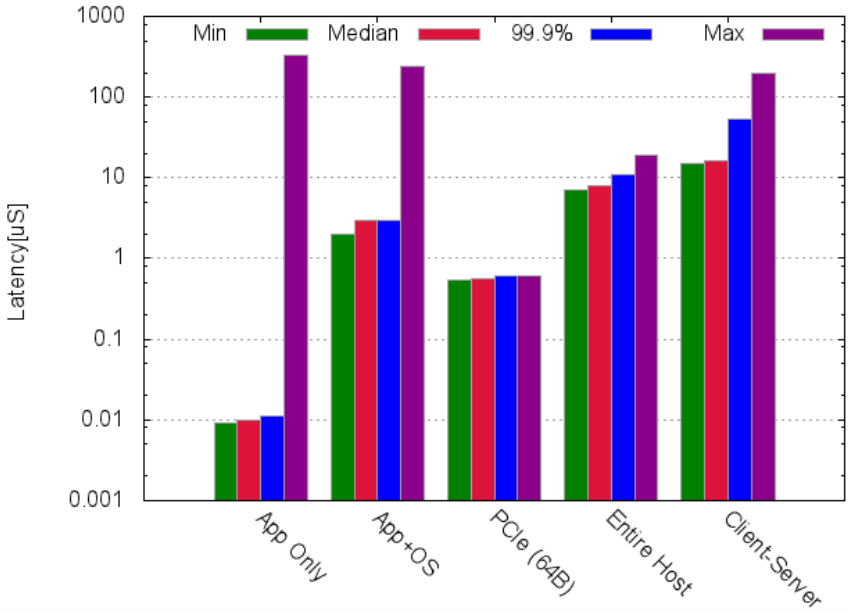
"There is an old network saying: Bandwidth problems can be cured with money. Latency problems are harder because the speed of light is fixed - you can't bribe God."

-- David Clark, MIT


250
200
150
100
50
0
Latency[uS]
Min
Median
99.9%
Max
Client-Server(SW)
Client-Server(HW)
Service Only(HW)


Host
App
OS
Driver
PCIe
NIC /
NetFPGA
Application Latency
Application+ OS Latency
Interconnect Latency
Host Latency

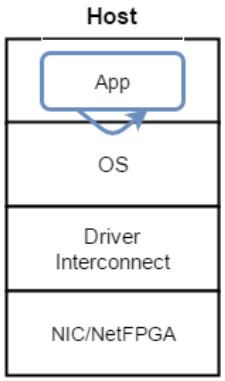
Thank You!


1000
100
10
1
0.1
0.01
0.001
Latency[uS]
Min
Median
99.9%
Max
App Only
App+OS
PCIe (64B)
Entire Host
Client-Server

Application Only

```
1   while (!done)
2   {
3       //Read TSC twice, one immedately after the other
4       do_rdtscp(tsc, cpu);
5       do_rdtscp(tsc2,cpu2);
6       //If the gap between the two reads is above a
            certain threshold, save it
7       if ((tsc2 - tsc > threshold) && (cpu == cpu2))
8          buffer[samples++] = tsc2-tsc;
9   }
```

- Code used for the evaluation
- Accurate measurement of time gaps
- May miss events

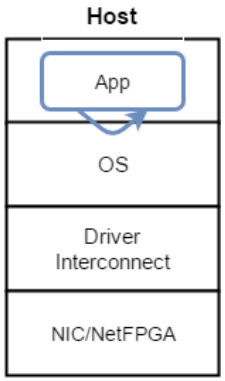


Application Only

```
1    while (!done)
2    {
3        //Read TSC once
4        do_rdtscp(tsc, cpu);
5        //If the gap between the current and the previous
             TSC value is above a certain threshold, save
             it
6        if ((tsc - last > threshold) && (cpu == lastcpu))
7           buffer[samples++] = tsc-last;
8        last = tsc;
9        lastcpu = cpu;
10   }
```

- *Alternate code: accounts for all events*
- *Also measures code-induced events*

- Min gap: +55% (9ns →14ns)
- 99.9%, “cold” buffer: +125% (4us → 9us [6us at 99%])
- Max gap, using mlock: x2 (~20us → ~40us)
- Max gap, no mlock: x16 (~20us → ~320us)



Network Latency

Datasheet Numbers:

	64B	1024B	Comments
NIC, Low latency	<1us		Solarflare SFN8522 Plus
Switch, Store-Forward	511ns	717ns	Broadcom Tomahawk, using 25GE
Switch, Cut-Through	328ns	381ns	Mellanox Spectrum, using 100GE
Switch, Cut-Through, HFT	110ns		Exablaze Fusion
Switch, L1 (Patch and Tap)	5ns		Exablaze Fusion
Transmit, 10Gbps	51ns	819ns	“Raw”, no FEC, no 64b/66b, …
Transmit, 100Gbps	5.1ns	81.9ns	
Fibre, 1m	5ns	5ns	
Fibre, 100m	500ns	500ns	

Care to calculate the latency over FatTree?