Data Acquisition Networks for Large Experiments

› **04/11/2014**

Grzegorz Jereczek

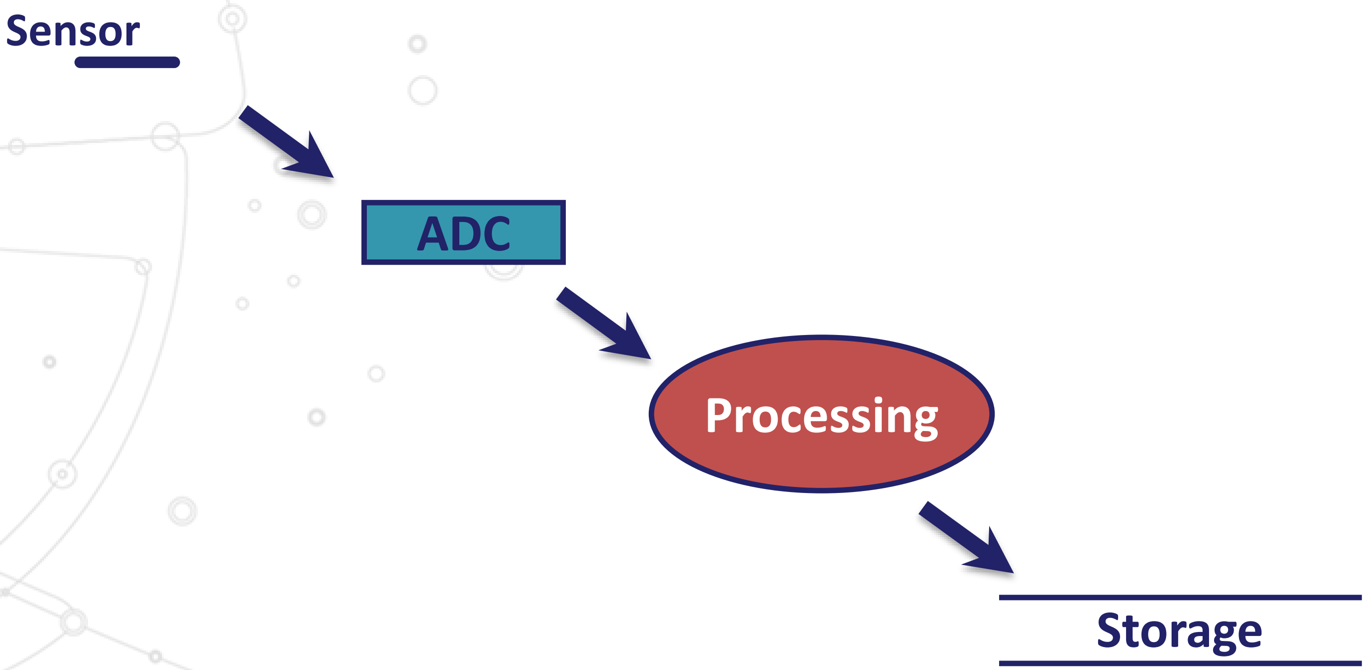
Intel-CERN European Doctorate Industrial Program

CERN openlab

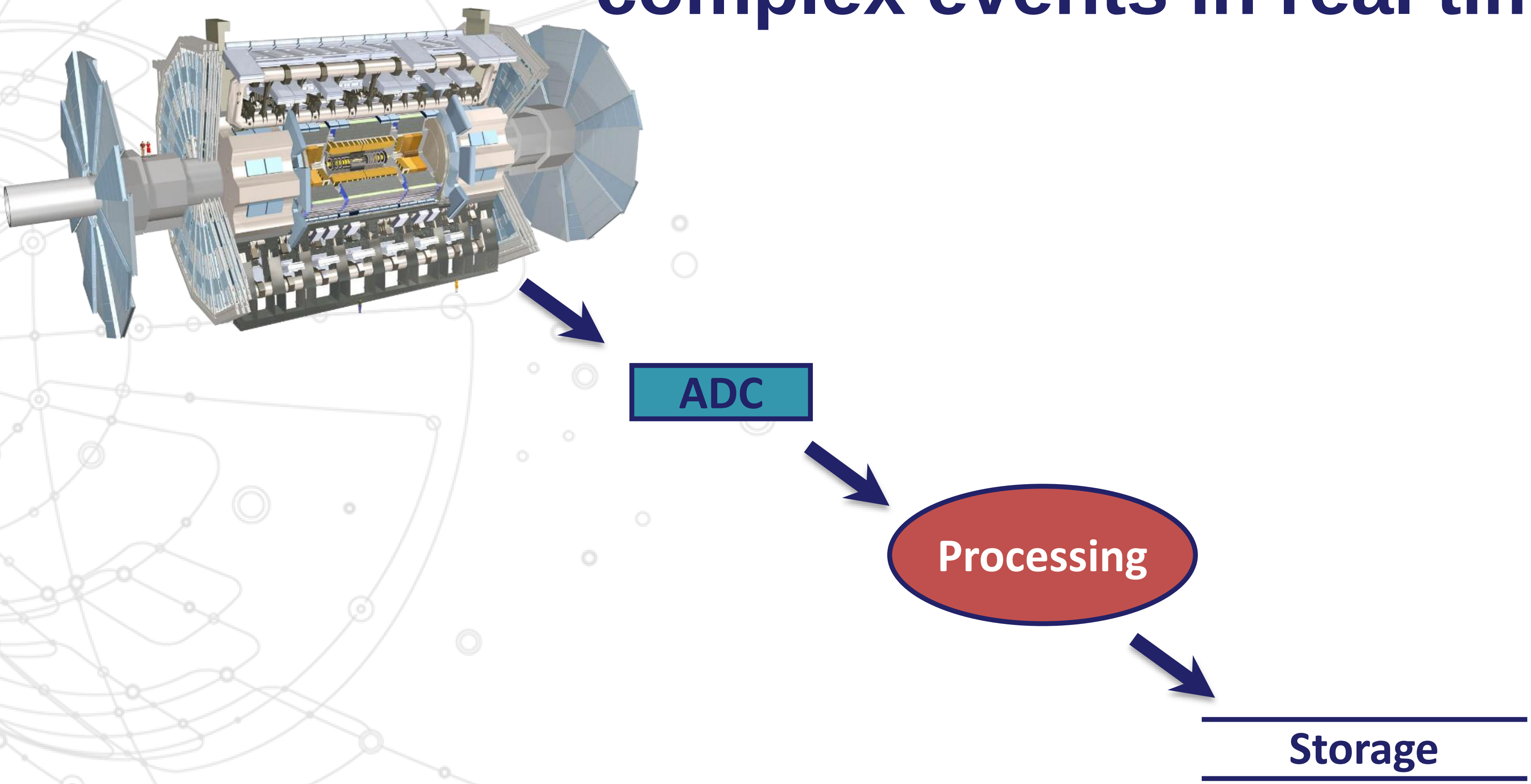
ICE-DIP is a European Industrial Doctorate project funded by the European Community's 7th Frameworkprogramme Marie Curie Actions under grant PITN-GA-2012-316596

640 Tbps

Reconstruct, analyse and select complex events in real time

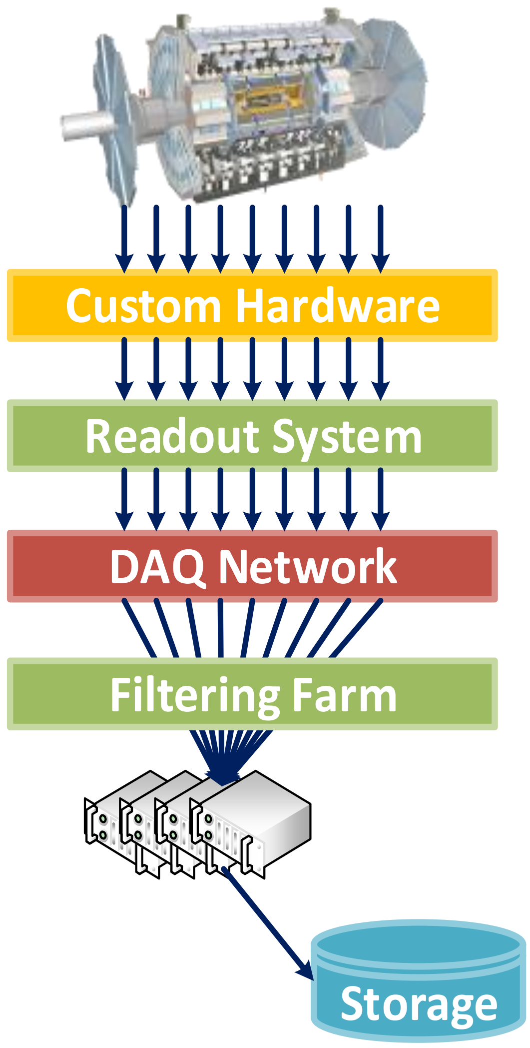


Reconstruct, analyse and select complex events in real time

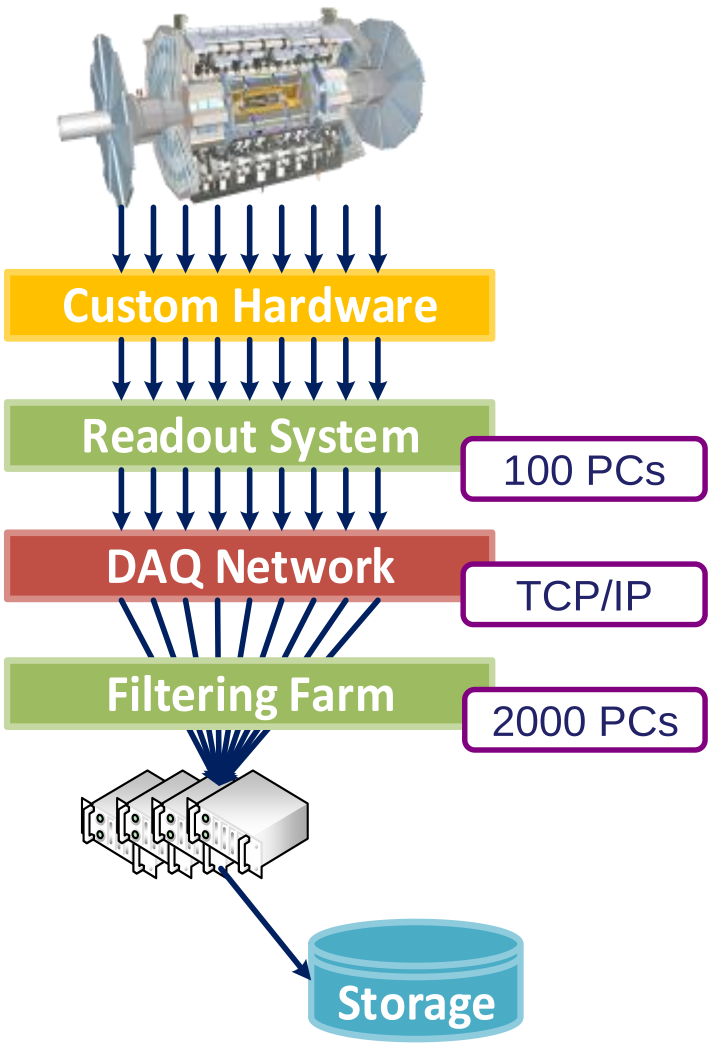


What can be done with commodity hardware?

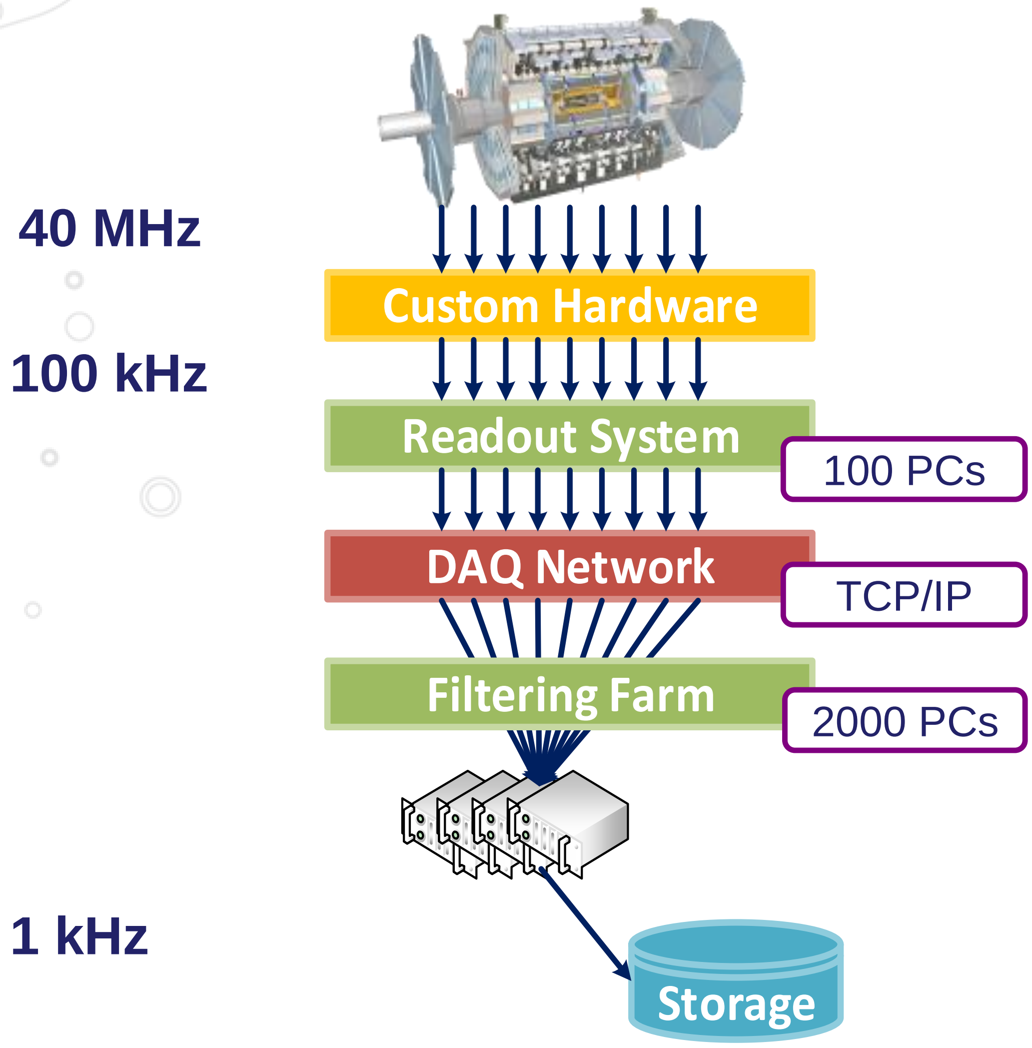
Data flow in the ATLAS experiment



Data flow in the ATLAS experiment

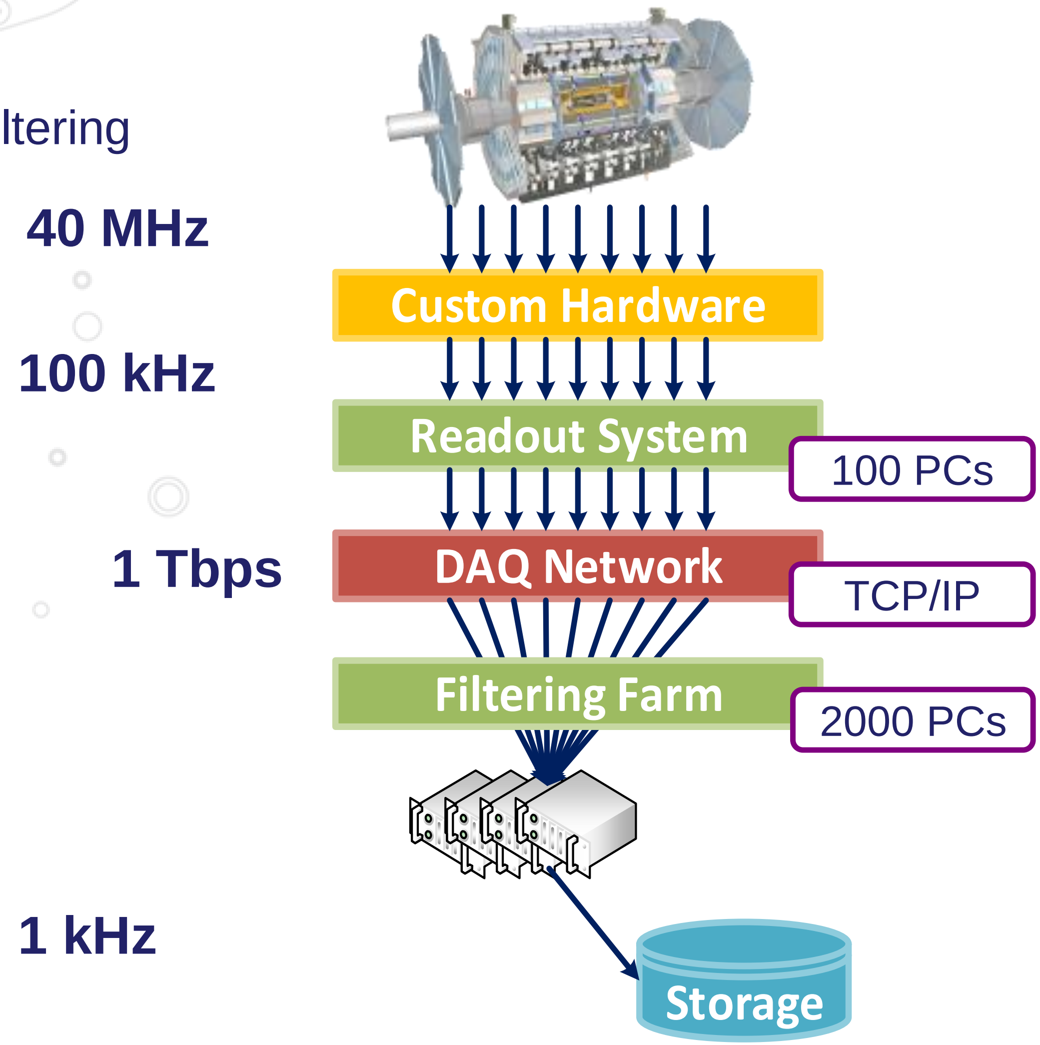


Data flow in the ATLAS experiment



Data flow in the ATLAS experiment

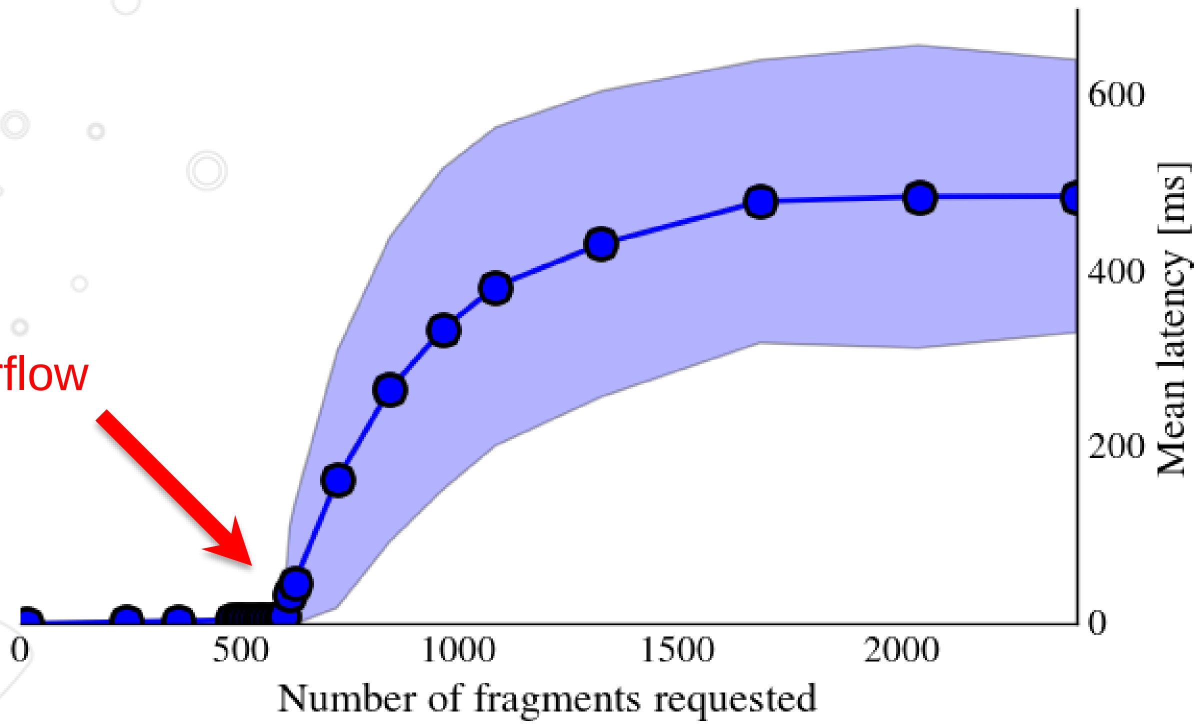
Total single event size: 2 MB
50% required on average for filtering



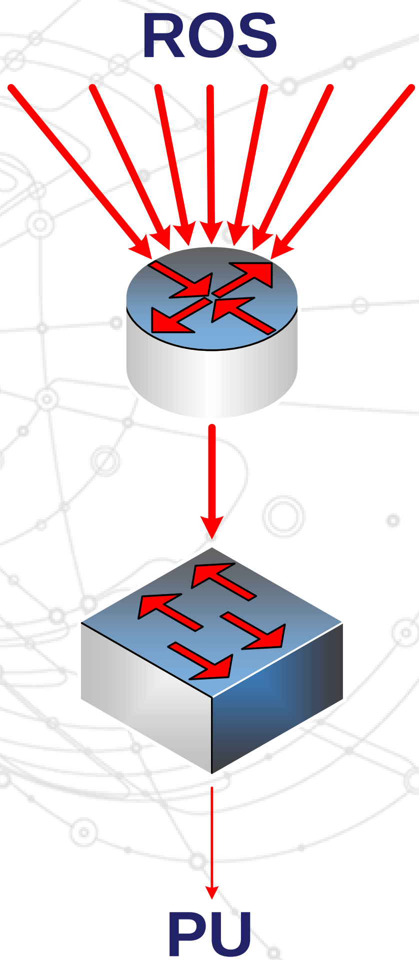
Event data collection latency

It must be kept under control
Jitter is critical

Switch buffers start to overflow

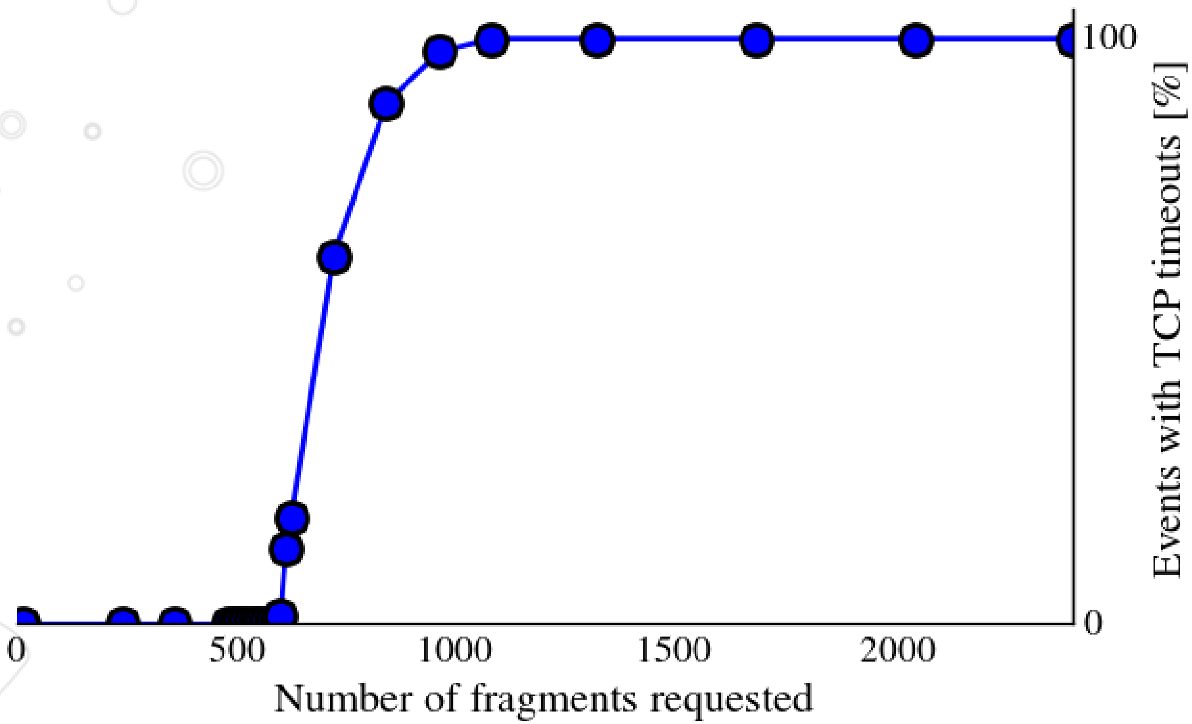


Many-to-one communication pattern



Packet drops lead to 200 ms TCP timeouts

TCP flows are too small to trigger fast retransmissions



The problem already defined as *TCP incast* in data centers

Preconditions

Large number of relatively small and synchronized flows

Sum of their windows exceeding the network's capacity

$$BDP + BufferSize < \sum_{i=1}^{n} wnd_i$$

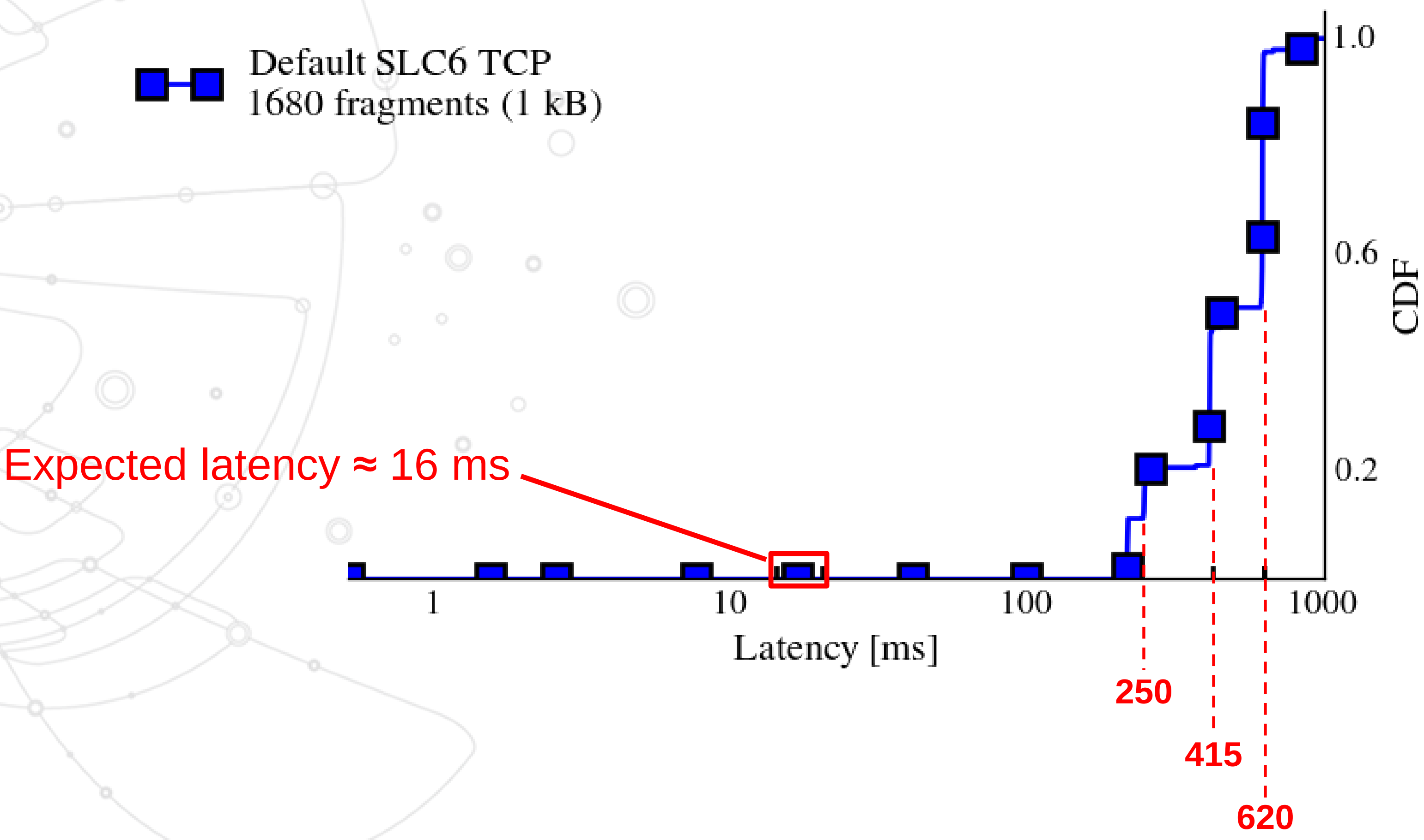
The ATLAS DAQ network

RTT: 200 microseconds

BDP: 17 TCP segments (25 kB)

For only 1 segment per flow the BDP is exceeded by a factor of 5!

Default TCP congestion control suffers from retransmission timeouts



Ways to avoid incast

$$BDP + BufferSize \geq \sum_{i=1}^{n} wnd_i$$

Increase the link speeds

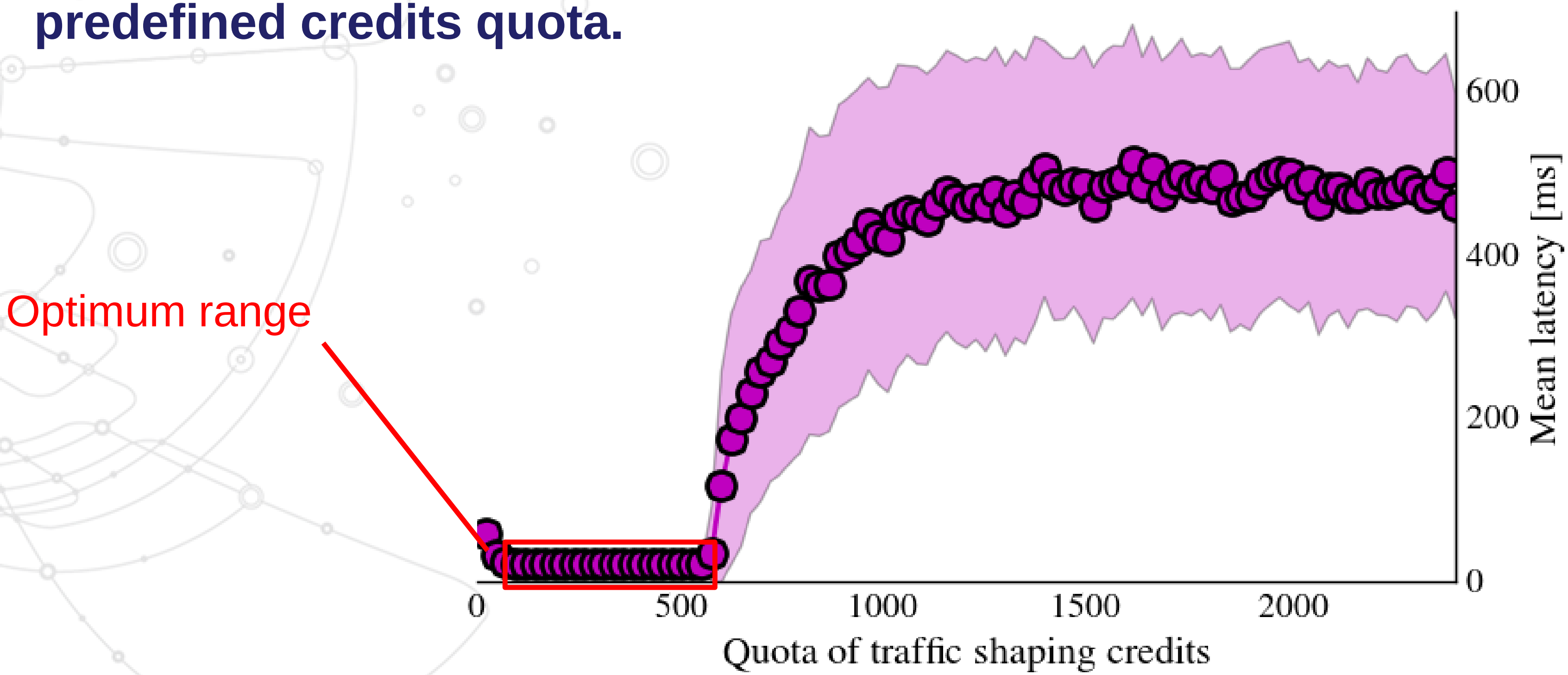
Extend the buffers

Keep the global window under control at the:

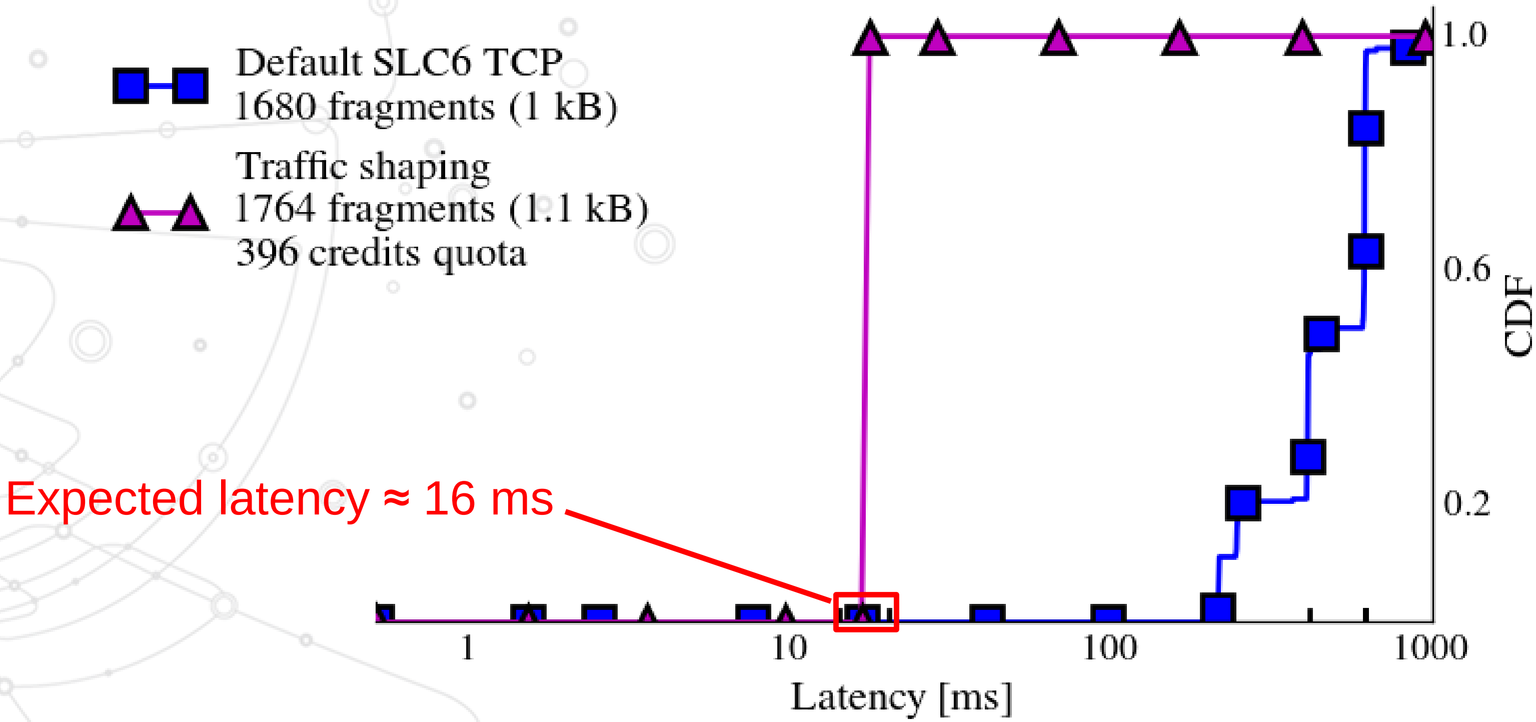
- Link layer
- Transport layer
- Application layer

Traffic shaping at the application layer

The number of outstanding event data requests from a single filtering PC must not exceed a predefined credits quota.



Traffic shaping prevents from TCP incast



Static configuration of the TCP congestion window

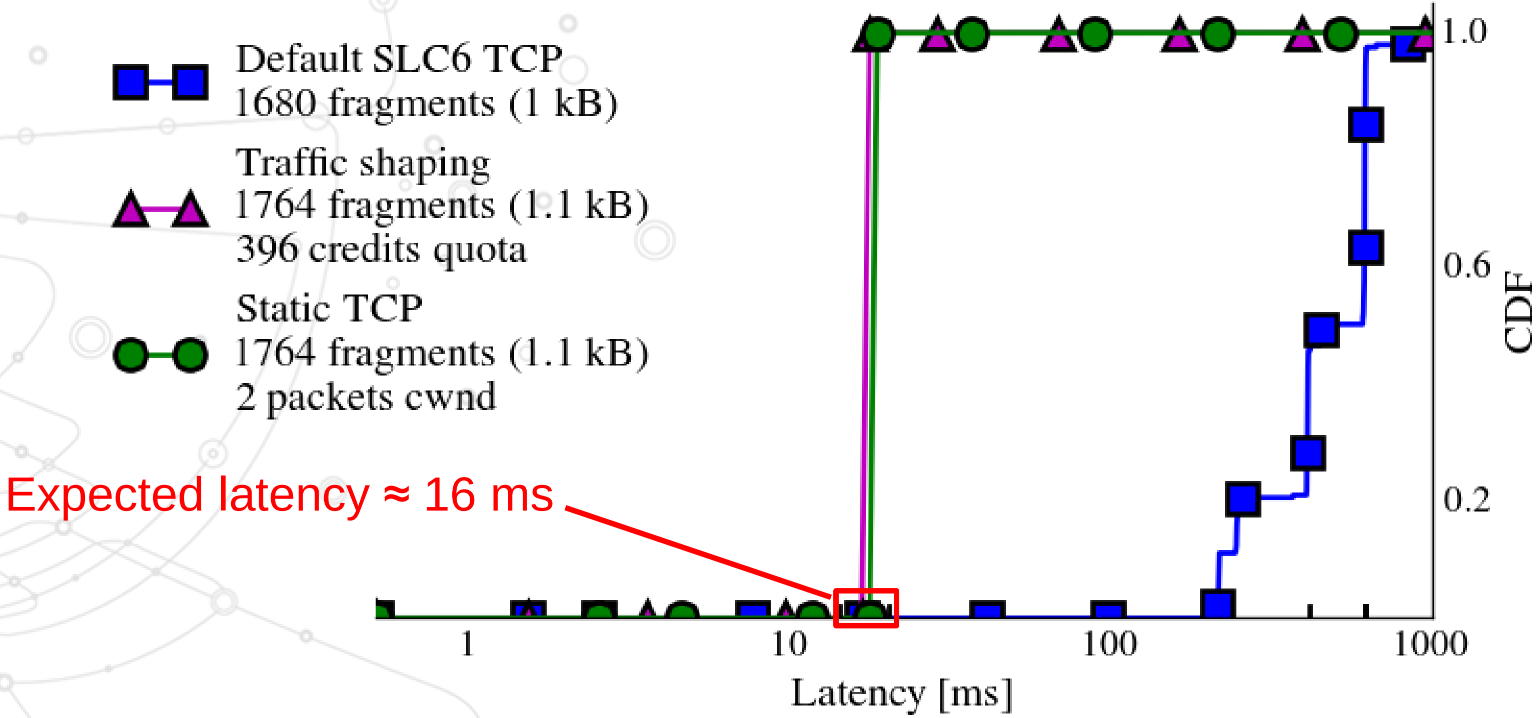
$$\boldsymbol{BDP + BufferSize} \geq \sum_{i=1}^{n} wnd_i = \boldsymbol{n \cdot cwnd} > \boldsymbol{BDP}$$

Prevents from buffer overflows (BDP + BufferSize ≥ Σ wnd_i)

Keeps the network fully utilized (n · cwnd > BDP)

Implementation as a simple loadable kernel module in Linux

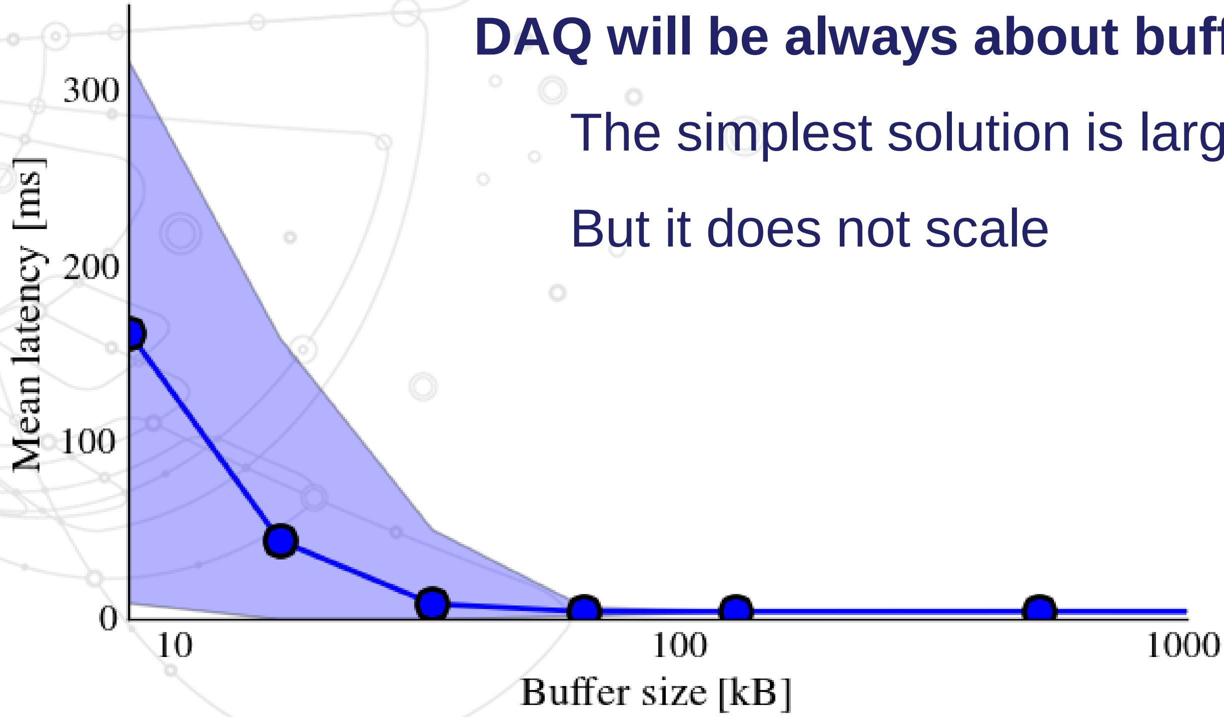
Static TCP congestion window is a valid alternative



Extending the buffers

Even with "TCP-hacks" the real problem remains unsolved

Buffer pressure moves from network to the ROS



DAQ will be always about buffering

The simplest solution is large buffers

But it does not scale

Expensive core routers can be replaced with commodity servers

The SDN/NFV trends are boosting the advance of software-based packet processing and forwarding on commodity servers.

Fast packet processing on x86 in userland

- Intel DPDK http://dpdk.org/

A solution tailored for DAQ can be therefore designed

- Huge buffering capabilities
- Per flow queues

120 Gbps IP forwarding with 12 cores on a commodity server

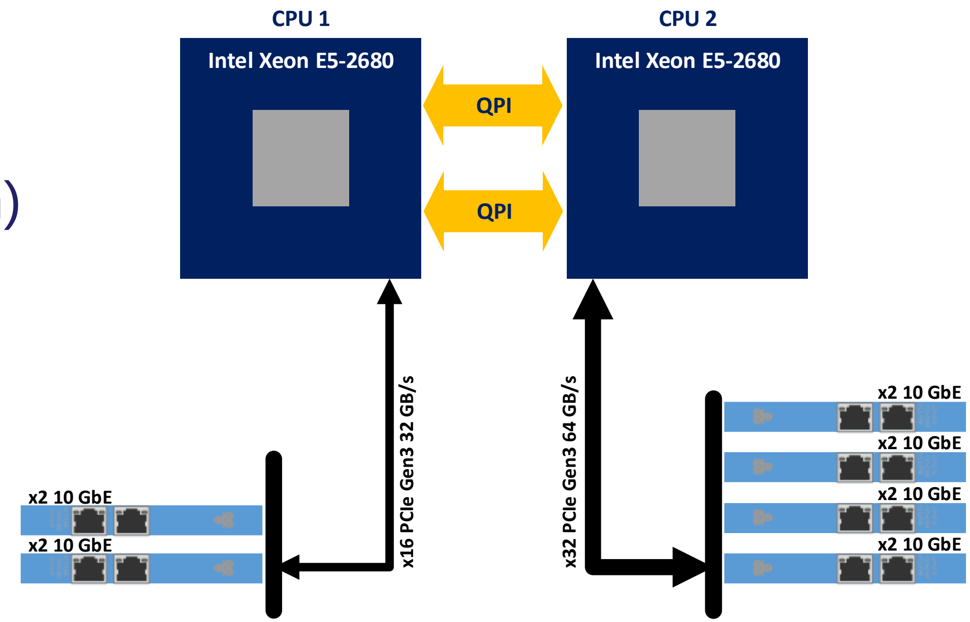
DPDK-based software switch

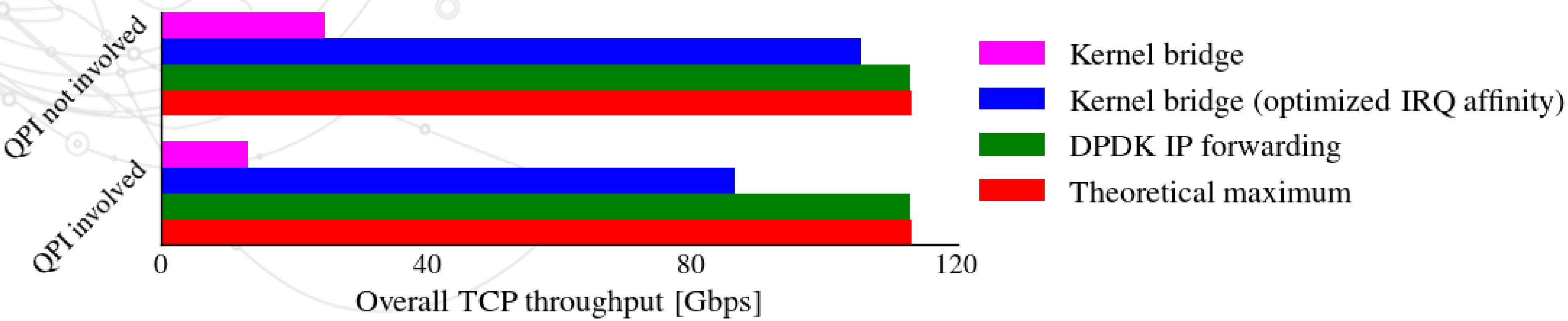
2x Xeon 2.70 GHz (8 cores each)

12x 10GbE (Intel 82599)

Single port load

20 Gbps bidirectional





Questions?