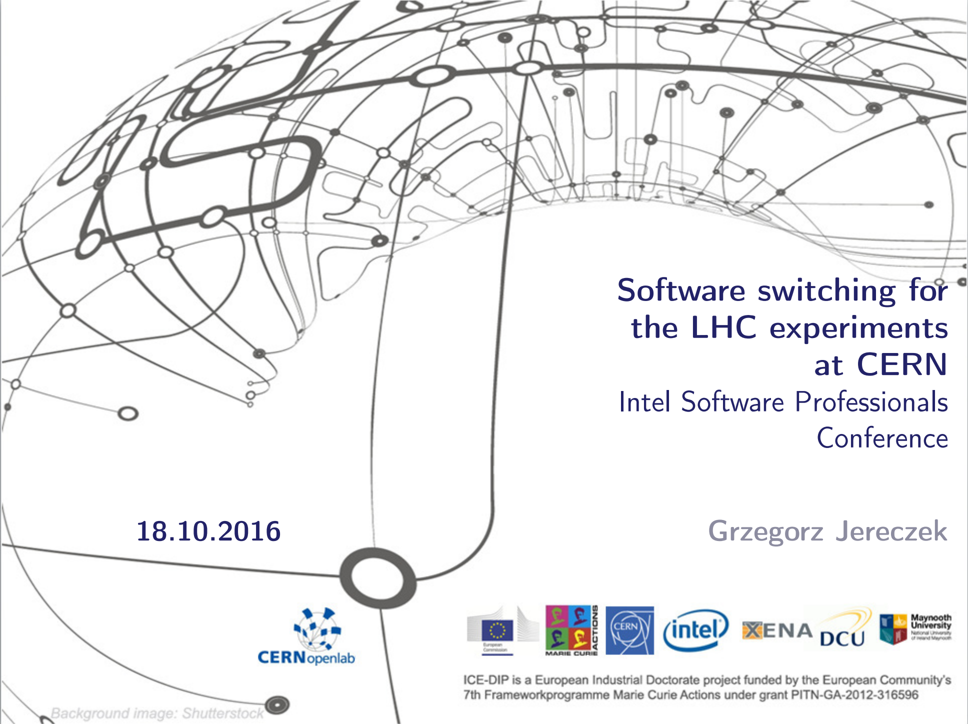


Software switching for the LHC experiments at CERN

Intel Software Professionals Conference

18.10.2016

Grzegorz Jereczek

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

Background image: Shutterstock

Image: CERN

640 Tbps?

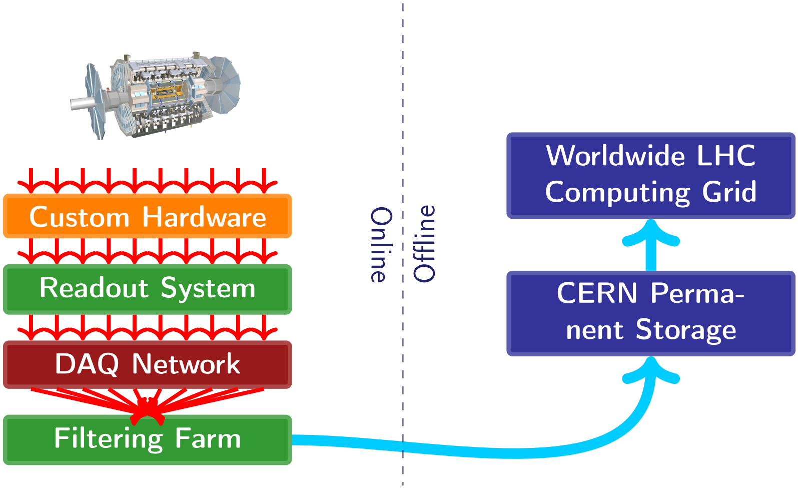
Image: CERN

Outline

1. Introduction
2. A lossless switch for data acquisition networks
3. A lossless network for data acquisition
4. Conclusions and outlook

Introduction

Data flow of the ATLAS experiment



Custom Hardware

Readout System

DAQ Network

Filtering Farm

Online

Offline

Worldwide LHC Computing Grid

CERN Permanent Storage

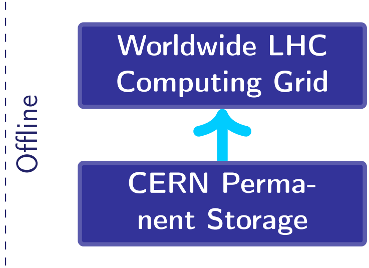
Reconstruct, analyse and select complex events in real time.

SDN already entering offline processing

25 PB of data per year stored by the LHC experiments.

Networks distribute the data to users around the world for offline analysis.

SDN can help the identification of elephant flows to optimize the distributed data analysis.



Offline

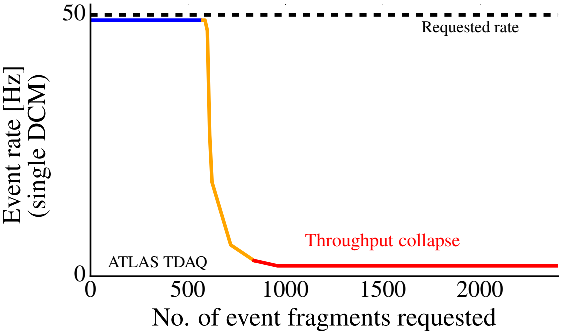
Worldwide LHC Computing Grid

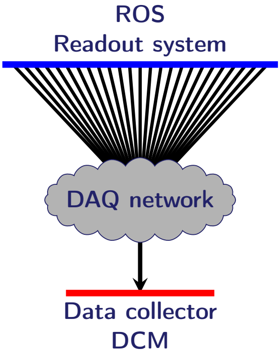
CERN Permanent Storage

More information: *Research community looks to SDN to help distribute data from the Large Hadron Collider*

Incast congestion in data acquisition networks

Synchronized many-to-one bursts from ROS overflow packet buffers in the network.





General approaches

Flow control: Ethernet Pause/PFC, InfiniBand
Designed to absorb fluctuations,
HoL blocking

Congestion control: traffic shaping, TCP variants, Ethernet DCB
HW/SW support, dependent on fragment sizes/counts and
network architecture, sender-side buffering

Deep buffers
Best throughput, simple push architecture,
but rare and/or expensive devices

General approaches

Flow control: Ethernet Pause/PFC, InfiniBand
Designed to absorb fluctuations,
HoL blocking

Congestion control: traffic shaping, TCP variants, Ethernet DCB
HW/SW support, dependent on fragment sizes/counts and
network architecture, sender-side buffering

Deep buffers
Best throughput, simple push architecture,
but rare and/or expensive devices

Can we use the DRAM memory as a packet buffer?

COTS-servers as network switches for DAQ

High I/O performance of modern servers

Memory: 540 Gbps (DDR4-2133, 4 channels/CPU)

PCIe: 63 Gbps, even 10 slots on a board (PCIe Gen3 x8)

COTS-servers as network switches for DAQ

✓ **High I/O performance of modern servers**

Software availability

Production quality software switch: **Open vSwitch (OvS)**

Frameworks for fast packet processing: **DPDK**

Network control: **Software Defined Networking (SDN)**

COTS-servers as network switches for DAQ

✓ **High I/O performance of modern servers**

✓ **Software availability**

Production quality software switch: **Open vSwitch (OvS)**
→ Optimize for throughput

Frameworks for fast packet processing: **DPDK**
→ Buffering mechanism

Network control: **Software Defined Networking (SDN)**
→ Use the global view of the network

Goal: *Lossless network based on software switches with large packet buffers in DRAM optimized for DAQ*

A lossless switch for data acquisition networks

Optimizing Open vSwitch for DAQ

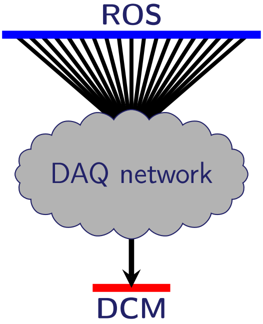
Some optimizations to datapath for high-throughput.

Queueing

Packets queued in the DPDK's rings.

A single ring dedicated to a single DCM.

Rings are distinct ports (*daqring port*).



Optimizing Open vSwitch for DAQ

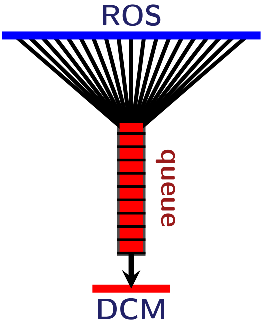
Some optimizations to datapath for high-throughput.

Queueing

Packets queued in the DPDK's rings.
→ Size of an event

A single ring dedicated to a single DCM.

Rings are distinct ports (*daqring port*).



Optimizing Open vSwitch for DAQ

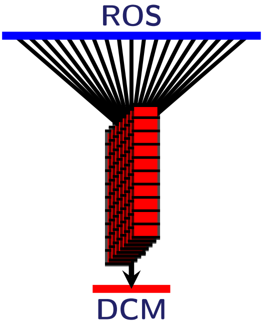
Some optimizations to datapath for high-throughput.

Queueing

Packets queued in the DPDK's rings.
→ Size of an event

A single ring dedicated to a single DCM.
→ Hundreds of rings for the entire system
→ Rate limitation possible

Rings are distinct ports (*daqring port*).



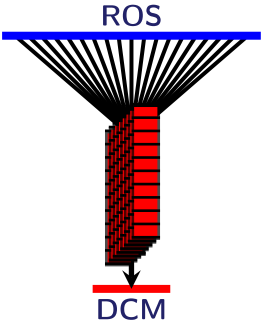
Optimizing Open vSwitch for DAQ

Some optimizations to datapath for high-throughput.

Queueing

Packets queued in the DPDK's rings.
→ Size of an event

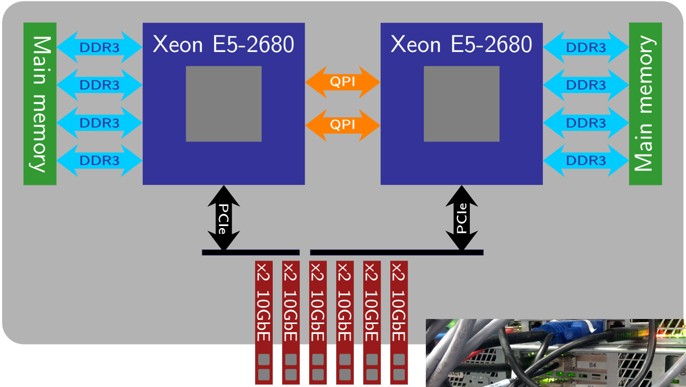
A single ring dedicated to a single DCM.
→ Hundreds of rings for the entire system
→ Rate limitation possible



Rings are distinct ports (*daqring port*).

→ Programming and optimizing flows with **OVSDB** and **OpenFlow**

12 x 10GbE prototype



Note: all results for large packets (MTU: 1500B).

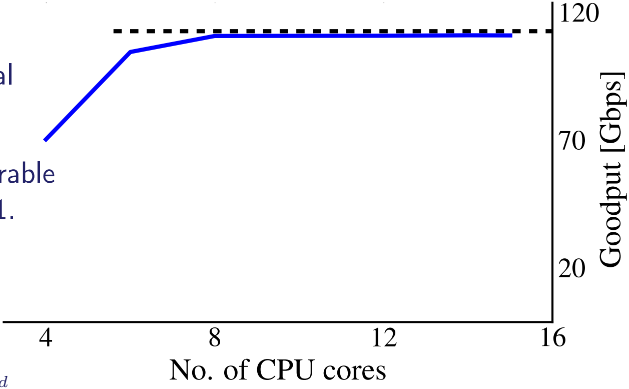
All-to-all incast: 12 ROSes and 144 DCMs

No packet drops: **lossless operation**.

98% of theoretical goodput with 8 CPU cores.

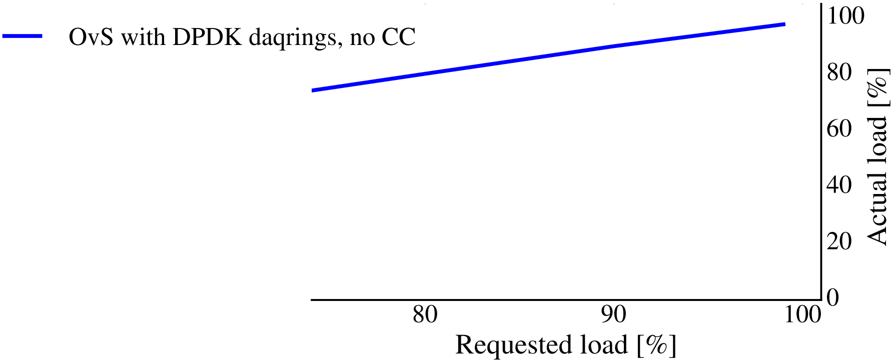
Utilizing full bidirectional bandwidth of 120Gbps.

Bandwidth-wise, comparable to ATLAS DAQ in run 1.

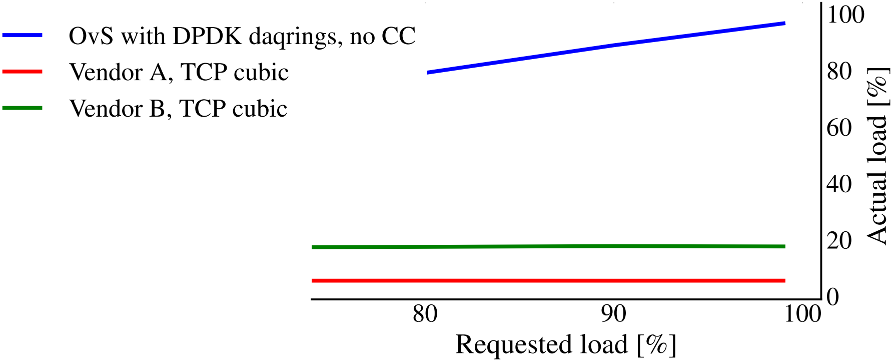


$$Goodput = \frac{event\ data\ collected}{collection\ time}$$

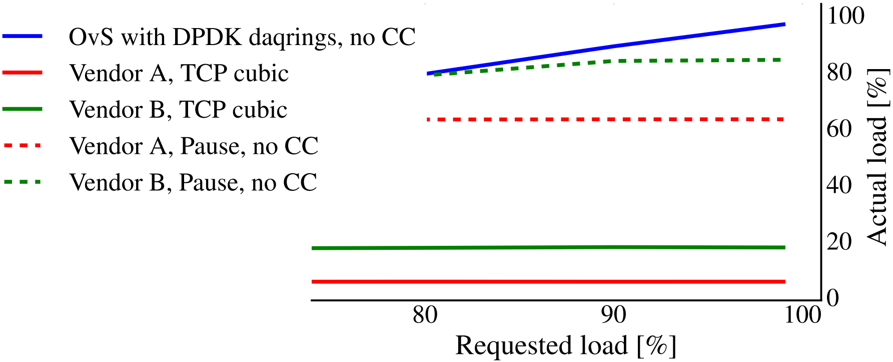
The lossless software switch outperforms regular switches with hardware flow control



The lossless software switch outperforms regular switches with hardware flow control



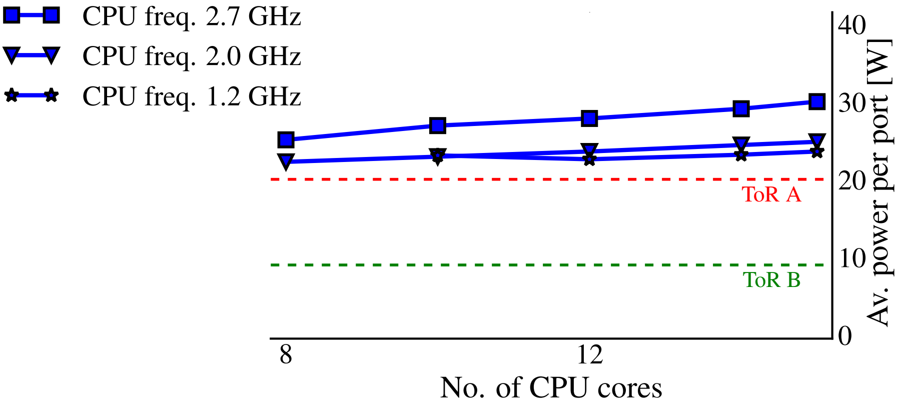
The lossless software switch outperforms regular switches with hardware flow control



Power consumption

Min. 95% of theoretical DAQ goodput in all cases.

Can be further optimized (*less polling*).



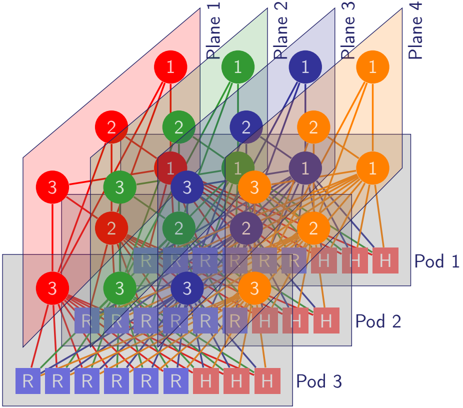
A lossless network for data acquisition

Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

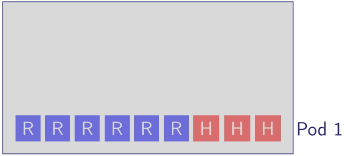


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

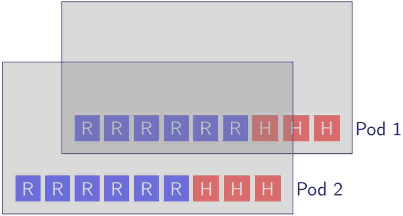


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

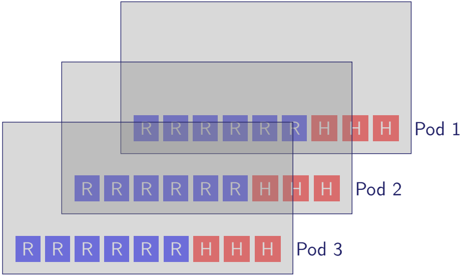


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

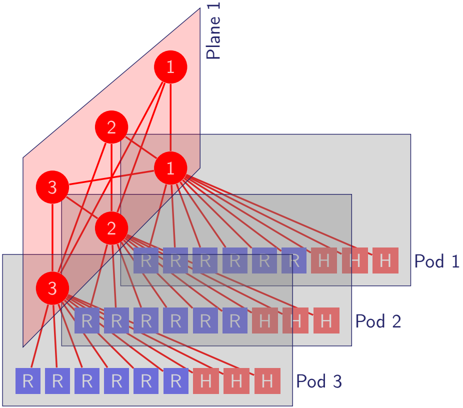


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

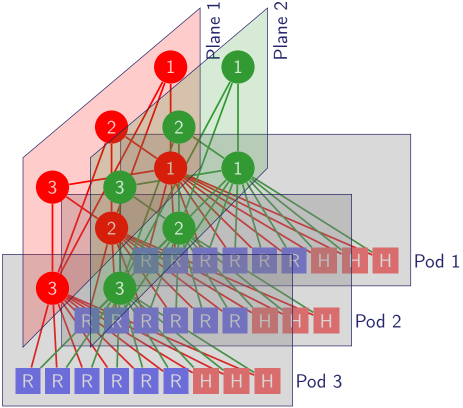


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

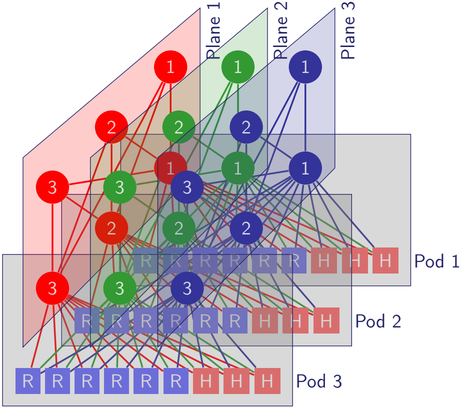


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

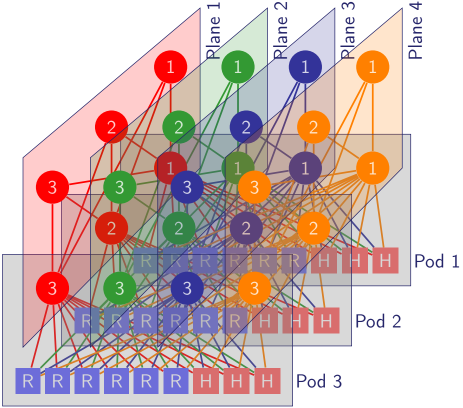


Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).



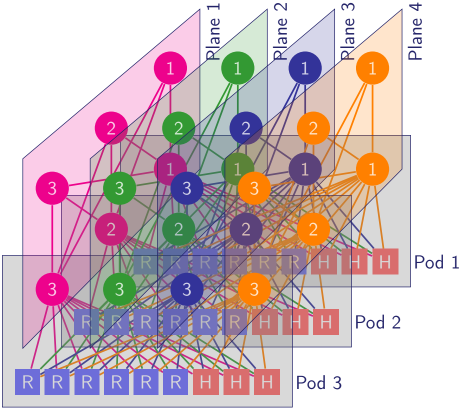
Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.



Parallel leaf-spine planes

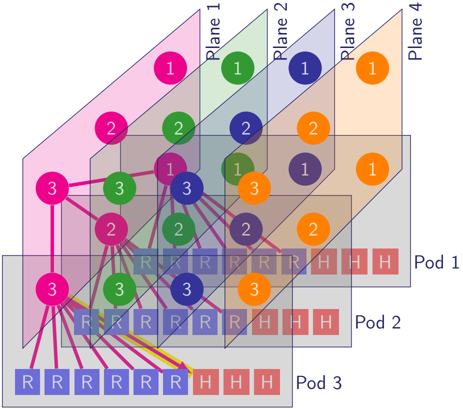
Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.

DCM flows distributed across available paths and daqrings (*waterfilling*).



Parallel leaf-spine planes

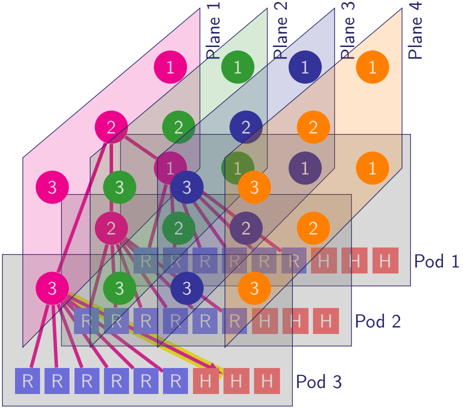
Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.

DCM flows distributed across available paths and daqrings (*waterfilling*).



Parallel leaf-spine planes

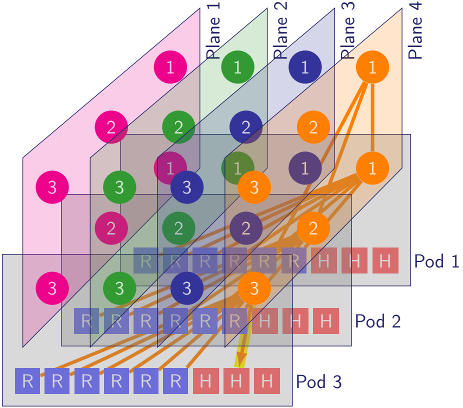
Topology based on Facebook's datacenter fabric.

Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.

DCM flows distributed across available paths and daqrings (*waterfilling*).



Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

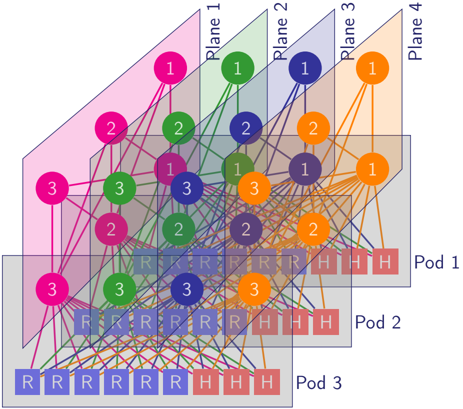
Applying in DAQ:

Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.

DCM flows distributed across available paths and daqrings (*waterfilling*).

OvS also on the end-nodes.



Parallel leaf-spine planes

Topology based on Facebook's datacenter fabric.

Applying in DAQ:

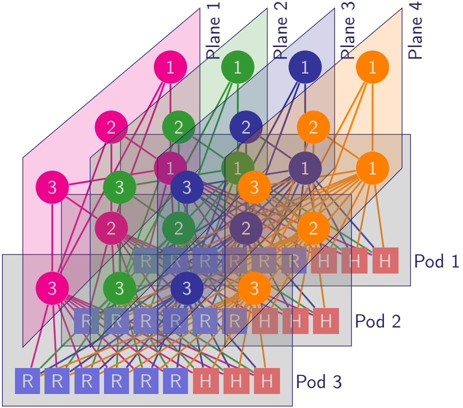
Data flow from ROS (R) to racks of DCMs (H).

OpenFlow L3-only network.

DCM flows distributed across available paths and daqrings (*waterfilling*).

OvS also on the end-nodes.

No need to use ECMP, LAG, or MLAG (no hashes!).

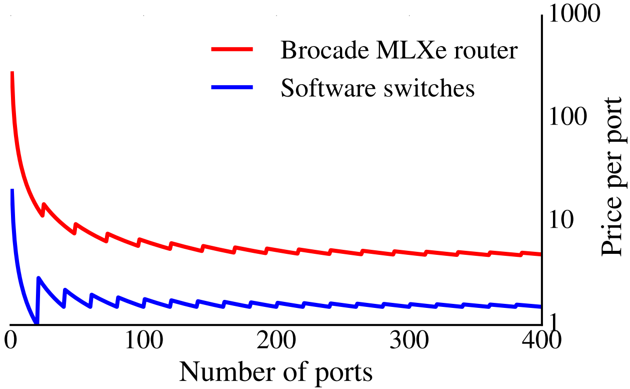


Rough cost estimates

Full non-blocking topology.

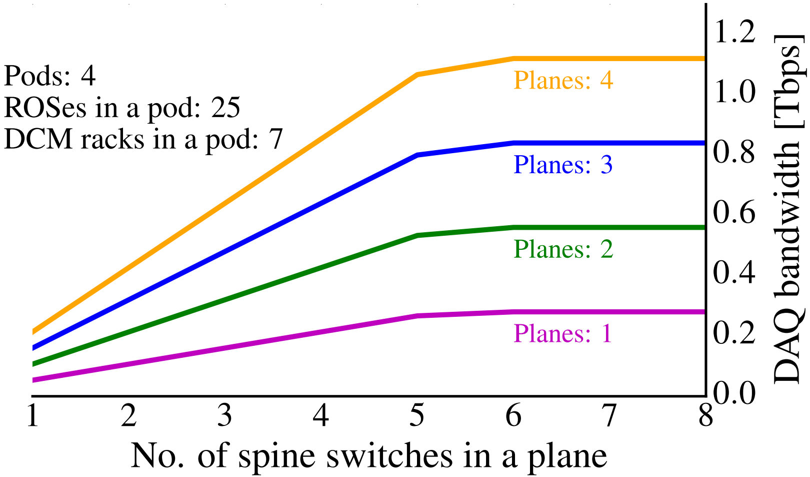
Traditional network without redundancy.

Further optimizations possible.

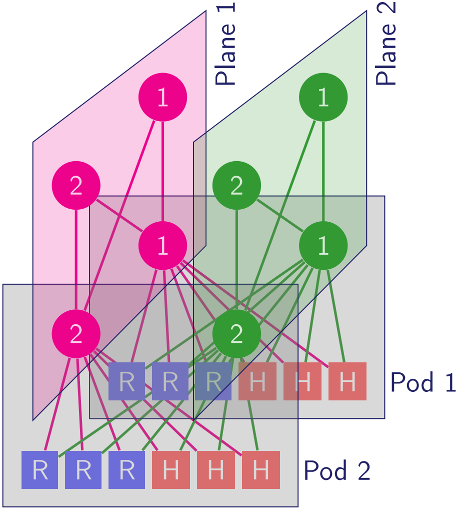


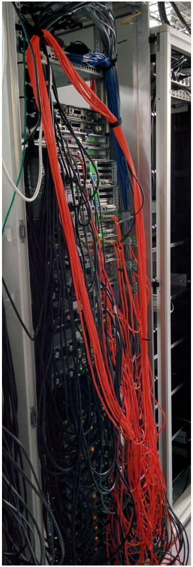
Note: Costs of cables and transceivers not included.

An example: offered DAQ bandwidth

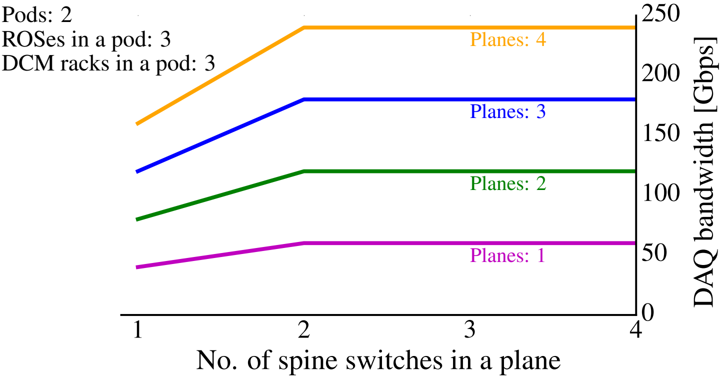


Prototype topology (8 switches)

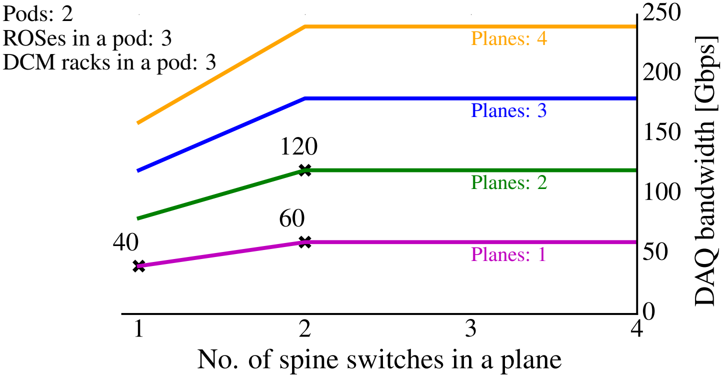




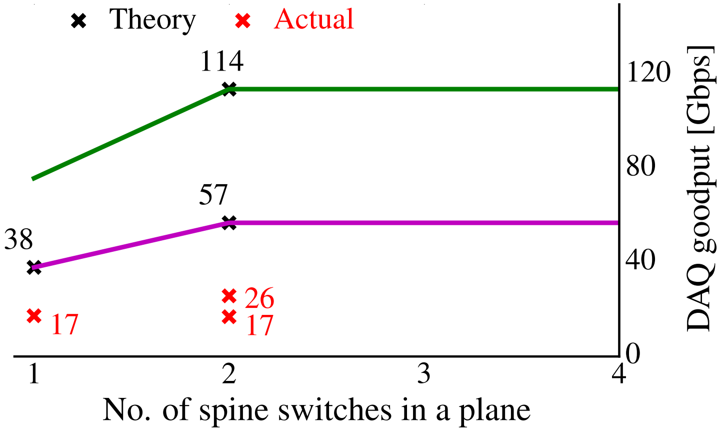
Offered DAQ bandwidth (theory)



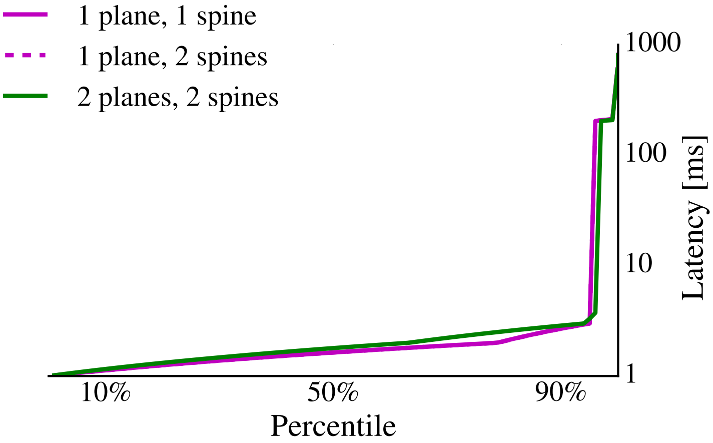
Offered DAQ bandwidth (theory)



Offered DAQ goodput (actual)

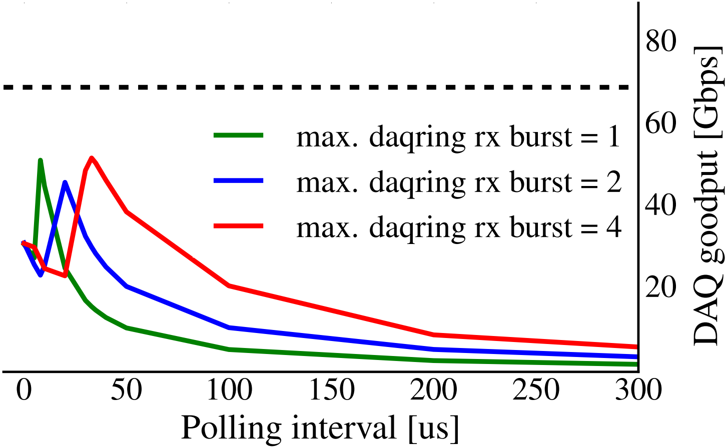


A problem: PCIe gen1 in the end nodes

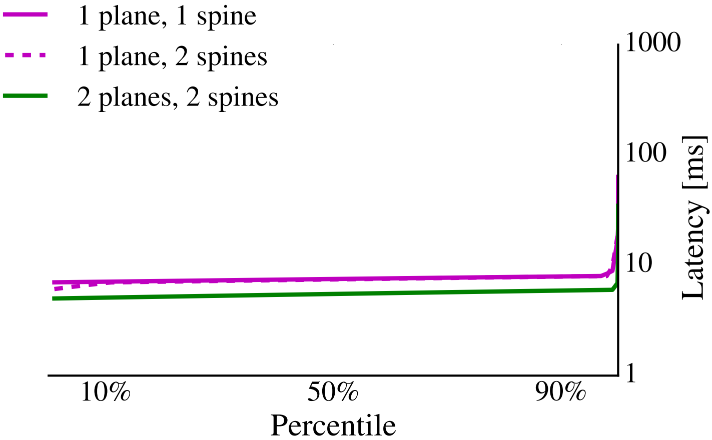


A problem: PCIe gen1 in the end nodes

Solution: Rate-limited daqrings

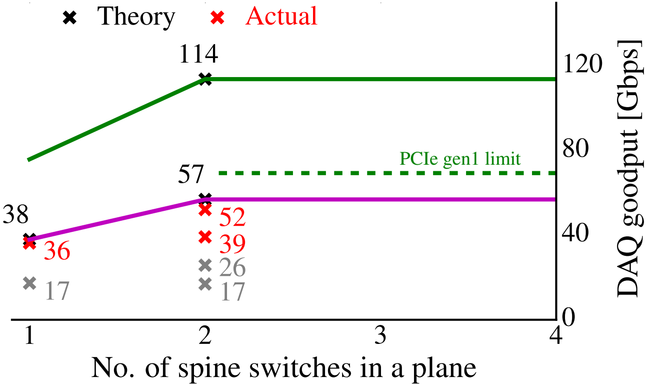


A problem: PCIe gen1 in the end nodes



Offered DAQ goodput (actual)

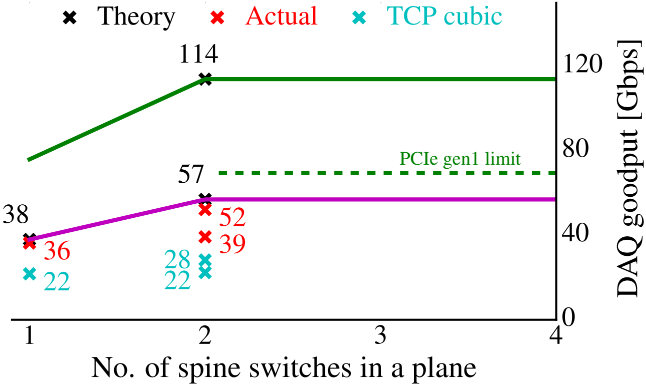
With rate-limited daqrings performance improved, but still limited (see 2 planes). Limit set by PCIe gen1.



Offered DAQ goodput (actual)

With rate-limited daqrings performance improved, but still limited (see 2 planes). Limit set by PCIe gen1.

Default TCP congestion control (TCP Cubic) performs poorly.



Conclusions and outlook

Trying to prevent incast congestion in DAQ

DRAM memory provides large enough
and cheap packet buffers.

Dedicated queueing to optimize the entire network.

First prototype offers **lossless operation** and **120Gbps bandwidth** for DAQ-specific network traffic with a single server.

Second prototype demonstrates the configuration and management of a **larger topology**.

Outlook

Generalized algorithm for load balancing.

Different service disciplines of DCM queues.

Fault tolerance.

Achievable port density.

The future

New family of Intel Ethernet products:

FM10000

Provides multiple Ethernet ports **AND** host PCIe interfaces.

Example - FM10840:

36 x 10GbE ports,

4 x 8-lane PCIe gen3 interfaces,

Approx. 200 Gbps over PCIe,

Ethernet Multi-host Controllers

Perfect match for building larger topologies with packet buffers in host memory?

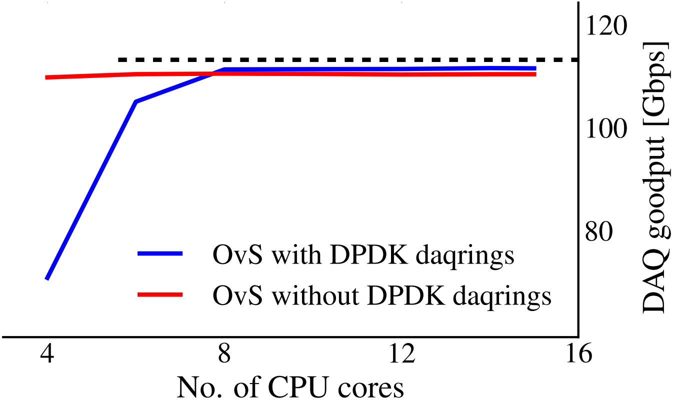
Questions?

Backup

Performance penalty with *daqrings*?

Better fairness among all data collectors.

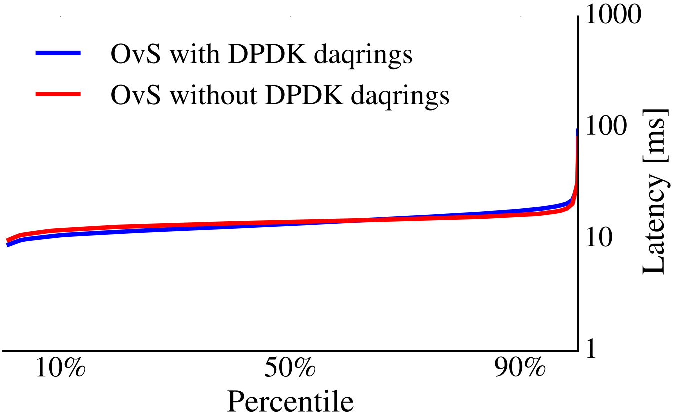
More CPU cycles required due to additional port send/recv and OpenFlow lookups



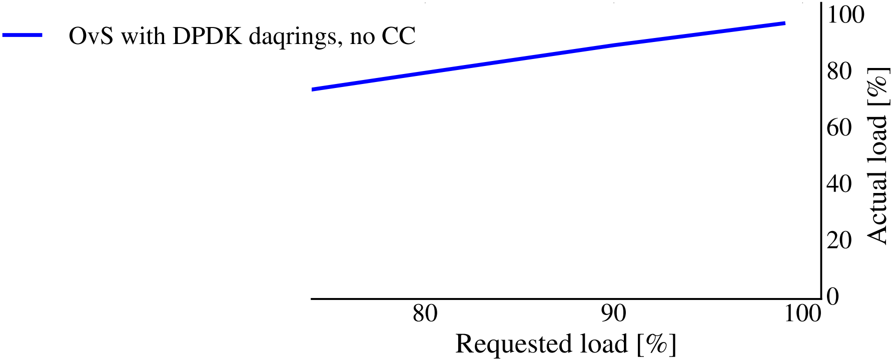
Performance penalty with *daqrings*?

Better fairness among all data collectors.

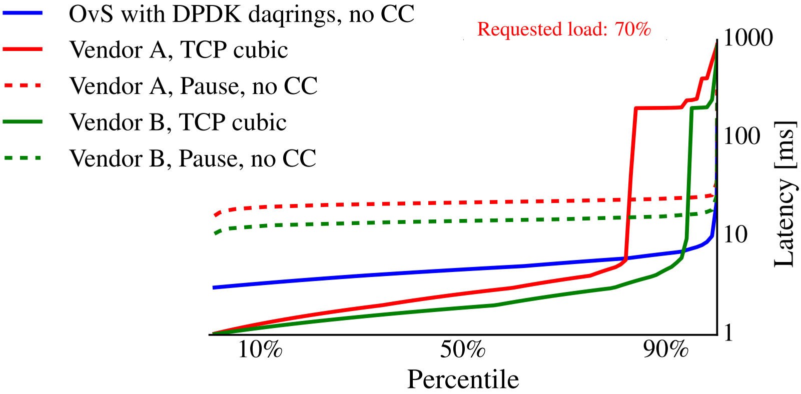
More CPU cycles required due to additional
port send/recv and OpenFlow lookups



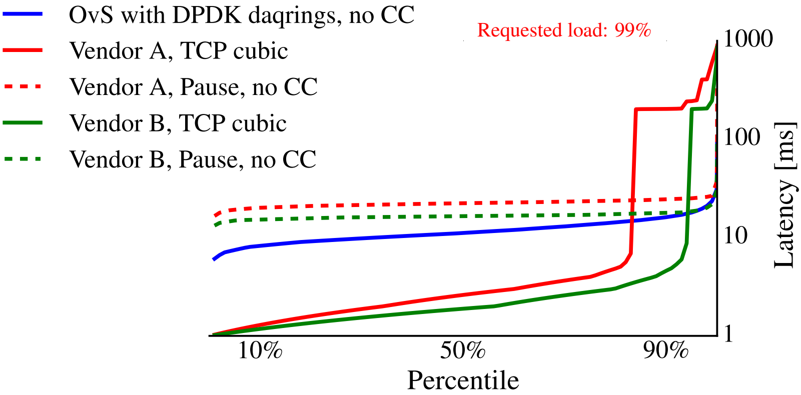
The lossless software switch outperforms regular switches with hardware flow control



The lossless software switch outperforms regular switches with hardware flow control



The lossless software switch outperforms regular switches with hardware flow control


OvS with DPDK daqrings, no CC
Vendor A, TCP cubic
Vendor A, Pause, no CC
Vendor B, TCP cubic
Vendor B, Pause, no CC
Requested load: 99%
Latency [ms]
Percentile