Online Software

Experience with IB and Ethernet from Mellanox

15th January 2013
Technical workshop on InfiniBand for Trigger/DAQ

Authors: Luciano Orsini, Andrea Petrucci
CERN (PH/CMD)
Contributors: Andre Georg Holzner (UCSD),
Petr Zejdl CERN (PH/CMD),
Christopher Wakefield CERN (PH/CMD)

Outline

- Introduction to CMS upgrade
- Network technologies
- Test setups
- Testware
- Boosting performance on NUMA architecture
- Preliminary measurements
- Mellanox experience
- Summary

Motivations for upgrade of CMS DAQ

- Aging of existing hardware (PCs and Networks at least 5 years old)
- Accommodate sub-detectors with upgraded off-detector electronics

Upgrade of DAQ system (I)

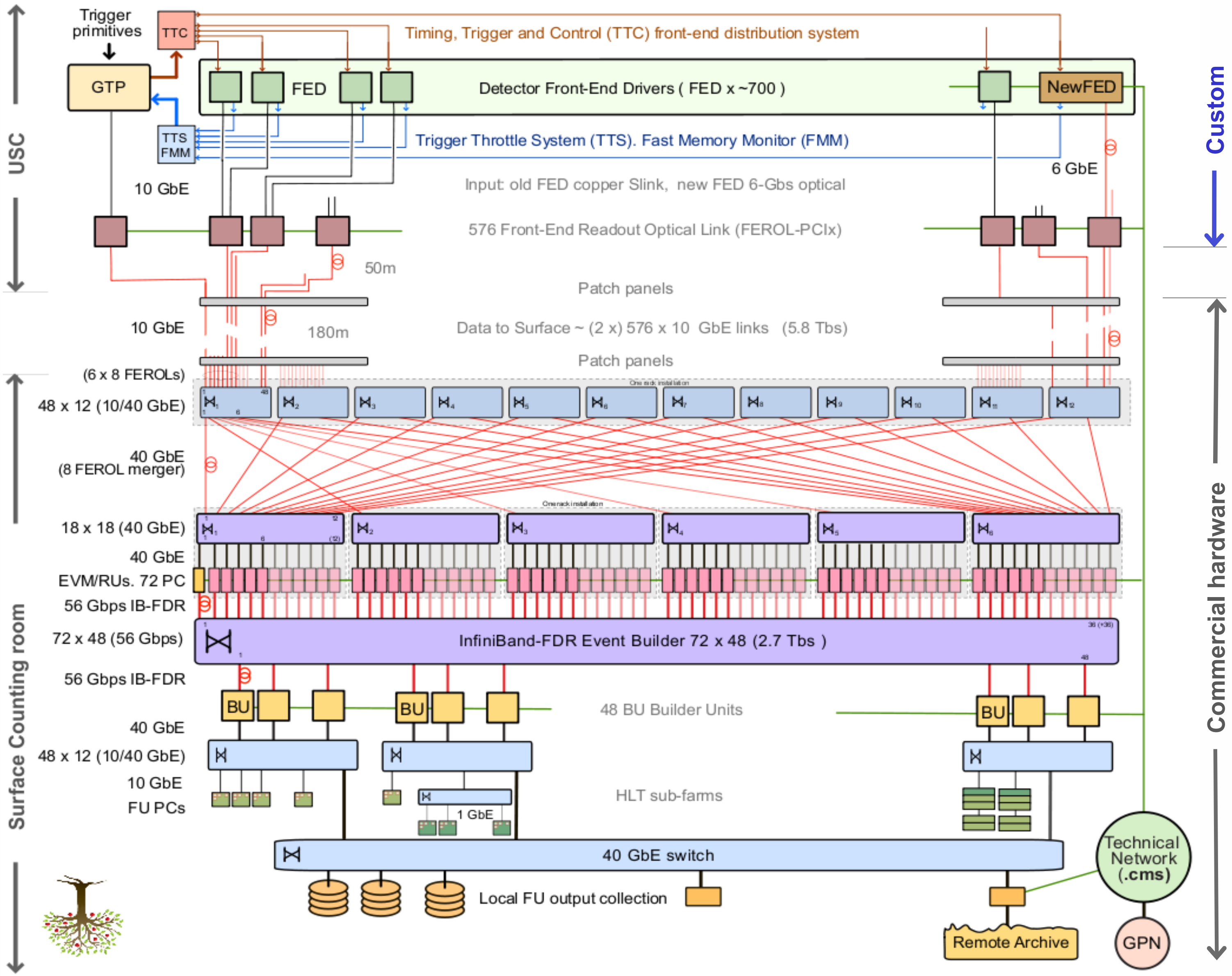
Requirements

- L1 rate of 100 KHz
- Detector Front End Drivers (FED)
  - ~552 Legacy FEDs (fragment size increases from 2 kB to 4 kB due to pile-up)
  - 37 (TRG, HCAL, HF) + 40 (Pixel - 2 x 10 GbE links) new readout links from new FEDs (expected maximal fragment size 8kB)
- Frontend Readout Links (FRLs)
  - ~360 FRLs (Legacy FEDs, ~400 MB/s)
  - ~120 FRLs (new FEDs, ~640 MB/s)
- High availability (redundancy, load balancing, failover, etc.)

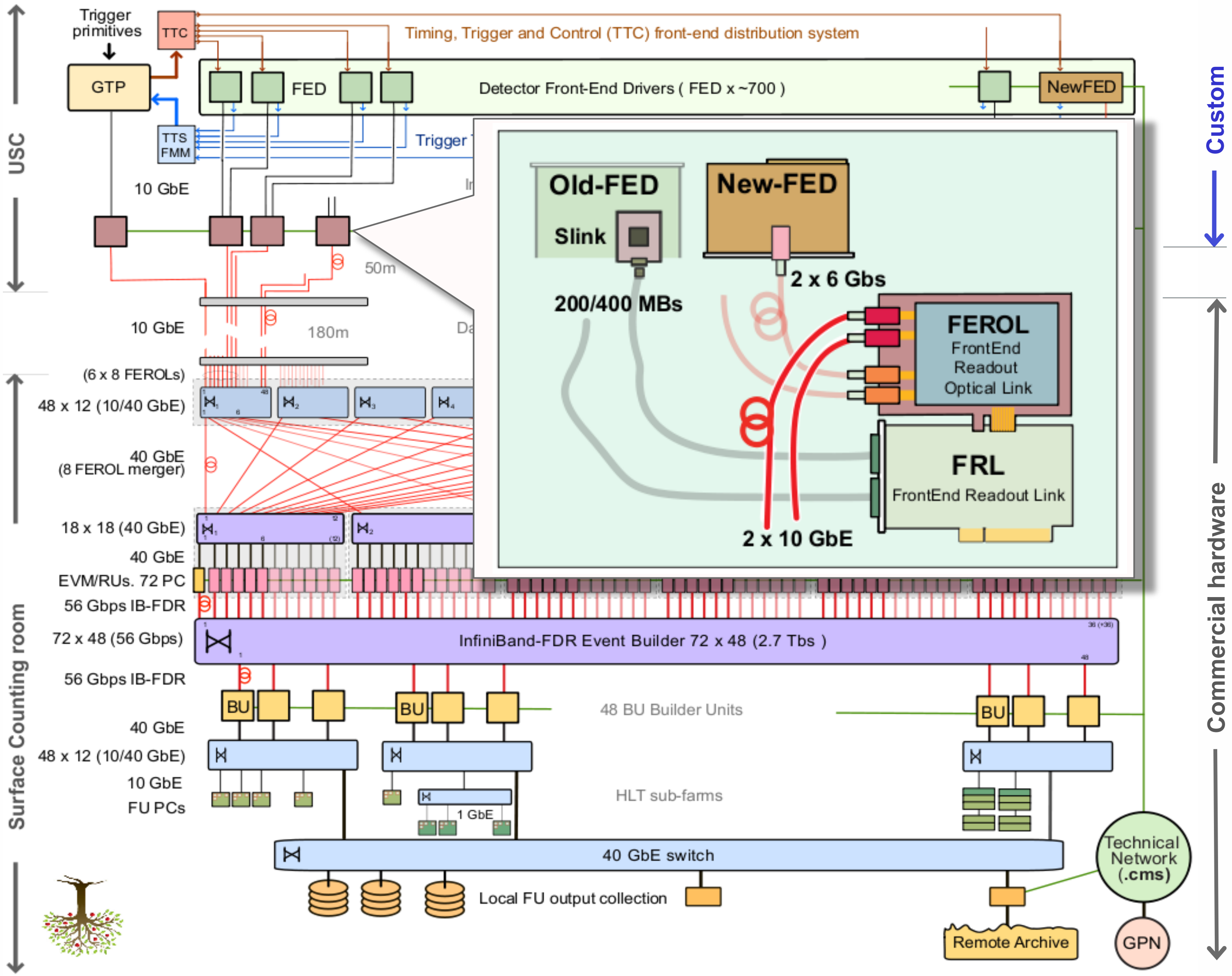
DAQ plans for upgrade

- Replace myrinet-based fedbuilder with 10/40 GE
- Replace event builder network with 40 GE or Infiniband
- New architecture between Event Builder and Filter Farm
- Replace of the Storage manager hardware

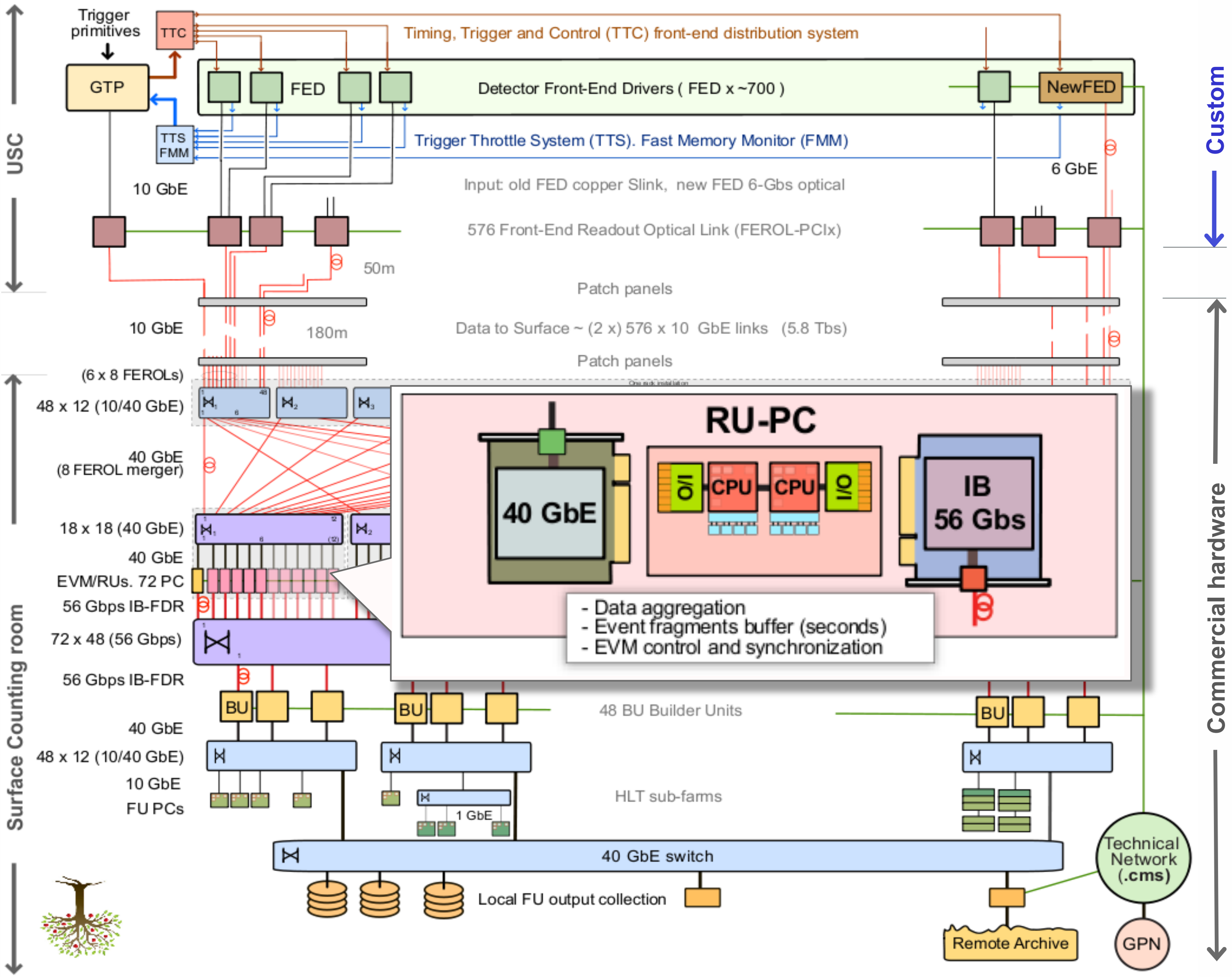
Upgrade of DAQ system (II)



Upgrade of DAQ system (II)

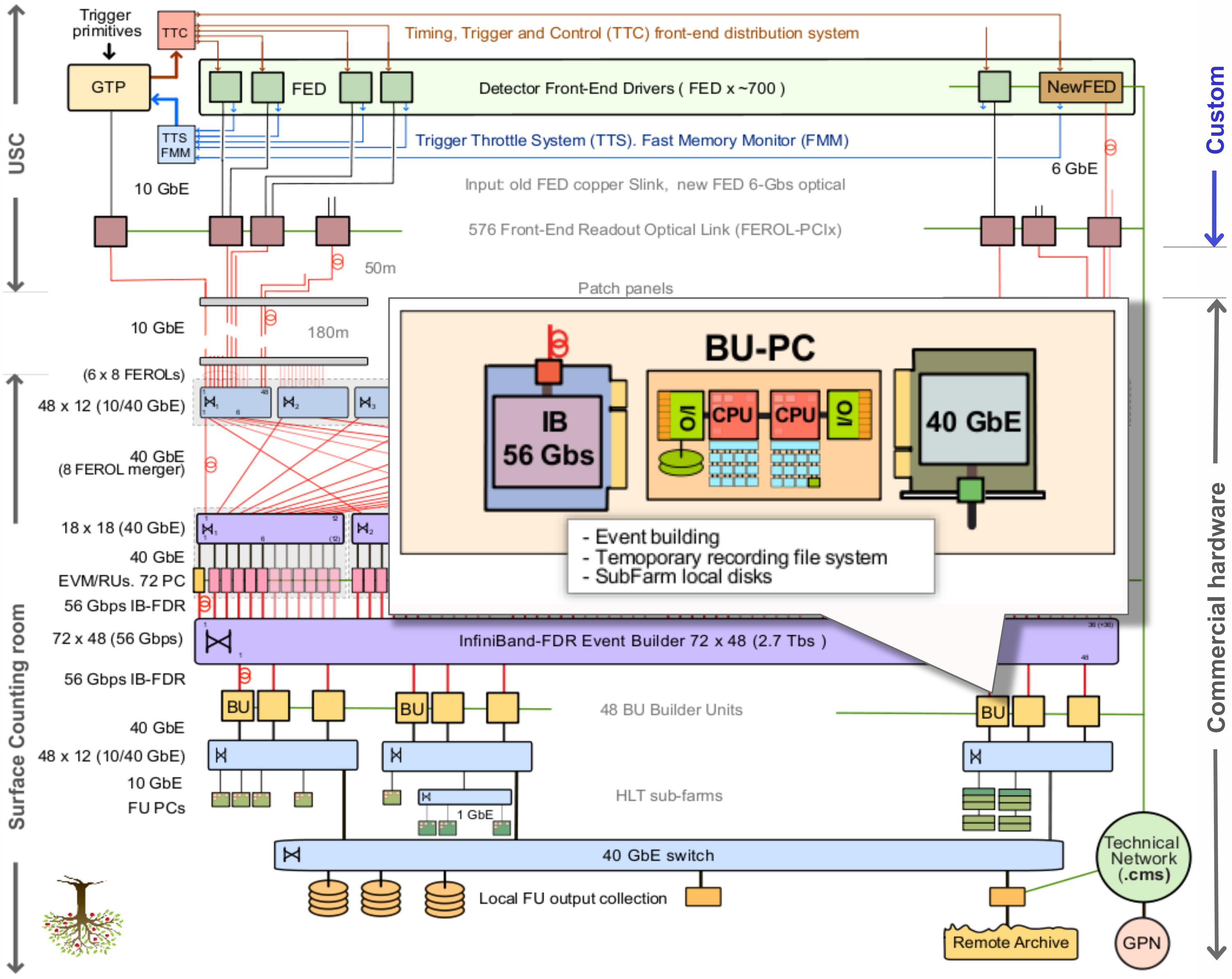


Upgrade of DAQ system (III)



- Data aggregation
- Event fragments buffer (seconds)
- EVM control and synchronization

Upgrade of DAQ system (IV)



Networking technologies

Networking technologies

Our feasibility studies are focused in two network technologies:

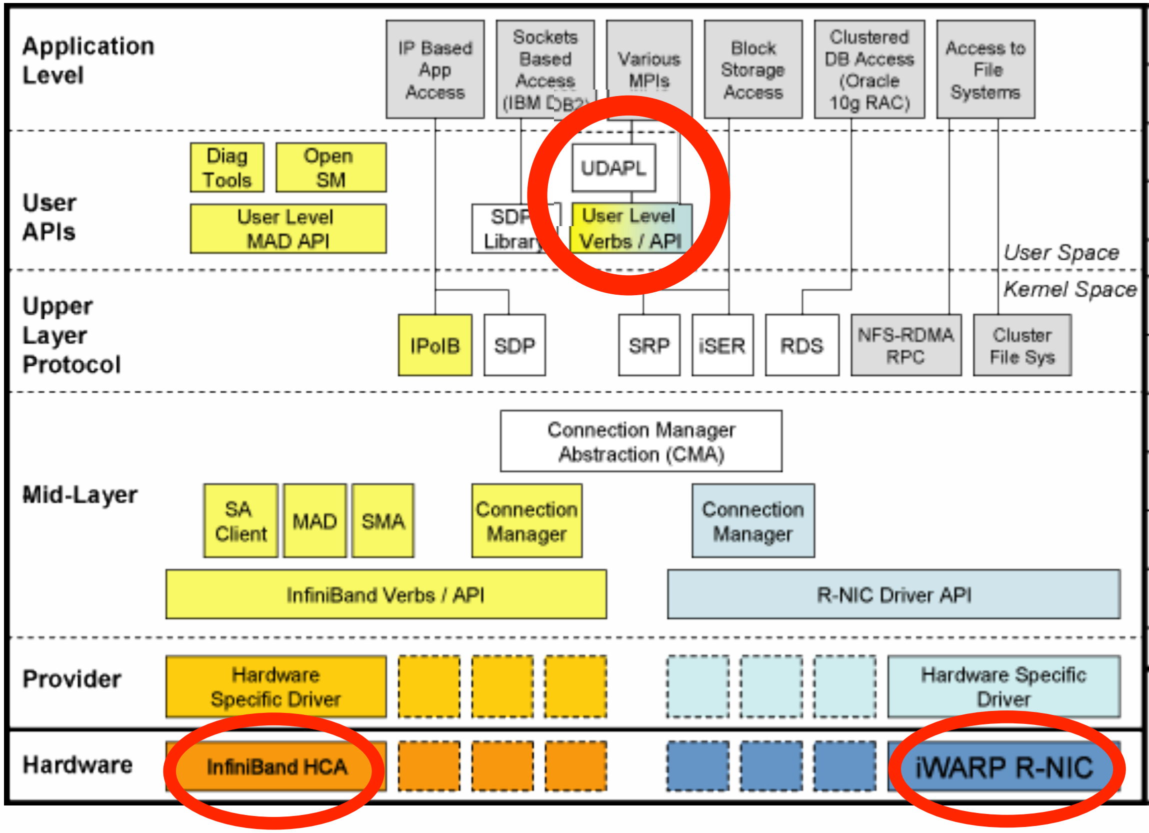
- **Ethernet**
  - 10/40 Gigabit Ethernet (different vendors)
  - iWARP (RDMA) – TCP/IP full offload (Chelsio T4 Unified Wire Adapters)
  - performance measurements using **TCP/IP** and **DAPL** (Direct Access Programming Library- OpenFabrics)
- **Infiniband**
  - 4x quad data rate (QDR)
    - 40 Gb/s - 8B/10B encoding -32 Gb/s data rate
  - 4x fourteen data rate (FDR)
    - 56 Gb/s – 64B/66B encoding – 54.54 Gb/s data rate
  - performance measurements using **DAPL** (Direct Access Programming Library- OpenFabrics) and IPoIB (IP over InfiniBand)

The OFED Stack (source: OpenFabrics Alliance)

A unified, cross-platform, transport-independent software stack for RDMA and kernel bypass

- http://www.openfabrics.org/

OPENFABRICS ALLIANCE

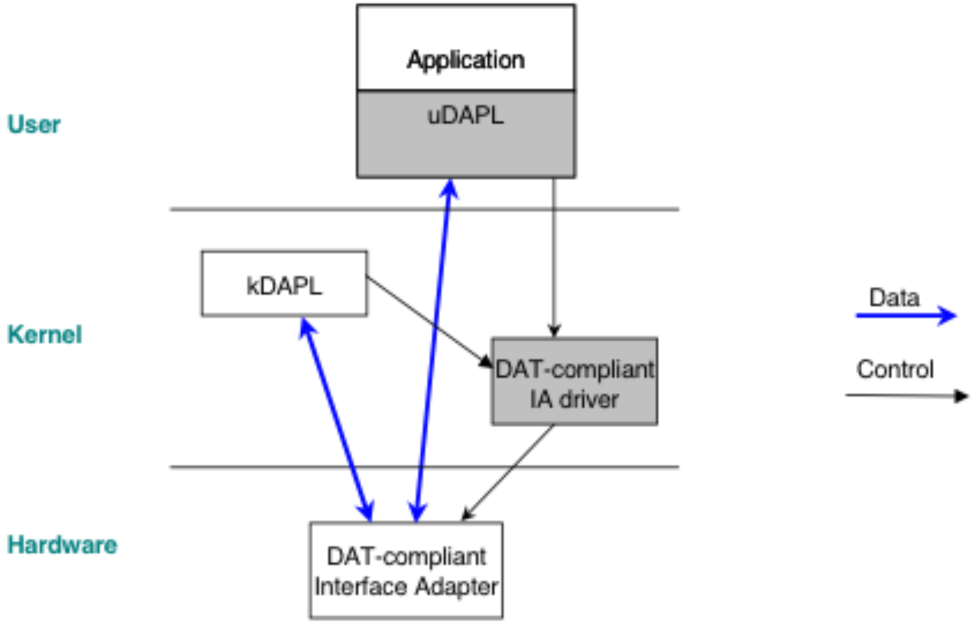


SA	Subnet Administrator
MAD	Management Datagram
SMA	Subnet Manager Agent
PMA	Performance Manager Agent
IPoIB	IP over InfiniBand
SDP	Sockets Direct Protocol
SRP	SCSI RDMA Protocol (Initiator)
iSER	iSCSI RDMA Protocol (Initiator)
RDS	Reliable Datagram Service
UDAPL	User Direct Access Programming Lib
HCA	Host Channel Adapter
R-NIC	RDMA NIC

Key
- Common
- InfiniBand
- iWARP
- Apps & Access Methods for using OF Stack

DAT Model (source DAT Collaborative)

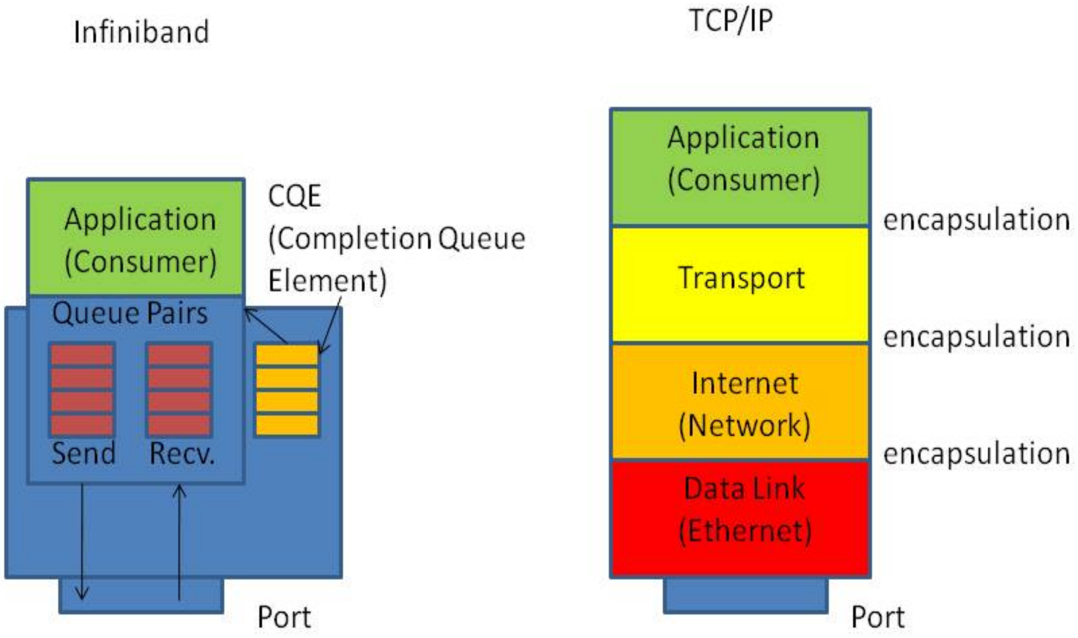
- Developed by DAT collaborative
  - http://www.datcollaborative.org/
- Transport and platform (OS) independent
- Define user (uDAPL) and kernel (kDAPL) APIs
- DAT supports reliable connection
- Data Transfer Operations send, receive, rdma_read, rdma_write
- uDAPL Version 2.x, January, 2007



DAT collaborative

Comparison of the Stacks (Infiniband vs TCP/IP)

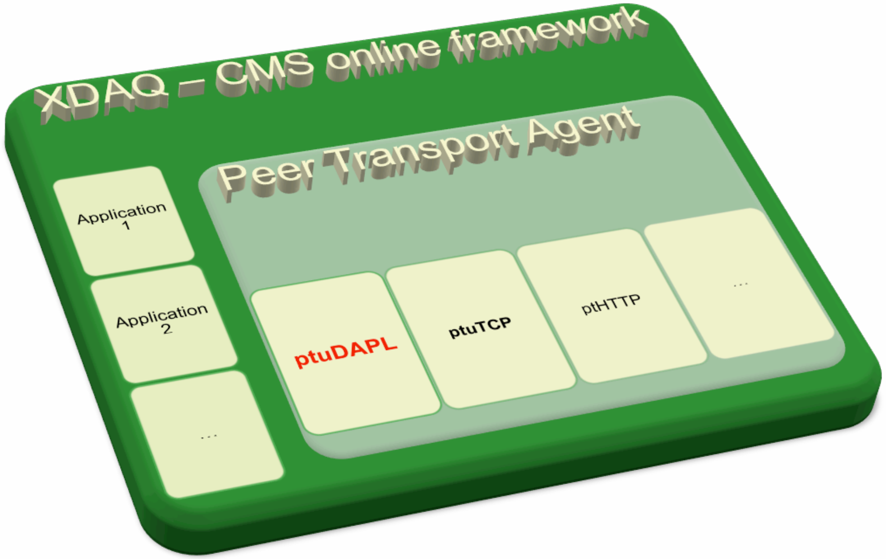
- The protocol is defined as a very thin set of zero copy functions when compared to thicker protocol implementations such as TCP/IP



Pluggable Peer-Transport for DAT library

The ptuDAPL is a pluggable component to transparently access the DAT library in XDAQ – CMS online framework

- All I/O operations centered on dedicated memory pool based on uDAPL memory region allocator
- Profiting for inherent non blocking and queuing of uDAPL API for minimizing latency
- Full zero-copy between CMS online applications and DAPL driver
- Based on DAT Spec 2.x



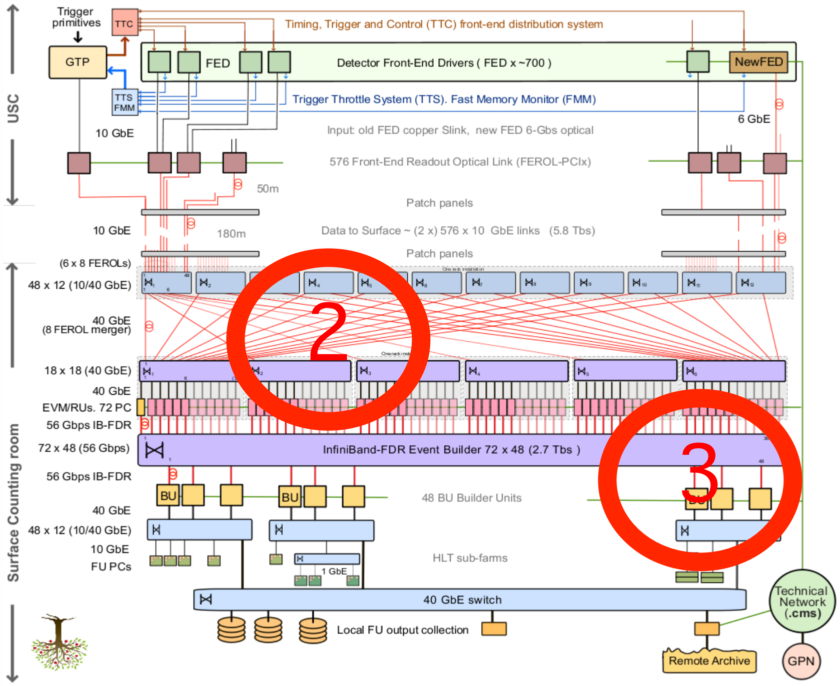
Test setups

Test setups

- Setup 1 (LHCb)
  - Initial software development environment (ptuDAPL)
  - first tests with Infiniband (QDR) and 10 GE (TCP, iWARP)

- Setup 2 (FEROL test)
  - Front-End Readout Optical link merger
  - 16 x 10 GE inputs to 1 x 40 GE

- Setup 3 (Event builder)
  - DAQ Event building
  - Scalability
  - N inputs to M outputs (IB or 40 GE)



USC
Trigger primitives
TTC
GTP
Timing, Trigger and Control (TTC) front-end distribution system
FED
Detector Front-End Drivers (FED x ~700)
NewFED
TTS FMM
Trigger Throttle System (TTS). Fast Memory Monitor (FMM)
6 GbE
10 GbE
Input: old FED copper Slink, new FED 6-Gbs optical
576 Front-End Readout Optical Link (FEROL-PCIx)
50m
Patch panels
10 GbE
180m
Data to Surface ~ (2 x) 576 x 10 GbE links (5.8 Tbs)
Patch panels
(6 x 8 FEROLs)
48 x 12 (10/40 GbE)
40 GbE
(8 FEROL merger)
2
18 x 18 (40 GbE)
40 GbE
EVM/RUs. 72 PC
56 Gbps IB-FDR
72 x 48 (56 Gbps)
InfiniBand-FDR Event Builder 72 x 48 (2.7 Tbs)
3
56 Gbps IB-FDR
BU
BU
48 BU Builder Units
40 GbE
48 x 12 (10/40 GbE)
10 GbE
FU PCs
1 GbE
HLT sub-farms
Surface Counting room
40 GbE switch
Local FU output collection
Technical Network (.cms)
Remote Archive
GPN

Setup 1 (LHCb)

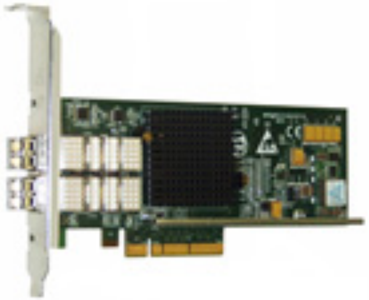
	Setup 1	
Nodes	8	
Type	DELL R710	
CPU	Xeon E5530 2x 4-core at 2.27 GHz	
Memory	3 GB	
Network	Ethernet	Infiniband
Adapter	Chelsio T420-CR 10GBASE-SFP RNIC (iWarp)	Qlogic HCA, qle7340 4x QDR PCIe
Switch	Voltaire Vantage 6048, 48 ports, 10 GbE	Qlogic 12300-BS01, 36 ports, 4x QDR

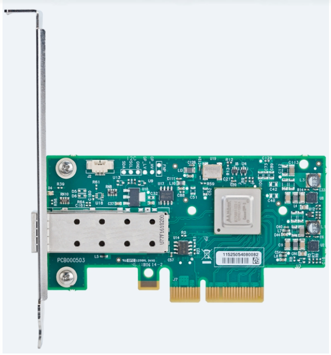
DELL R310/R620

- **Operating System:** Scientific Linux CERN SLC release 5.3 (Boron)
- **Linux version:** 2.6.18-164.6.1.el5
- **OFED version**: OFED.1.5.2.x.x
- **XDAQ version:** release 11

Setup 2 (Hardware)

	Setup 2	
Nodes	16	1
Type	DELL R310	DELL R620
CPU	Xeon X3450 1x 6-core at 2.67 GHz	Xeon E5-2670 2x 8-core at 2.6 GHz
Memory	4 GB	32 GB
Network	10 GE	40 GE
Adapters	Silicom PE210G2SPi9 Intel Corporation 82599EB 10-Gigabit SFI/SFP+	Mellanox - ConnectX-3 VPI MCX353A-FCBT
Switches	Mellanox 36 - QSFP40 GbE - MSX1036B-1SFR	





Setup 2 (Firmware/Software)

DELL R310/R620

- **Operating System:** Scientific Linux CERN SLC release 6.2 beta (Carbon)
- **Linux version:** 2.6.32-220.2.1.el6.x86_64
- **OFED version**: OFED.1.5.3.3.1.0
- **Ethernet driver:** mlx4_en version 1.5.8.3
- **XDAQ version:** release 11
- **TCP test**: sock application http://www.icir.org/christian/sock.html

Silicom PE210G2SPi9-SR

- Firmware version: 1.8-0

Mellanox - ConnectX-3 VPI

- Firmware version: 2.11.500

Mellanox 36 – MSX1036B-1SFR

- Firmware version: 9.1.6294
- Mellanox MLNX-OS™ version: 3.2.0506

Setup 3 (Hardware)

	Setup 3	
Nodes	32	
Type	DELL C6220	
CPU	Xeon E5-2670 2x 8-core at 2.6 GHz	
Memory	32 GB	
Network	IB FDR 4x/40 GE	
Adapters	Mellanox - ConnectX-3 VPI MCX353A-FCBT (# 4)	DELL mezzanine Mellanox FDR CX3 (# 24)
Switches	Mellanox 36 - QSFP FDR based Infiniband - MSX1036F-1SFR Mellanox 36 - QSFP40 GbE - MSX1036B-1SFR	

Setup 3 (Firmware/Software)

DELL C6220

- **Operating System:** Scientific Linux CERN SLC release 6.2 beta (Carbon)
- **Linux version:** 2.6.32-220.2.1.el6.x86_64
- **OFED version**: OFED.1.5.3.3.1.0
- **Ethernet driver:** mlx4_en version 1.5.8.3
- **XDAQ version:** release 11
- **TCP test**: sock application http://www.icir.org/christian/sock.html

Mellanox - ConnectX-3 VPI

- Firmware version: 2.11.500

DELL mezzanine Mellanox FDR CX3

- Firmware version: 2.10.4492

Mellanox 36 – MSX1036F-1SFR

- Firmware version: 9.1.3190
- Mellanox MLNX-OS™ version: 3.2.0300

Mellanox 36 – MSX1036B-1SFR

- Firmware version: 9.1.6294
- Mellanox MLNX-OS™ version: 3.2.0506

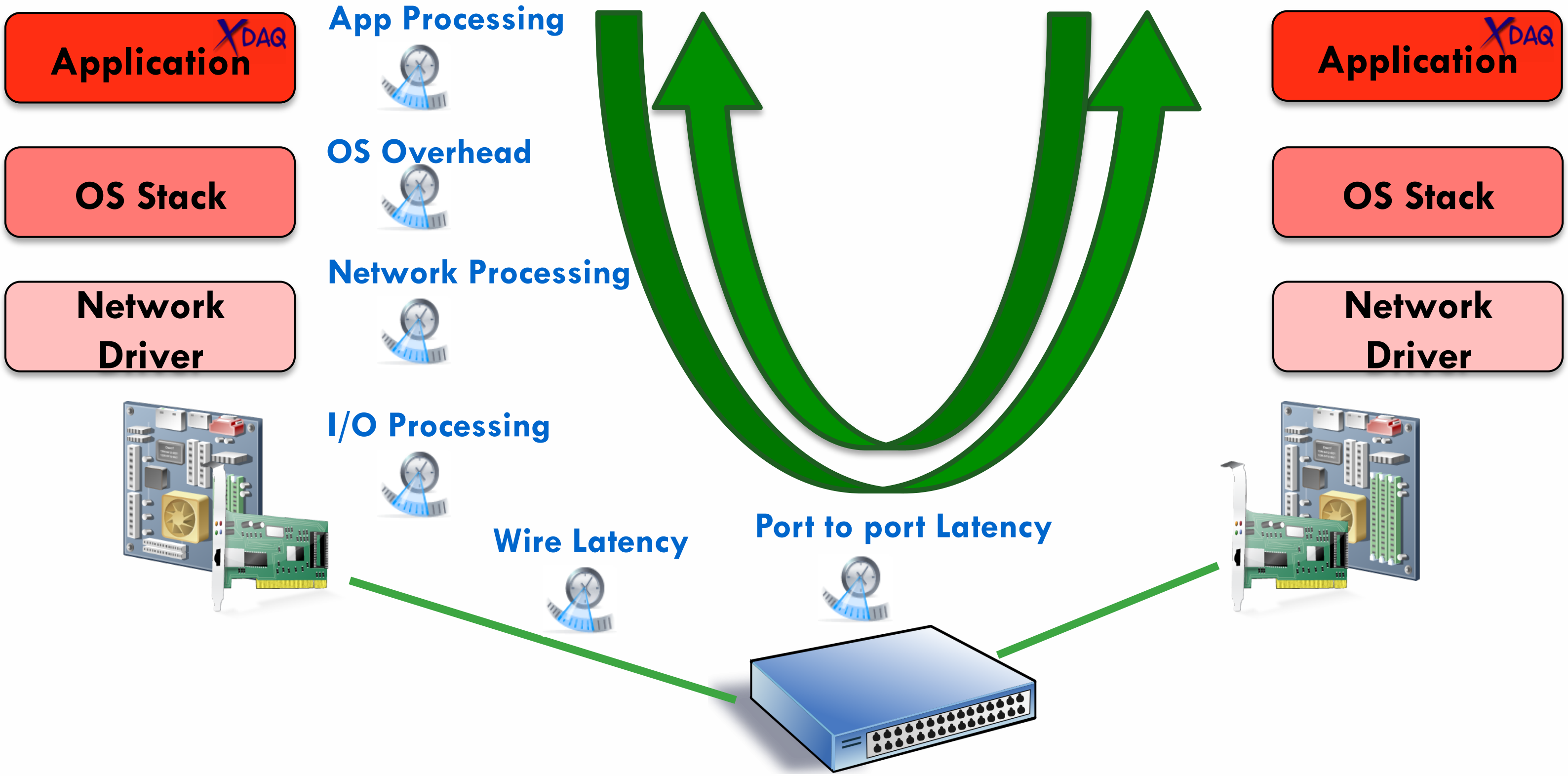
Testware

Testware

- Roundtrip
  - Used to measure latency
- MStreamIO
  - Used to measure throughput (N to M combinations)
  - FEROL merger (N to 1)
- Event Builders
  - Used to measure event building throughput
  - GEVB Generic Event Builder (N x M)

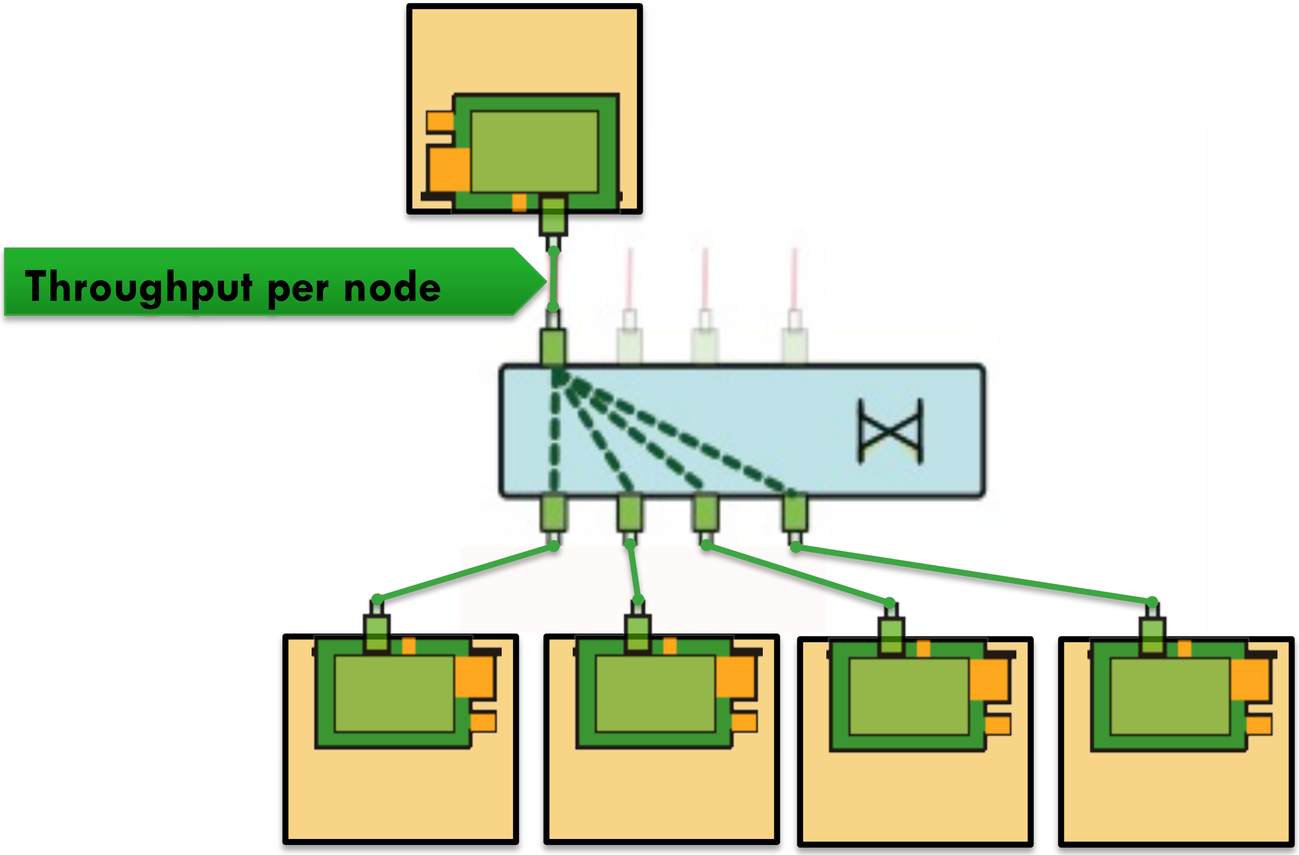
Roundtrip

- Simple XDAQ application to compute the One-way delay
- Time packet to travel from a specific source to a specific destination and back again
- One-way latency is measured by timing a round-trip message and dividing the obtained result by two



Multi-Stream I/O

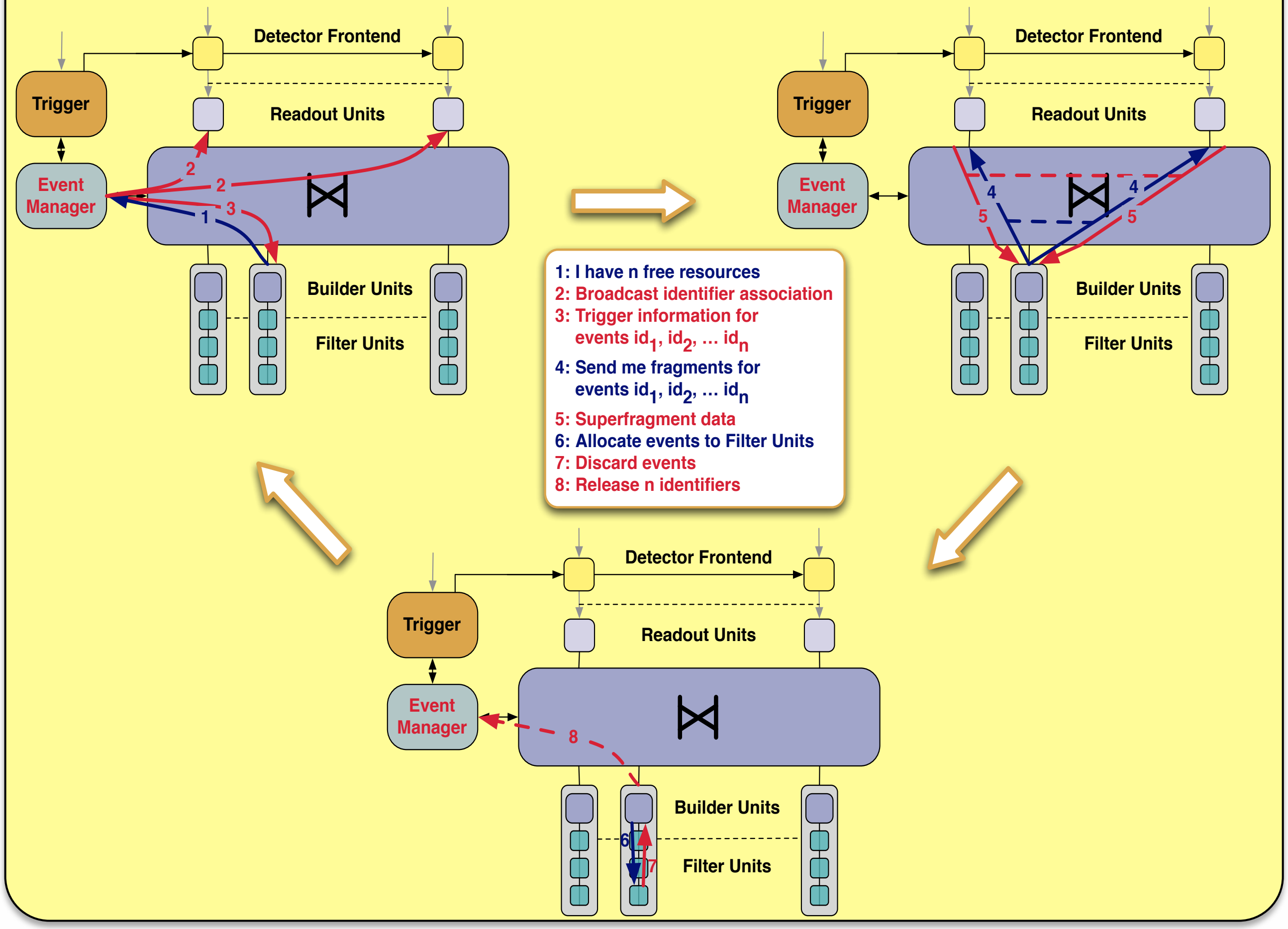
- Unidirectional throughput (bandwidth) is measured using a unidirectional send of N messages to M receivers.
- Time sampling is done at senders and receivers sides.



Event Builder

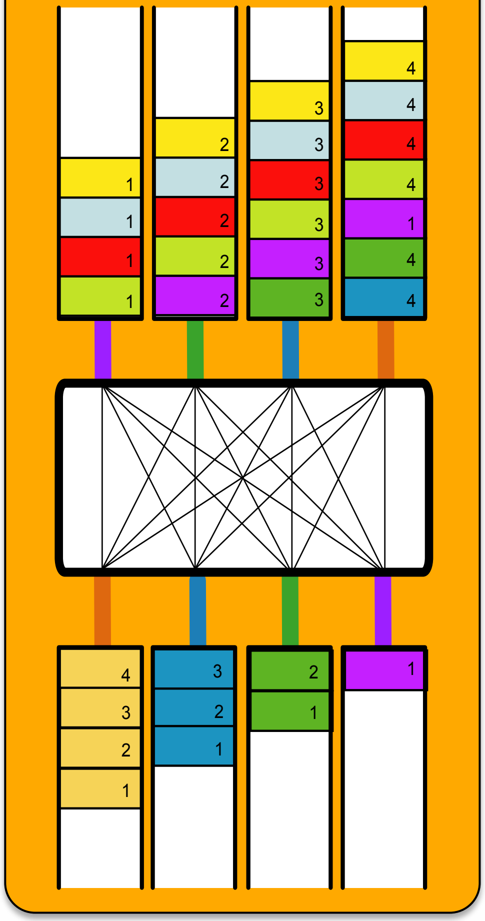
- CMS RU-builder
  - Currently used in CMS DAQ for data taking

RU-builder protocol – Token passing



1: I have n free resources
2: Broadcast identifier association
3: Trigger information for events id_1, id_2, ... id_n
4: Send me fragments for events id_1, id_2, ... id_n
5: Superfragment data
6: Allocate events to Filter Units
7: Discard events
8: Release n identifiers

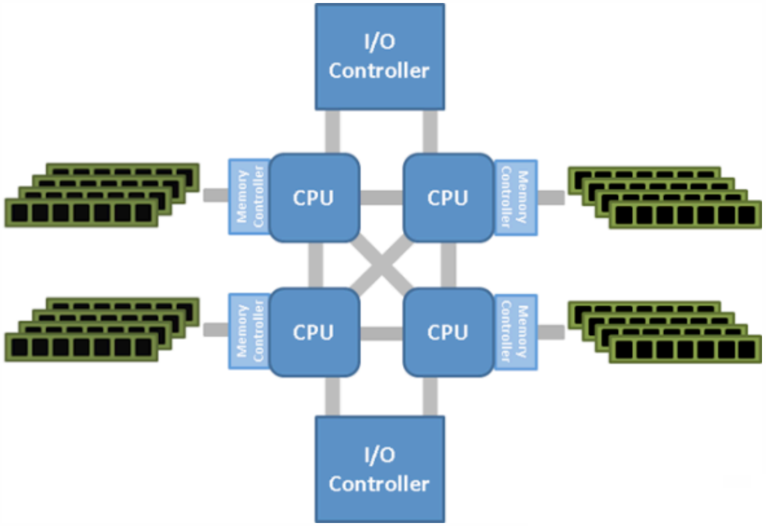
Event building traffic



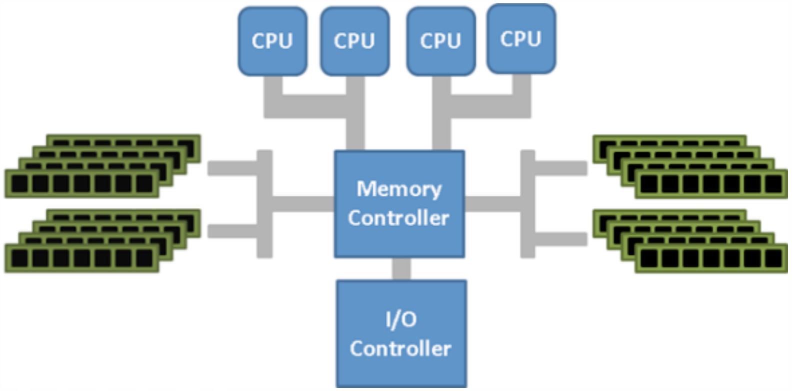
Boosting performance on NUMA architecture

NUMA vs UMA

Non-Uniform Memory Access (NUMA)

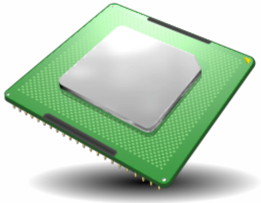


Uniform Memory Access (UMA)



Performance factors

Interrupt affinity



CPU affinity

Memory affinity

Affinity example

IRQ Affinity

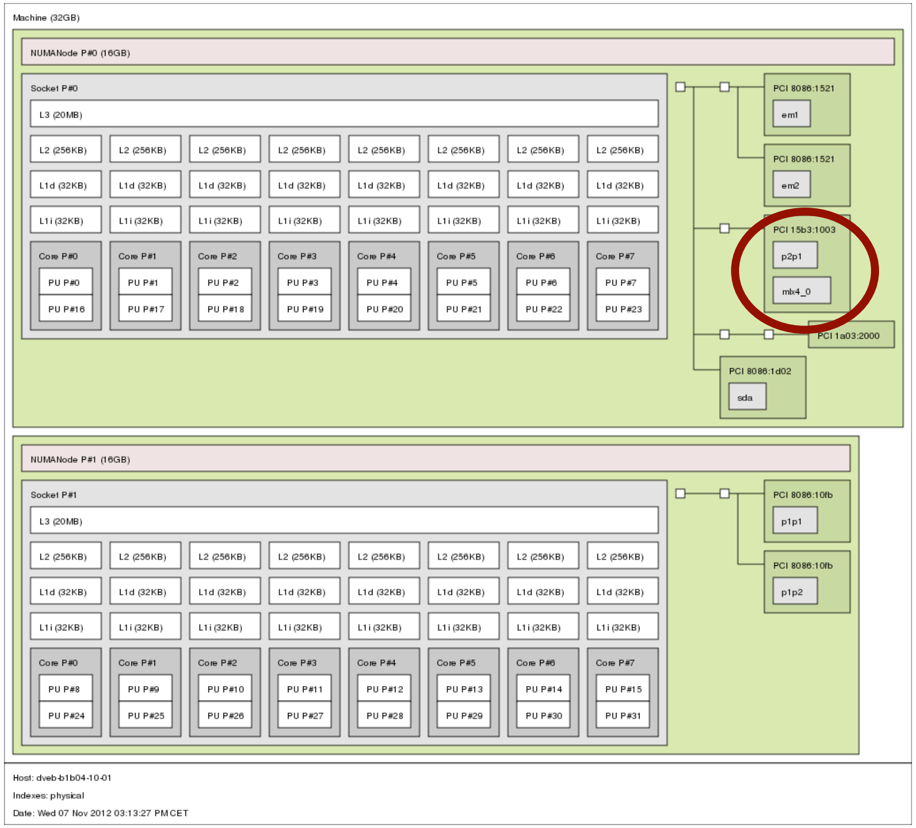
- Set one core for all IRQ queues

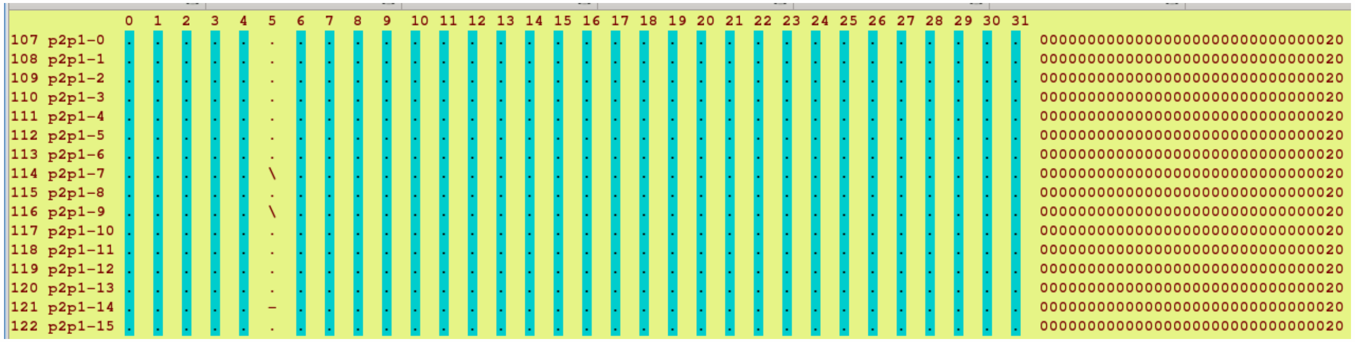
Processor Affinity

- Bind application threads on cores in the same socket where “IRQ core”

Memory Affinity

- Bind application memory on the same NUMA node where “IRQ core”





```
              0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
107 p2p1-0    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
108 p2p1-1    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
109 p2p1-2    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
110 p2p1-3    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
111 p2p1-4    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
112 p2p1-5    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
113 p2p1-6    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
114 p2p1-7    .  .  .  .  .  \  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
115 p2p1-8    .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
116 p2p1-9    .  .  .  .  .  \  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
117 p2p1-10   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
118 p2p1-11   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
119 p2p1-12   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
120 p2p1-13   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
121 p2p1-14   .  .  .  .  .  -  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
122 p2p1-15   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  00000000000000000000000000000020
```

40 GE Tuning: IPv4 Traffic Performance

# Disable TCP timestamp support to reduce CPU usage:

```
sudo /sbin/sysctl -w net.ipv4.tcp_timestamps=1
sudo /sbin/sysctl -w net.ipv4.tcp_sack=1
```

# Increase the TCP maximum and default buffer sizes using setsockopt():

```
sudo /sbin/sysctl -w net.core.rmem_max=104857600
sudo /sbin/sysctl -w net.core.wmem_max=104857600
sudo /sbin/sysctl -w net.core.rmem_default=16777216
sudo /sbin/sysctl -w net.core.wmem_default=16777216
```

# Increase memory thresholds to prevent packet dropping:

```
sudo /sbin/sysctl -w net.ipv4.tcp_mem="16777216 16777216 16777216"
```

# Set minimum, default, and maximum TCP buffer limits:

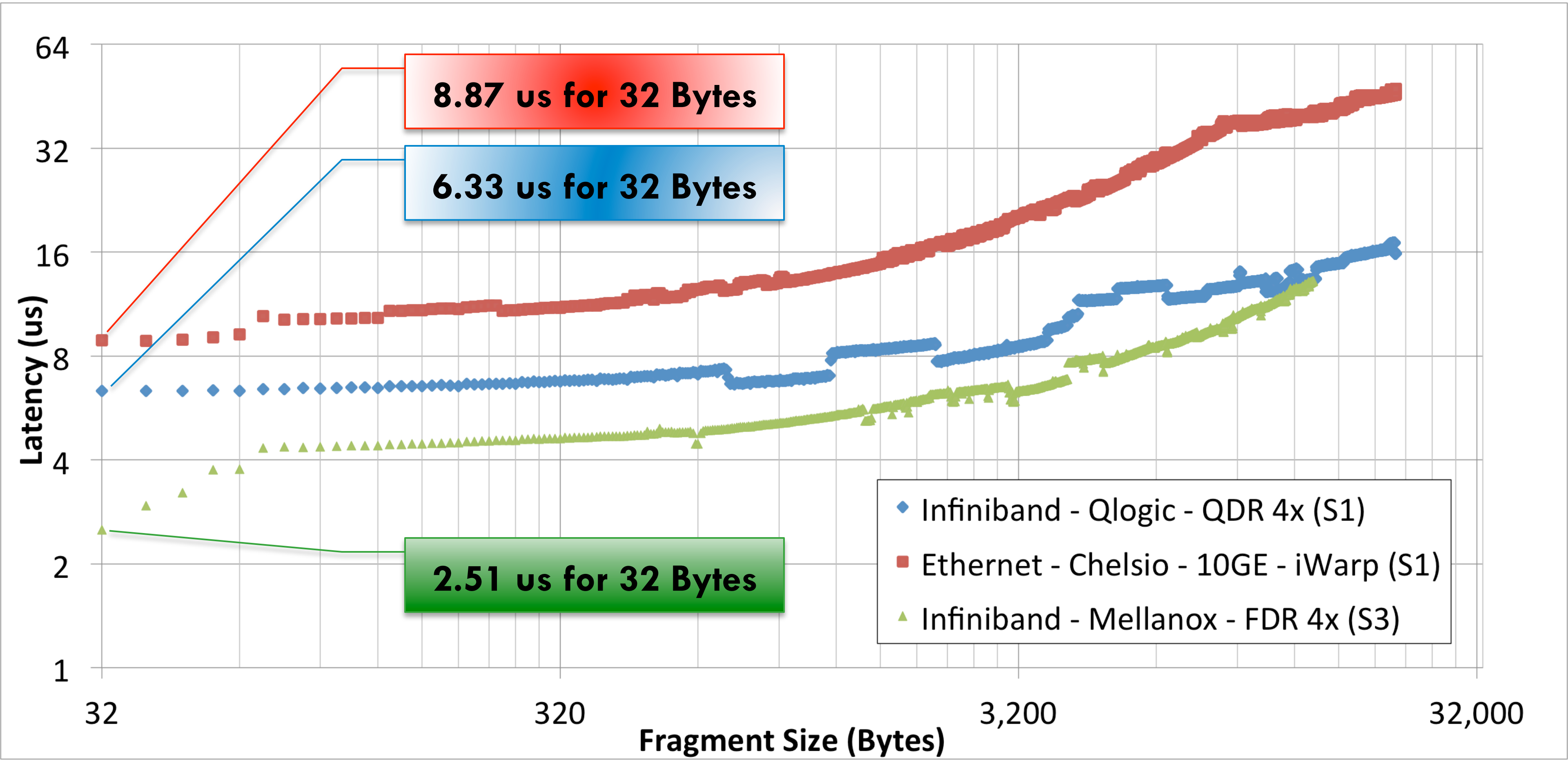
```
sudo /sbin/sysctl -w net.ipv4.tcp_rmem="4096 524288 104857600"
sudo /sbin/sysctl -w net.ipv4.tcp_wmem="4096 524288 104857600”
# Set maximum network input buffer queue length
sudo /sbin/sysctl -w net.core.netdev_max_backlog=250000
```

# Disable caching of TCP congestion state (2.6 only); fixes a bug in some Linux stacks:

```
sudo /sbin/sysctl -w net.ipv4.tcp_no_metrics_save=0
```

Preliminary Measurements

Latency measurements for ptuDAPL

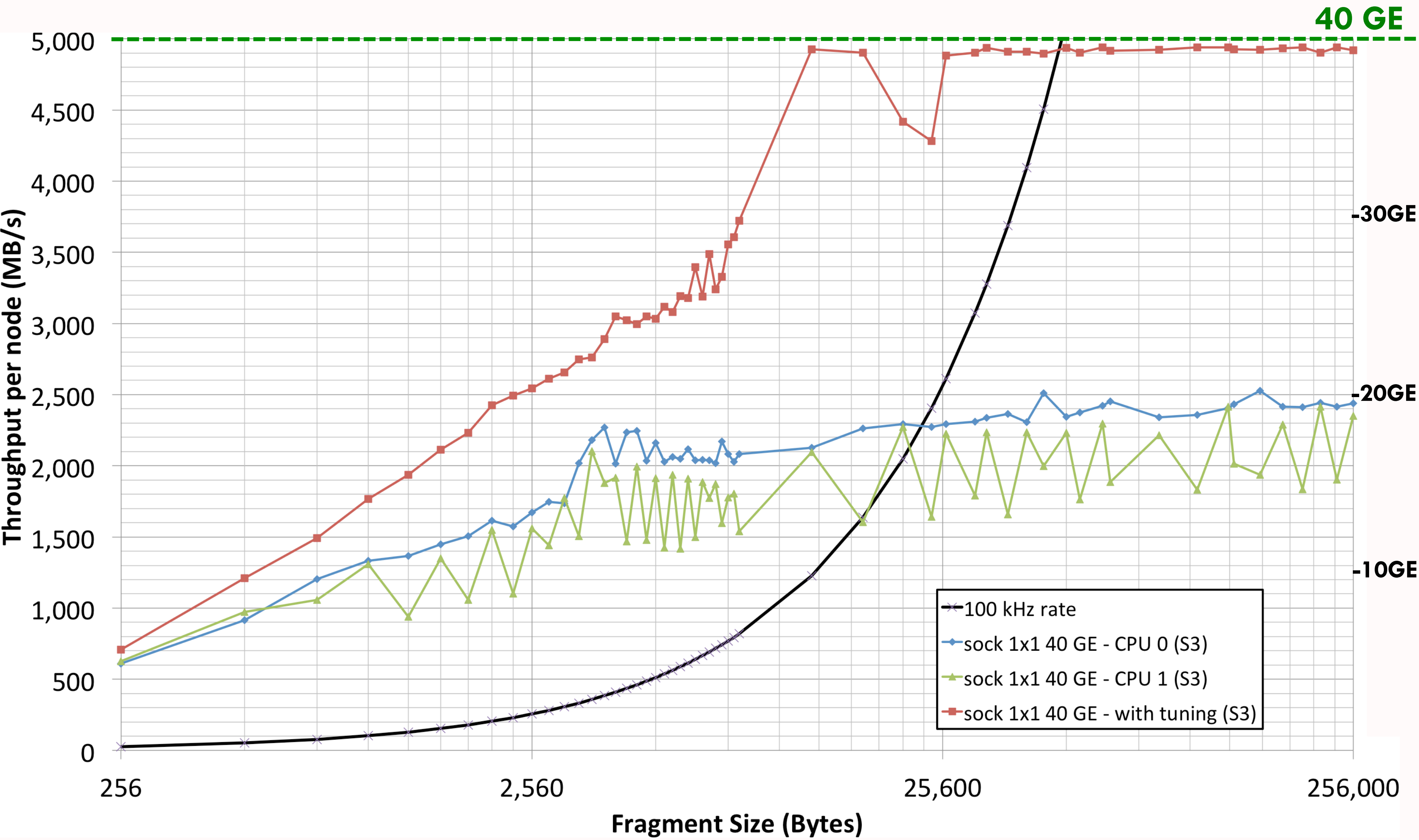


8.87 us for 32 Bytes

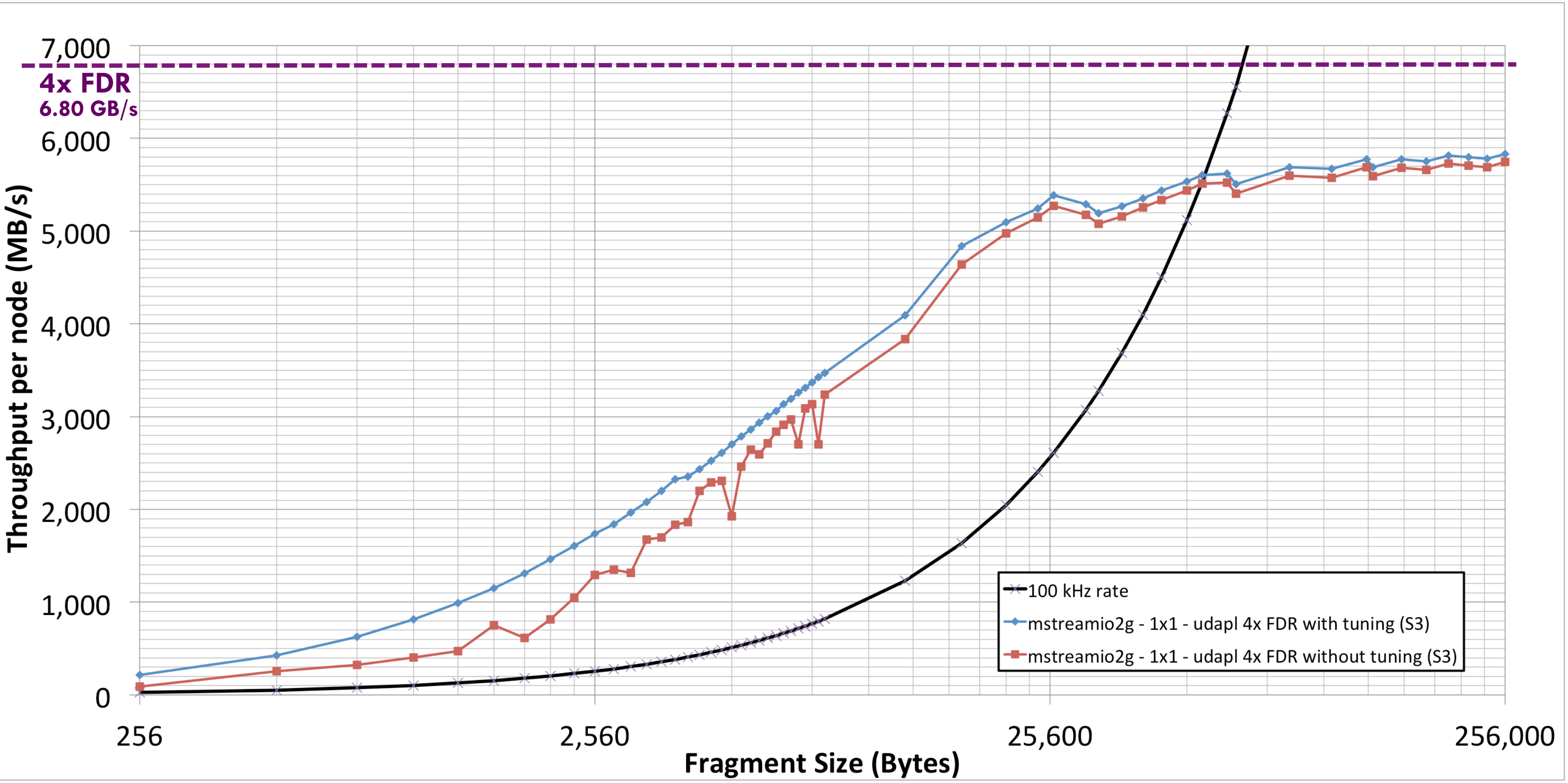
6.33 us for 32 Bytes

2.51 us for 32 Bytes

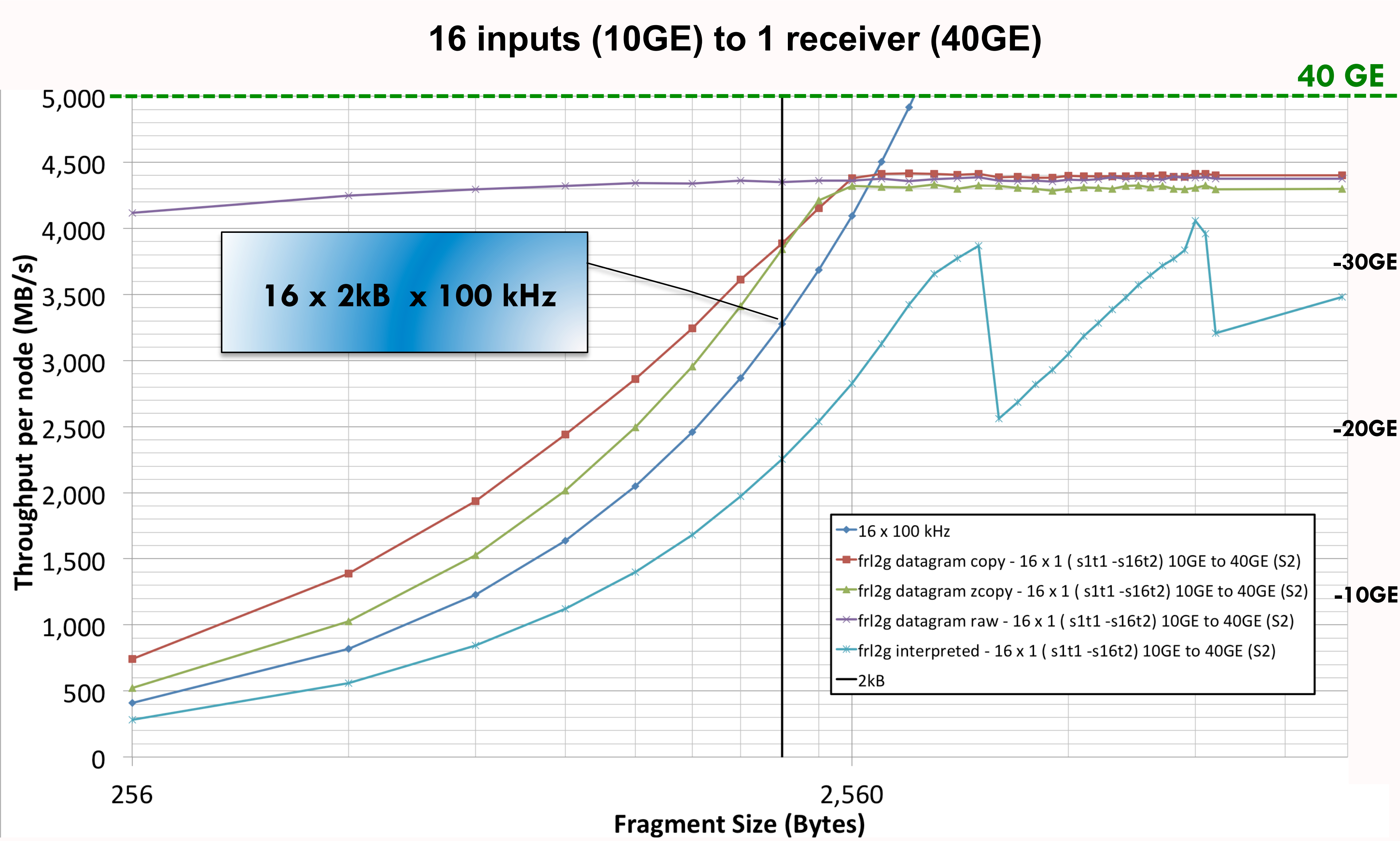
TCP Point to Point – 40GE



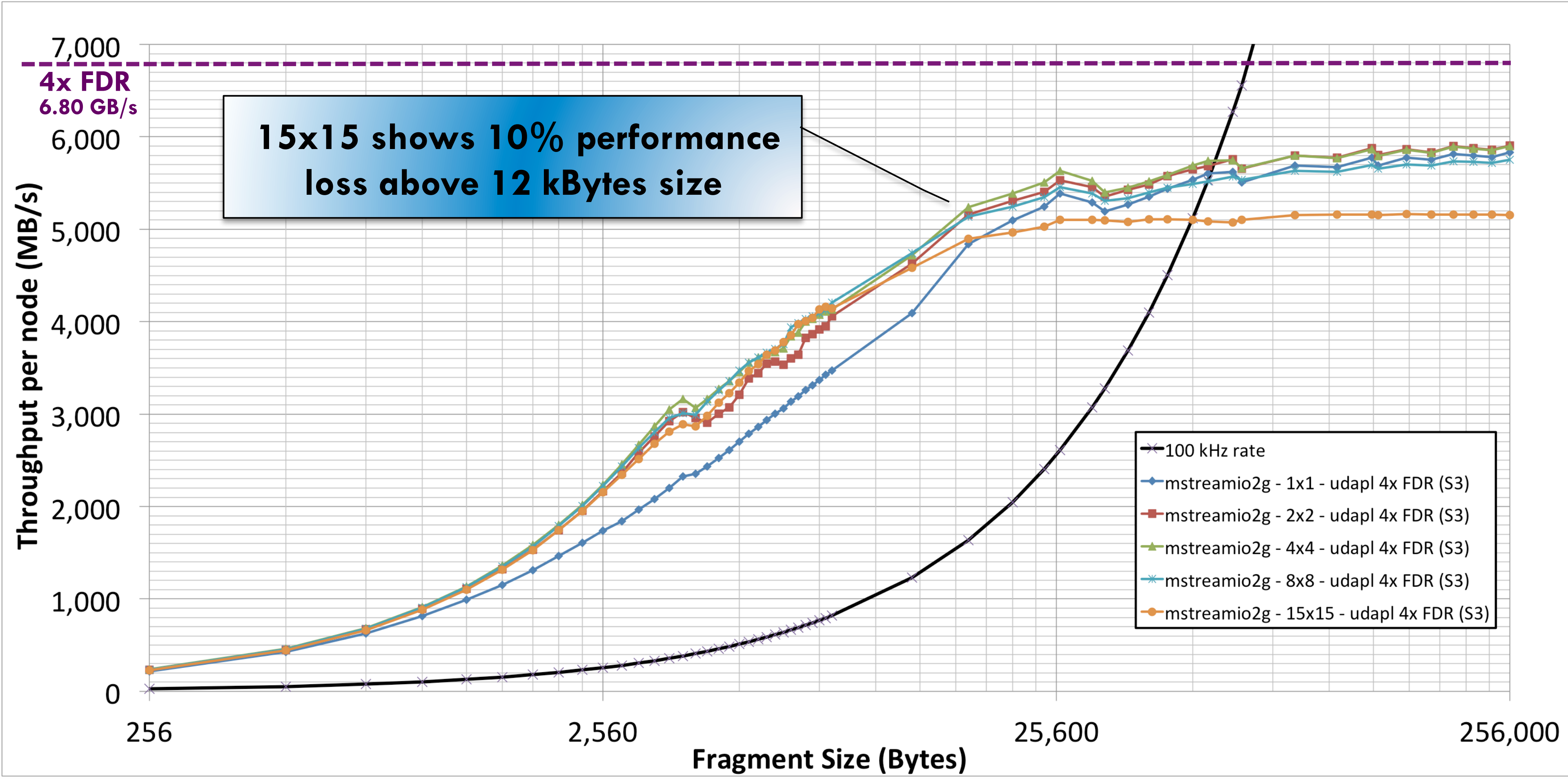
uDAPL Point to Point – IB



Ferol Merger

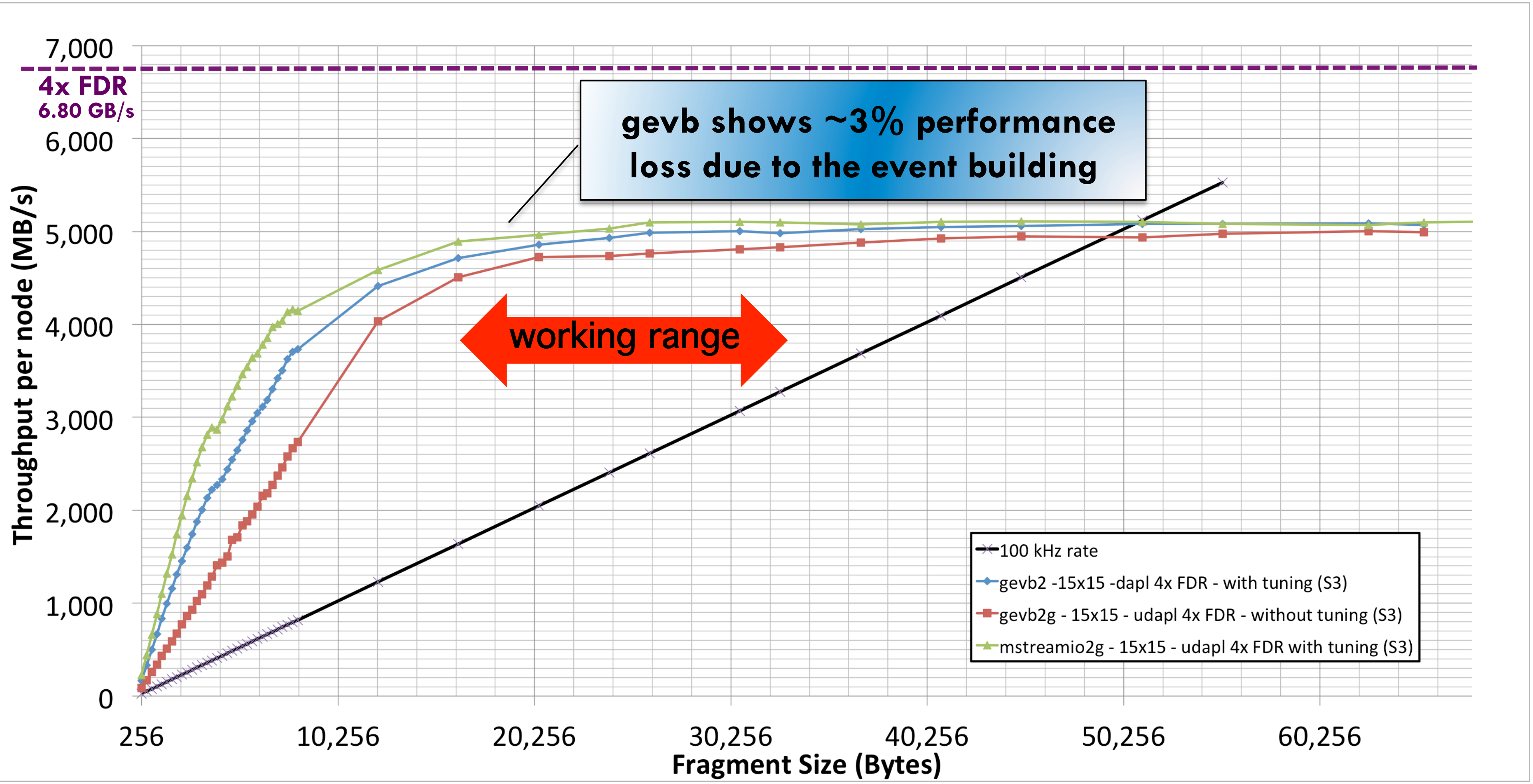


IB Scalability: from 1x1 to 15x15



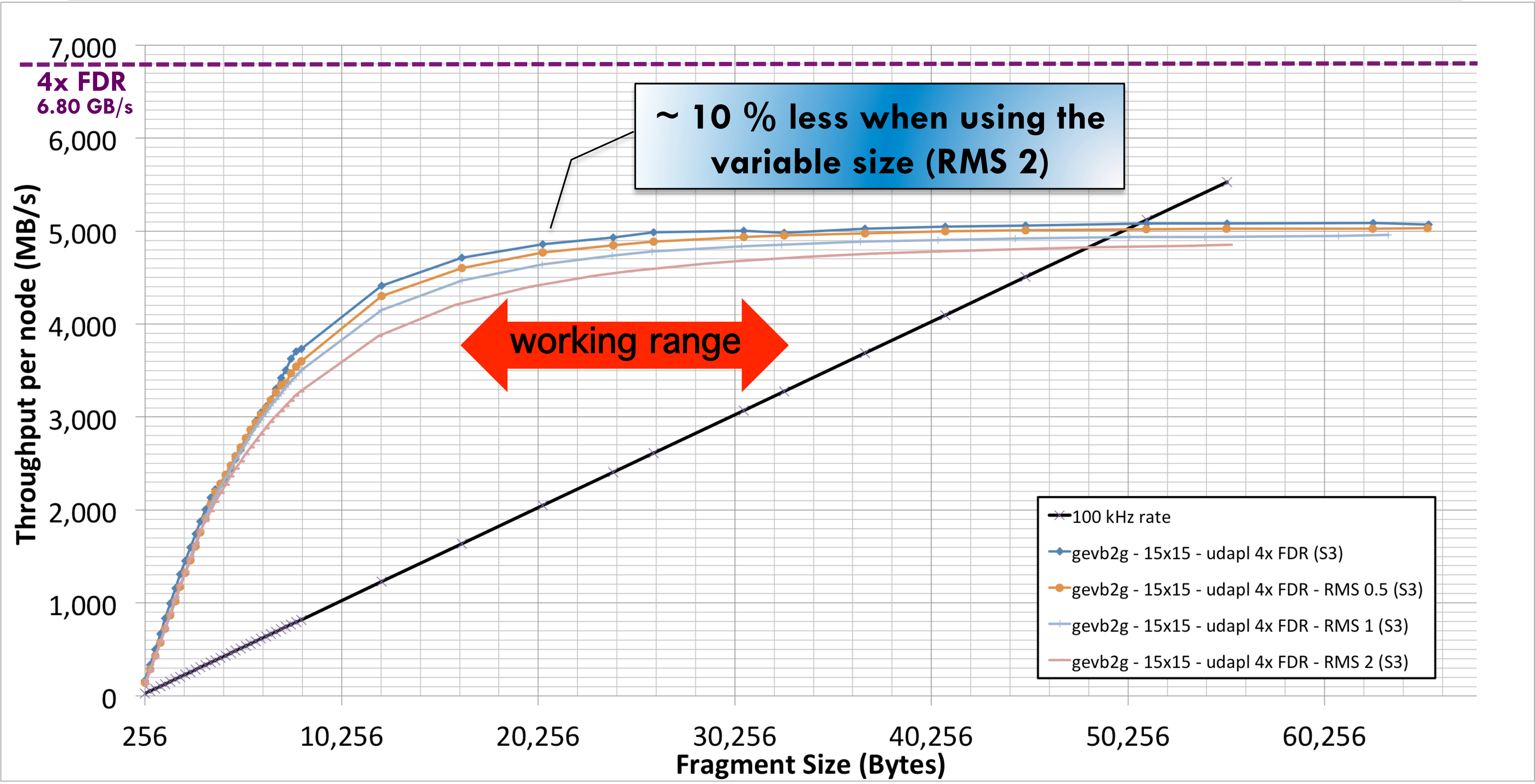
15x15 shows 10% performance loss above 12 kBytes size

IB Event building



gevb shows ~3% performance loss due to the event building

IB Event building with variable fragment size

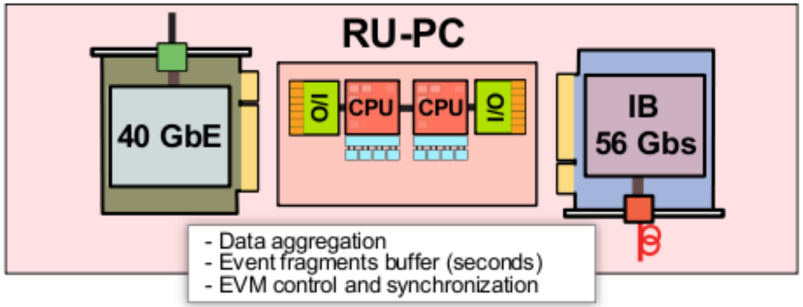


~ 10 % less when using the variable size (RMS 2)

Fragment size - Bytes	100 kHz MB/s	Fixed	RMS 0.5	RMS 1	RMS 2
16384	1638	4715	4604	4450	4209
32768	3277	4983	4951	4850	4701

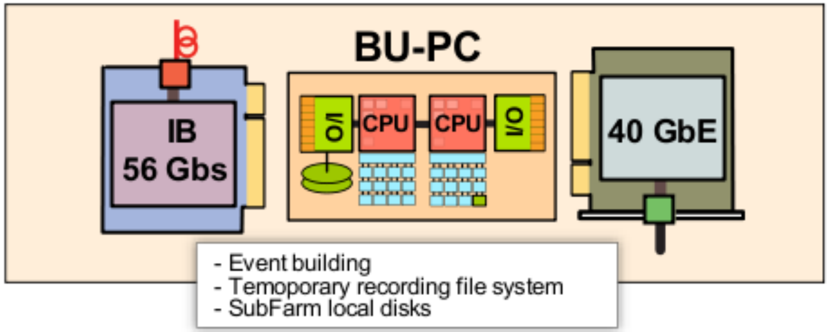
Pending Issues

- Simultaneous input/output on RU



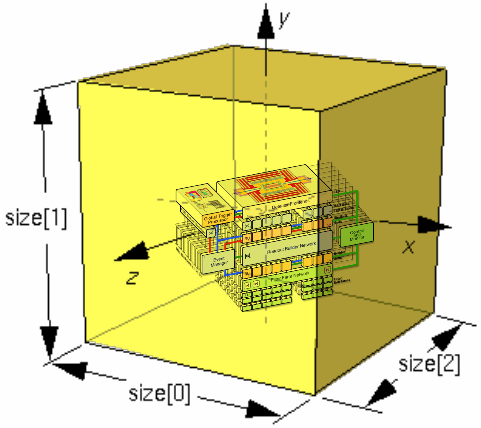
- Data aggregation
- Event fragments buffer (seconds)
- EVM control and synchronization

- Simultaneous input/output on BU



- Event building
- Temoporary recording file system
- SubFarm local disks

- Scaling of EVB from 15x15 to 72x48



Mellanox experience

- Open 9 cases
  - explain some misunderstanding
    - CX3 - 4K mtu setting on HCA limits capable vl's to 0-1 only

- Ethernet Driver
  - Distribution OFED and Ethernet tarball are not consistent
  - 10 GE TCP streams stop after a few hours with 40GE NIC and 40 GE switch (EN driver version 1.5.9)
  - Connection timeout under heavy traffic load (suppose to be fixed in version 1.5.8.3)

- ConnectX3 OEM we are using has ~6 months delay with firmware

Summary

OFED API experience

- DAPL(OFED stack) library stable and reliable on all tested environments and technologies
- Thin code implementation as compared to socket programming
- Similar approach to sockets for connection establishment (asynchronous)
- Standard portable code (Infiniband and Ethernet)
- Reliable datagram support fits nicely XDAQ - CMS online framework
- DAPL SEND/RCV preferred to RDMA (not fitting our application domain and framework)