Experience with Infiniband for CMS Event Building

Andrea Petrucci – CERN (PH/CMD)

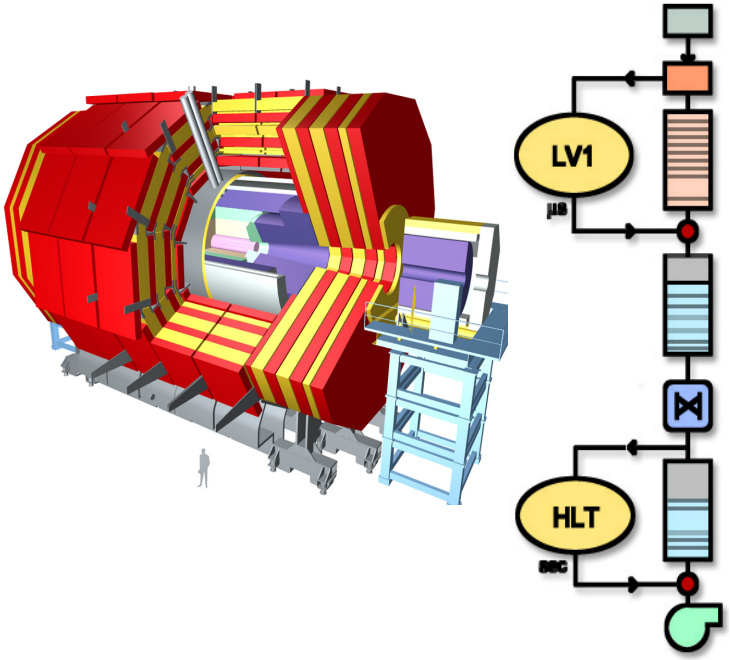
Istituto Nazionale di Fisica Nucleare (INFN), CNAF Seminar
19 September 2014, Sala Garisenda, Via Ranzani n. 13/2, Bologna, Italy

Outline

- CMS Experiment in LHC run 2
  - Event builder changes
- Feasibility studies
- Infiniband Verbs (ptIBVerbs)
- DAPL (ptUDAPL)
- IB Verbs vs DAPL
- Performance tuning
- Preliminary Results
- Conclusion

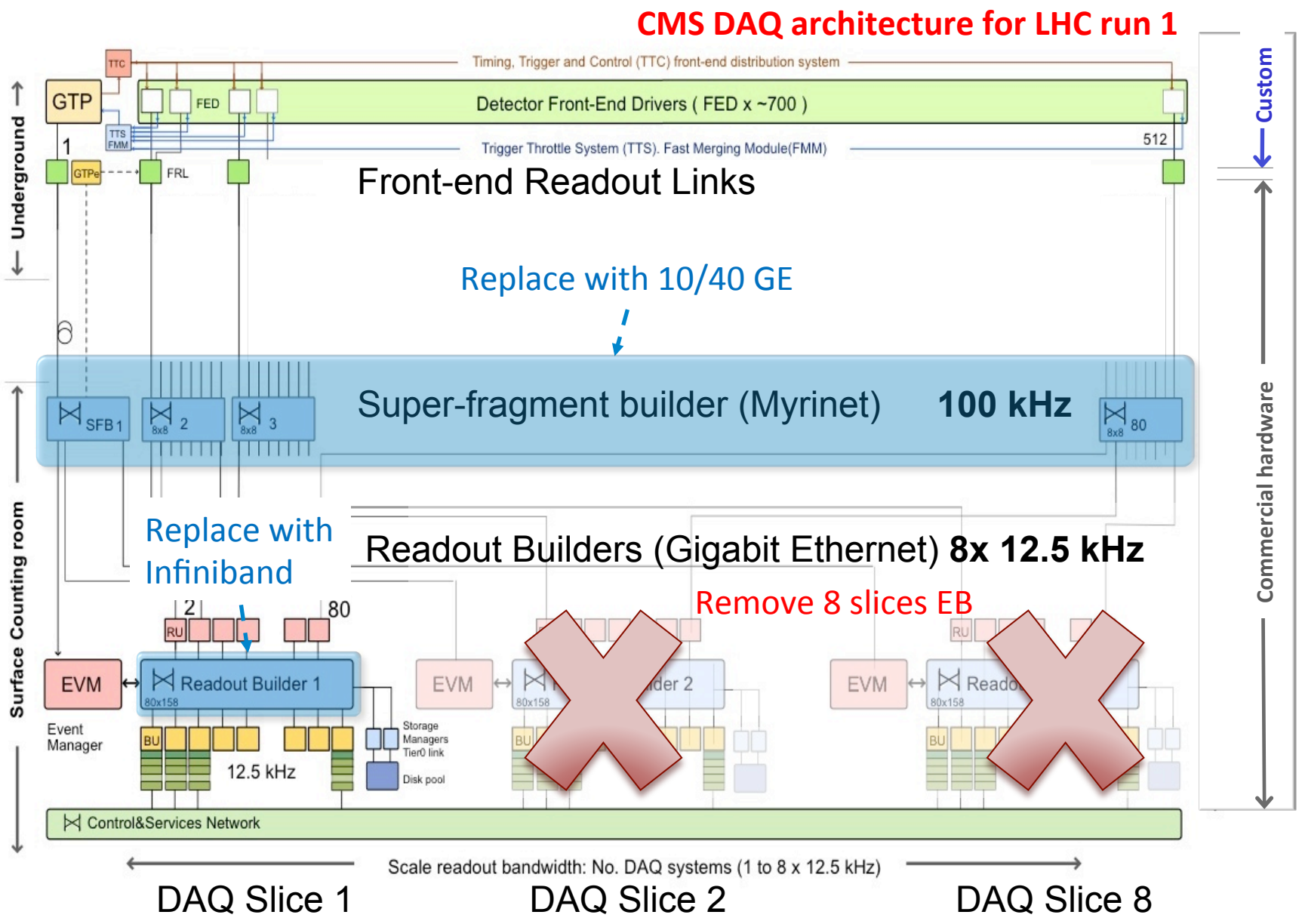
CMS Experiment in LHC run 2

CMS DAQ Requirements for LHC Run 2

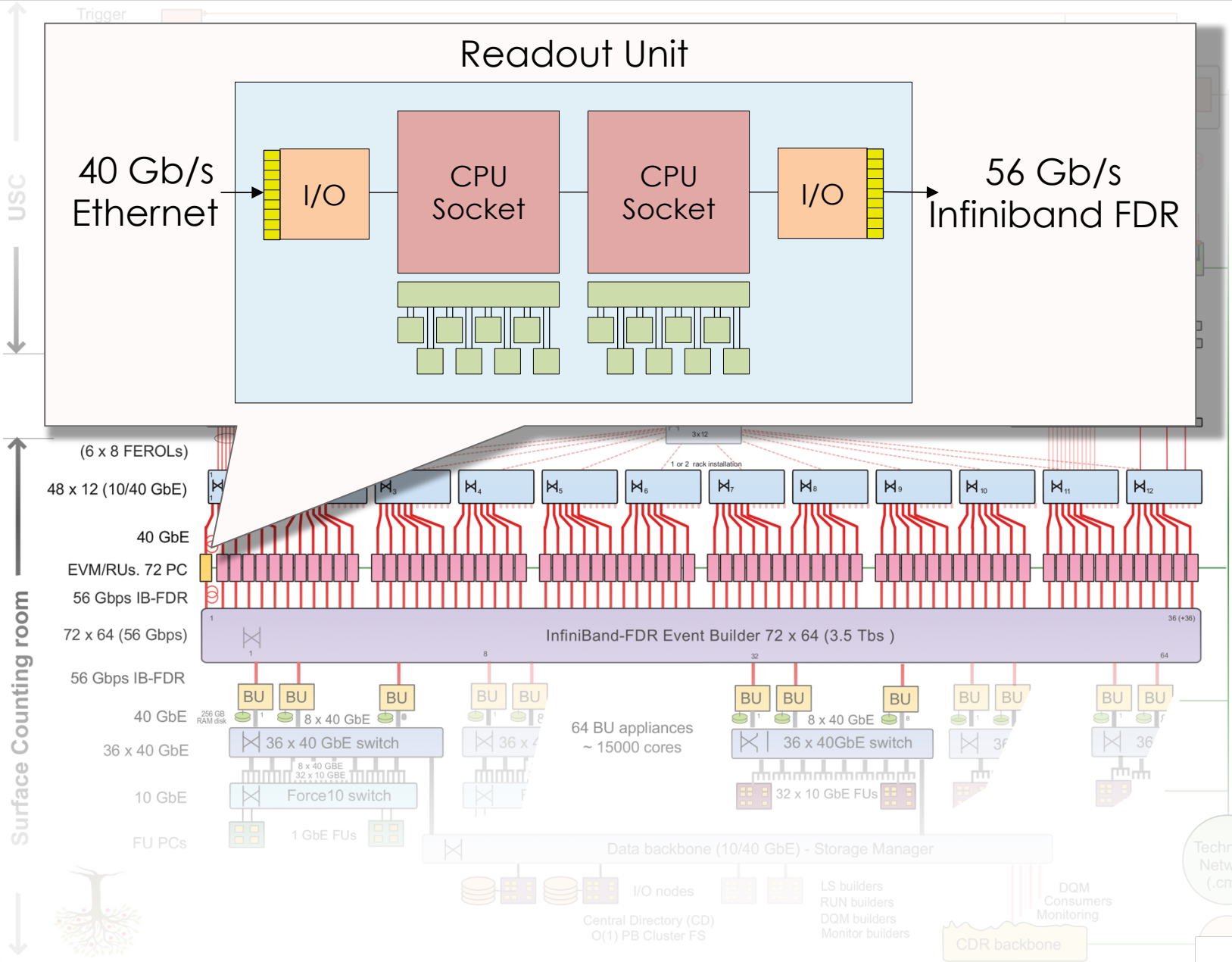


Parameters	
Data Sources (FEDs)	~ 620
Trigger levels	2
First Level rate	100 kHz
Event size	1 to 2 MB
Readout Throughput	200 GB/s
High Level Trigger	1 kHz
Storage Bandwidth	2 GB/s

Event builder changes



CMS DAQ System for LHC Run 2



Feasibility studies

Networking technologies

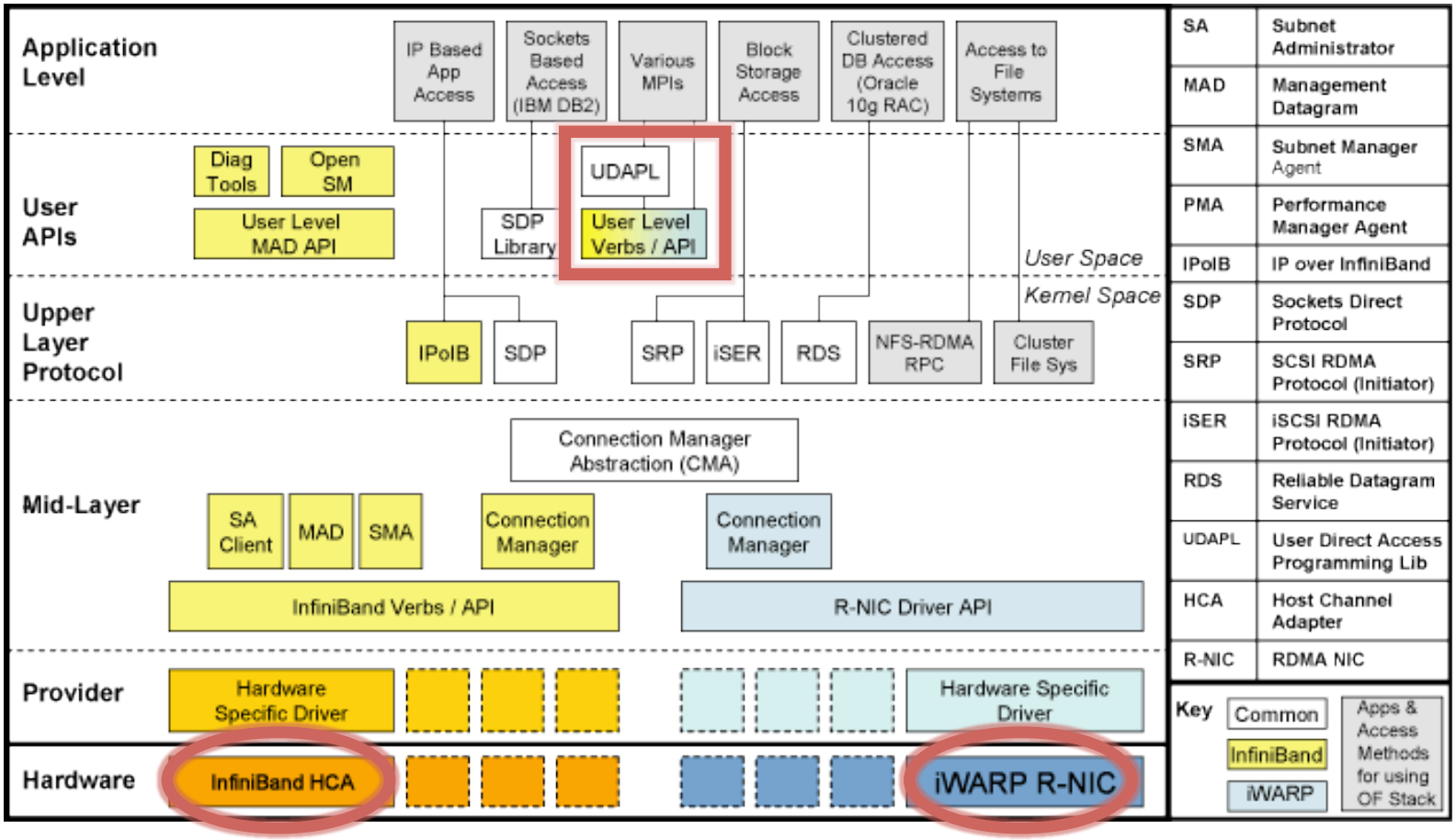
Our feasibility studies are focused in two network technologies

- **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), **IB verbs** (Infiniband verbs) and IPoIB (IP over InfiniBand)

The OFED Stack (source: OpenFabrics Alliance)

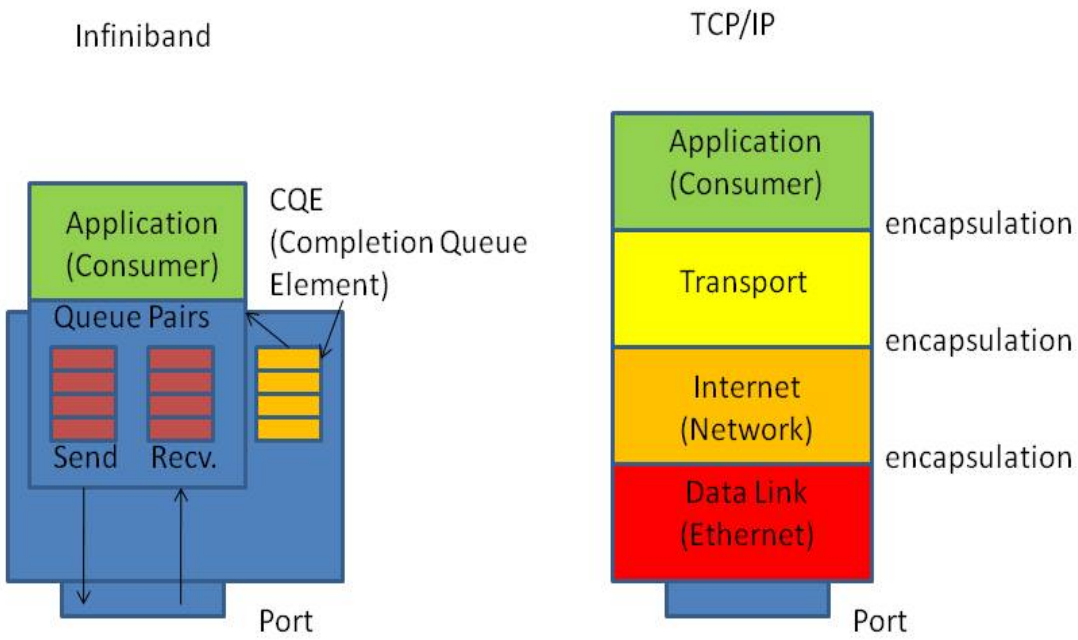
A unified, cross-platform, transport-independent software stack for RDMA and kernel bypass

- http://www.openfabrics.org/



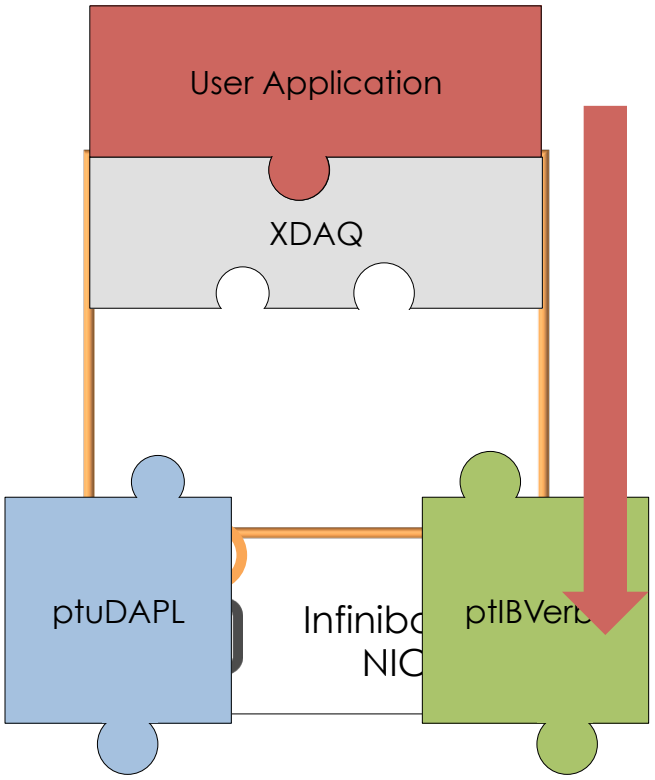
Comparison of the Stacks (Infiniband vs TCP/IP)

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



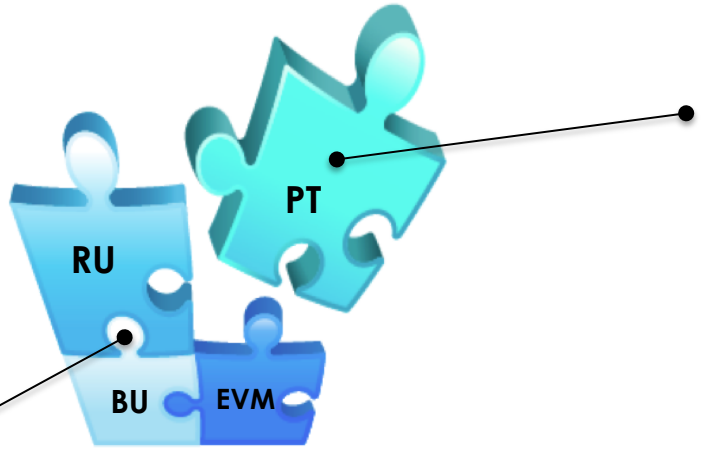
Infiniband Peer Transports

- Two new XDAQ Peer Transports
  - uDAPL -> ptuDAPL
  - verbs -> ptIBVerbs
- Full integration into XDAQ framework
- Event based API
- Send/receive with reliable connections
- Buffer loaning – zero-copy
- Memory pools automatically register memory with the NIC
  - Translation of virtual to physical addresses
  - Pinning memory to avoid swapping



Event builder software

The ptuDAPL and ptIBVerbs are pluggable components to transparently access the IB network in XDAQ – CMS online framework



Re-use and adapt same Event Builder software

Infiniband verbs (ptIBVerbs)

IB verbs specification

- RDMA read and write requires additional communication to alert remote processes of completed transactions. Therefore,

- We are using the 'Send' and 'Receive' operations with 'Reliable Connections'

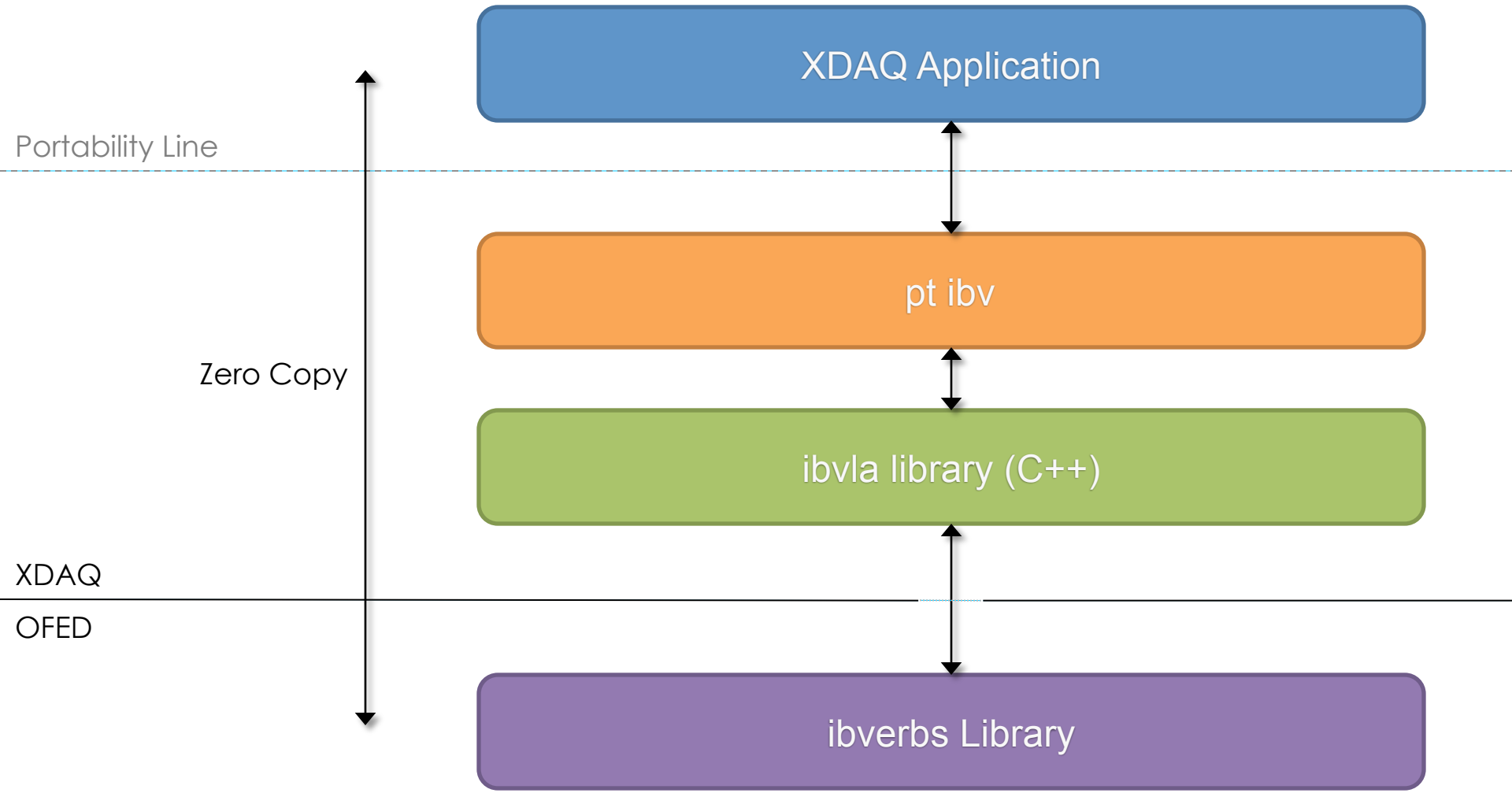
Operation	UD	UC	RC	RD
Send	x	x	x	x
Receive	x	x	x	x
RDMA Write		x	x	x
RDMA Read			x	x
Atomic: Fetch and Add/Cmp and Swap			x	x
Max Message Size	MTU	2GB	2GB	2GB

- Reliable Datagrams is not currently implemented in Mellanox HCA

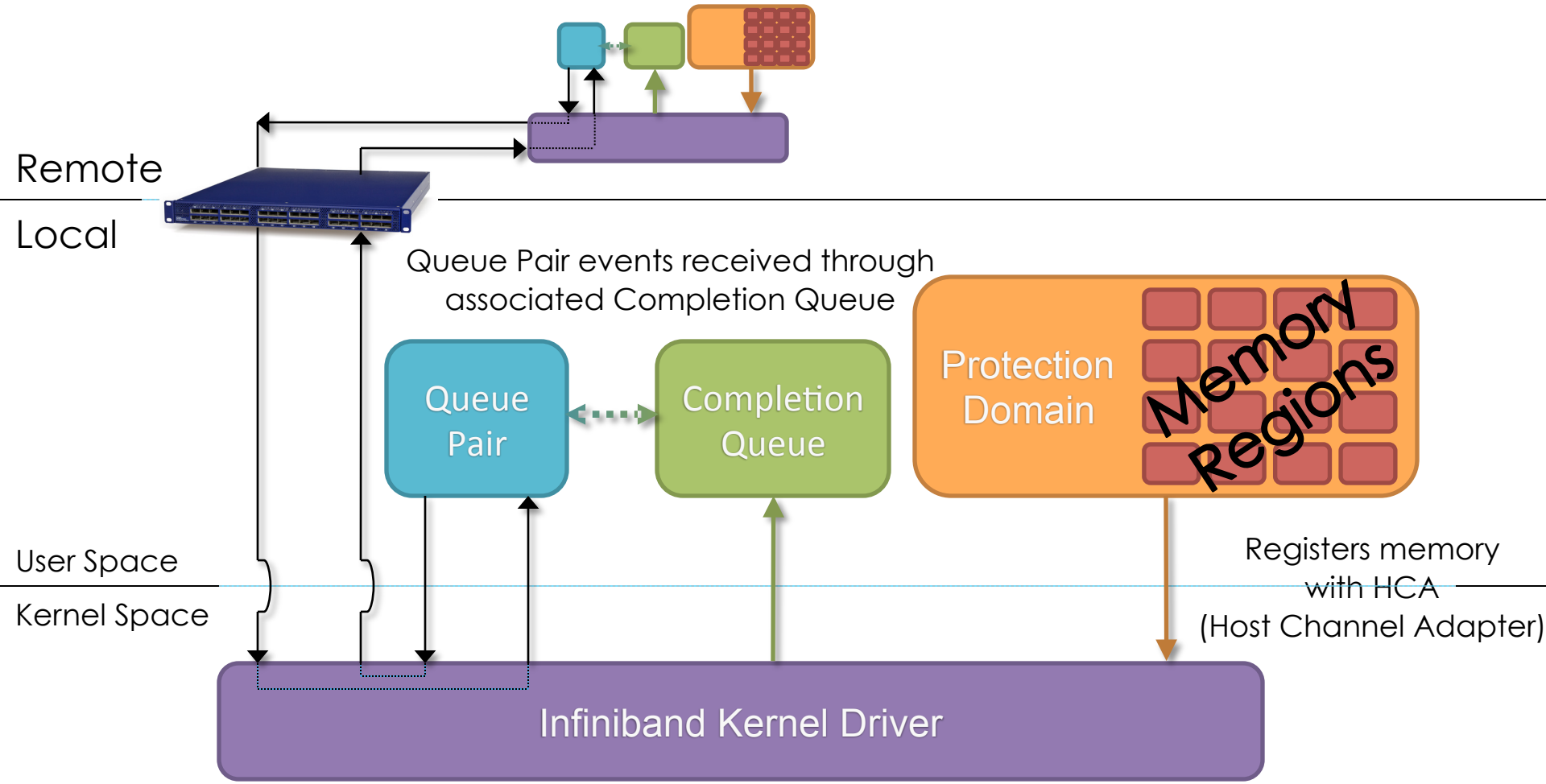
Infiniband Verbs Concepts

- **Device** – The physical Host Channel Adapter (HCA)
- **Context** – A user 'context' within which a user can interact with the HCA (1 per device per application)
- **Protection Domain** – Memory is registered to a Protection Domain, which in turn registers the memory with the device
- **Memory Region** – A Memory Region represents a buffer that is registered to a Protection Domain
- **Queue Pair** – The IBV equivalent to a socket which internally consists of a send queue and a receive queue
- **Work Request** – Send and receive requests are represented as Work Requests
- **Completion Queue** – Queue Pairs report completion of Work Requests through Completion Queues (there can be many Queue Pairs for a single Completion Queue)
- **Asynchronous Events Queue** – Errors and other asynchronous events are reported through a special asynchronous event queue

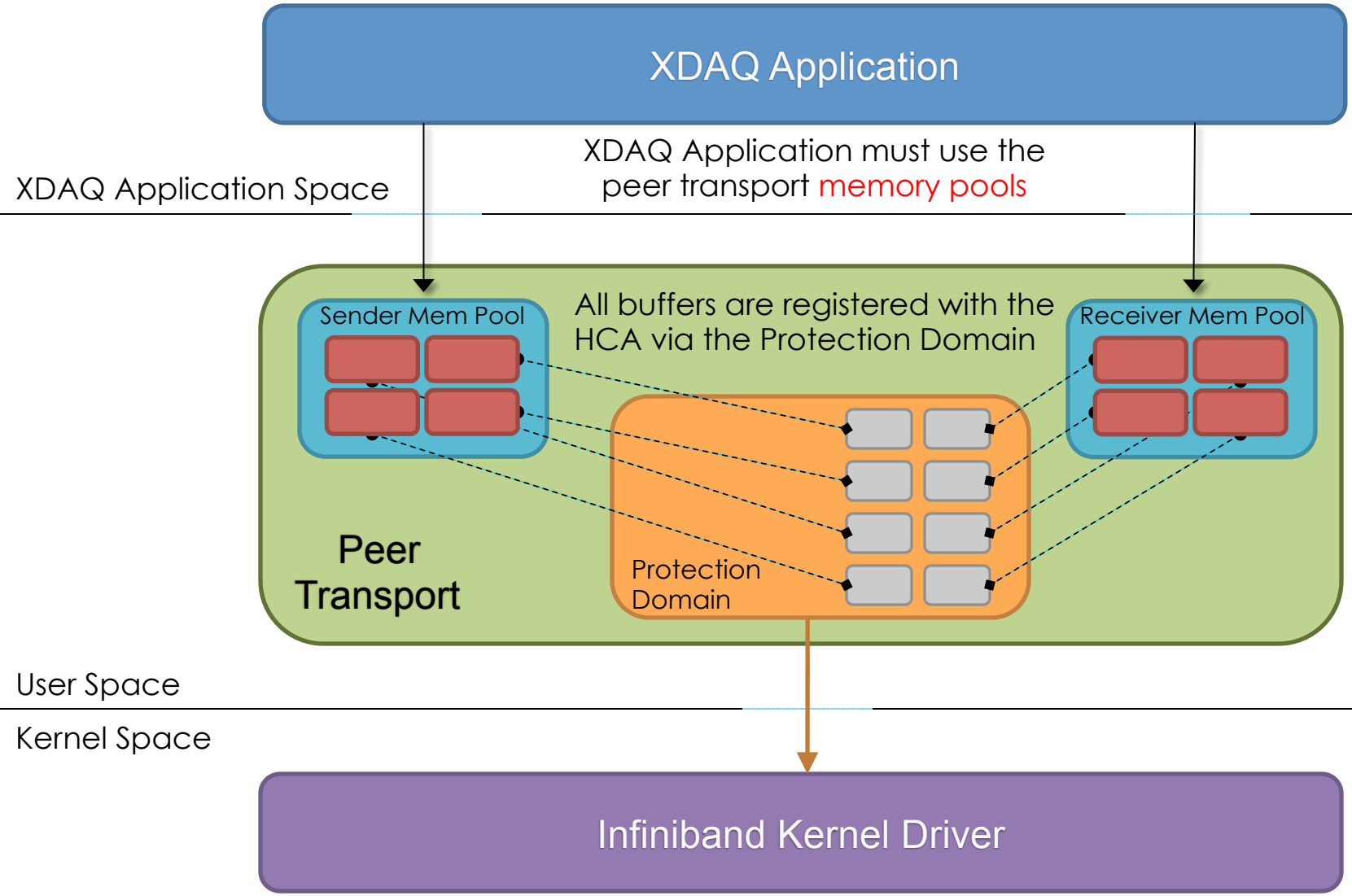
Software Stack



Infiniband Architecture Principles

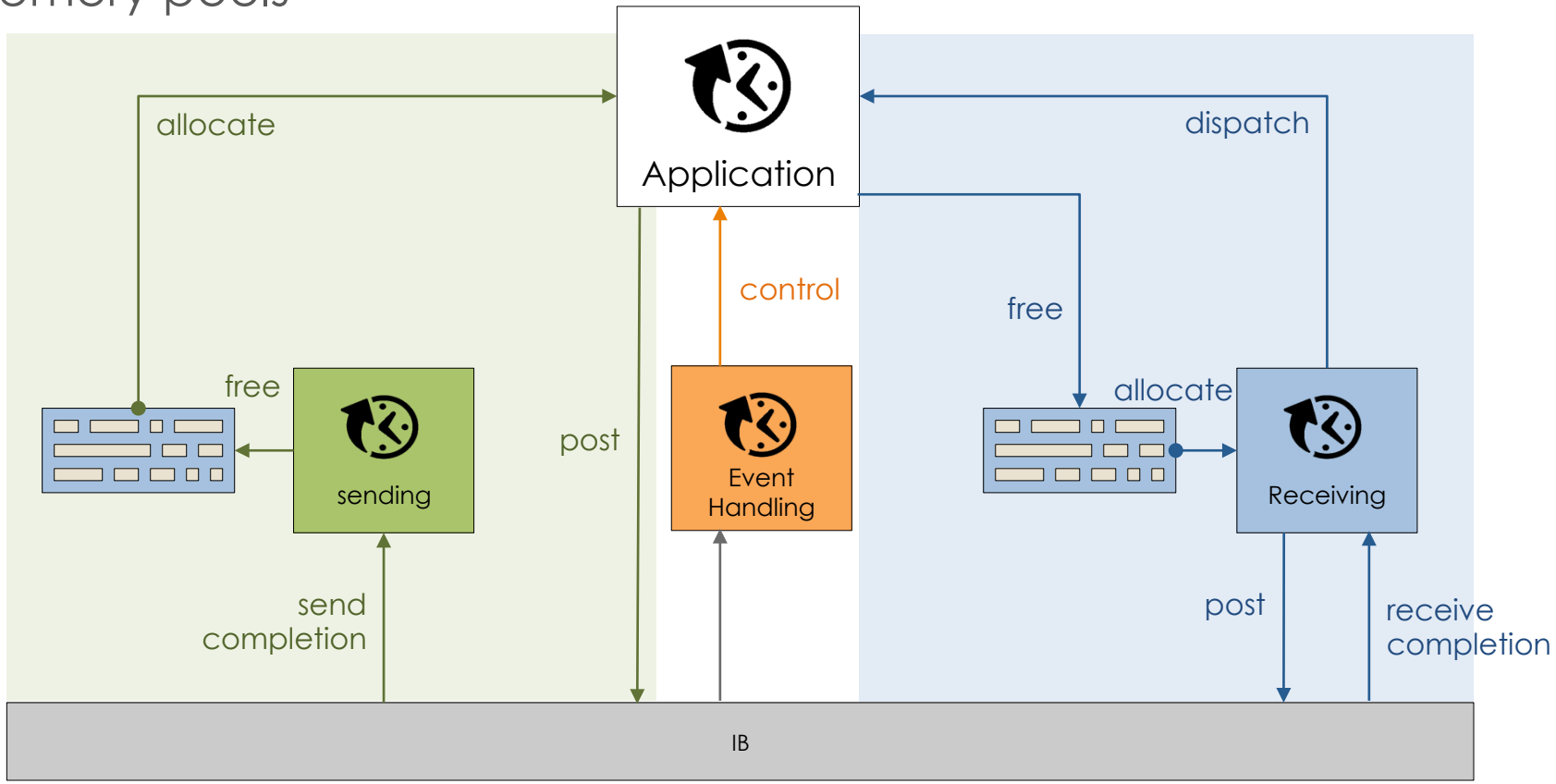


Peer Transport Memory Management



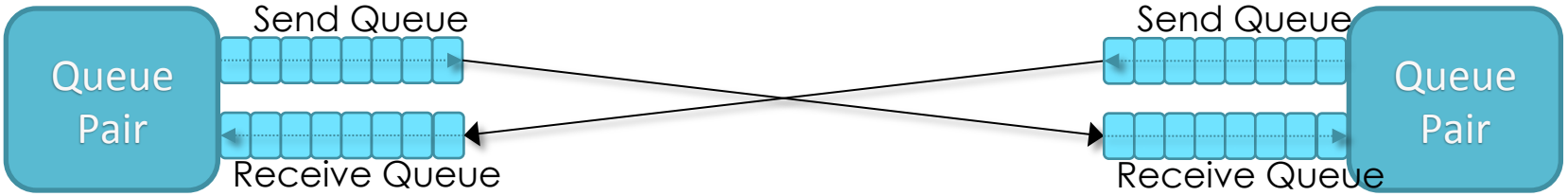
Parallelism

- Work is distributed across several XDAQ workloops
- Workloops are bound to run on one or more cores
- Sending/receiving operations separated and use different memory pools



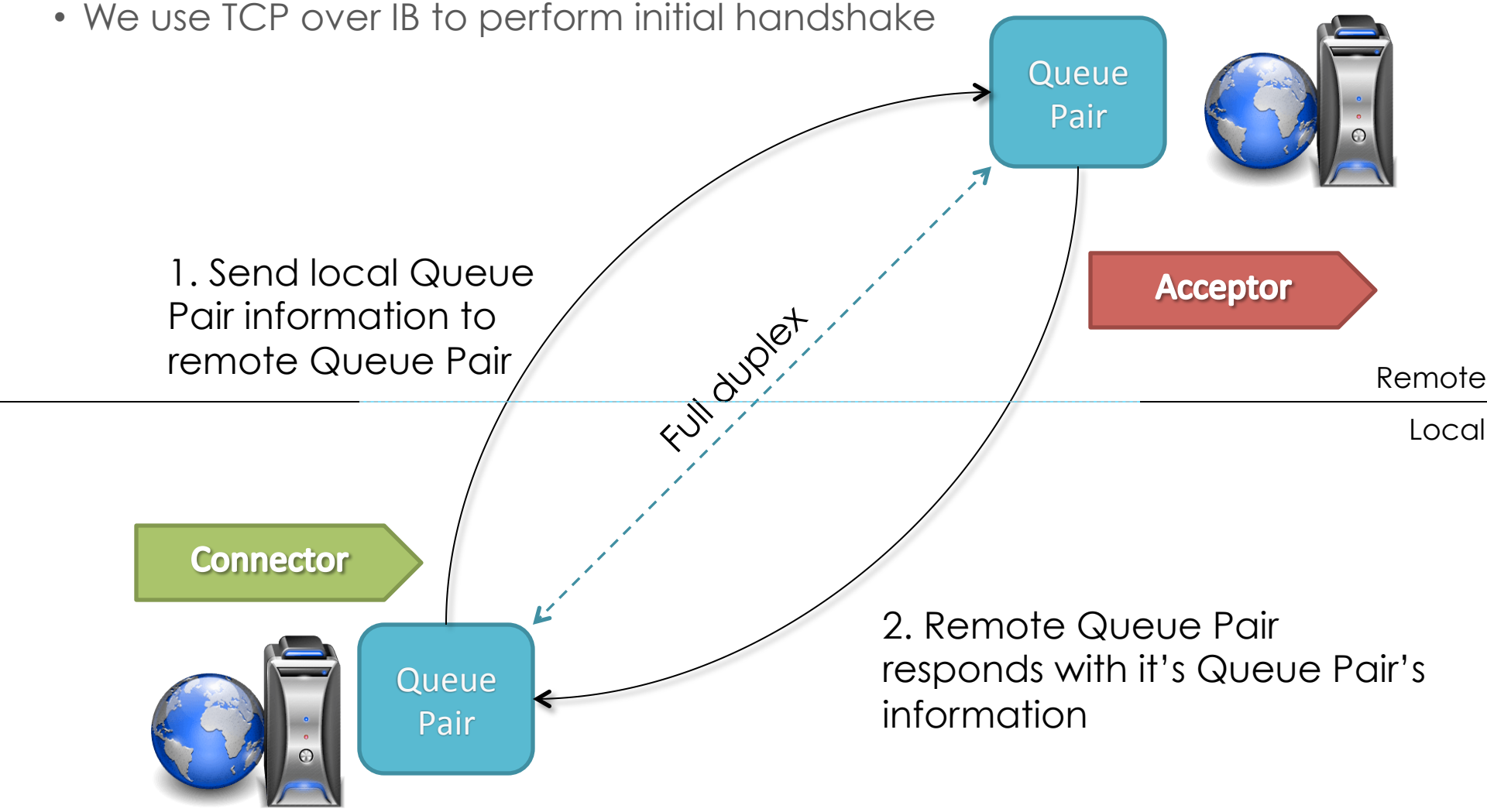
Infiniband Connections

- Infiniband supports "reliable connections", but they are really reliable, connectionless datagram messaging services
- Queue Pairs store routing information for the remote Queue Pair, which is used to route all outgoing packets
- A connected pair of Queue Pair's effectively have their internal send and receive queues connected to each other respectively
- The relationship between Queue Pairs is 1 to 1 (no multicast)



ptIBV Connection Setup

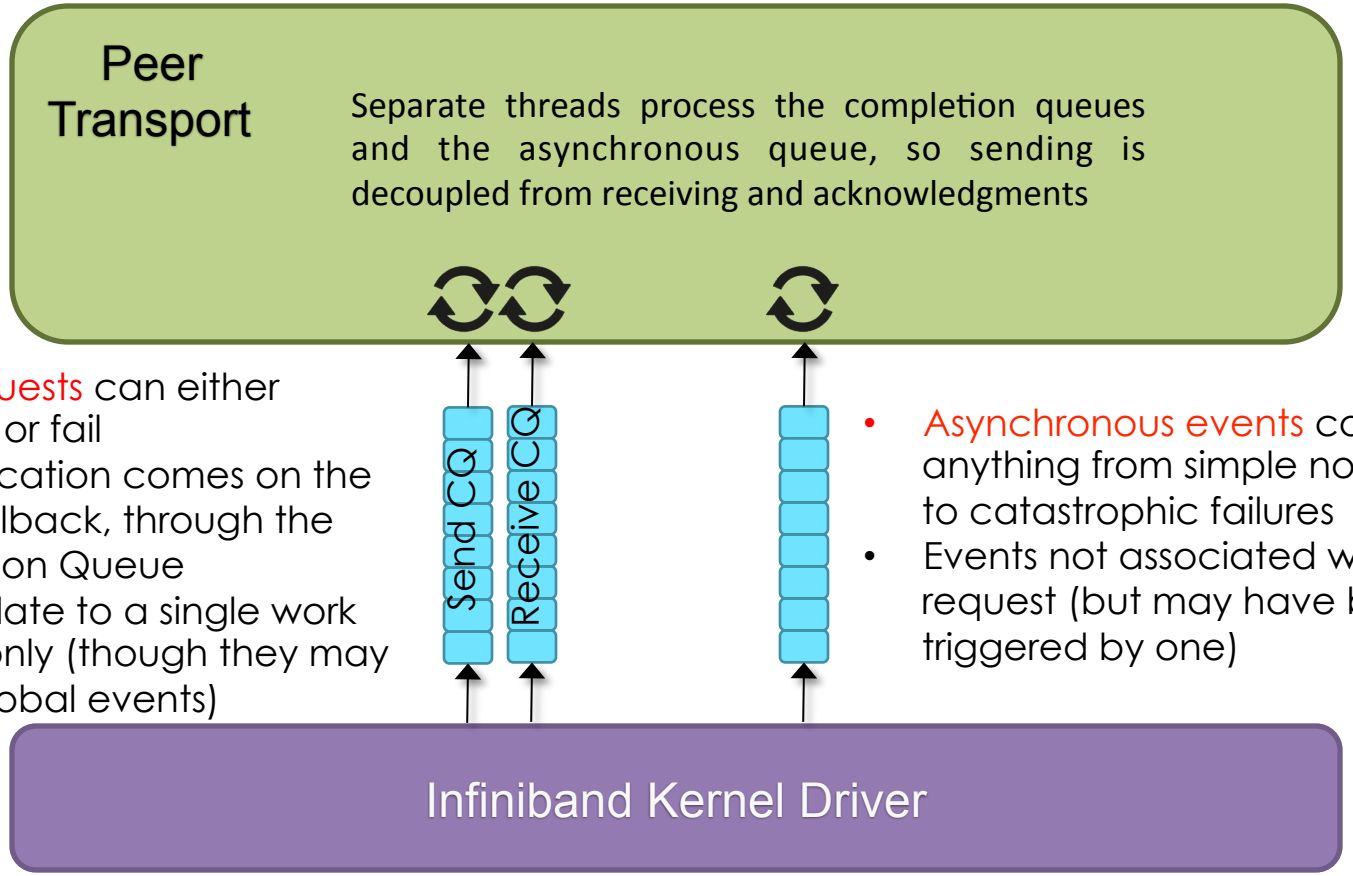
- No proprietary ib verbs connection mechanism
- We use TCP over IB to perform initial handshake



Connector/Acceptor based on ACE (The ADAPTIVE Communication Environment)

Events and Error Handling

- Two types of notification, work completions and asynchronous event, which are sent through separate queues



- Work requests can either succeed or fail
- The notification comes on the same callback, through the Completion Queue
- Events relate to a single work request only (though they may trigger global events)

- Asynchronous events can relate to anything from simple notifications to catastrophic failures
- Events not associated with a work request (but may have been triggered by one)

The Life of a Queue Pair

1. QP created.
2. QP initialized.
3. Make QP RTS or RTR (connection).
4. On error, move into the error state. This flushes all outstanding work requests.
5. QP stays in error state. A new QP is created.

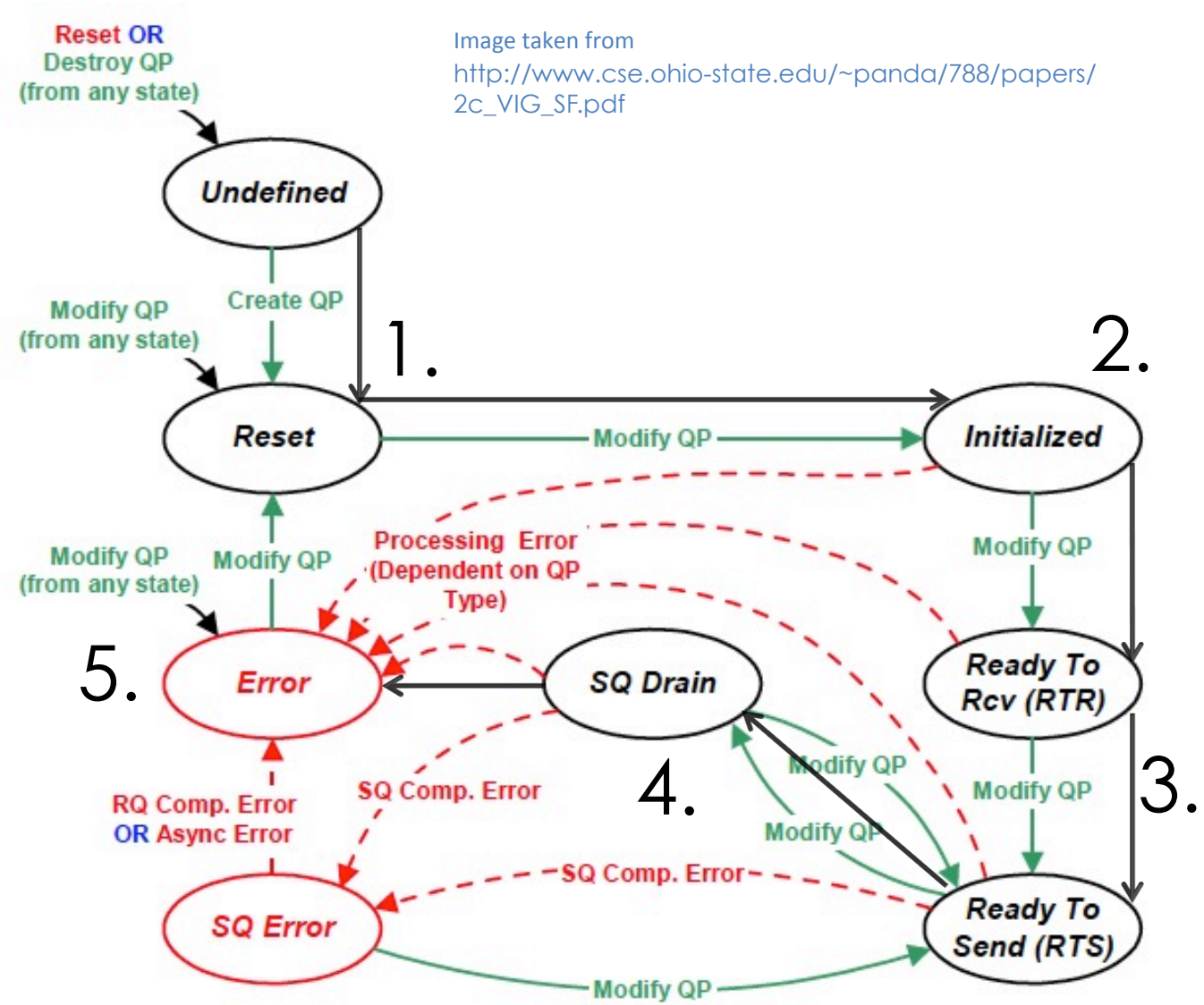
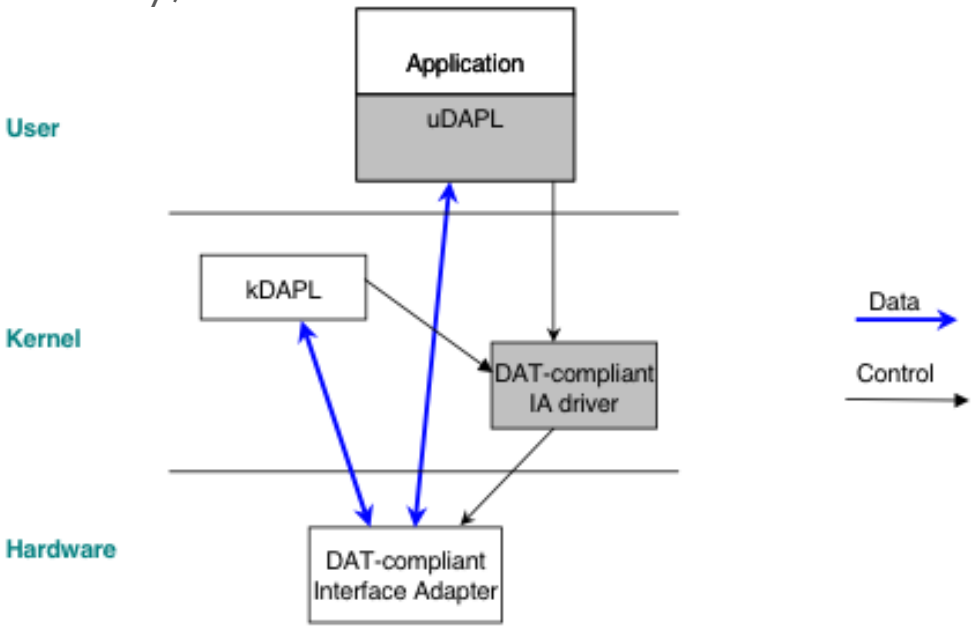


Image taken from
http://www.cse.ohio-state.edu/~panda/788/papers/2c_VIG_SF.pdf

DAPL (ptUDAPL)

DAT Model (source DAT Collaborative)

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



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

IB verbs vs DAPL

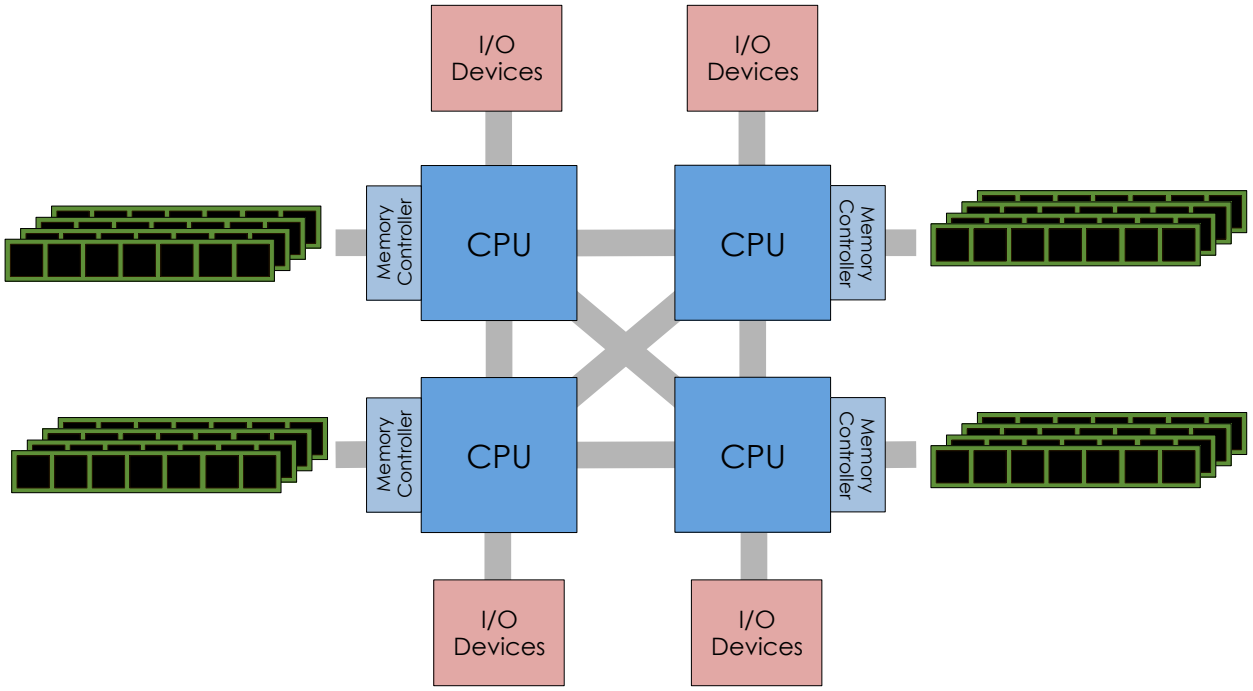
ptIBV vs ptuDAPL

ptIBV	ptuDAPL
Complex API	Simple API
High Performance	High performance
High stability	Average stability
High flexibility	Moderate flexibility
Well documented	Poorly documented
Open connection method	Connection method
Core Infiniband Specification	3rd Party

Performance Tuning

Non-Uniform Memory Access (NUMA)

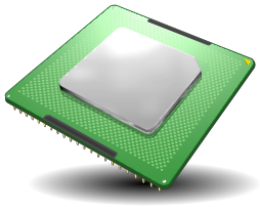
- The distance between processing cores and memory or I/O interrupts varies within Non-Uniform Memory Access (NUMA) architectures



Non-Uniform Memory Access (NUMA)

Performance Factors

I/O Interrupt affinity

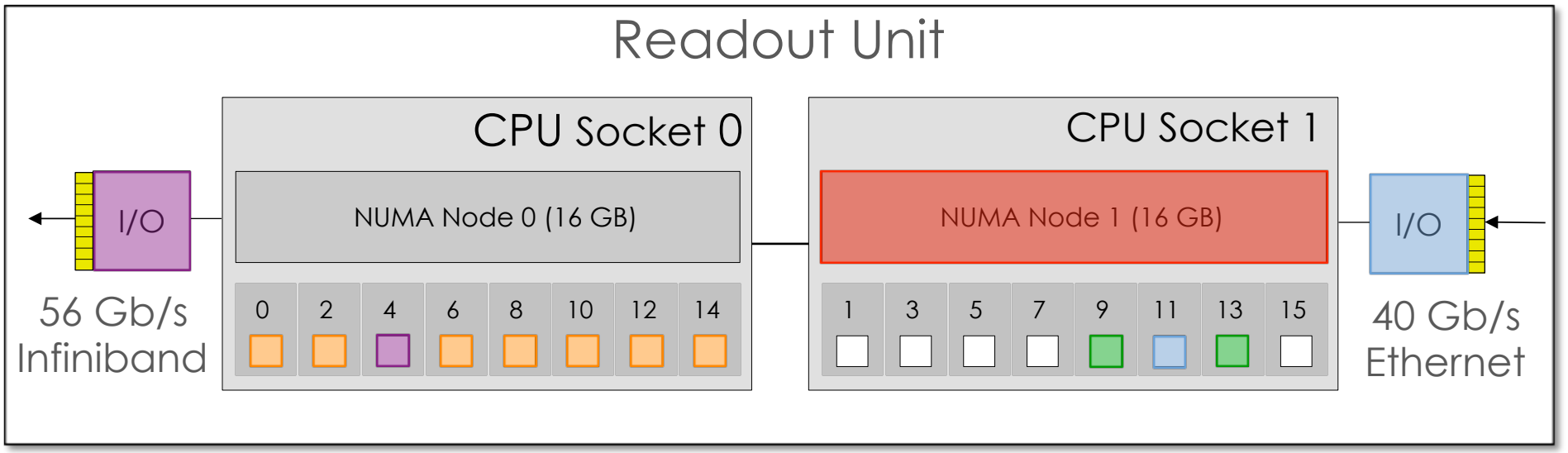


CPU affinity

Memory affinity

TCP Custom Kernel Settings

Readout Unit Machine Affinities



Affinities

- 40 GbE - I/O Interrupts
- 40 GbE - Socket reading
- Infiniband - I/O Interrupts
- DAQ threads
- DAQ Memory allocation
- Cores available to operating system

Preliminary Results

Test Setup

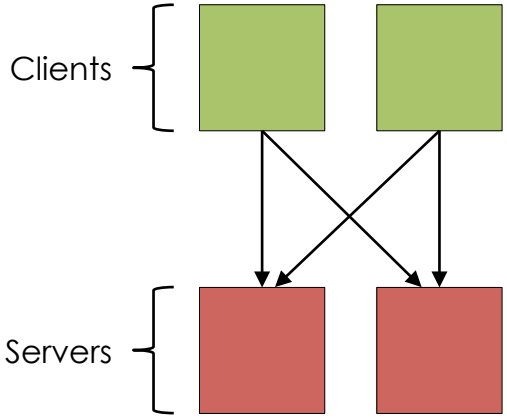
- Small scale with 4 nodes on 1 switch
  - 1x1 and 2x2 (RU x BU) tests for…
- N-to-N and event building tests
- Each node has…
  - Dual socket Intel Xeon E5-2670 8-core processors @ 2.6 GHz
  - 16 GB RAM per socket (NUMA)
  - Mellanox Connect X-3 VPI Infiniband FDR network card
  - OFED v 2.x
  - Scientific Linux (CERN) 6

	SDR	DDR	QDR	FDR-10	FDR	EDR
1X	2	4	8	9.67	13.64	25
4X	8	16	32	38.79	54.54	100
12X	24	48	96	116.36	163.64	300

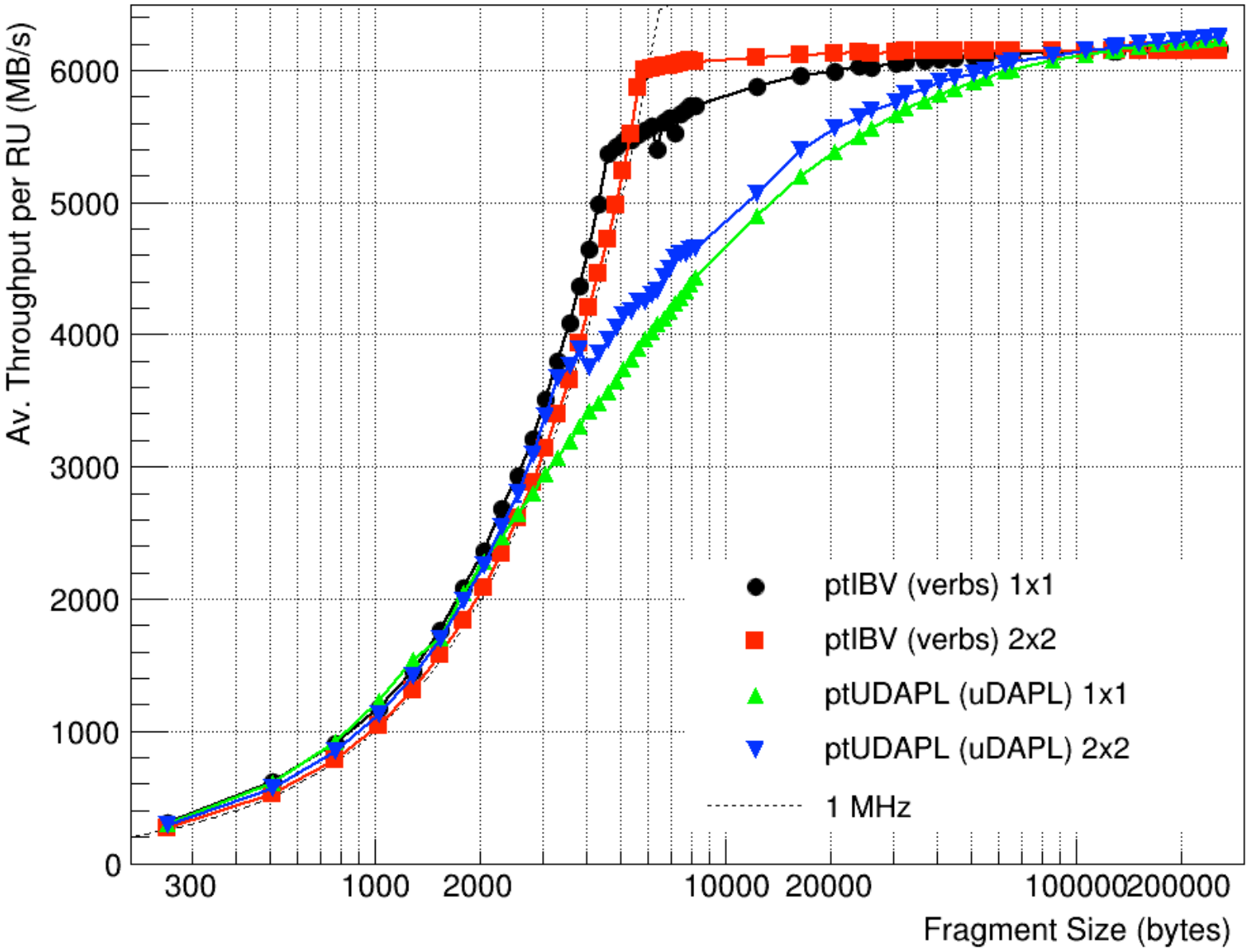
Effective unidirectional theoretical throughput in Gb/s

N-to-N Setup

- N clients each send to N servers for each ‘message’
- The measurement is the rate of receiving in the receivers
- No additional processing
- Fixed sized messages, round robin dispatching in the senders
- Test to show the performance for unidirectional throughput

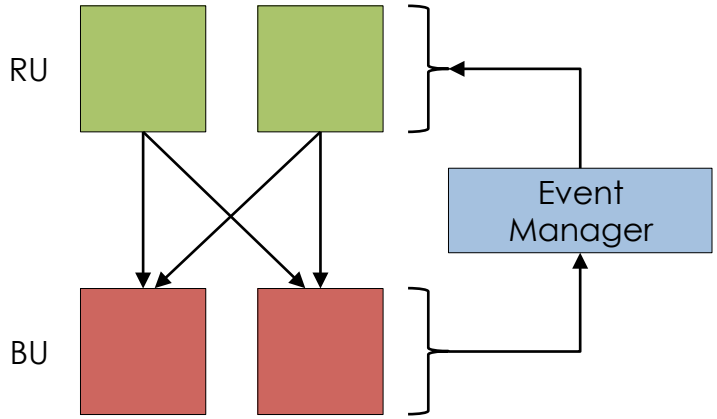


N-to-N

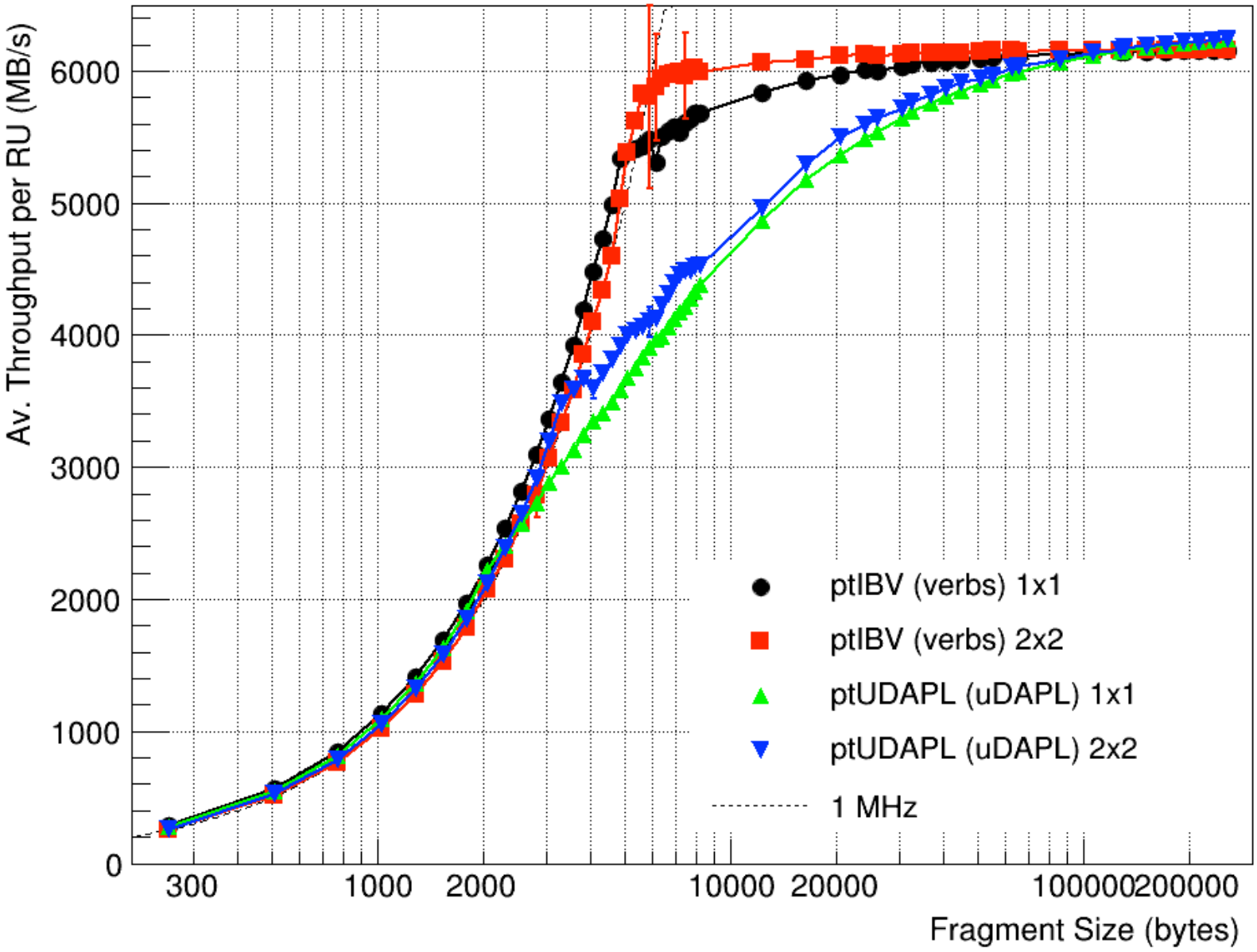

Av. Throughput per RU (MB/s)
0
1000
2000
3000
4000
5000
6000
300
1000
2000
10000
20000
100000
200000
Fragment Size (bytes)
ptIBV (verbs) 1x1
ptIBV (verbs) 2x2
ptUDAPL (uDAPL) 1x1
ptUDAPL (uDAPL) 2x2
1 MHz

Event Building Setup

- Event fragments are generated in the RU's
- Fully built event are dropped in the BU's
- The measurement is the rate of receiving in the BU's
- Additional control messages with Event Manager

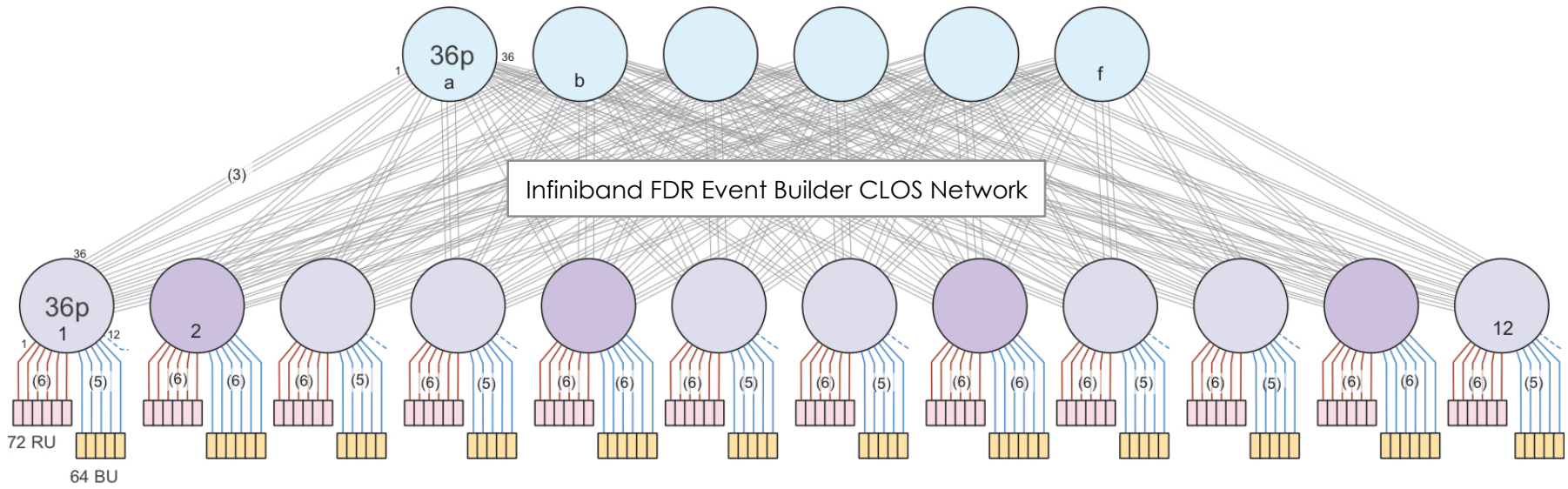


Event Building



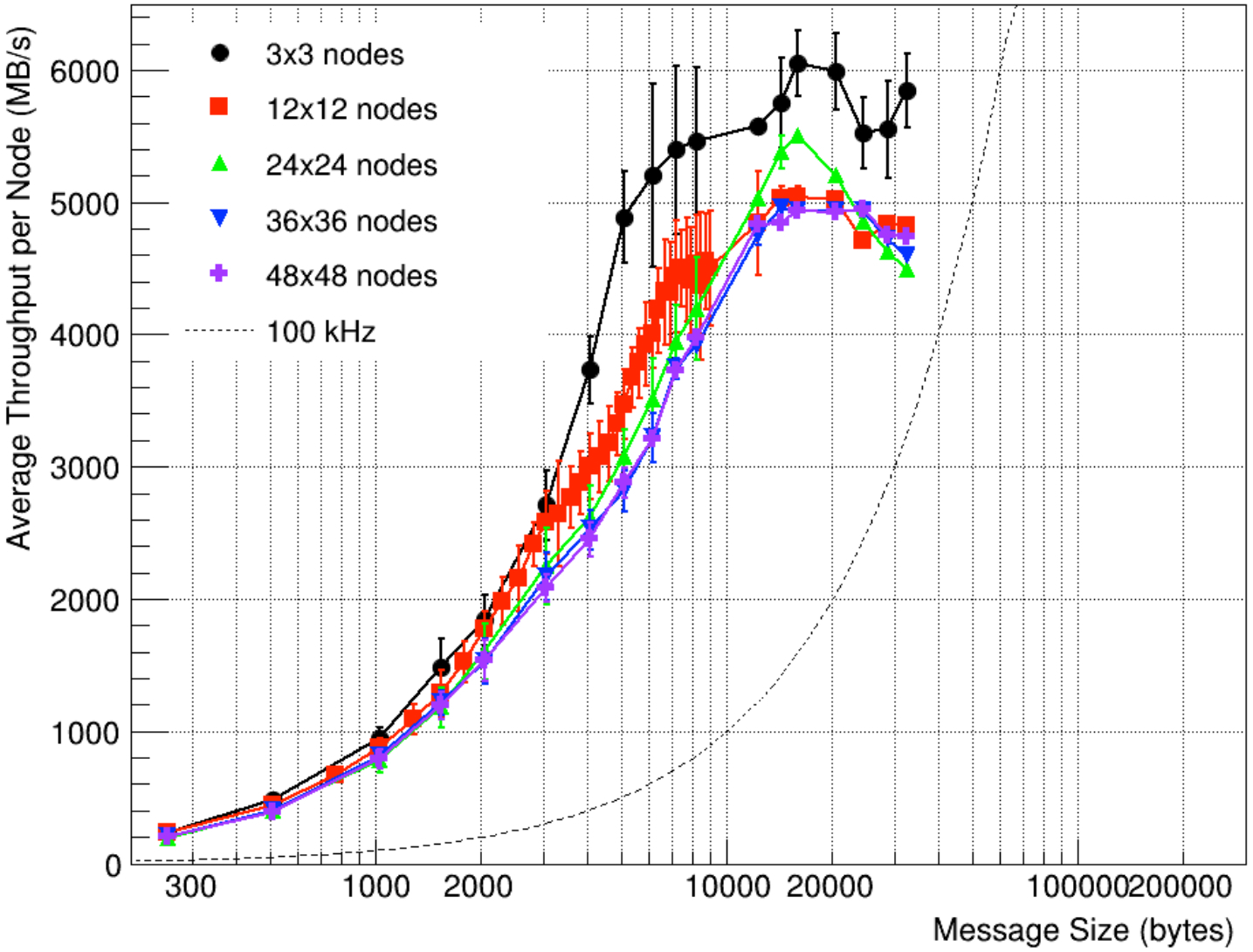
Larger Scale Tests

- ptIBVerbs used for larger scale tests
  - up to 48x48 using Infiniband CLOS network
- Preliminary N-to-N tests

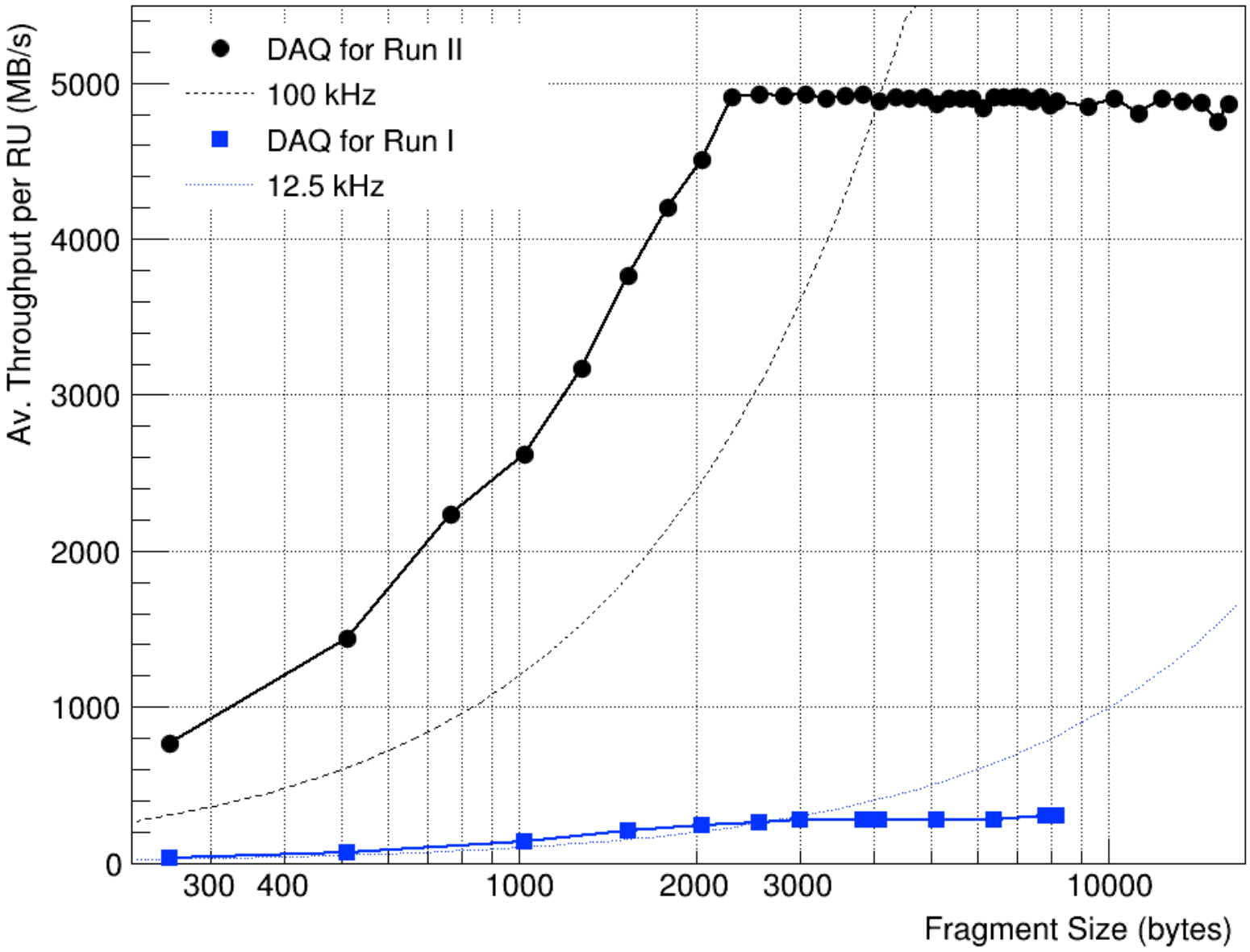


ptIBVerbs Performance – N-to-N

Message Size v Throughput



DAQ1 v DAQ2 RU Performance



Conclusion

Conclusions

- Infiniband works well with event building applications
- and the CMS Online Software framework (XDAQ)
- CMS DAQ will be using ptIBV for data flow in LHC run 2
- Performance compared to DAQ 1 allows for an order of magnitude of reduction in physical resources for event building

- In the future…
  - Full DAQ2 tests

Questions ?

- Thank you for listening
- Are there any questions?

Backup Slides

Links and References

- CMS Online Software Support
  - Trac https://svnweb.cern.ch/trac/cmsos
  - Mailing list (e-group) cms-os-users@cern.ch
  - User's Support Guide https://edms.cern.ch/file/1001791/
  - Documentation available on EDMS
    - https://edms.cern.ch/nav/CMS-0000000001/CMS-0000009924

- Links to additional documentation and tools
  - Book *"Version Control with Subversion"* http://svnbook.red-bean.com/
  - Subversion http://subversion.tigris.org/
  - Trac http://trac.edgewall.org/