Boosting Event Building Performance Using Infiniband FDR for CMS Upgrade

Andrew Forrest – CERN (PH/CMD)

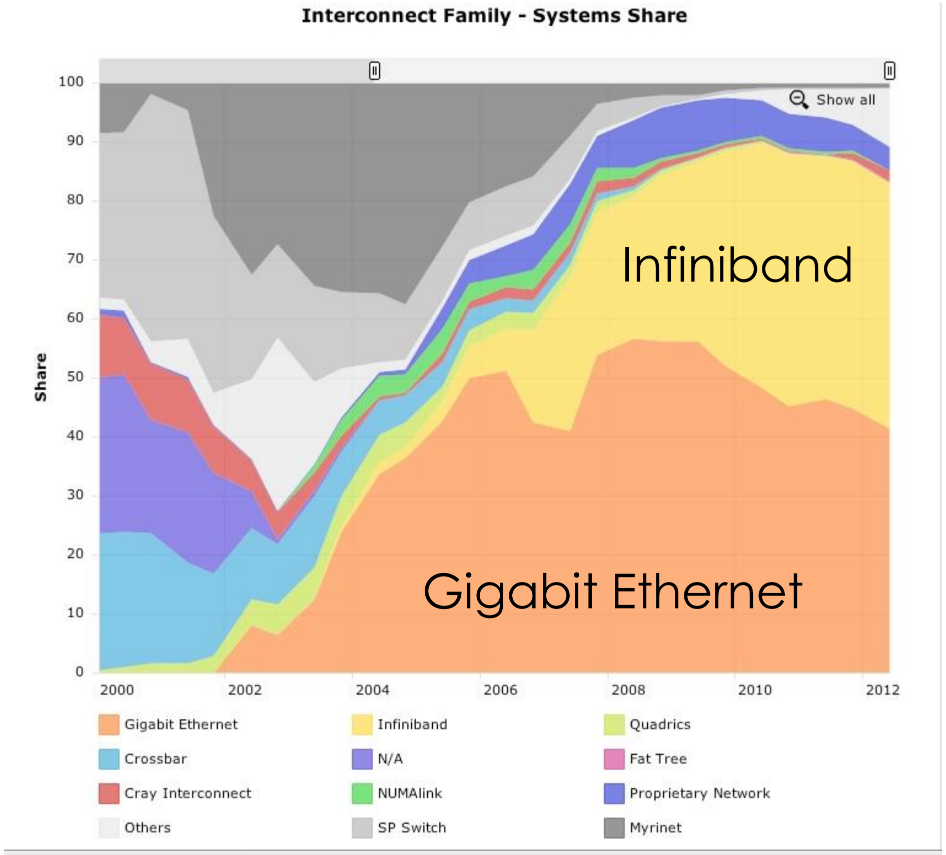
Technology and Instrumentation in Particle Physics Conference
2 June 2014, Amsterdam

Outline

- Motivations
- A Quick Overview of Infiniband
- CMS Data Acquisition
  - Event Building
- CMS Online Software Framework (XDAQ)
- Integrating Infiniband
- Preliminary Results
- Conclusions

Motivations

- DAQ hardware (PC's) from run 1 at end-of-life (>5 years old)
- Cost-effective solution that meets the requirements for run 2
- Opportunity to take advantage of technological advances



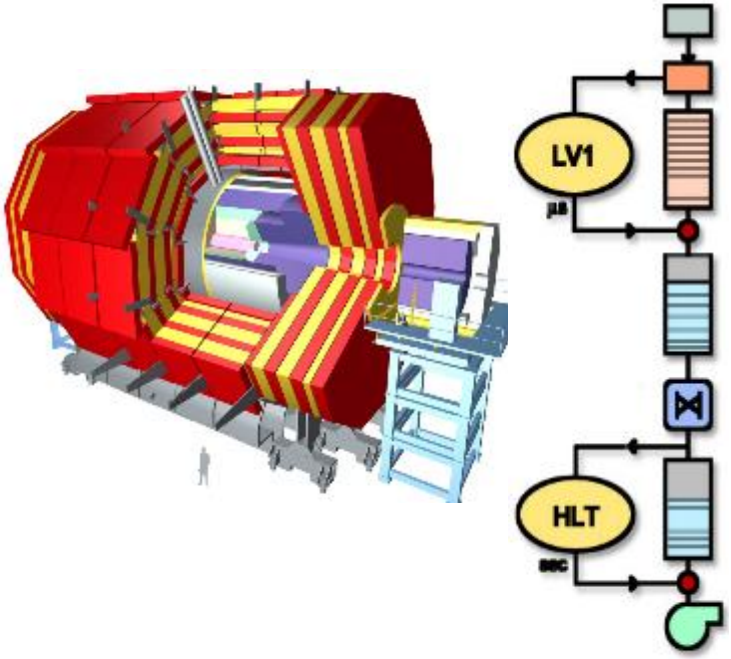
top500.org

A Quick Overview of Infiniband

- Infiniband is…
  - A switched fabric computer network communications link for high performance
  - Reliable communication
  - Supports message based transfer using send/receive semantic
- Multiple programming methods
  - Infiniband verbs
  - uDAPL (user-level Direct Access Programming Library)
  - IPoIB
- Software available as part of the OFED (OpenFabrics Enterprise Distribution)

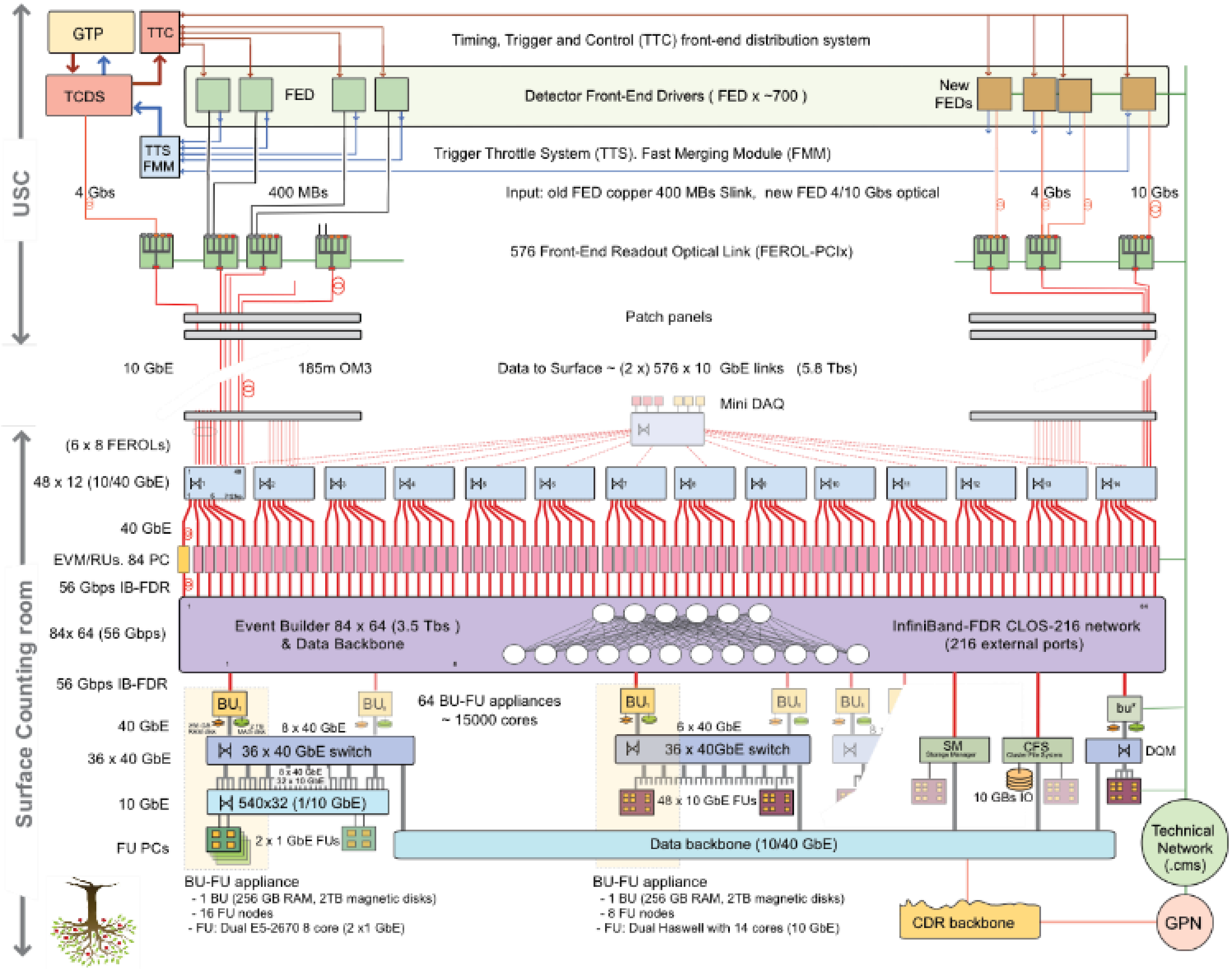
CMS Data Acquisition

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

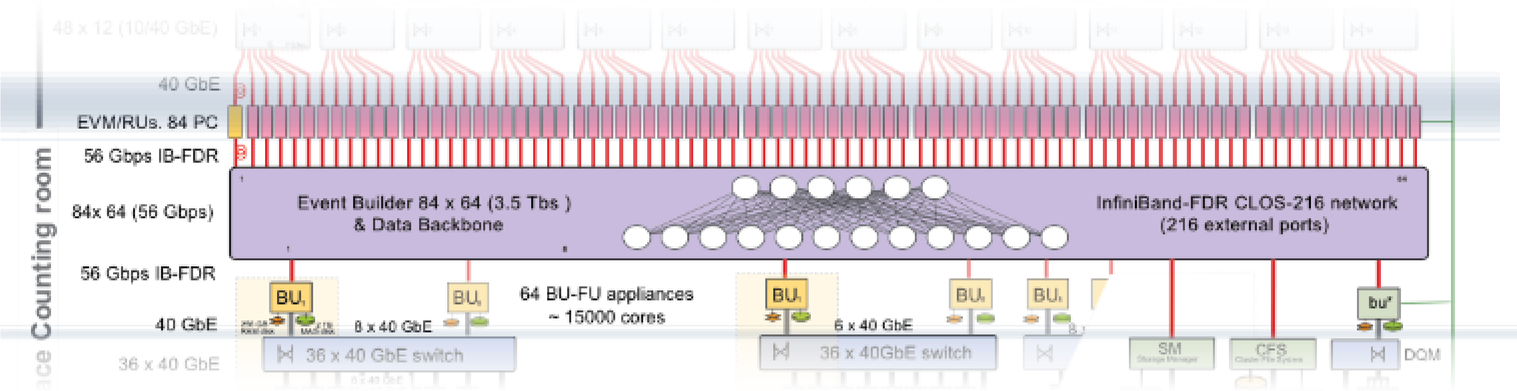
CMS DAQ2 Layout



BU-FU appliance
- 1 BU (256 GB RAM, 2TB magnetic disks)
- 16 FU nodes
- FU: Dual E5-2670 8 core (2 x1 GbE)

BU-FU appliance
- 1 BU (256 GB RAM, 2TB magnetic disks)
- 8 FU nodes
- FU: Dual Haswell with 14 cores (10 GbE)

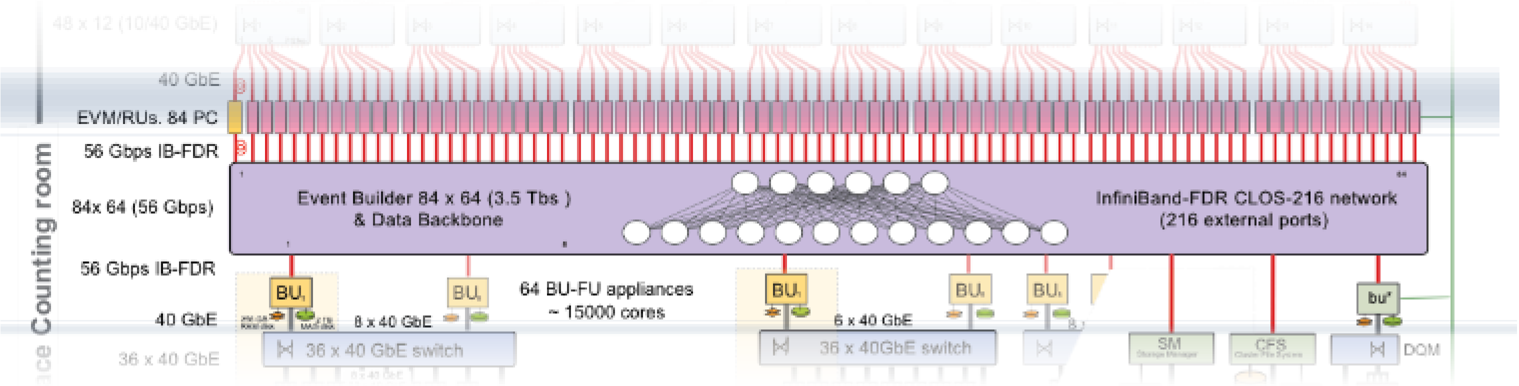
CMS Event Builder Network



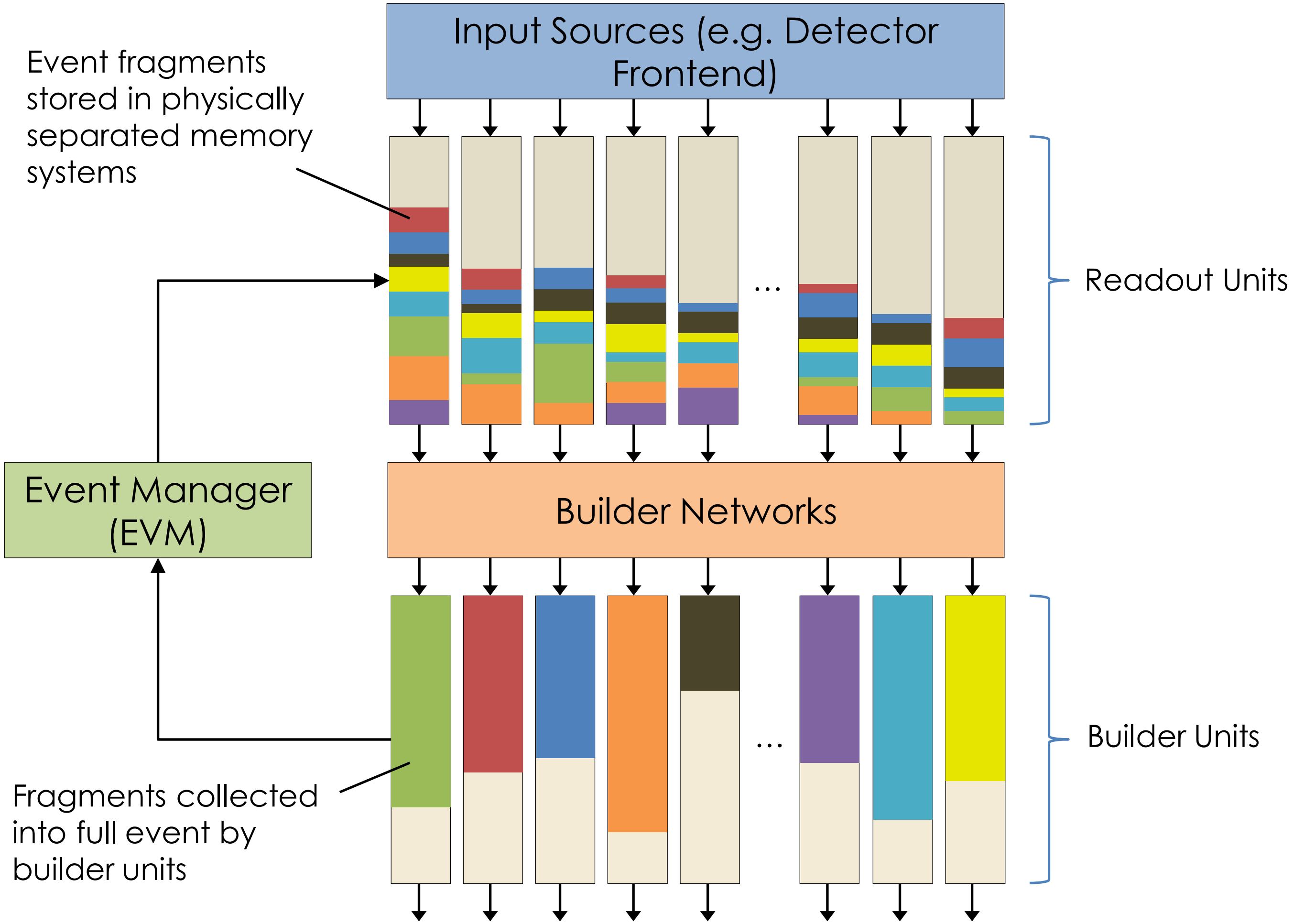
CMS Event Builder Network

Resulting Required Physical Resources

Resources Type	Run 1	Run 2
RU	~640	84
BU	~1260	64
FU		~1260
Throughput Requirement	100 Gb/s	200 Gb/s



CMS Event Builder Layout



Event fragments stored in physically separated memory systems

Fragments collected into full event by builder units

CMS Online Software Framework (XDAQ)

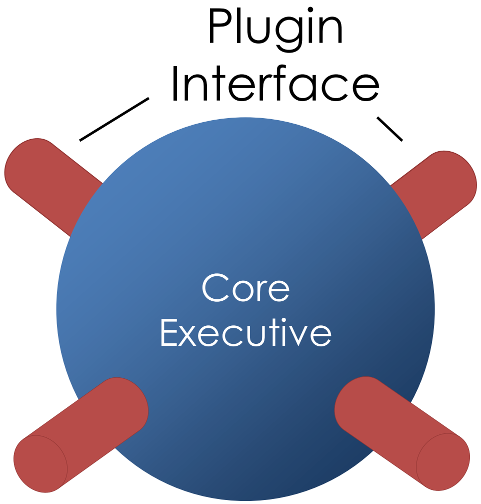
XDAQ Framework

- The XDAQ is software platform created specifically for the development of distributed data acquisition systems
- Implemented in C++, developed by the CMS DAQ group
- Provides platform independent services, tools for inter-process communication, configuration and control
- Builds upon industrial standards, open protocols and libraries, and is designed according to the object-oriented model

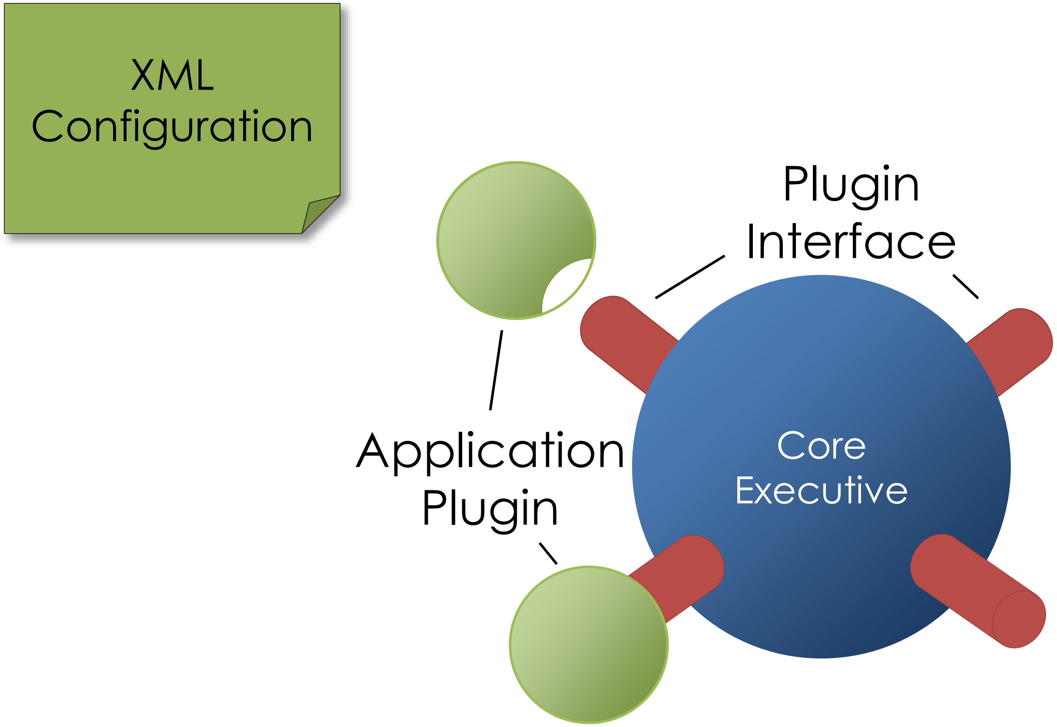
For further information about XDAQ see:

J. Gutleber, S. Murray and L. Orsini, Towards a homogeneous architecture for high-energy physics data acquisition systems published in Computer Physics Communications, vol. 153, issue 2, pp. 155-163, 2003
http://www.sciencedirect.com/science/article/pii/S0010465503001619

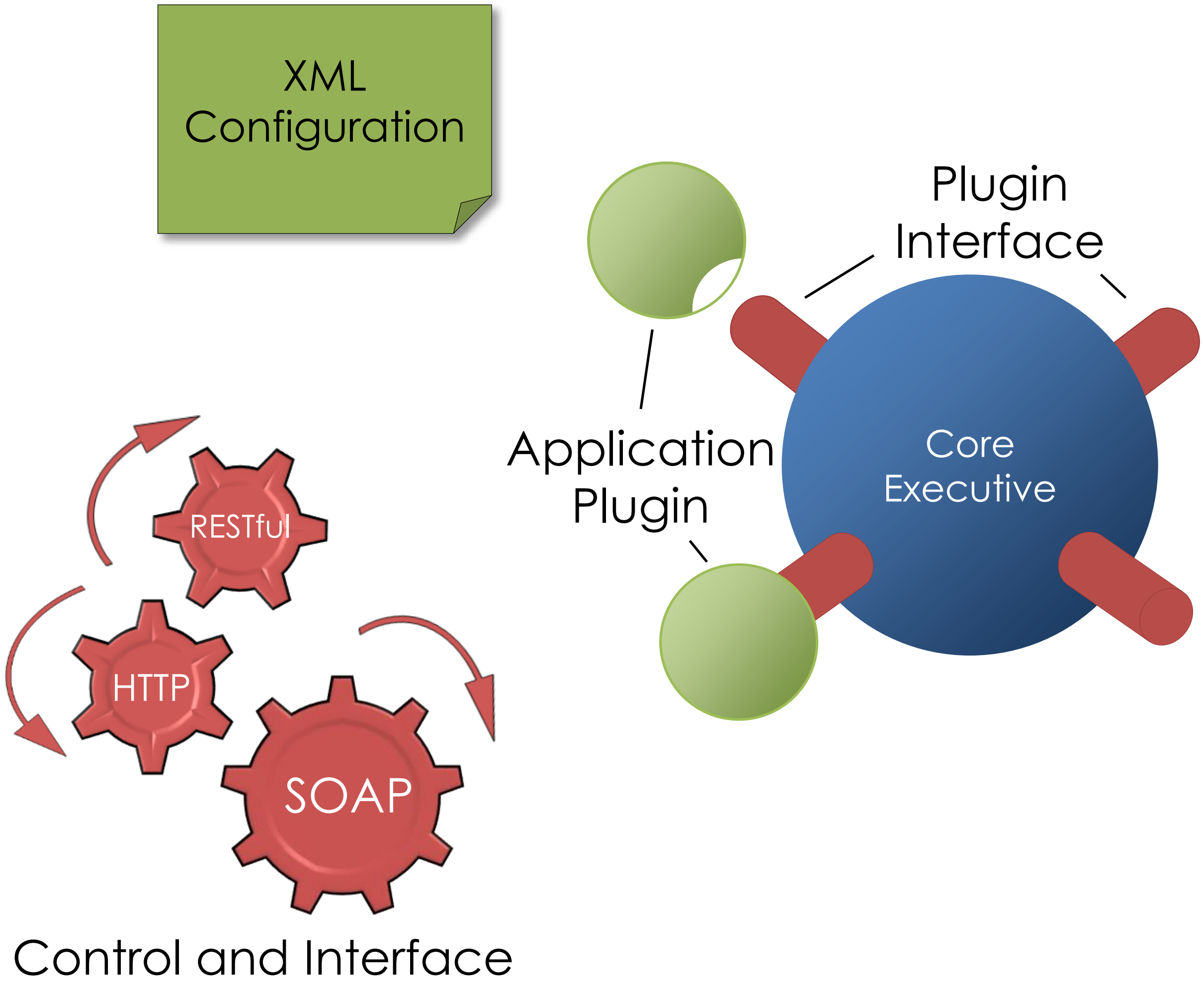
XDAQ Architecture Foundation



XDAQ Architecture Foundation

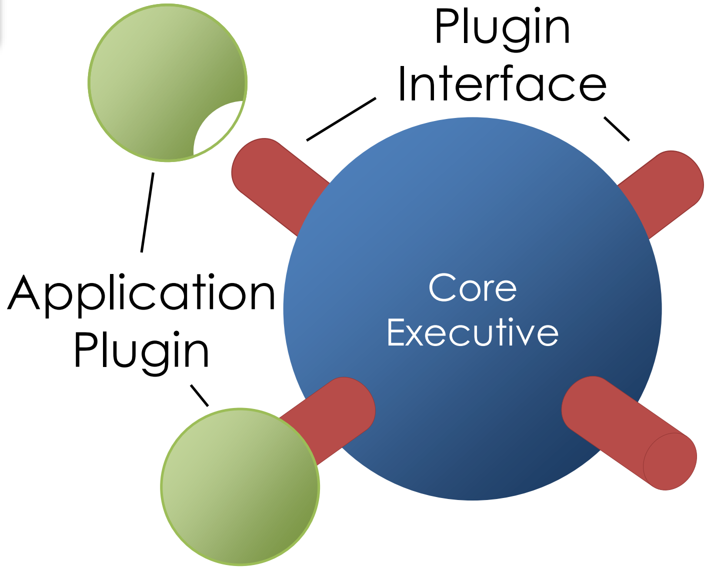


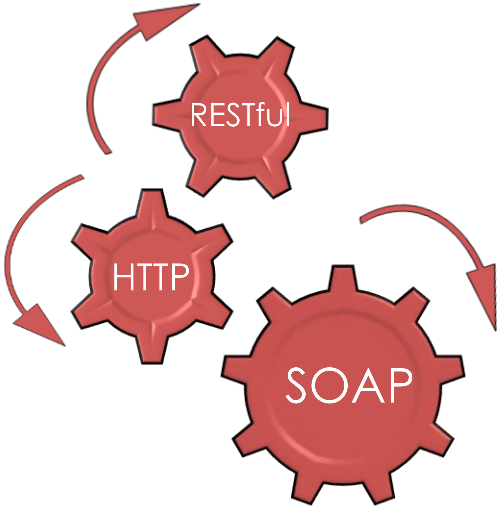
XDAQ Architecture Foundation



XDAQ Architecture Foundation

XML Configuration





Control and Interface

Hardware Access

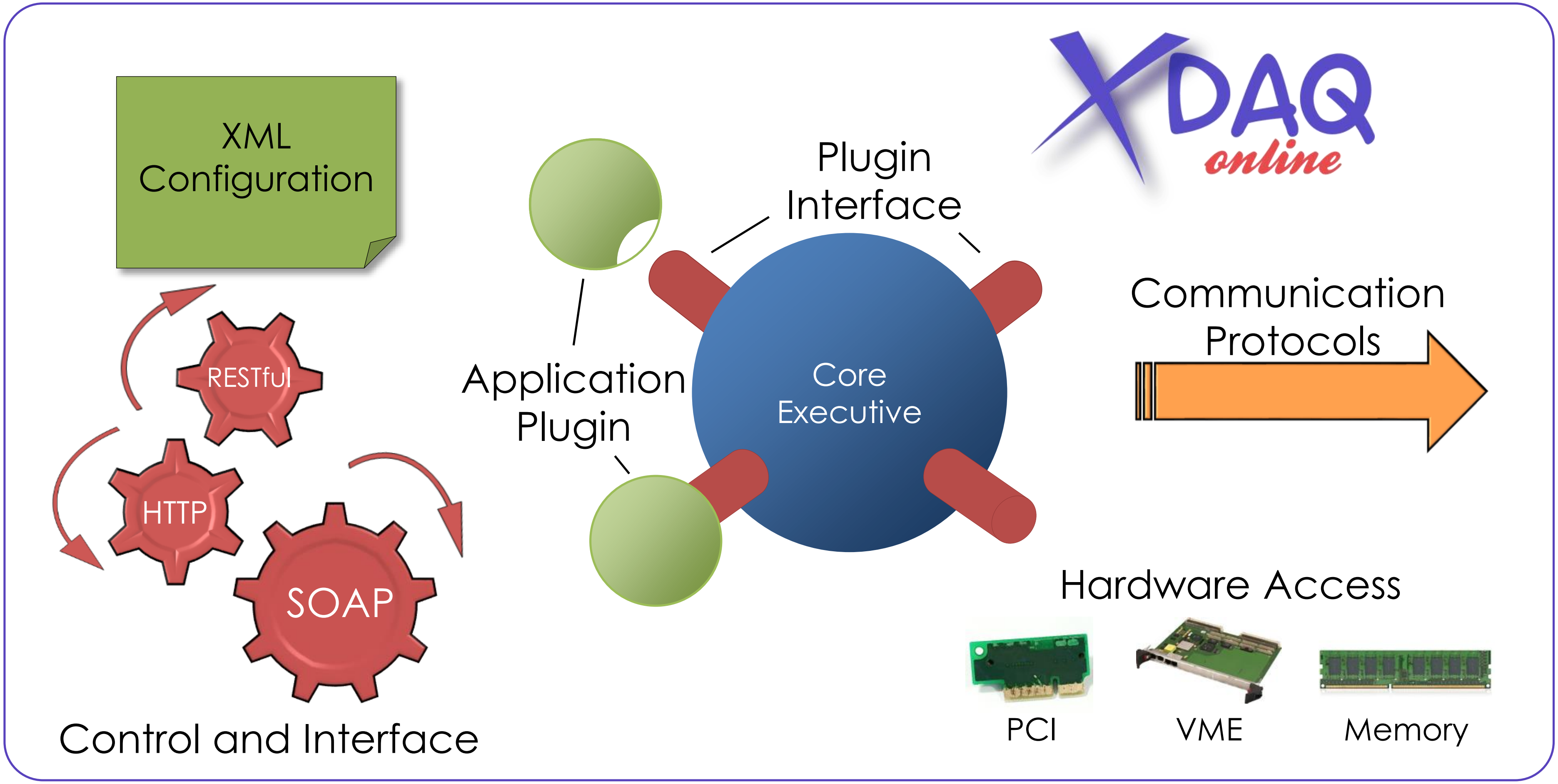
PCI

VME

Memory

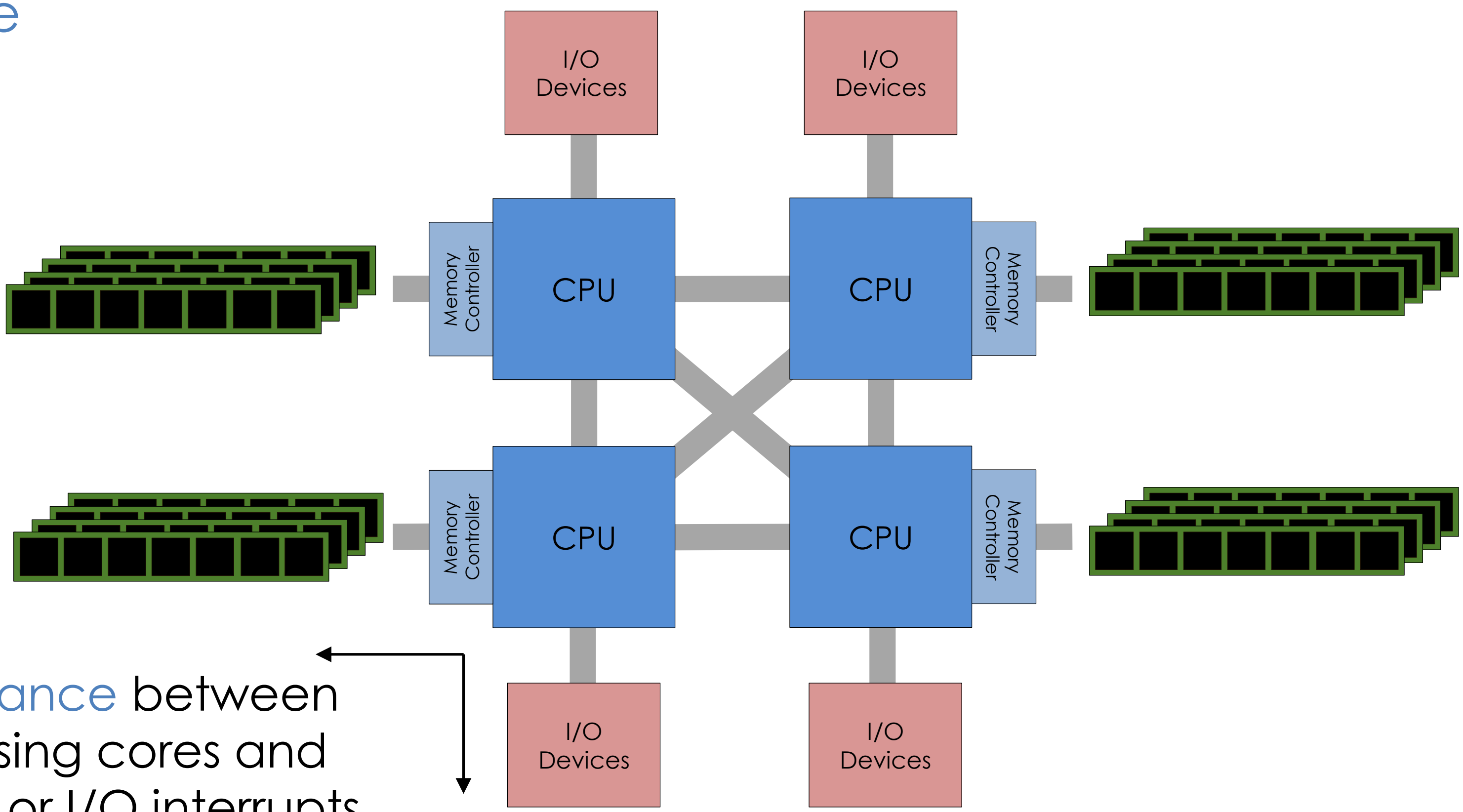
XDAQ Architecture Foundation

- Uniform building blocks - One or more executives per computer contain application and service components



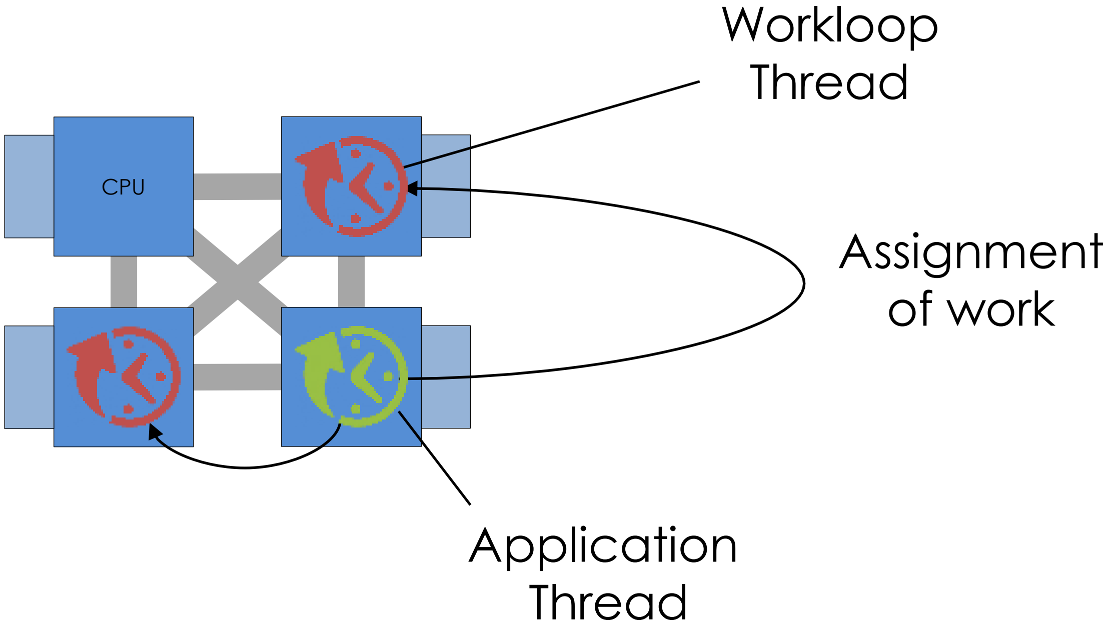
Hardware Architecture - NUMA

- Non Uniform Memory Access (NUMA)
- XDAQ provides utilities to take advantage



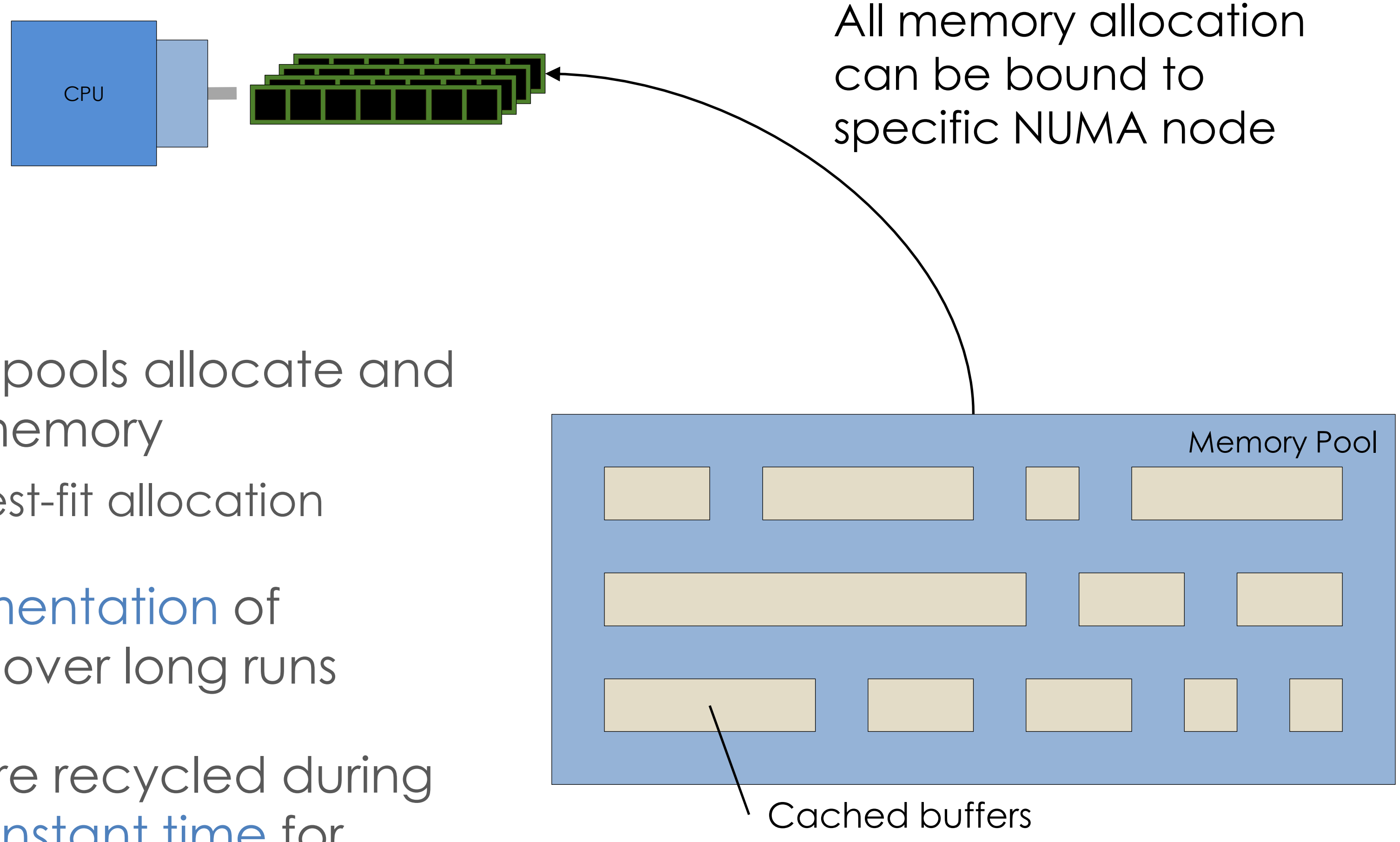
The distance between processing cores and memory or I/O interrupts varies for each core

Multithreading



- Workloops can be bound to run on specific CPU cores by configuration
- Work assigned by application
- Workloops provide easy use of threads

Memory – Memory Pools



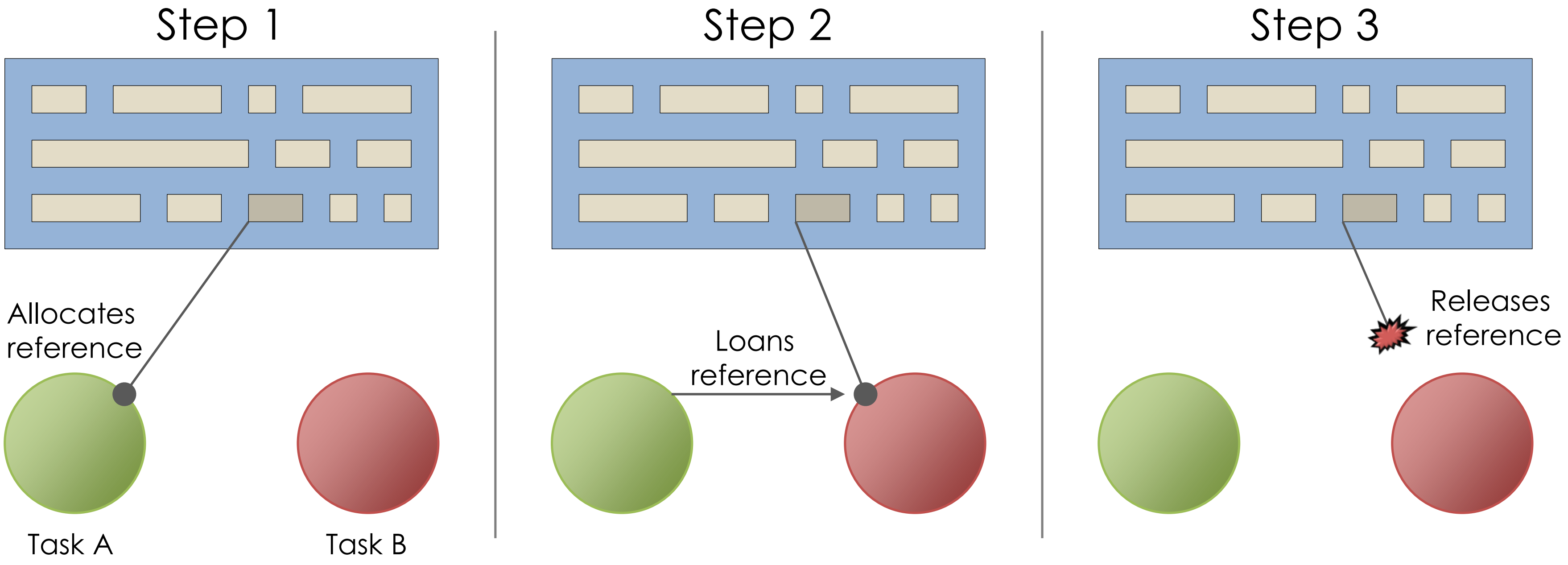
All memory allocation can be bound to specific NUMA node

Cached buffers

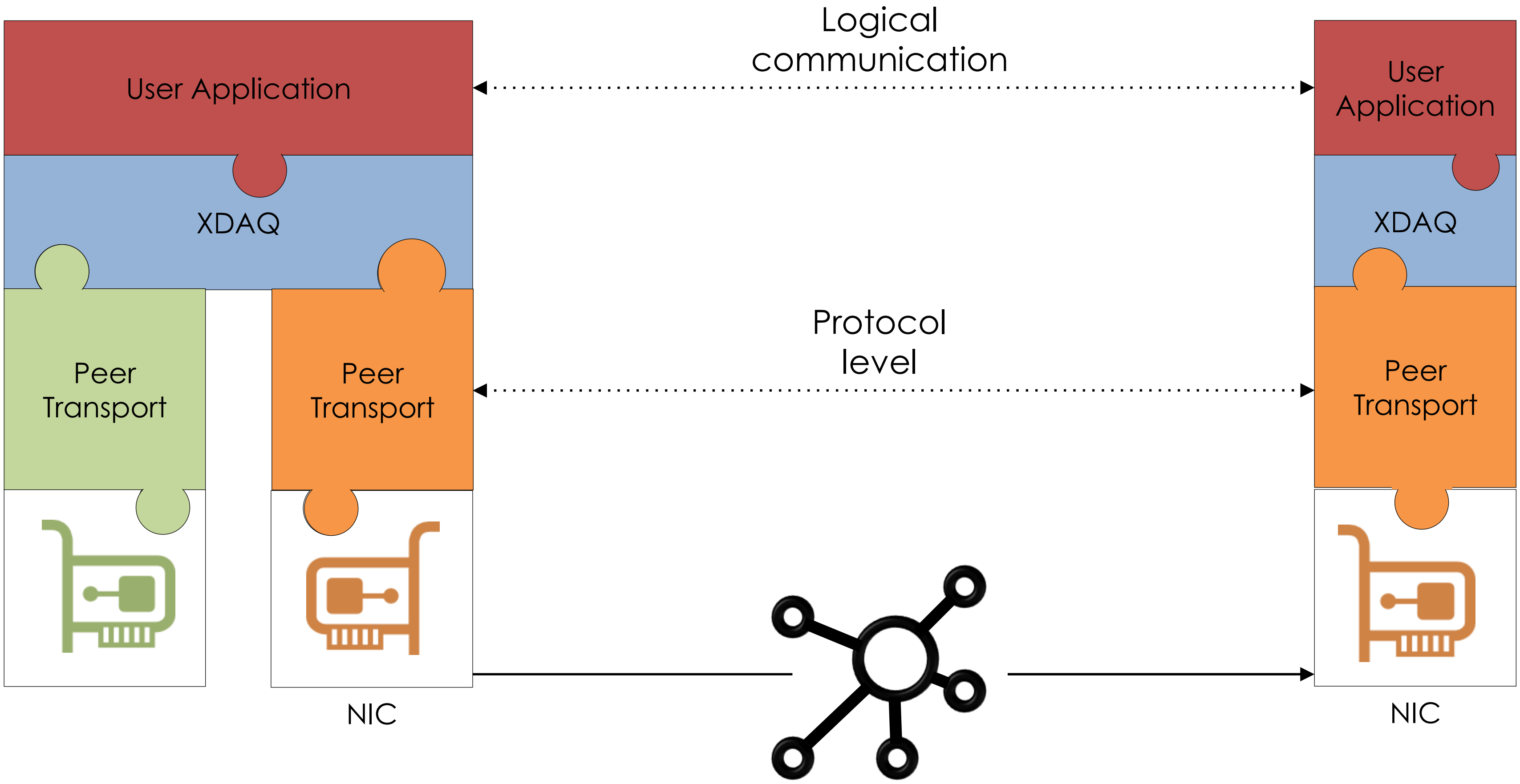
- Memory pools allocate and cache memory
  - $\log_2$ best-fit allocation
- No fragmentation of memory over long runs
- Buffers are recycled during runs – constant time for retrieval

Memory – Buffer Loaning

- Buffer loaning allows zero-copy of data between software layers and processes



Data Transmission – Peer Transports

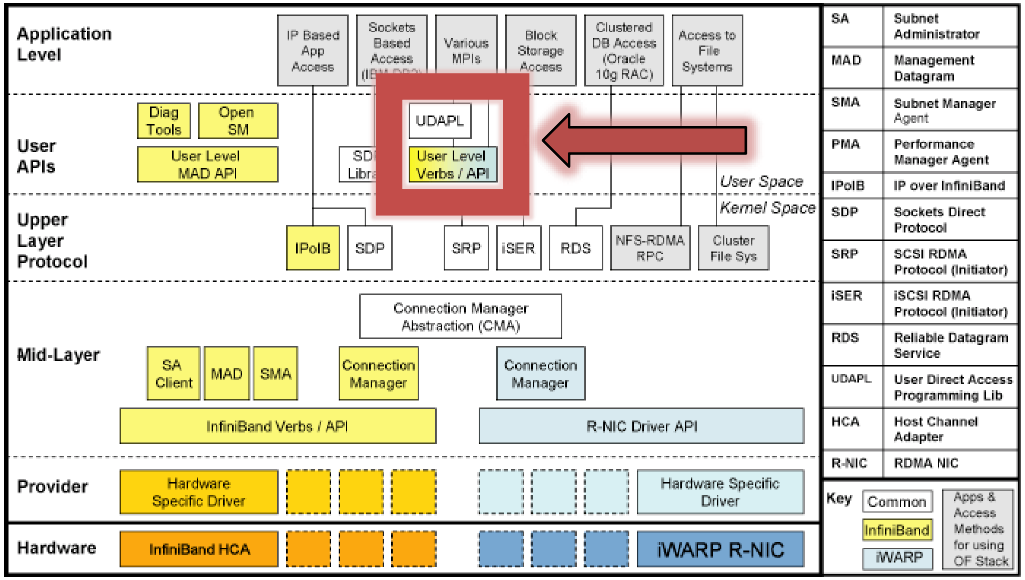


- User application is network and protocol independent
- Routing defined by XDAQ configuration
- Connections setup through Peer to Peer model

Integrating Infiniband

Integration Aims

- Evaluation of two programming libraries
  - uDAPL
  - verbs
- Both from OpenFabric distribution
- uDAPL has connection support
- verbs is lower level in the software stack



Application Level

IP Based App Access

Sockets Based Access (IBM DB2)

Various MPIs

Block Storage Access

Clustered DB Access (Oracle 10g RAC)

Access to File Systems

User APIs

Diag Tools

Open SM

User Level MAD API

SDP Library

UDAPL

User Level Verbs / API

User Space

Kernel Space

Upper Layer Protocol

IPoIB

SDP

SRP

iSER

RDS

NFS-RDMA RPC

Cluster File Sys

Mid-Layer

Connection Manager Abstraction (CMA)

SA Client

MAD

SMA

Connection Manager

Connection Manager

InfiniBand Verbs / API

R-NIC Driver API

Provider

Hardware Specific Driver

Hardware Specific Driver

Hardware

InfiniBand HCA

iWARP R-NIC

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

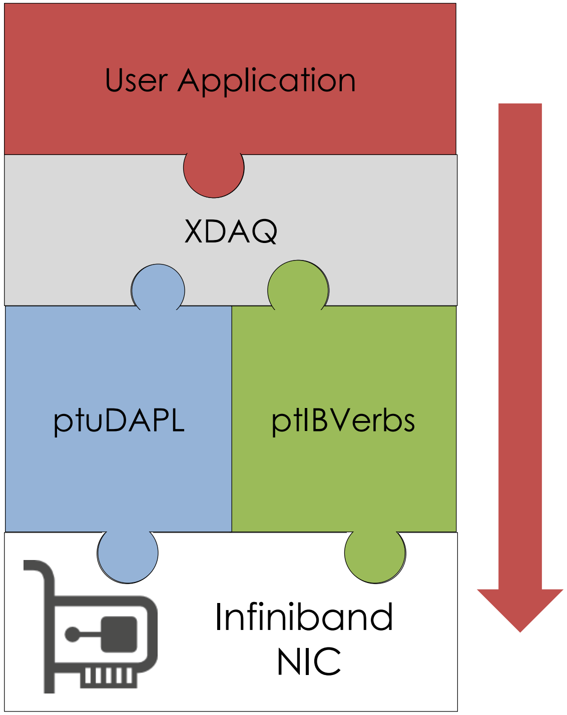
InfiniBand

iWARP

Apps & Access Methods for using OF Stack

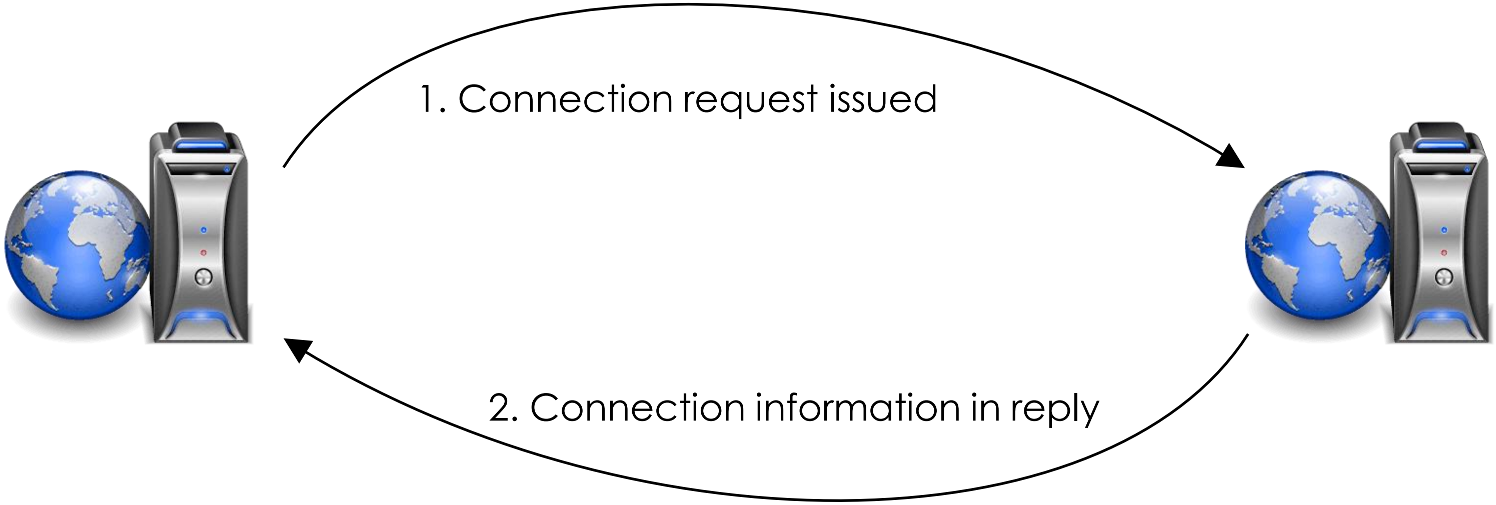
Infiniband Peer Transports

- 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



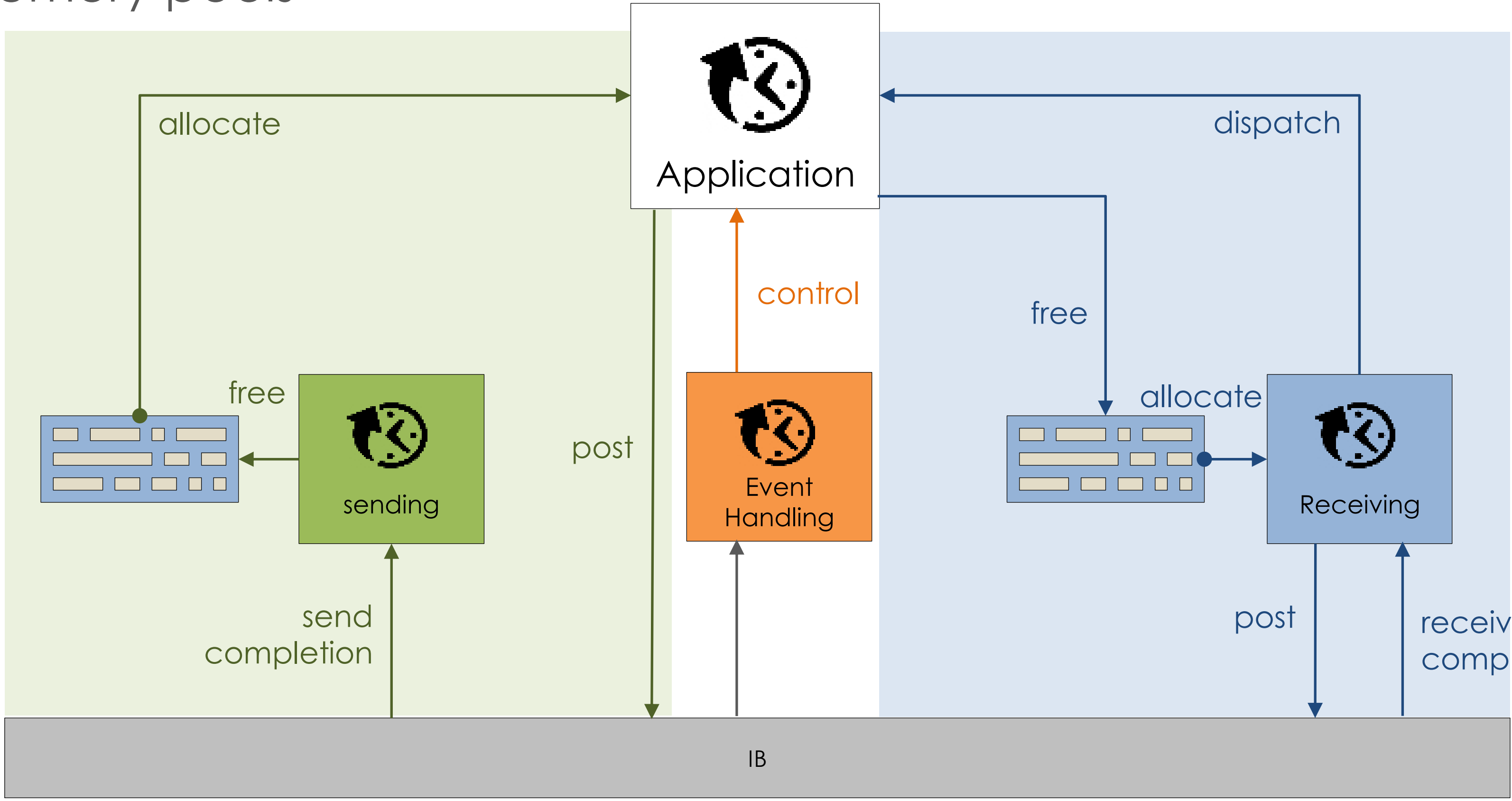
Connection Handling

- uDAPL provides IP address based connections
- verbs leaves the question of connecting peers open…
- For ptIBVerbs, a custom connection mechanism was implemented based upon IPoIB


1. Connection request issued
2. Connection information in reply

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



Preliminary Results

Test Setup

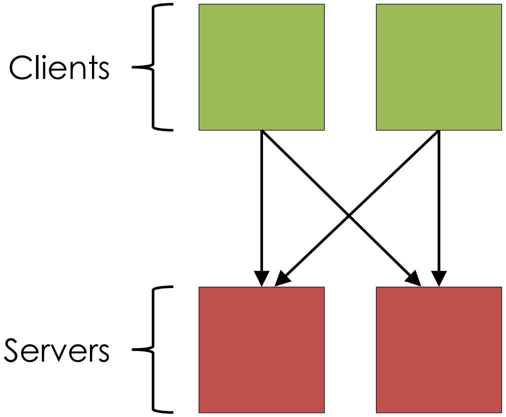
- Small scale with 4 nodes on 1 switch
  - 1x1 and 2x2 (RU x BU) tests for…
- N-to-N and event building tests
- Each node has…
  - Dual socket Intel Xeon E5-2670 8-core processors @ 2.6 GHz
  - 16 GB RAM per socket (NUMA)
  - Mellanox Connect X-3 VPI Infiniband FDR network card
  - OFED v 2.x
  - Scientific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

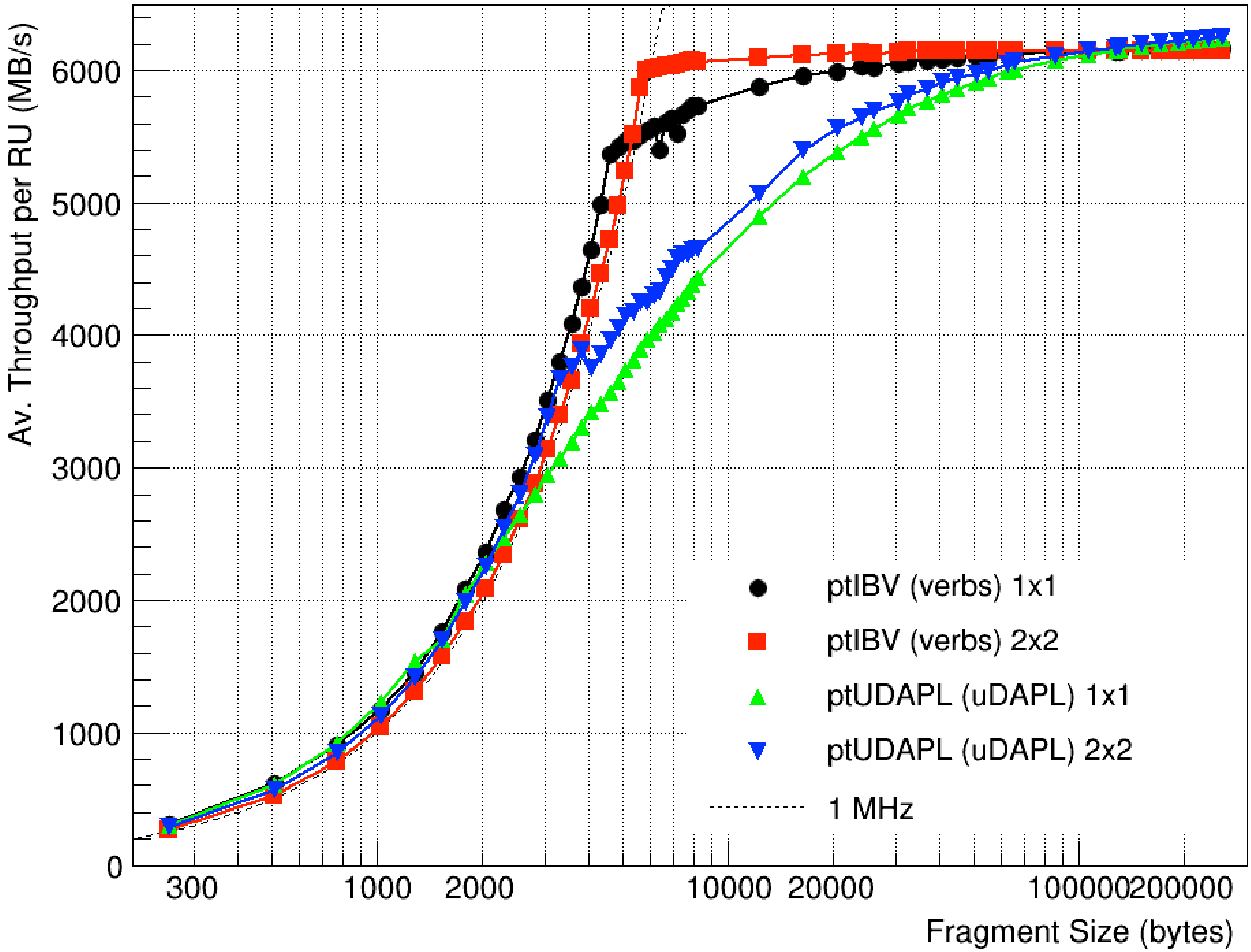
Effective unidirectional theoretical throughput in Gb/s

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

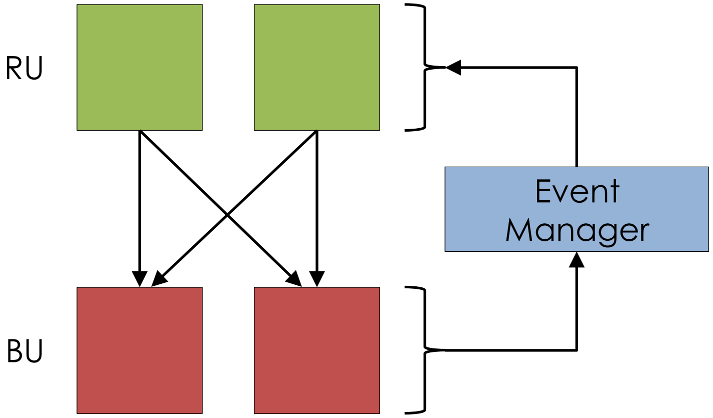


N-to-N

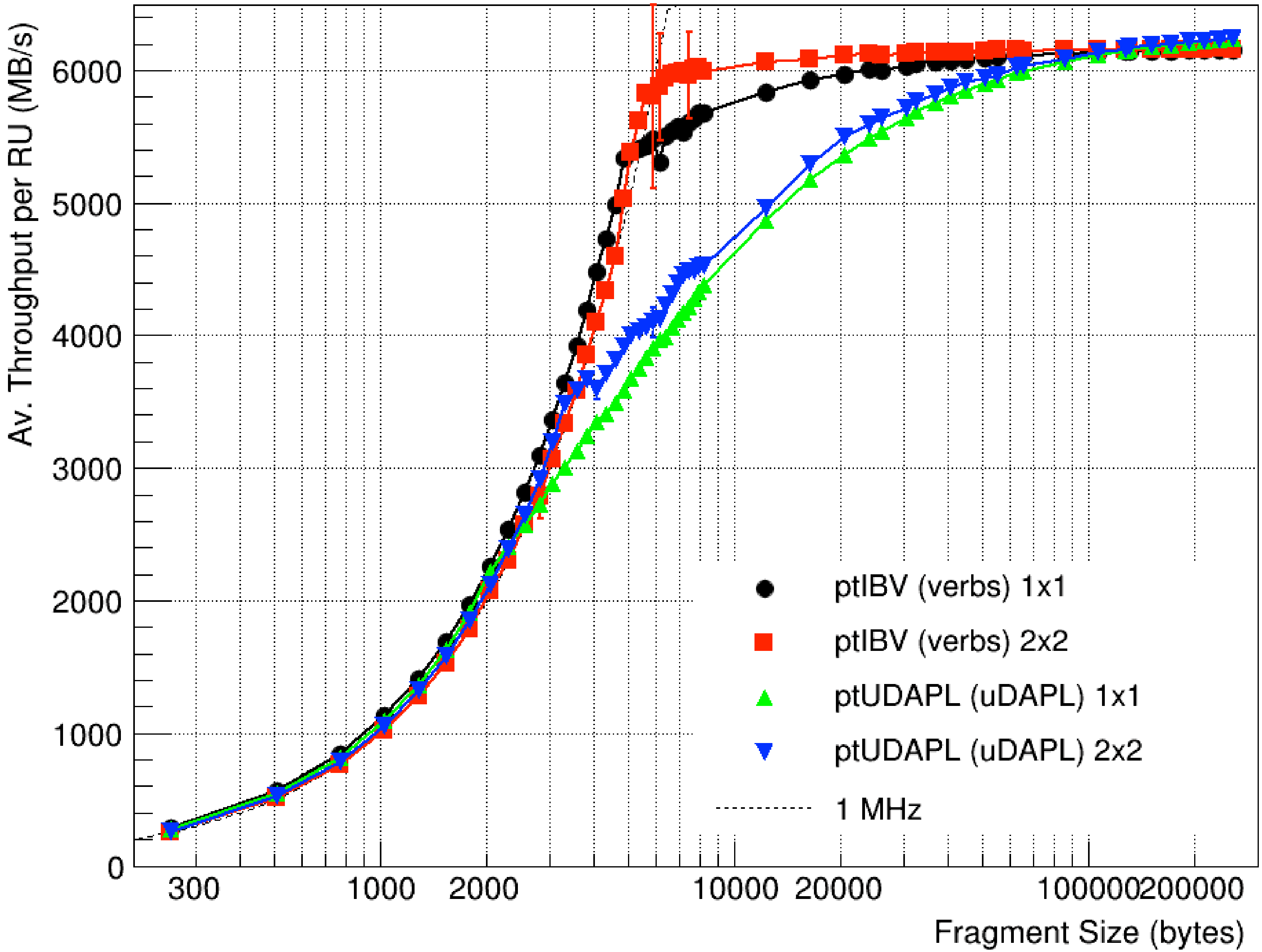

Av. Throughput per RU (MB/s)
0
1000
2000
3000
4000
5000
6000
300
1000
2000
10000
20000
100000
200000
Fragment Size (bytes)
● ptIBV (verbs) 1x1
■ ptIBV (verbs) 2x2
▲ ptUDAPL (uDAPL) 1x1
▼ ptUDAPL (uDAPL) 2x2
1 MHz

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

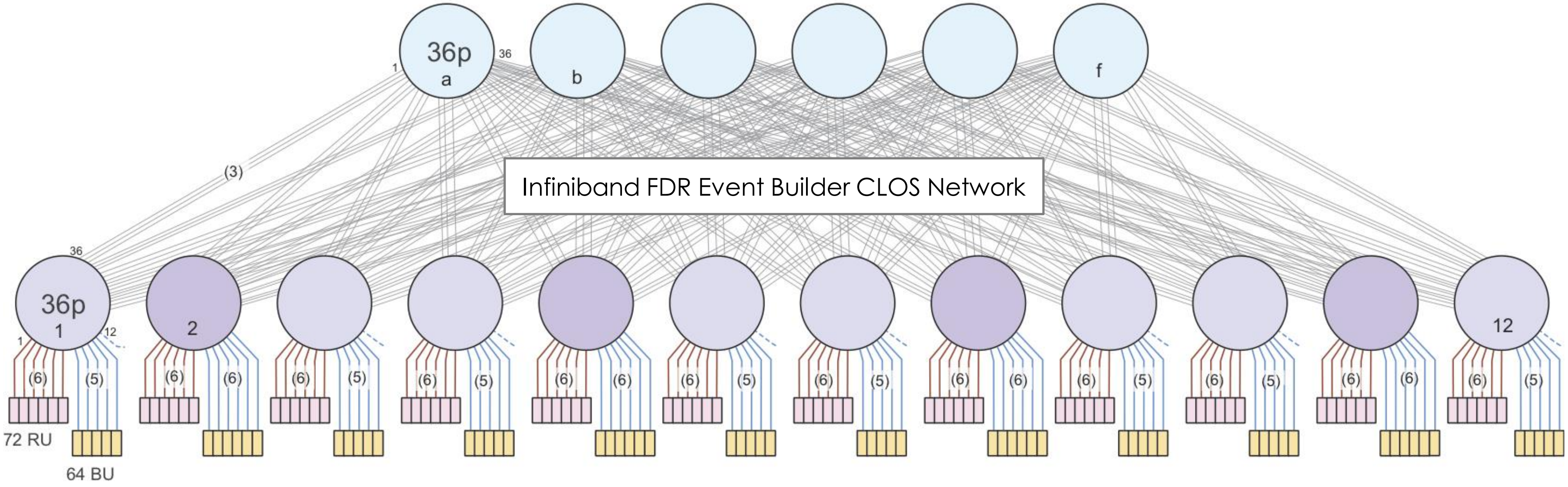


Event Building

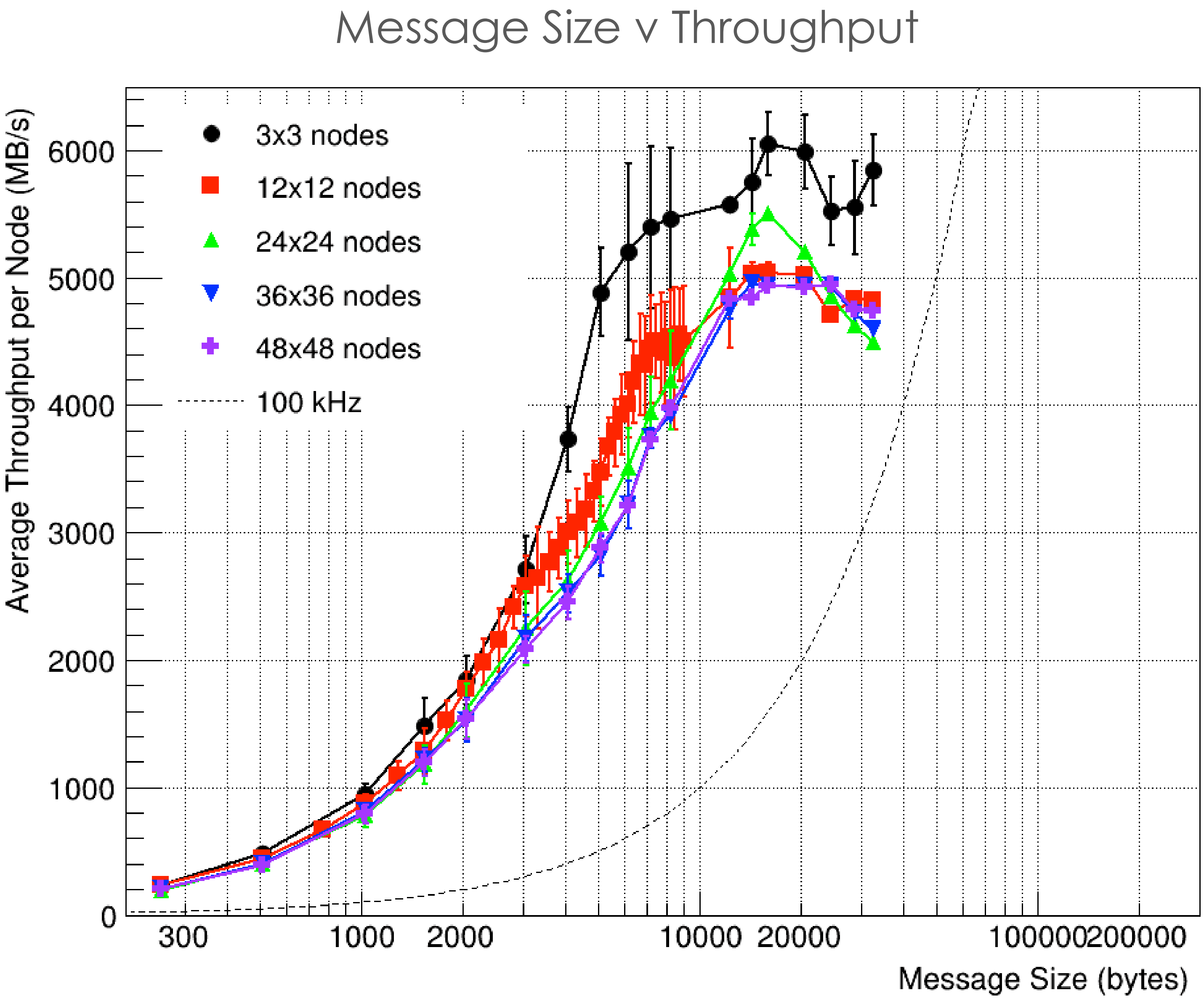


Larger Scale Tests

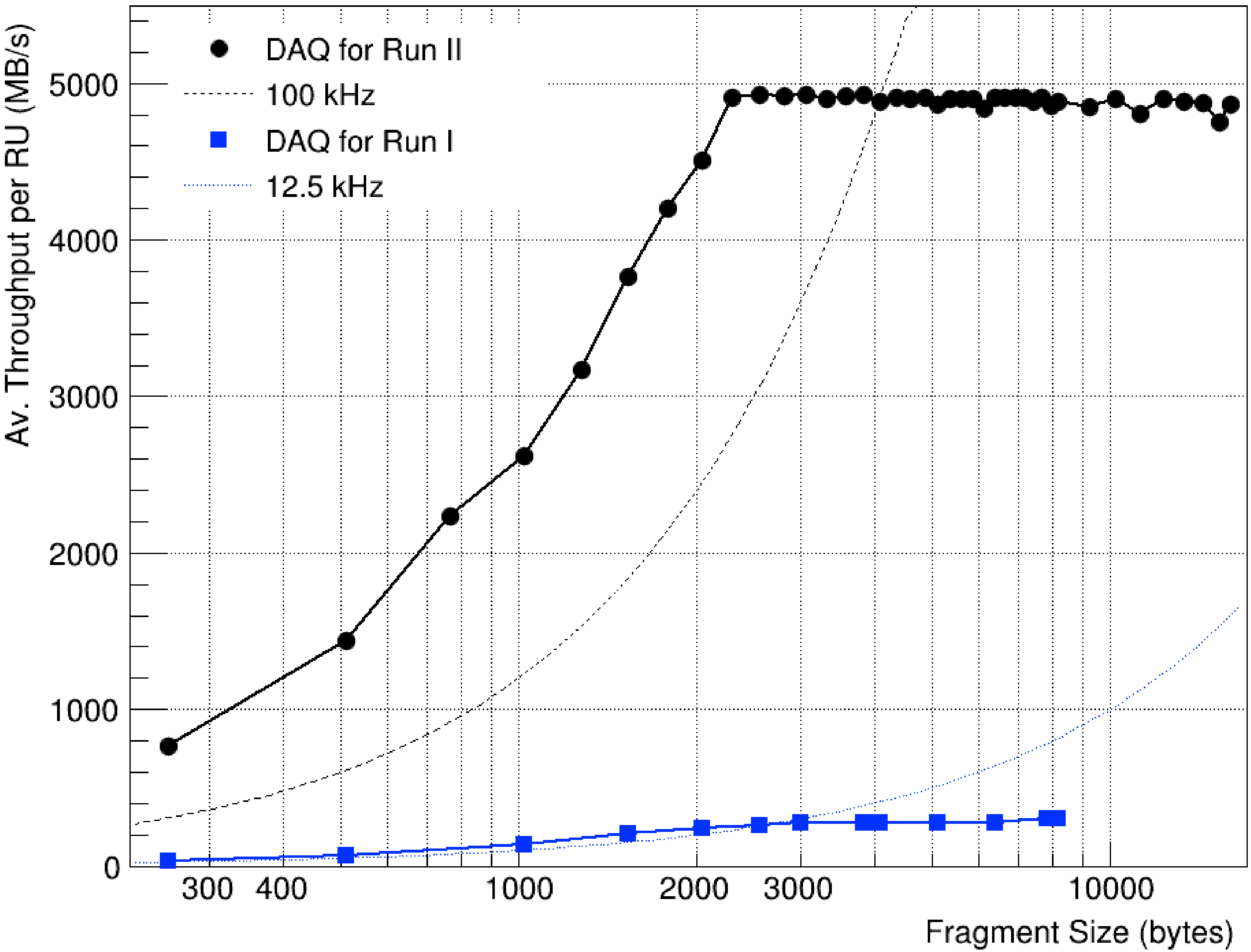
- ptIBVerbs used for larger scale tests
  - up to 48x48 using Infiniband CLOS network
- Preliminary N-to-N tests



ptIBVerbs Performance – N-to-N



DAQ1 v DAQ2 RU Performance



Conclusions

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

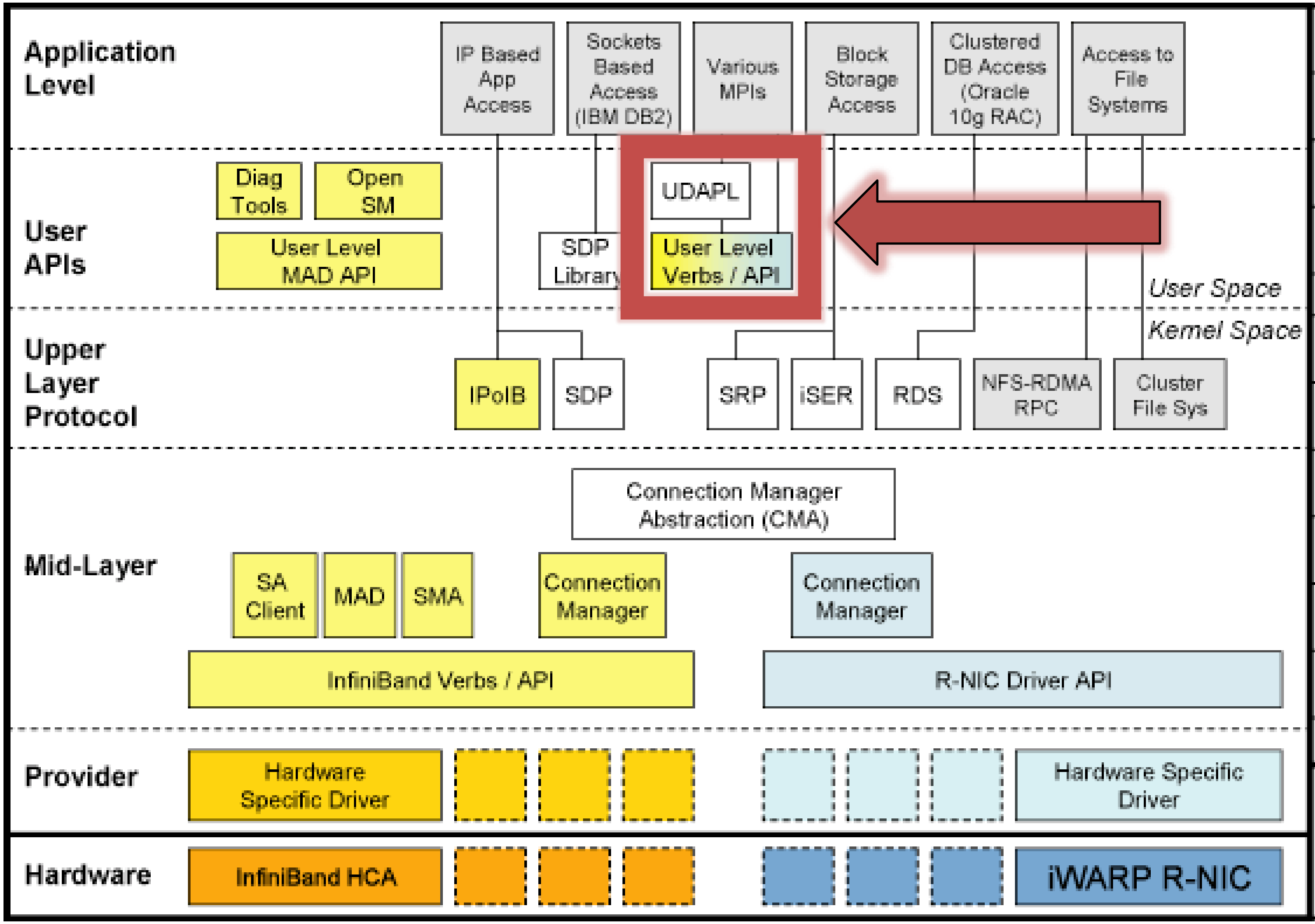
- In the future…
  - Full DAQ2 tests

Thank You For Listening

- Questions?

Additional Materials

OFED



Application Level

IP Based App Access

Sockets Based Access (IBM DB2)

Various MPIs

Block Storage Access

Clustered DB Access (Oracle 10g RAC)

Access to File Systems

User APIs

Diag Tools

Open SM

User Level MAD API

SDP Library

UDAPL

User Level Verbs / API

User Space

Kernel Space

Upper Layer Protocol

IPoIB

SDP

SRP

iSER

RDS

NFS-RDMA RPC

Cluster File Sys

Mid-Layer

Connection Manager Abstraction (CMA)

SA Client

MAD

SMA

Connection Manager

Connection Manager

InfiniBand Verbs / API

R-NIC Driver API

Provider

Hardware Specific Driver

Hardware Specific Driver

Hardware

InfiniBand HCA

iWARP R-NIC

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