Application Acceleration Beyond Flash Storage

Session 303C – Mellanox Technologies

Flash Memory Summit | July 2014

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™

Accelerating Applications, Step-by-Step

First Steps

- Make compute fast – Moore’s Law
- Make storage fast – Flash

Don’t stop there! What next?

- Make network fast –16/40/56 Gb/s
- Reduce latency – RDMA
- Make RDMA more accessible
- Make applications more efficient
- Simplify communications programming

Make the Network Fast—Bandwidth

Server

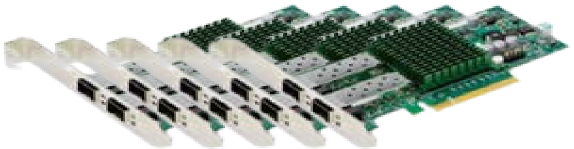
+

24 x 2.5” SAS 12Gb SSDs

= **24GB/s** =

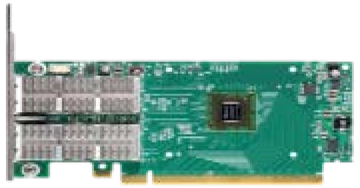
15 x 16Gb/s Fibre Channel Ports

OR



20 x 10Gb/s iSCSI Ports (with offload)

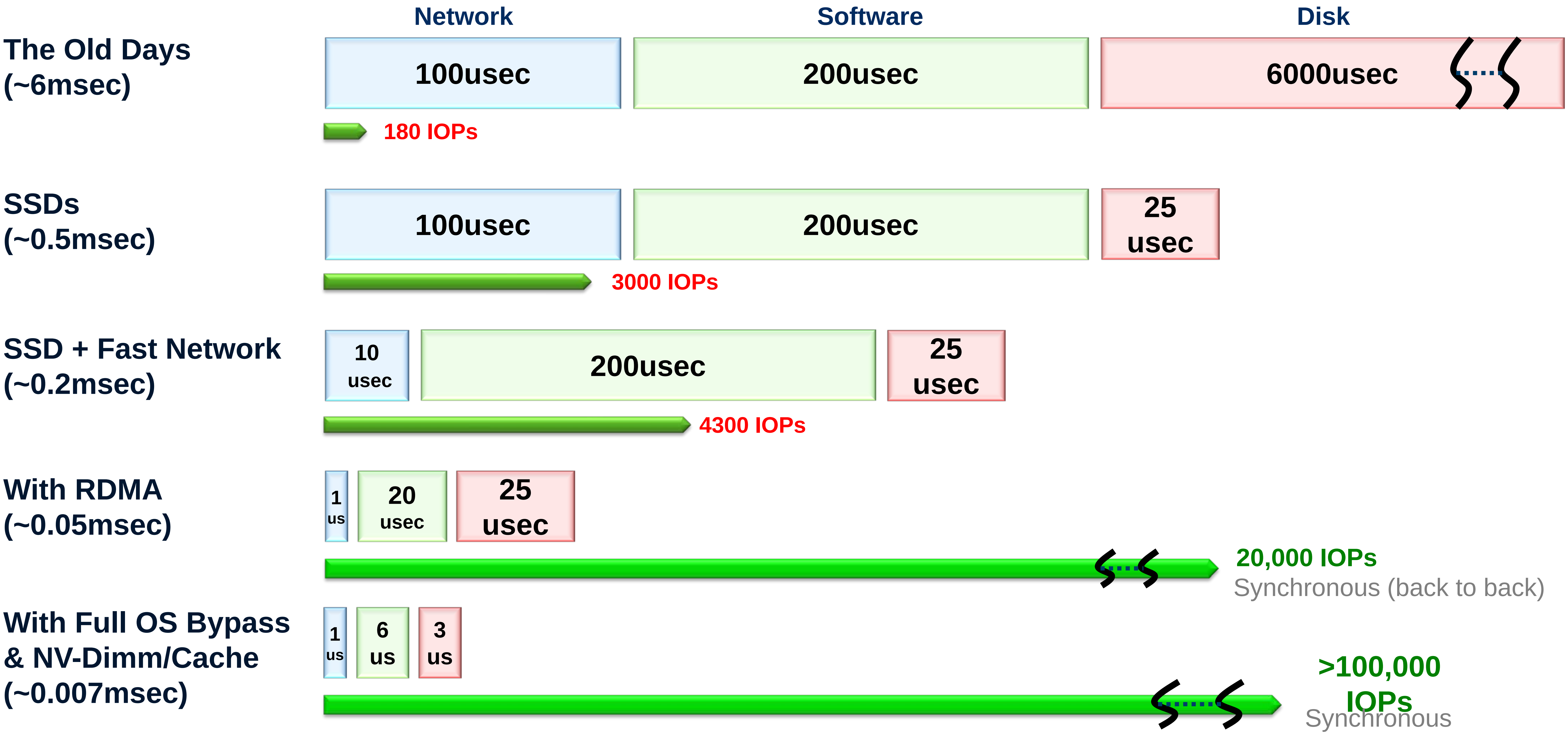
OR



4 x 40-56Gb/s IB/Eth port (with RDMA)

SSD & Flash Mandate High-Speed Interconnect

Make the Network Fast – Latency



Network

Software

Disk

The Old Days (~6msec)

100usec

200usec

6000usec

180 IOPs

SSDs (~0.5msec)

100usec

200usec

25 usec

3000 IOPs

SSD + Fast Network (~0.2msec)

10 usec

200usec

25 usec

4300 IOPs

With RDMA (~0.05msec)

1 us

20 usec

25 usec

20,000 IOPs

Synchronous (back to back)

With Full OS Bypass & NV-Dimm/Cache (~0.007msec)

1 us

6 us

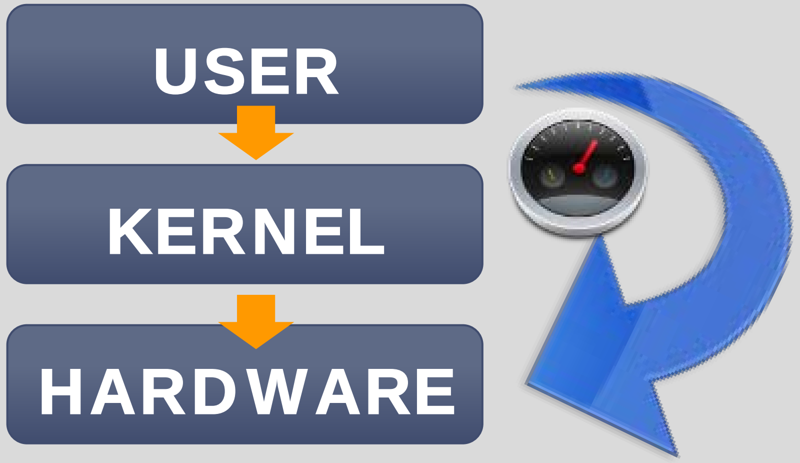
3 us

>100,000 IOPs

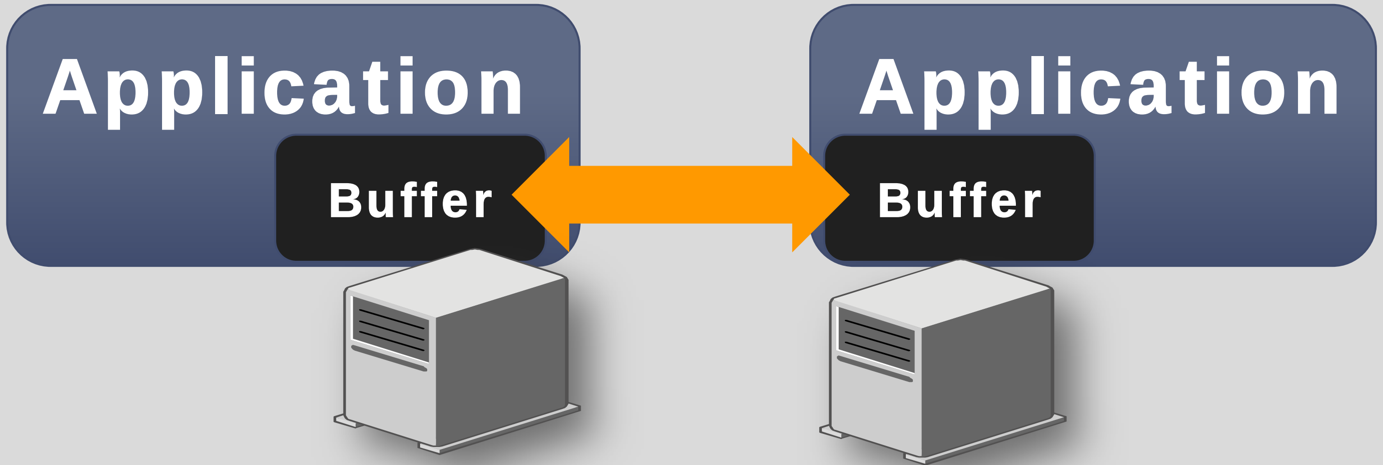
Synchronous

RDMA: Critical for Efficient use of Data Center Resources

ZERO Copy



Remote Data Transfer



Kernel Bypass Protocol Offload

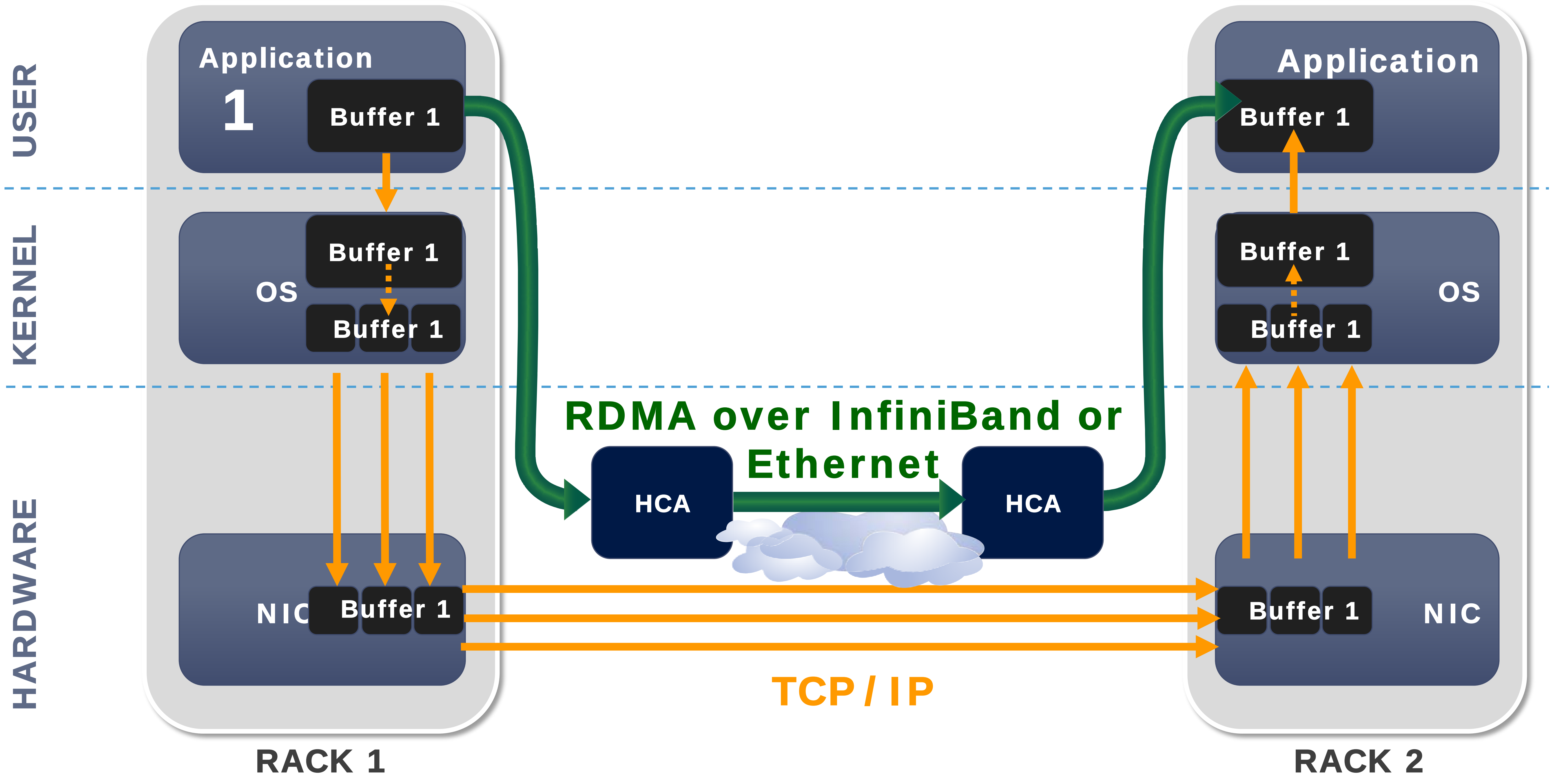
Low Latency, High Performance Data Transfers

InfiniBand - 56Gb/s

RoCE* – 40Gb/s

* RDMA over Converged Ethernet

RDMA – How it Works



Is Doing RDMA Easy or Hard? It Depends

- Block storage
  - iSCSI RDMA– iSER
  - SRP (InfiniBand only)
- File Storage
  - Windows – SMB Direct
  - Linux/Unix – NFS RDMA*
  - Lustre or GPFS

NetApp IBM GPFS X-IO violin MEMORY
lustre xyratex NIMBUS DATA
DataDirect NETWORKS ORACLE
vmware Windows Server 2012

- What about the rest?
  - Object, Hadoop, Ceph, xNBD
  - Storage clustering
  - Custom applications
  - Write to RDMA verbs

*NFS over RDMA is not yet a mature solution

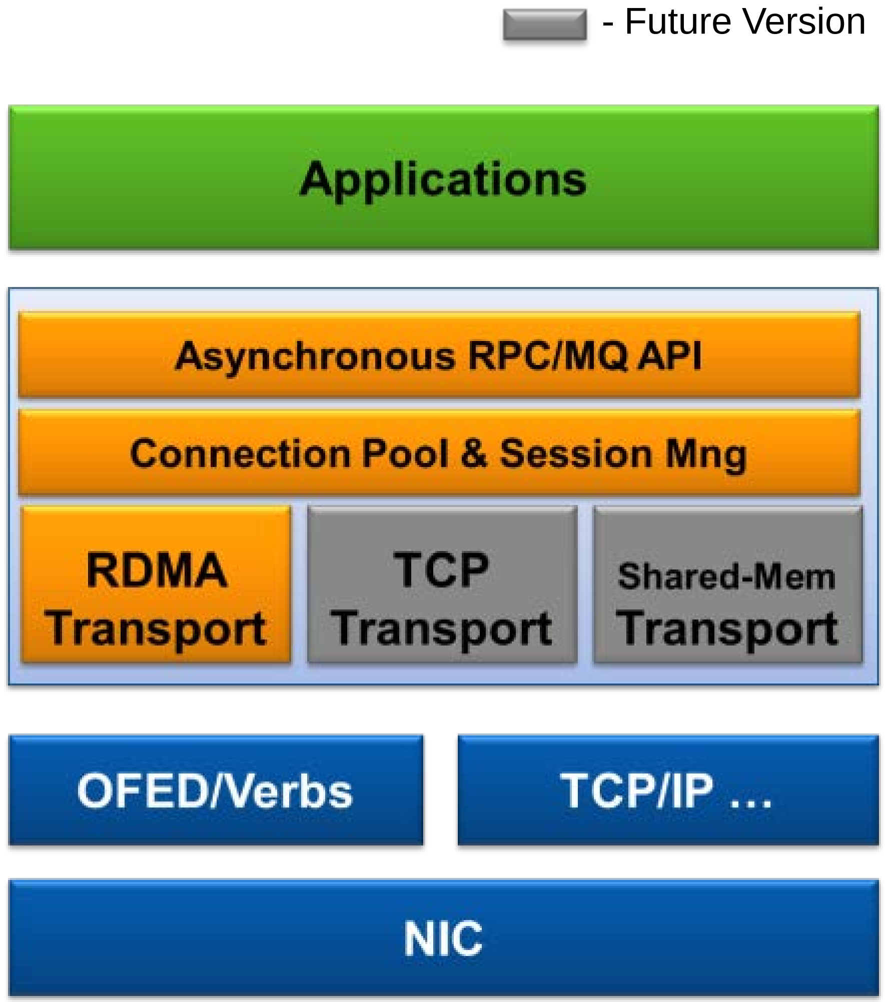
Accelio Makes RDMA Transport Easy

- RDMA-accelerated transport with minimal development effort
  - High-performance, Simple, Reliable Messaging and RPC Library
  - Optimal usage of CPU and Network hardware
  - Built in fault-tolerance, transaction reliability, and load-balancing

- The “Easy Button” of RDMA
  - No writing to verbs
  - Support user space, C/C++, Java

- Open Source Community project
  - Site: http://accelio.org
  - Code in: http://github.com/accelio
  - Project/Bug tracking: http://launchpad.net/accelio

- Version 1.0 GA in Feb 14th, 2014
  - Coming soon: Kernel space support



More details at: http://www.accelio.org/wp-content/themes/pyramid_child/pdf/WP_Accelio_OpenSource_IO_Message_and_RPC_Acceleration_Library.pdf

Accelio Goals

- Goals:
  - Maximize efficiency of modern CPU and NIC hardware
  - High performance data/message delivery middleware
  - Easy-to-use, reliable, scalable

- Key features:
  - Focus on high-performance asynchronous APIs
  - Reliable message delivery (end to end)
  - Request/Response (Transaction) or Send/Receive models
  - Connection and resource abstraction to max scalability and availability
  - Maximize multi-threaded application performance with dedicated HW resources per thread
  - Designed to maximize the benefits of RDMA, hardware offloads, and Multi-core CPUs
  - Native support for service and storage clustering/scale-out
  - Simple and abstract API

Why not stick to sockets

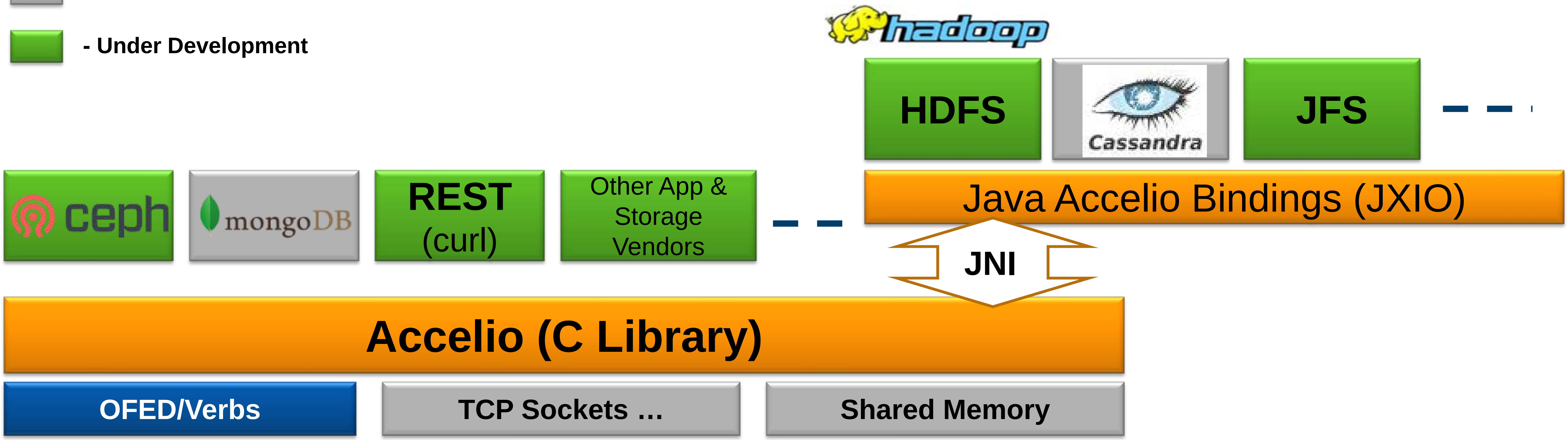
- Most data center protocols (HTTP, FTP, iSCSI, NFS, SQL, RPC) are transactional
  - Sockets API is byte streaming
- Socket APIs require heavy protocol processing and copies
  - No message boundaries; require data copy and credit/buffer management
- Can’t address growing CPU cores and NUMA
  - Applications must use multiple parallel connections and interrupt resources to scale
- Socket APIs don’t guarantee reliable delivery to the peer applications
  - Just reception by peer TCP stack; need extra application logic for reliability

- **New API semantics needed to leverage faster CPUs and Networking**

Accelio Integration With Other Applications/Projects

▨ - Planned

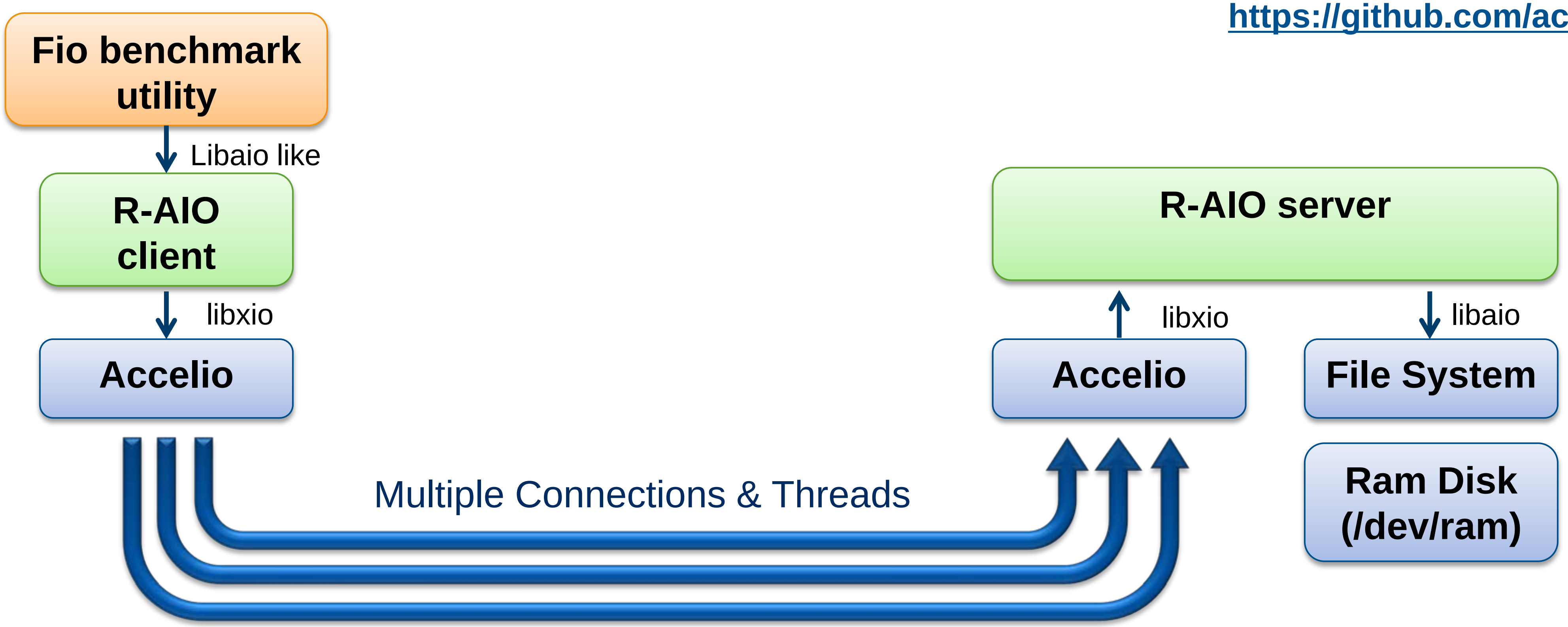
🟩 - Under Development



- Accelio being implemented today: Ceph, HDFS, xNBD

R-AIO Remote File Access Application Example

https://github.com/accelio/accelio/tree/master/examples/usr/raio



Performance

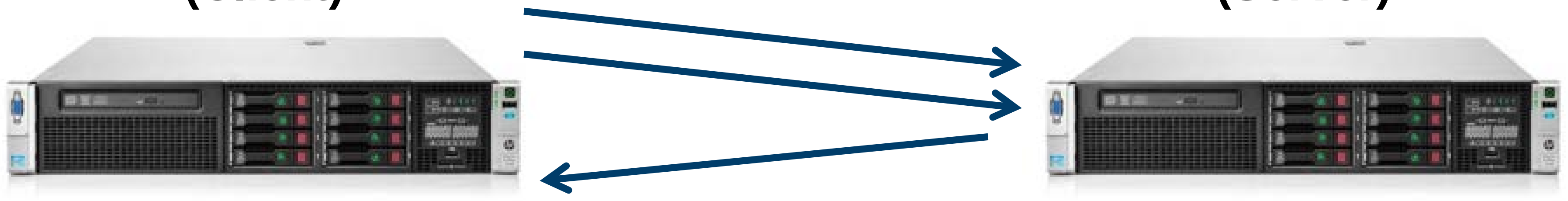
Max IOPs	2.5M
IO Latency	10us
Bandwidth	6GB/s

- Provide access to a remote file system by redirecting libaio (async file IO) commands to a remote server (which will issue the IO and return the results to the client)
- Deliver extraordinary performance to remote ram file (/dev/ram)
  - Using 4 CPUs & HW QPs for parallelism
  - Similar performance to local ram file access (i.e. minimal degradation due to communication)

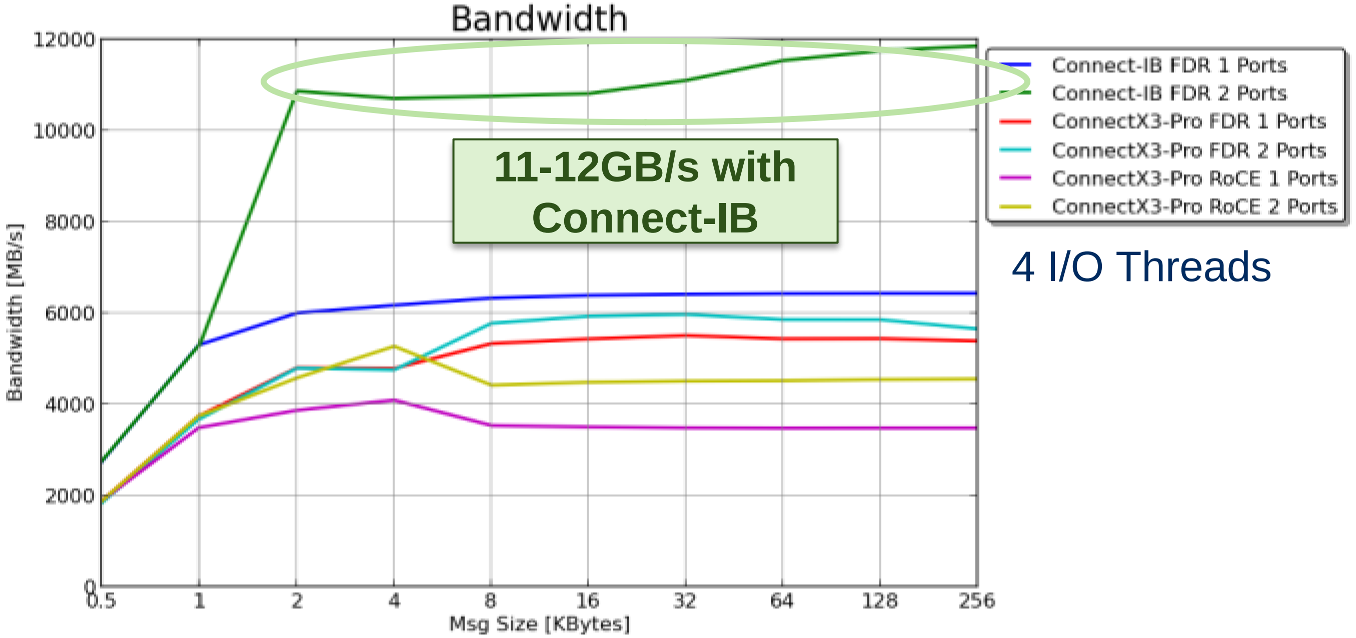
Test Configuration

- Server
  - HP ProLiant DL380p Gen8
  - 2 x Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz
  - 64 GB Memory
- Adapters
  - ConnectX3-Pro VPI (IB FDR or 40GbE)
  - Connect-IB 16x PCIe
  - OFED 2.1
- OS
  - RedHat EL 6.4
  - Kernel: 2.6.32-358.el6.x86_64
- Test
  - Accelio I/O test utility in C, User space
  - Request/Responce transactions (RPC)
  - Over 1 or 2 ports, using auto
    load balancing based on threads

Transaction flow



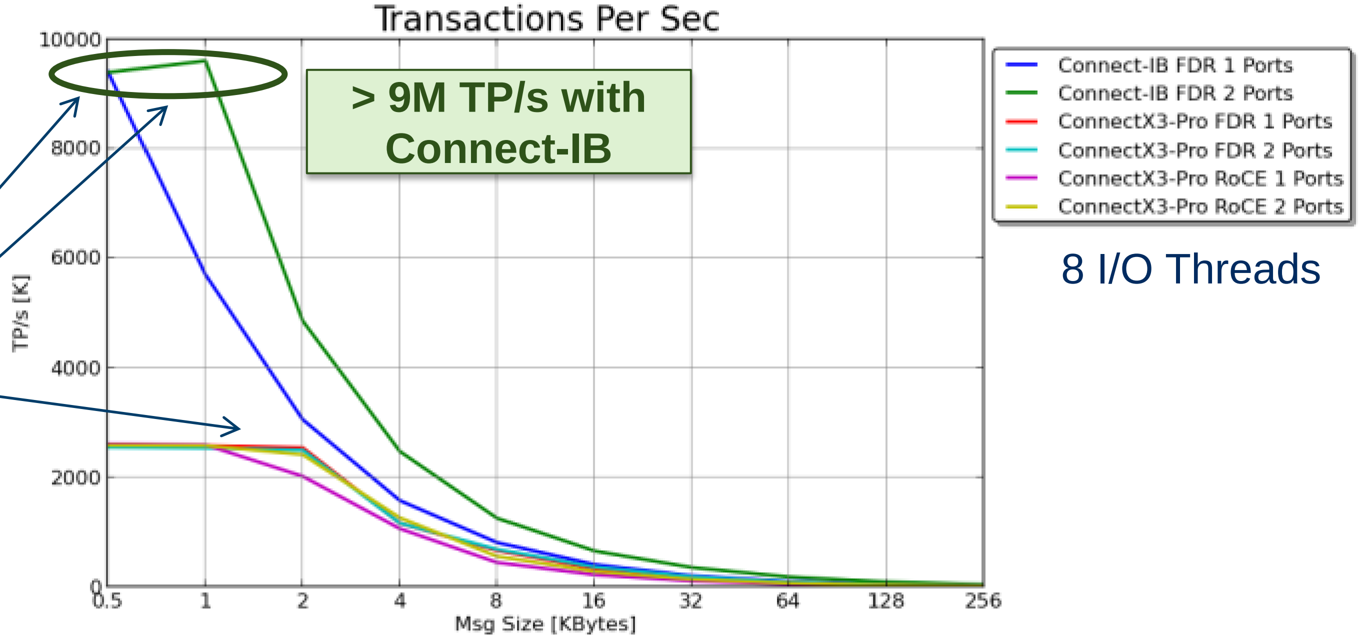
Bandwidth Results



Bandwidth

11-12GB/s with Connect-IB

4 I/O Threads

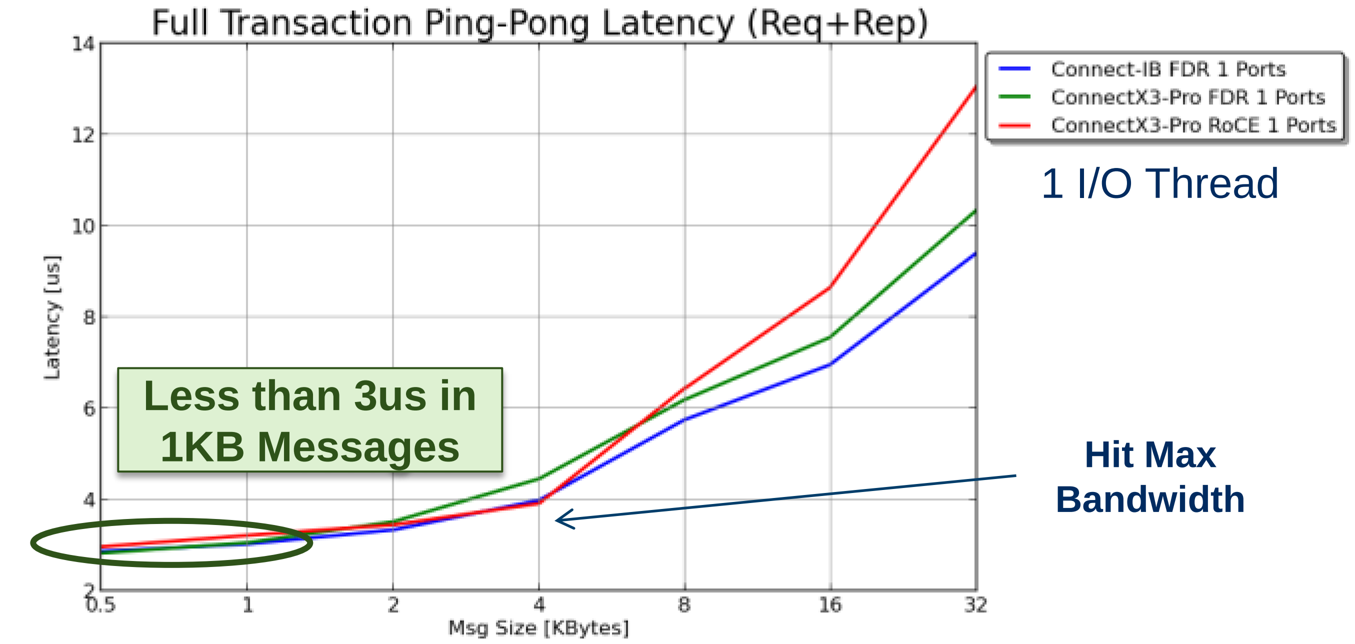


Transactions Per Sec

Hit Max Bandwidth

> 9M TP/s with Connect-IB

8 I/O Threads

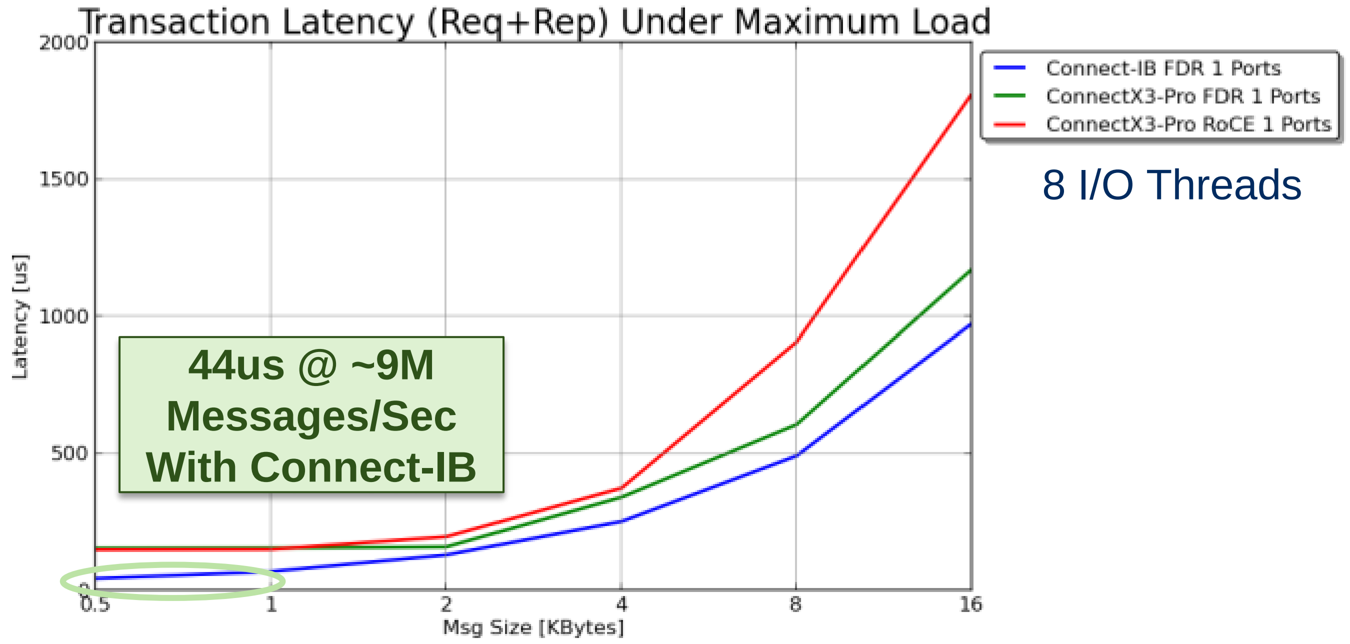


Full Transaction Ping-Pong Latency (Req+Rep)

1 I/O Thread

Less than 3us in 1KB Messages

Hit Max Bandwidth



Transaction Latency (Req+Rep) Under Maximum Load

8 I/O Threads

44us @ ~9M Messages/Sec With Connect-IB

Summary

- Flash is fast – you knew that!
- Fast storage needs fast networks
- Fast wire speed not enough—also need RDMA
  - RDMA for iSCSI and SMB Direct is already here and easy to use
  - Programming to verbs requires time investment
- Accelio is the “easy button” for adding RDMA
  - Very fast communications
  - Leverages multi-core, multi-queue
  - Being used today for Hadoop HDFS, Ceph, and custom applications

Thank You

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™