Accelerating and Benchmarking Big Data Processing on Modern Clusters

Open RG Big Data Webinar (Sept '15)

by

Dhabaleswar K. (DK) Panda

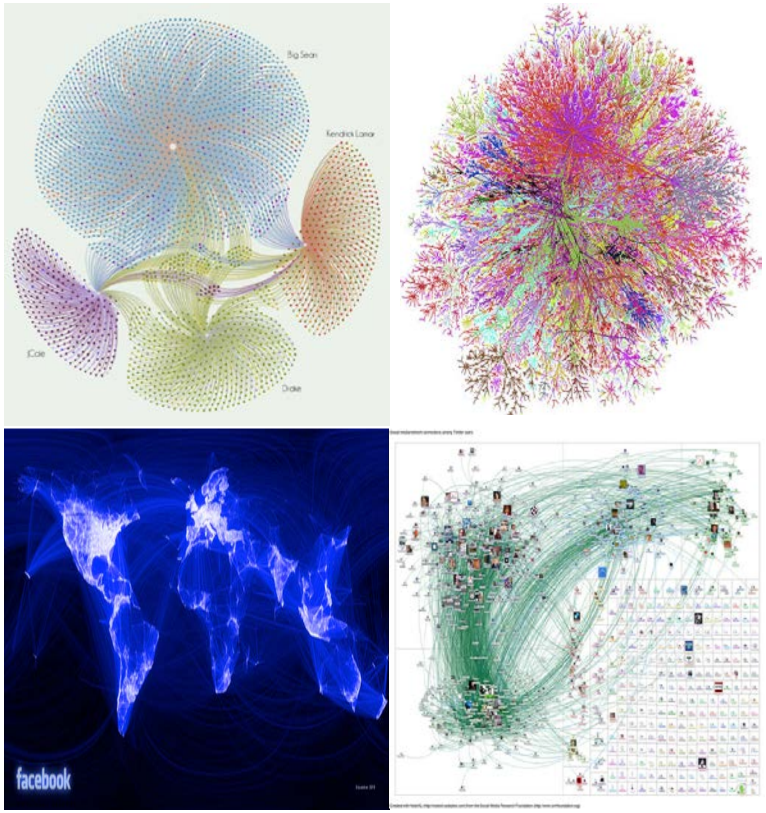
The Ohio State University

E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

Introduction to Big Data Applications and Analytics

- Big Data has become the one of the most important elements of business analytics
- Provides groundbreaking opportunities for enterprise information management and decision making
- The amount of data is exploding; companies are capturing and digitizing more information than ever
- The rate of information growth appears to be exceeding Moore's Law



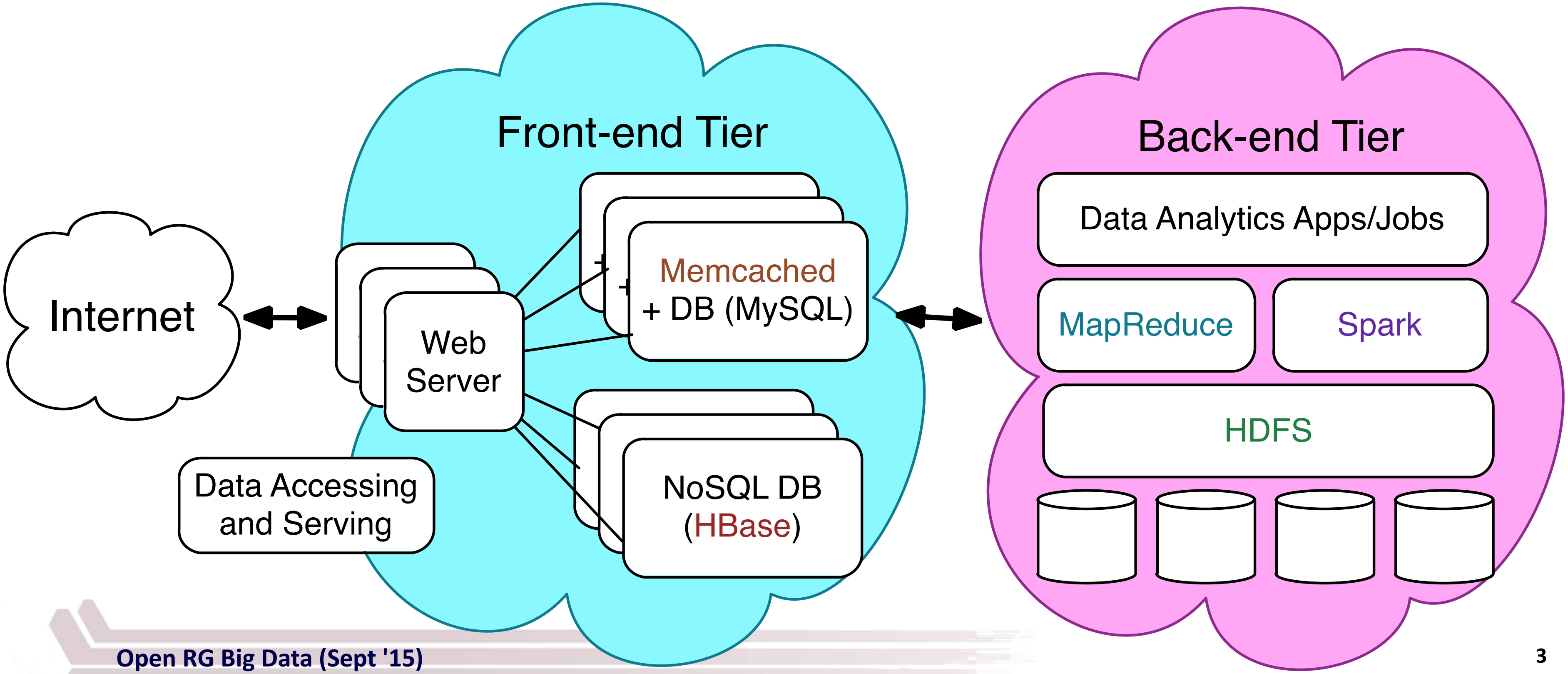
- Commonly accepted **3V**'s of Big Data
  - **Volume, Velocity, Variety**

Michael Stonebraker: Big Data Means at Least Three Different Things, http://www.nist.gov/itl/ssd/is/upload/NIST-stonebraker.pdf

- **5V**'s of Big Data – **3V** + **Value, Veracity**

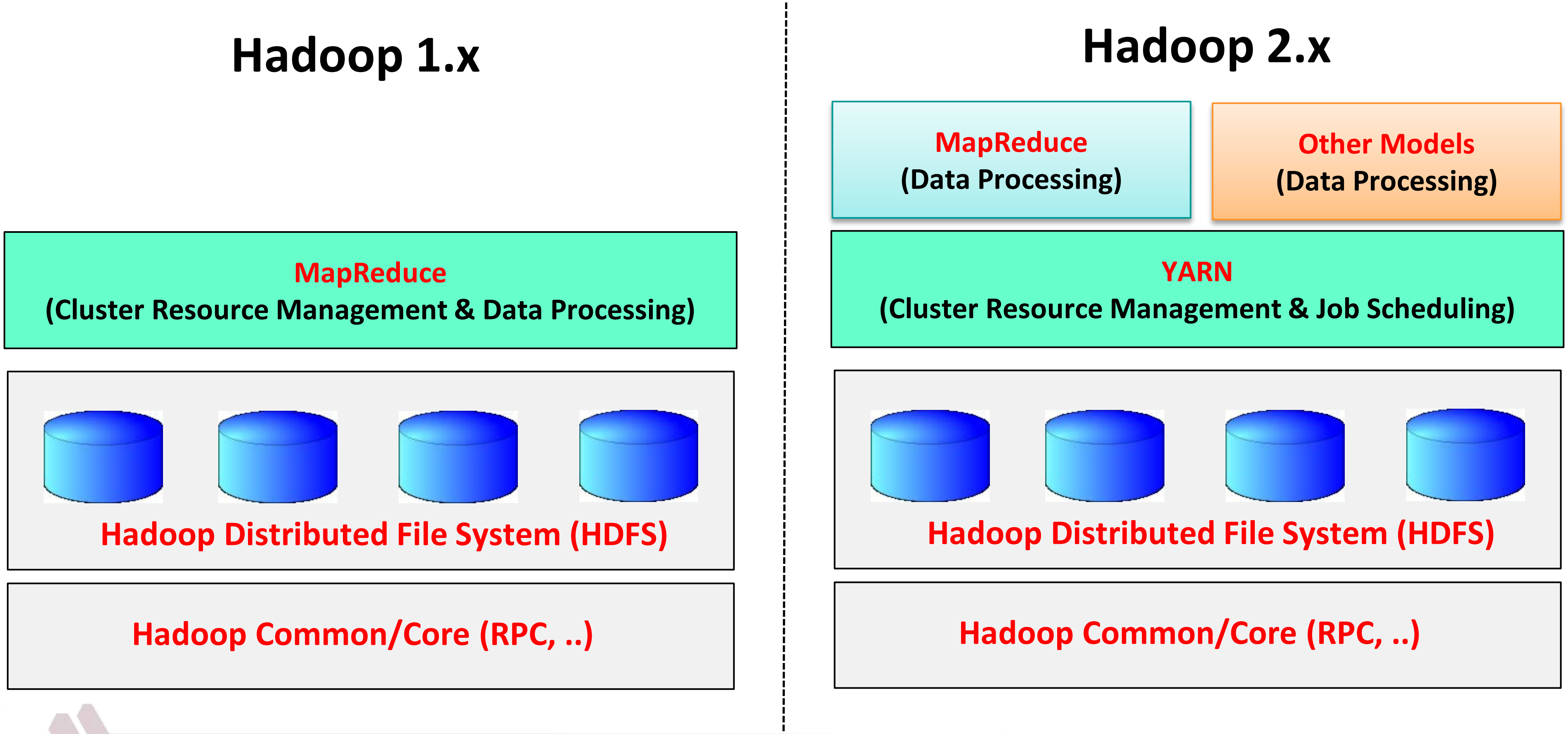
Data Management and Processing on Modern Clusters

- Substantial impact on designing and utilizing modern data management and processing systems in multiple tiers
  - Front-end data accessing and serving (Online)
    - Memcached + DB (e.g. MySQL), HBase
  - Back-end data analytics (Offline)
    - HDFS, MapReduce, Spark



Overview of Apache Hadoop Architecture

- Open-source implementation of Google MapReduce, GFS, and BigTable for Big Data Analytics
  - Hadoop Common Utilities (RPC, etc.), HDFS, MapReduce, YARN
- http://hadoop.apache.org



Hadoop 1.x

MapReduce
(Cluster Resource Management & Data Processing)

Hadoop Distributed File System (HDFS)

Hadoop Common/Core (RPC, ..)

Hadoop 2.x

MapReduce
(Data Processing)

Other Models
(Data Processing)

YARN
(Cluster Resource Management & Job Scheduling)

Hadoop Distributed File System (HDFS)

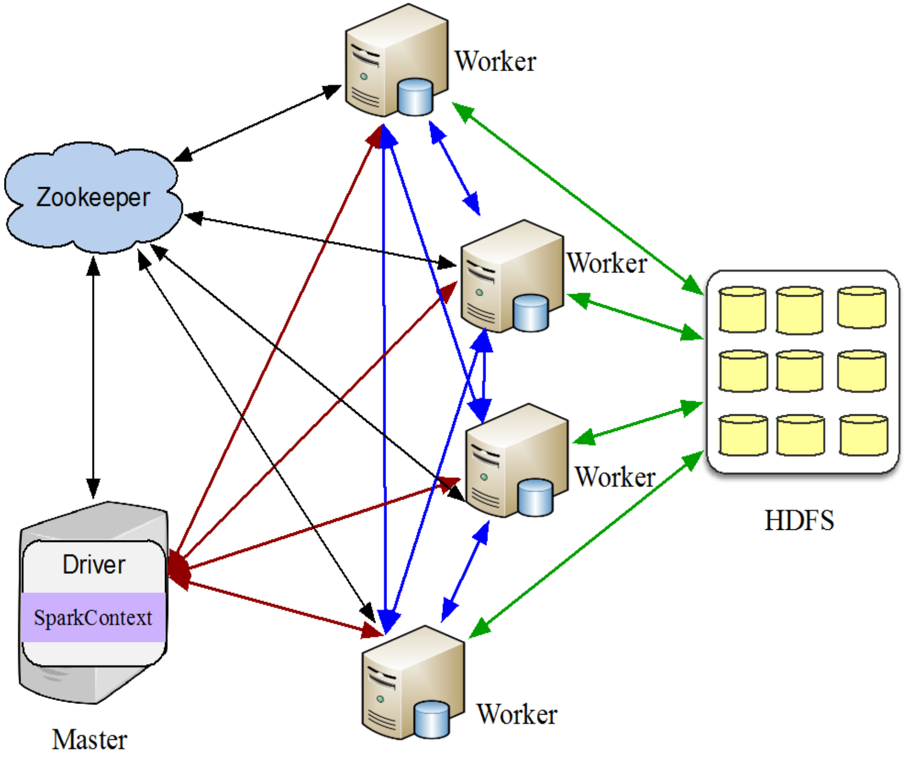
Hadoop Common/Core (RPC, ..)

HDFS and MapReduce in Apache Hadoop

- HDFS: Primary storage of Hadoop; highly reliable and fault-tolerant
  - NameNode stores the file system namespace
  - DataNodes store data blocks
- MapReduce: Computing framework of Hadoop; highly scalable
  - Map tasks read data from HDFS, operate on it, and write the intermediate data to local disk
  - Reduce tasks get data by shuffle, operate on it and write output to HDFS
- Adopted by many reputed organizations
  - eg: Facebook, Yahoo!
- Developed in Java for platform-independence and portability
- Uses Sockets/HTTP for communication!

Spark Architecture Overview

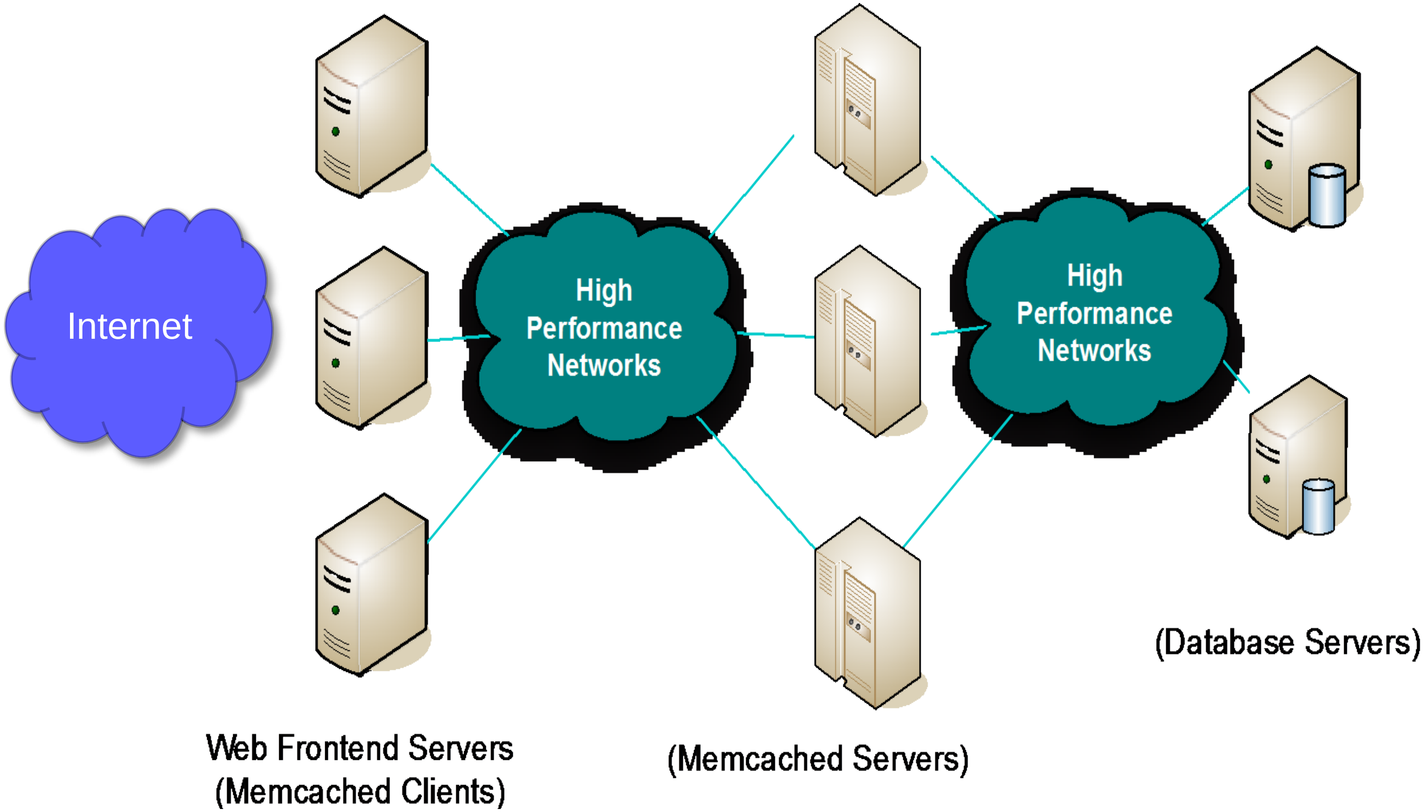
- An in-memory data-processing framework
  - Iterative machine learning jobs
  - Interactive data analytics
  - Scala based Implementation
  - Standalone, YARN, Mesos
- Scalable and communication intensive
  - Wide dependencies between Resilient Distributed Datasets (RDDs)
  - MapReduce-like shuffle operations to repartition RDDs
  - Sockets based communication



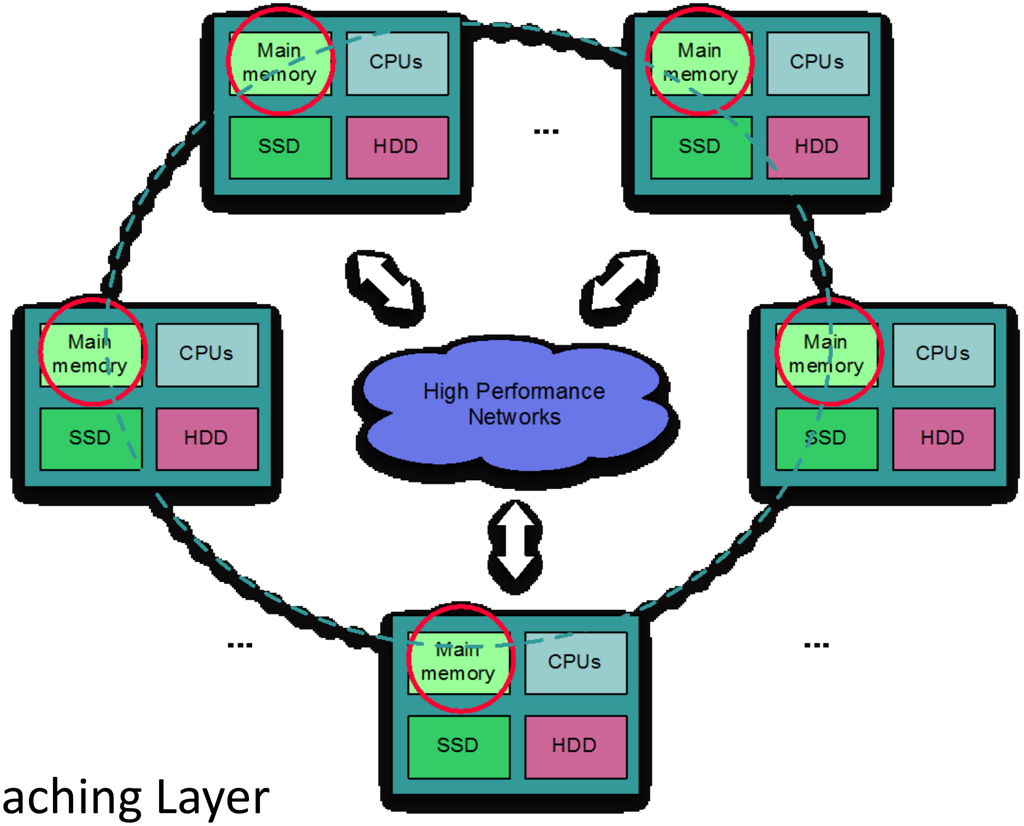
http://spark.apache.org

Overview of Web 2.0 Architecture and Memcached

- Three-layer architecture of Web 2.0
  - Web Servers, Memcached Servers, Database Servers
- Memcached is a core component of Web 2.0 architecture



Memcached Architecture

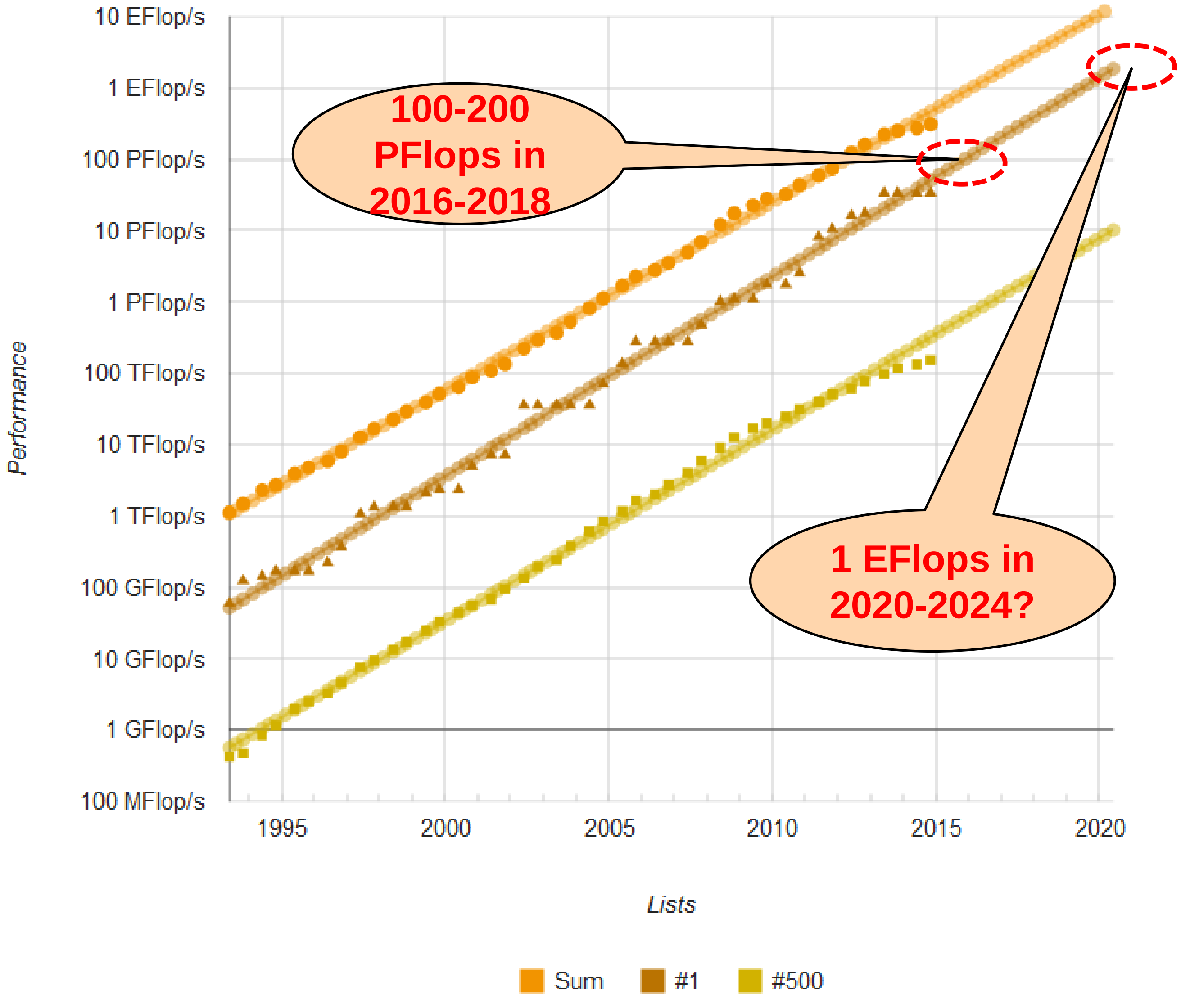


- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

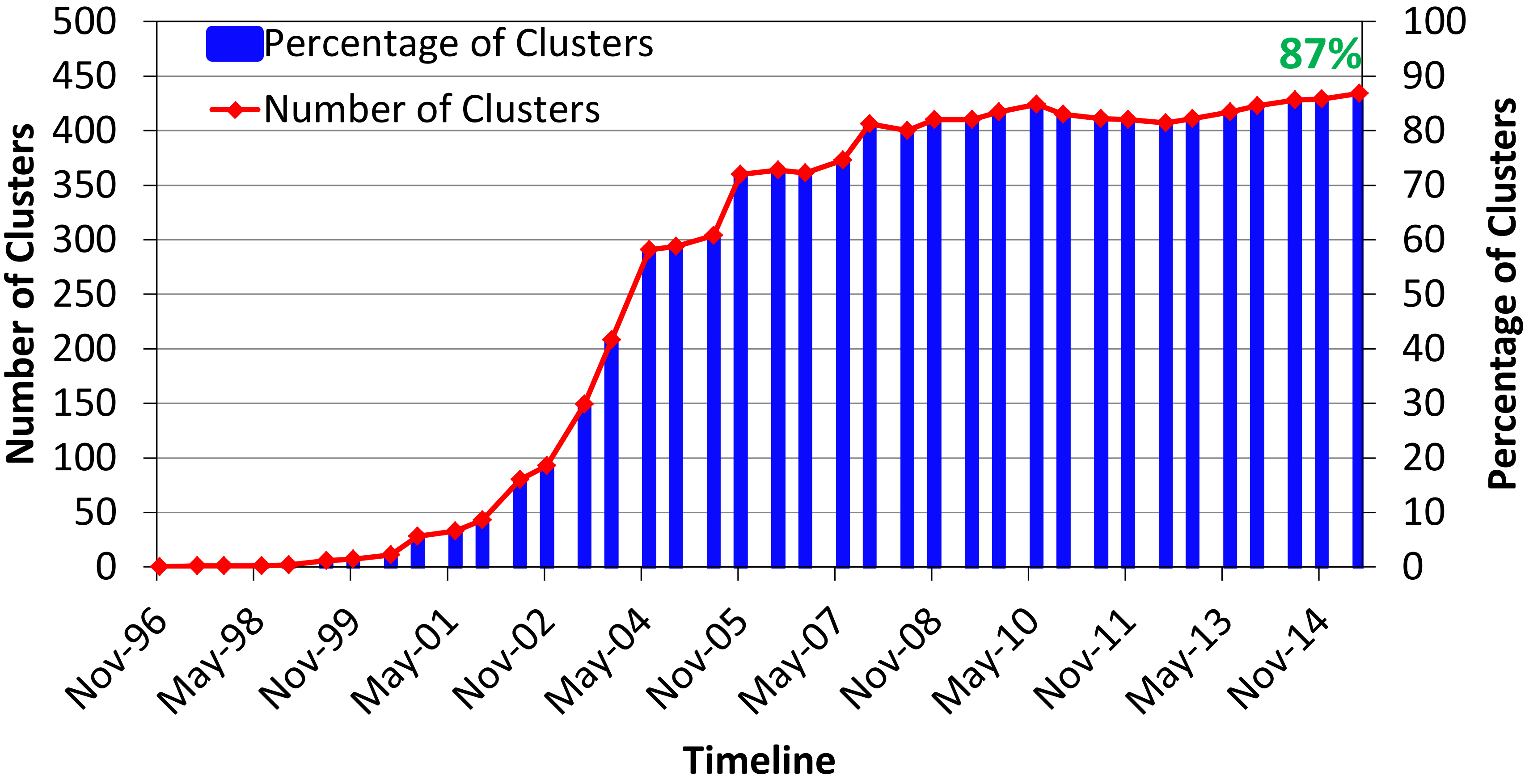
High-End Computing (HEC): PetaFlop to ExaFlop



100-200 PFlops in 2016-2018

1 EFlops in 2020-2024?

Trends for Commodity Computing Clusters in the Top 500 List (http://www.top500.org)



Drivers for Modern HPC Clusters

- High End Computing (HEC) is growing dramatically
  - High Performance Computing
  - Big Data Computing
- Technology Advancement
  - Multi-core/many-core technologies and accelerators
  - Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
  - Solid State Drives (SSDs) and Non-Volatile Random-Access Memory (NVRAM)
  - Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)

Tianhe – 2

Titan

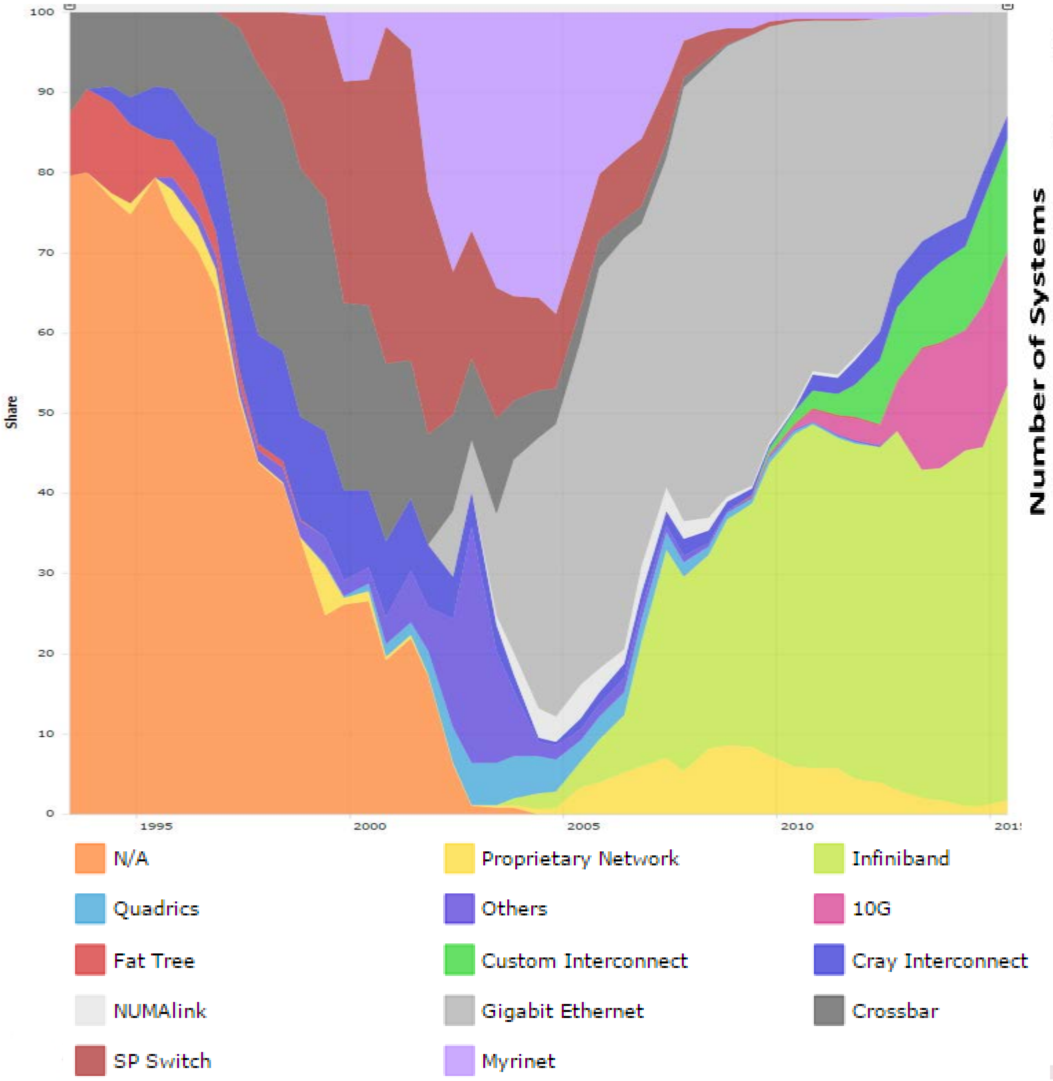
Stampede

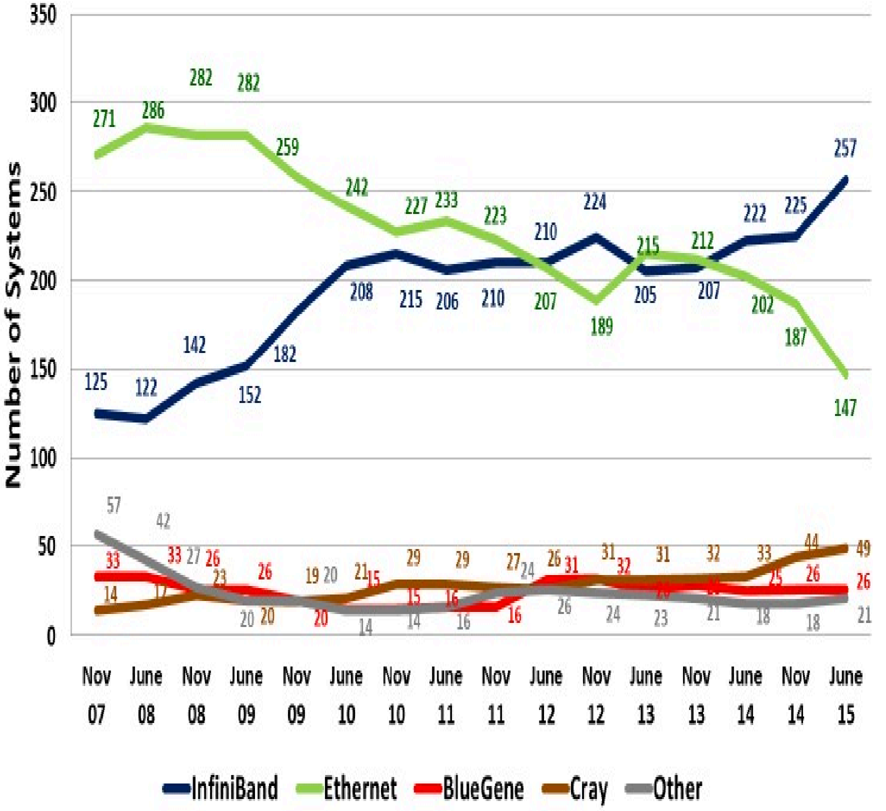
Tianhe – 1A

Trends of Networking Technologies in TOP500 Systems

Percentage share of InfiniBand is steadily increasing

Interconnect Family – Systems Share





Courtesy:
http://top500.org

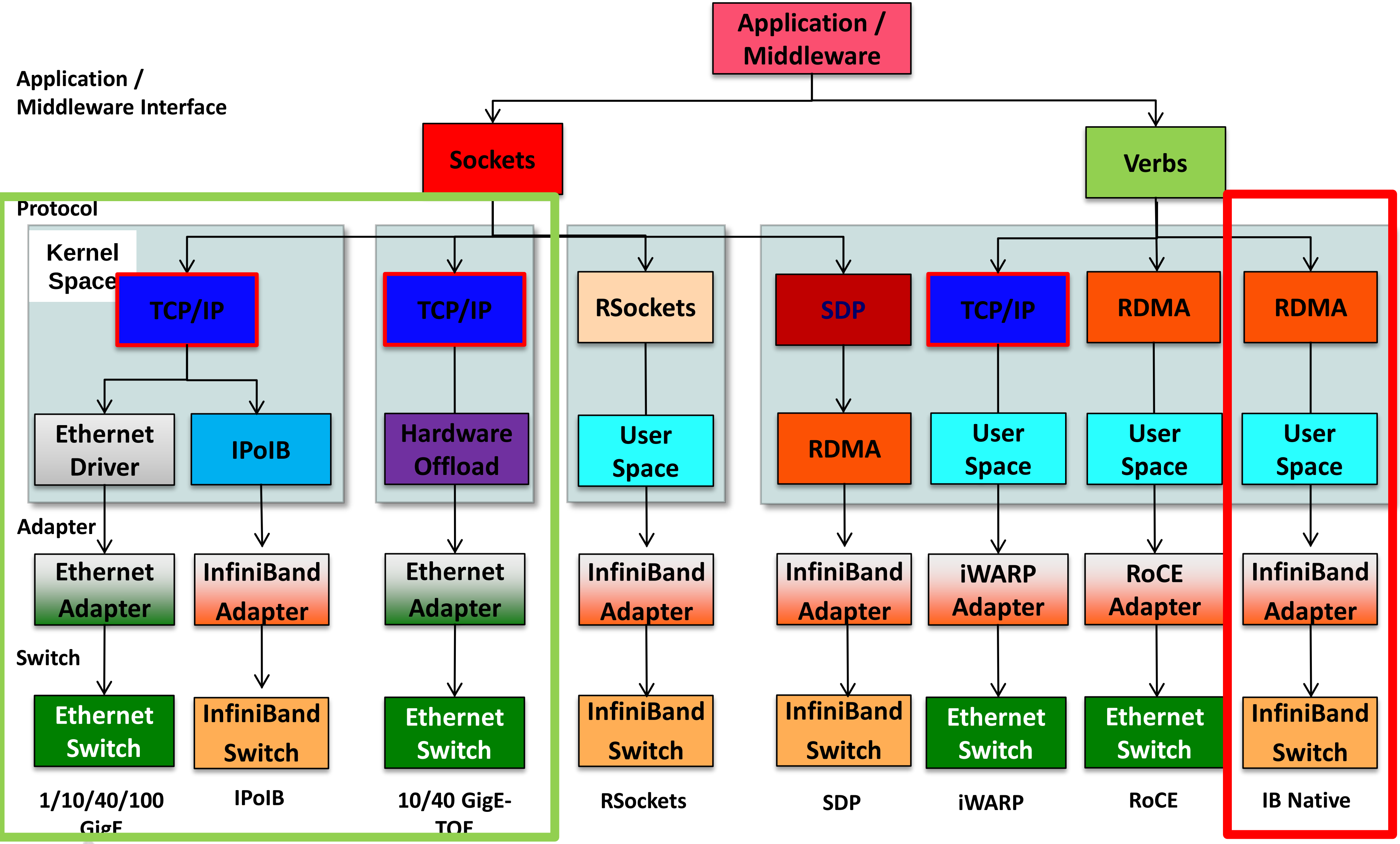
http://www.theplatform.net/2015/07/20/ethernet-will-have-to-work-harder-to-win-hpc/

Large-scale InfiniBand Installations

- 259 IB Clusters (51%) in the June 2015 Top500 list (http://www.top500.org)
- Installations in the Top 50 (24 systems):

519,640 cores (Stampede) at TACC (8th)	76,032 cores (Tsubame 2.5) at Japan/GSIC (22nd)
185,344 cores (Pleiades) at NASA/Ames (11th)	194,616 cores (Cascade) at PNNL (25th)
72,800 cores Cray CS-Storm in US (13th)	76,032 cores (Makman-2) at Saudi Aramco (28th)
72,800 cores Cray CS-Storm in US (14th)	110,400 cores (Pangea) in France (29th)
265,440 cores SGI ICE at Tulip Trading Australia (15th)	37,120 cores (Lomonosov-2) at Russia/MSU (31st)
124,200 cores (Topaz) SGI ICE at ERDC DSRC in US (16th)	57,600 cores (SwiftLucy) in US (33rd)
72,000 cores (HPC2) in Italy (17th)	50,544 cores (Occigen) at France/GENCI-CINES (36th)
115,668 cores (Thunder) at AFRL/USA (19th)	76,896 cores (Salomon) SGI ICE in Czech Republic (40th)
147,456 cores (SuperMUC) in Germany (20th)	73,584 cores (Spirit) at AFRL/USA (42nd)
86,016 cores (SuperMUC Phase 2) in Germany (21st)	**and many more!**

All interconnects and protocols in OpenFabrics Stack



Application / Middleware Interface

Protocol

Adapter

Switch

Application / Middleware

Sockets

Verbs

Kernel Space

TCP/IP

Ethernet Driver

IPoIB

Ethernet Adapter

InfiniBand Adapter

Ethernet Switch

InfiniBand Switch

1/10/40/100 GigE

IPoIB

TCP/IP

Hardware Offload

Ethernet Adapter

Ethernet Switch

10/40 GigE-TOE

RSockets

User Space

InfiniBand Adapter

InfiniBand Switch

RSockets

SDP

RDMA

InfiniBand Adapter

InfiniBand Switch

SDP

TCP/IP

User Space

iWARP Adapter

Ethernet Switch

iWARP

RDMA

User Space

RoCE Adapter

Ethernet Switch

RoCE

RDMA

User Space

InfiniBand Adapter

InfiniBand Switch

IB Native

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

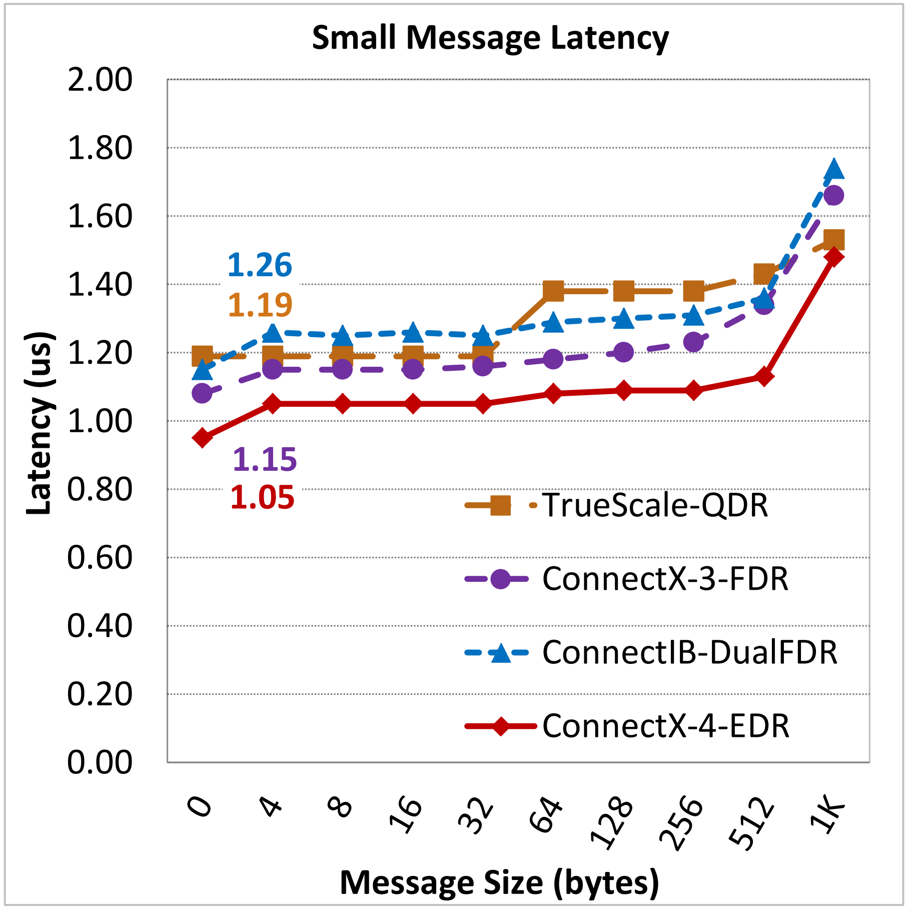
Wide Adoption of RDMA Technology

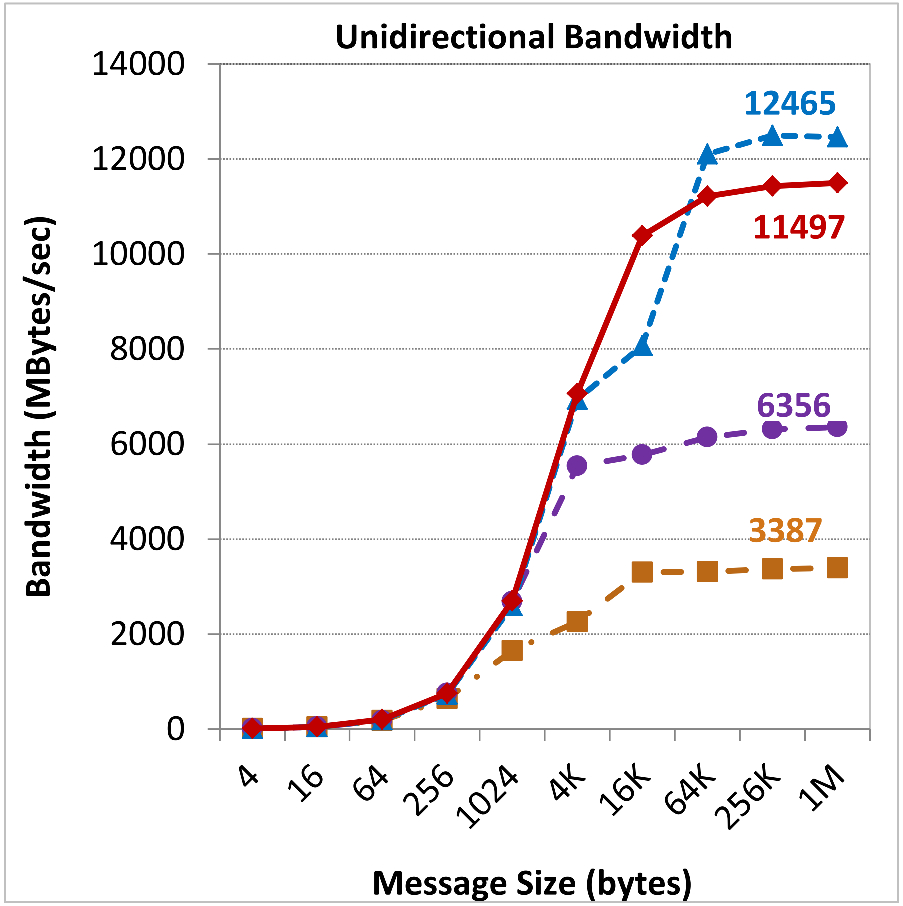
- Message Passing Interface (MPI) for HPC
- Parallel File Systems
  - Lustre
  - GPFS
- Delivering excellent performance:
  - < 1.0 microsec latency
  - 100 Gbps bandwidth
  - 5-10% CPU utilization
- Delivering excellent scalability

MVAPICH2 Software

- High Performance open-source MPI Library for InfiniBand, 10-40Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Available since 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - **Used by more than 2,450 organizations in 76 countries**
  - **More than 289,000 downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '15 ranking)
    - 8th ranked 519,640-core cluster (Stampede) at TACC
    - 11th ranked 185,344-core cluster (Pleiades) at NASA
    - 22nd ranked 76,032-core cluster (Tsubame 2.5) at Tokyo Institute of Technology and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Stampede at TACC (8th in Jun'15, 462,462 cores, 5.168 Plops)

Latency & Bandwidth: MPI over IB with MVAPICH2





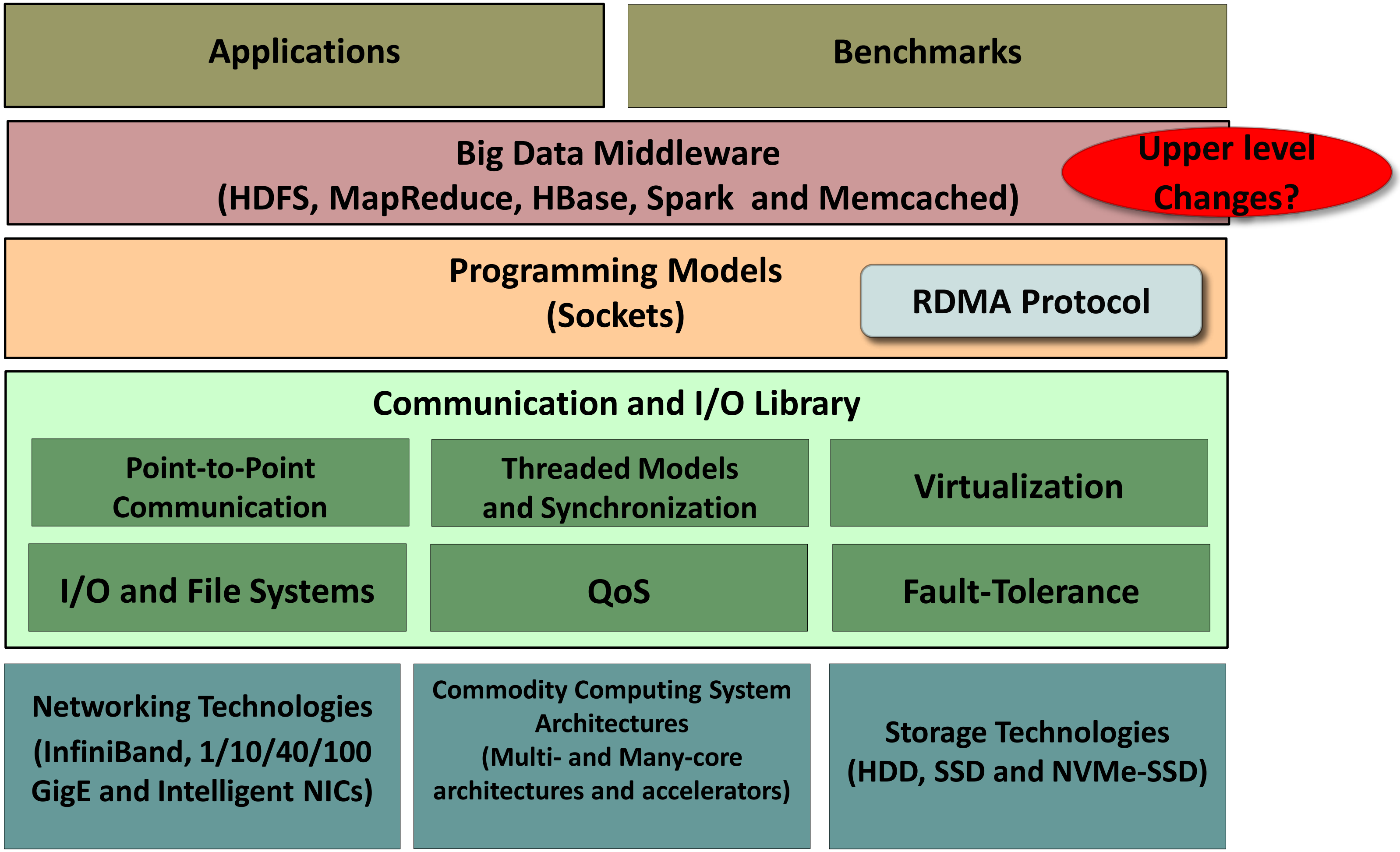
TrueScale-QDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

ConnectX-4-EDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 Back-to-back

Designing Communication and I/O Libraries for Big Data Systems: Solved a Few Initial Challenges



Applications

Benchmarks

Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)

Upper level Changes?

Programming Models
(Sockets)

RDMA Protocol

Communication and I/O Library

Point-to-Point Communication

Threaded Models and Synchronization

Virtualization

I/O and File Systems

QoS

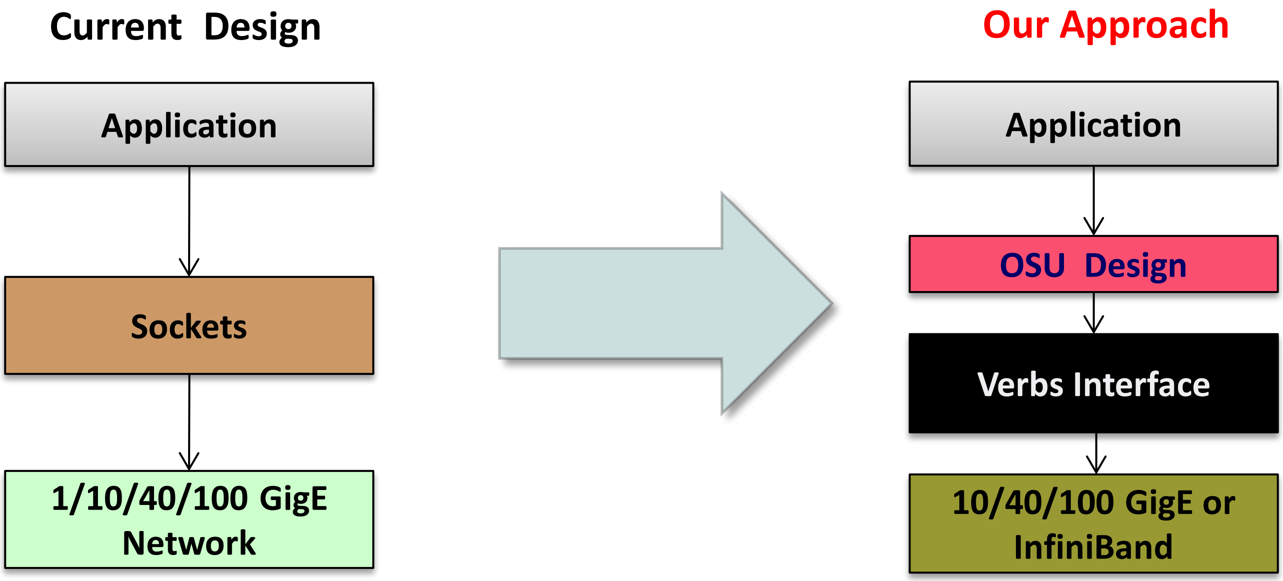
Fault-Tolerance

Networking Technologies (InfiniBand, 1/10/40/100 GigE and Intelligent NICs)

Commodity Computing System Architectures
(Multi- and Many-core architectures and accelerators)

Storage Technologies
(HDD, SSD and NVMe-SSD)

Can Big Data Processing Systems be Designed with High-Performance Networks and Protocols?



- Sockets not designed for high-performance
  - Stream semantics often mismatch for upper layers
  - Zero-copy not available for non-blocking sockets

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

The High-Performance Big Data (HiBD) Project

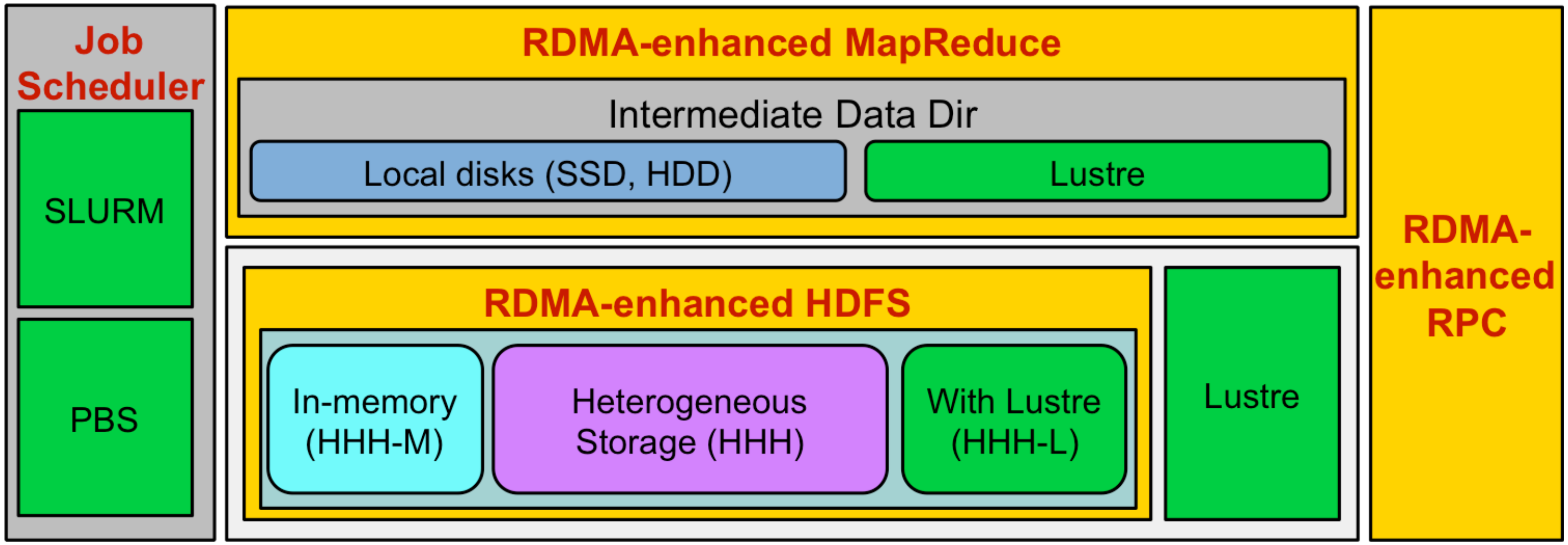
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache and HDP Hadoop distributions
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- RDMA for Memcached (RDMA-Memcached)
- OSU HiBD-Benchmarks (OHB)
  - HDFS and Memcached Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 130 organizations from 20 countries
- More than 13,000 downloads from the project site
- RDMA for Apache HBase, Spark and CDH

HiBD
High-Performance
Big Data

Network Based Computing
Laboratory

The Ohio State
University

Different Modes of RDMA for Apache Hadoop 2.x



- **HHH**: Heterogeneous storage devices with hybrid replication schemes are supported in this mode of operation to have better fault-tolerance as well as performance. This mode is enabled by **default** in the package.
- **HHH-M**: A high-performance in-memory based setup has been introduced in this package that can be utilized to perform all I/O operations in-memory and obtain as much performance benefit as possible.
- **HHH-L**: With parallel file systems integrated, HHH-L mode can take advantage of the Lustre available in the cluster.
- **MapReduce over Lustre, with/without local disks**: Besides, HDFS based solutions, this package also provides support to run MapReduce jobs on top of Lustre alone. Here, two different modes are introduced: with local disks and without local disks.
- **Running with Slurm and PBS**: Supports deploying RDMA for Apache Hadoop 2.x with Slurm and PBS in different running modes (HHH, HHH-M, HHH-L, and MapReduce over Lustre).

RDMA for Apache Hadoop 2.x Distribution

- High-Performance Design of Hadoop over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for HDFS, MapReduce, and RPC components
  - Enhanced HDFS with in-memory and heterogeneous storage
  - High performance design of MapReduce over Lustre
  - Plugin-based architecture supporting RDMA-based designs for Apache Hadoop and HDP
  - Easily configurable for different running modes (HHH, HHH-M, HHH-L, and MapReduce over Lustre) and different protocols (native InfiniBand, RoCE, and IPoIB)
- Current release: 0.9.8
  - Based on Apache Hadoop 2.7.1
  - Compliant with Apache Hadoop 2.7.1 and HDP 2.3.0.0 APIs and applications
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR and FDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - Different file systems with disks and SSDs and Lustre
  - http://hibd.cse.ohio-state.edu

RDMA for Memcached Distribution

- High-Performance Design of Memcached over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for Memcached and libMemcached components
  - High performance design of SSD-Assisted Hybrid Memory
  - Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)
- Current release: 0.9.3
  - Based on Memcached 1.4.22 and libMemcached 1.0.18
  - Compliant with libMemcached APIs and applications
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR and FDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - SSD
  - http://hibd.cse.ohio-state.edu

OSU HiBD Micro-Benchmark (OHB) Suite – HDFS & Memcached

- Micro-benchmarks for Hadoop Distributed File System (HDFS)
  - Sequential Write Latency (**SWL**) Benchmark, Sequential Read Latency (**SRL**) Benchmark, Random Read Latency (**RRL**) Benchmark, Sequential Write Throughput (**SWT**) Benchmark, Sequential Read Throughput (**SRT**) Benchmark
  - Support benchmarking of
    - Apache Hadoop 1.x and 2.x HDFS, Hortonworks Data Platform (HDP) HDFS, Cloudera Distribution of Hadoop (CDH) HDFS
- Micro-benchmarks for Memcached
  - **Get** Benchmark, **Set** Benchmark, and **Mixed** Get/Set Benchmark
- Current release: 0.8
- http://hibd.cse.ohio-state.edu

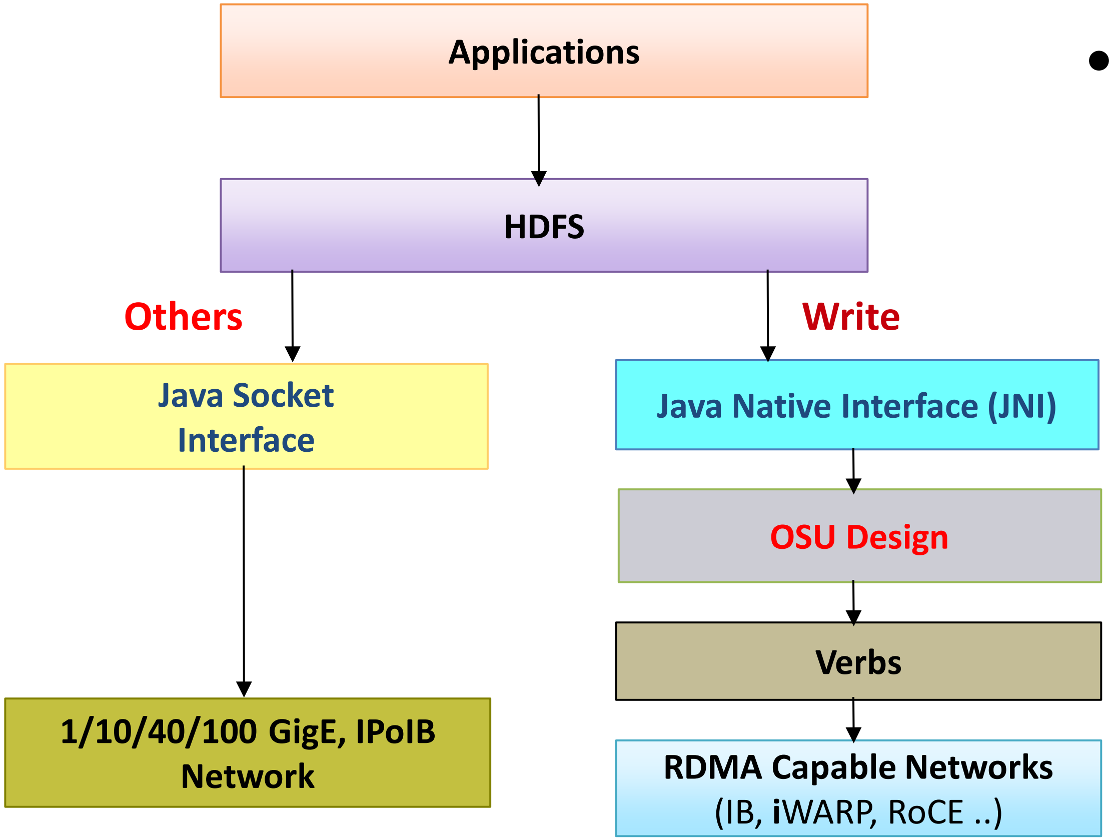
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark

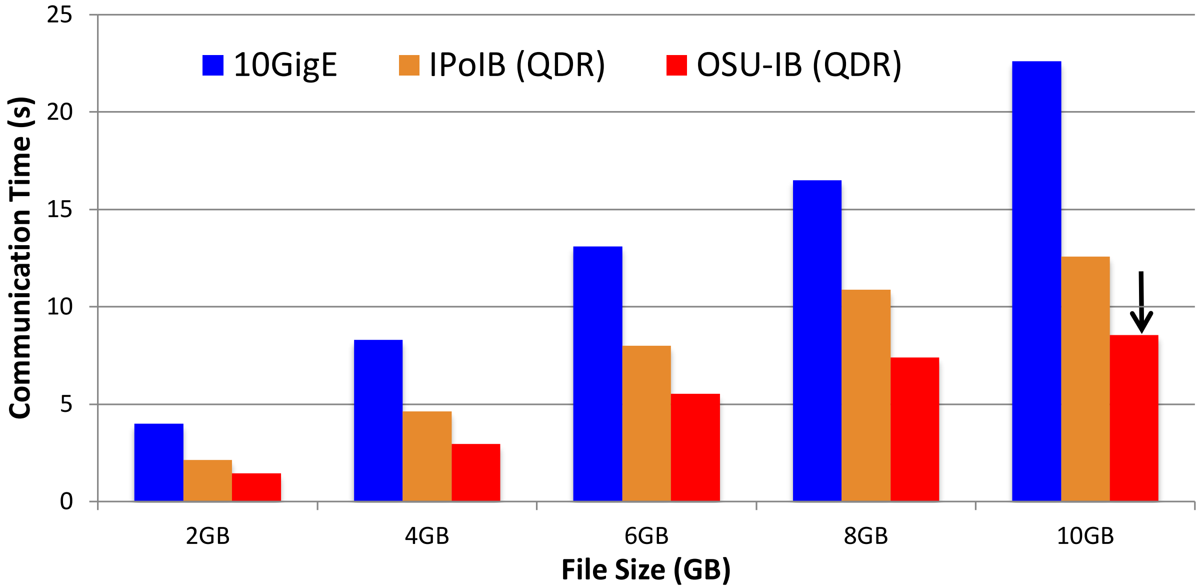
Design Overview of HDFS with RDMA



- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - Parallel replication support
  - On-demand connection setup
  - InfiniBand/RoCE support

- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Java based HDFS with communication library written in native code

Communication Times in HDFS



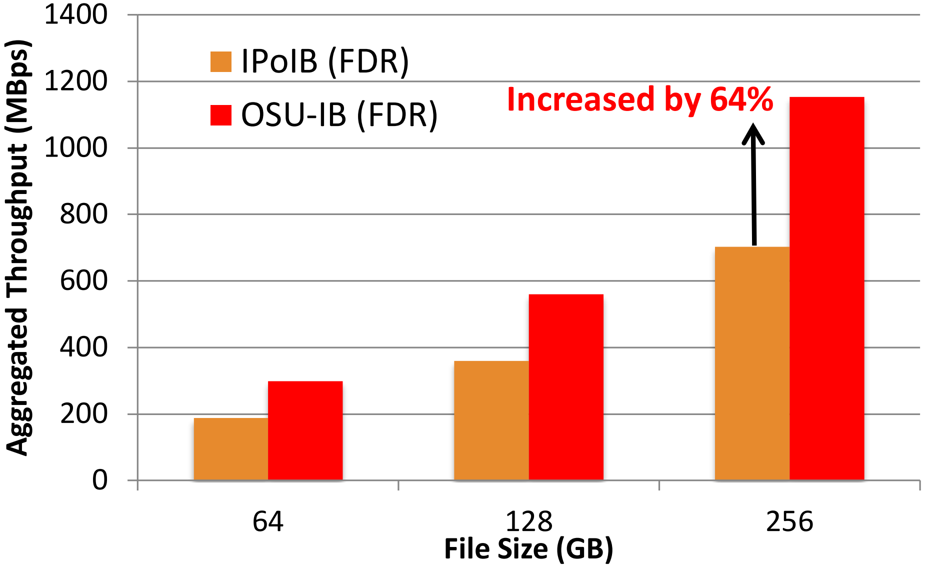
Reduced by 30%

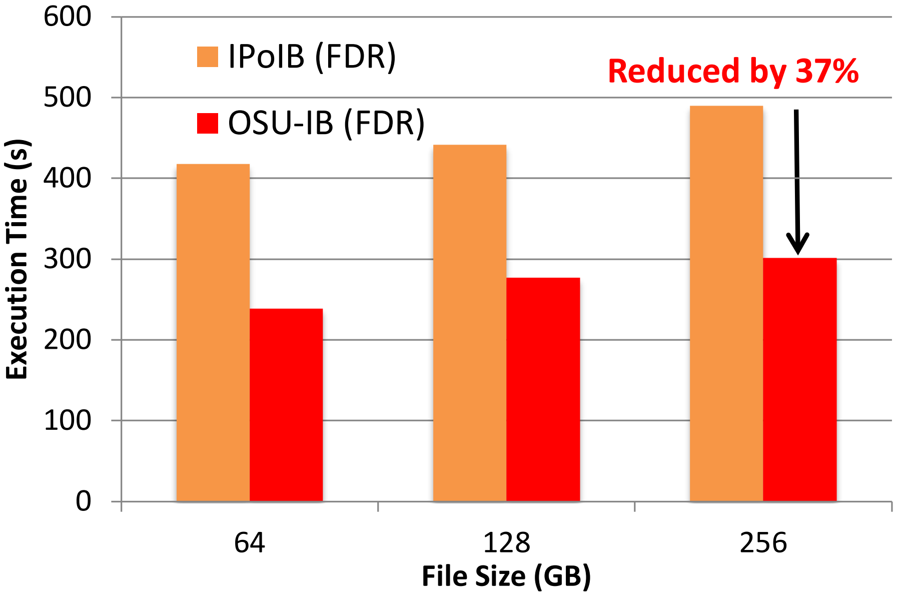
- Cluster with HDD DataNodes
  - **30%** improvement in communication time over IPoIB (QDR)
  - **56%** improvement in communication time over 10GigE
- Similar improvements are obtained for SSD DataNodes

N. S. Islam, M. W. Rahman, J. Jose, R. Rajachandrasekar, H. Wang, H. Subramoni, C. Murthy and D. K. Panda , High Performance RDMA-Based Design of HDFS over InfiniBand , Supercomputing (SC), Nov 2012

N. Islam, X. Lu, W. Rahman, and D. K. Panda, SOR-HDFS: A SEDA-based Approach to Maximize Overlapping in RDMA-Enhanced HDFS, HPDC '14, June 2014

Evaluations using Enhanced DFSIO of Intel HiBench on TACC-Stampede



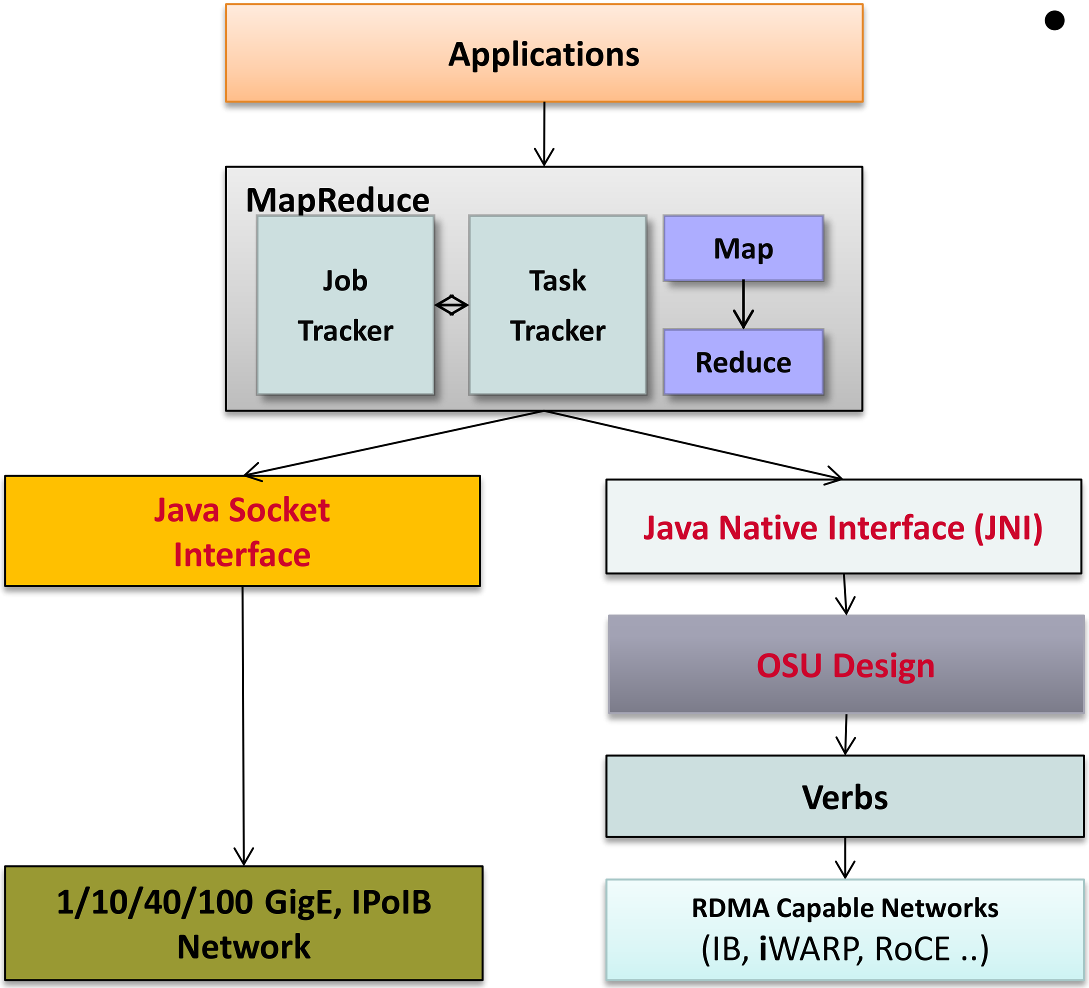


- Cluster with 64 DataNodes (1K cores), single HDD per node
  - **64%** improvement in throughput over IPoIB (FDR) for 256GB file size
  - **37%** improvement in latency over IPoIB (FDR) for 256GB file size

Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark

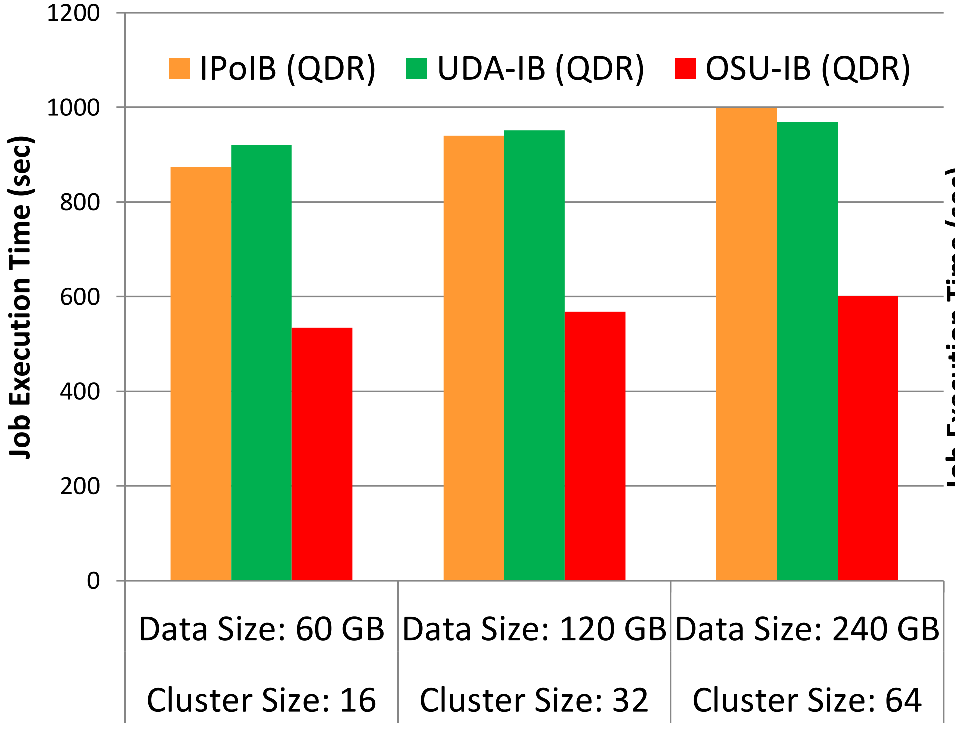
Design Overview of MapReduce with RDMA



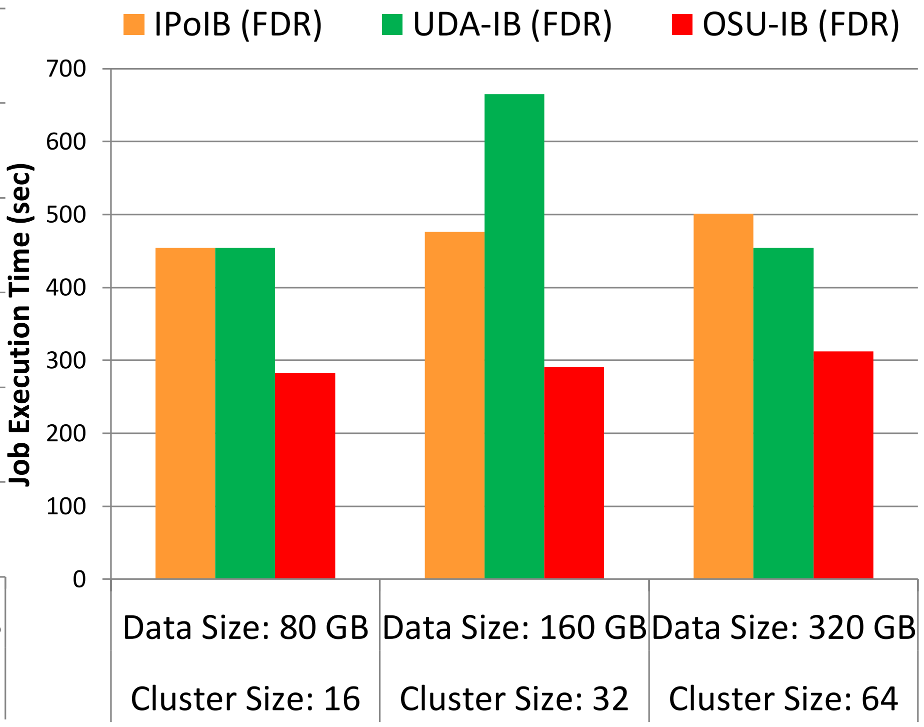
- Design Features
  - RDMA-based shuffle
  - Prefetching and caching map output
  - Efficient Shuffle Algorithms
  - In-memory merge
  - On-demand Shuffle Adjustment
  - Advanced overlapping
    - map, shuffle, and merge
    - shuffle, merge, and reduce
  - On-demand connection setup
  - InfiniBand/RoCE support

- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Java based MapReduce with communication library written in native code

Performance Evaluation of Sort and TeraSort



Sort in OSU Cluster

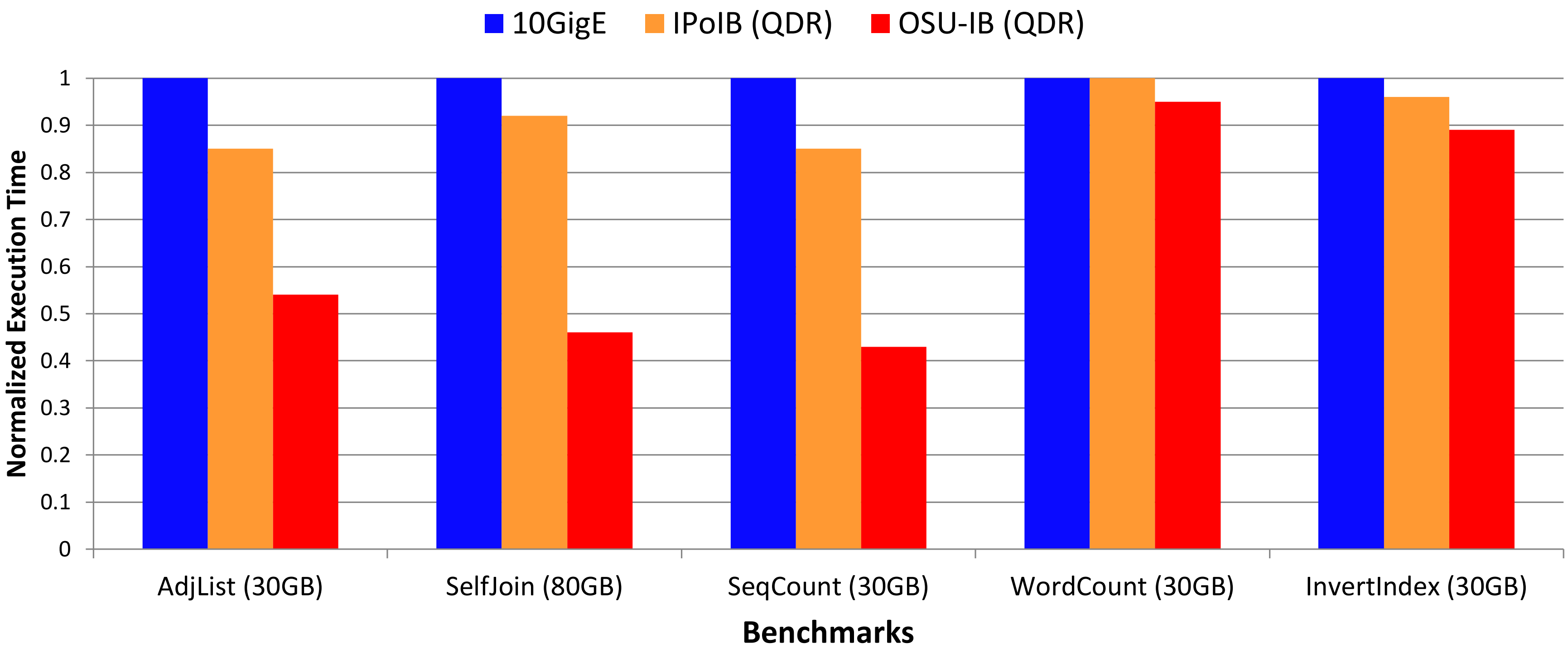


TeraSort in TACC Stampede

- For 240GB Sort in 64 nodes (512 cores)
  - 40% improvement over IPoIB (QDR) with HDD used for HDFS

- For 320GB TeraSort in 64 nodes (1K cores)
  - 38% improvement over IPoIB (FDR) with HDD used for HDFS

Evaluations using PUMA Workload

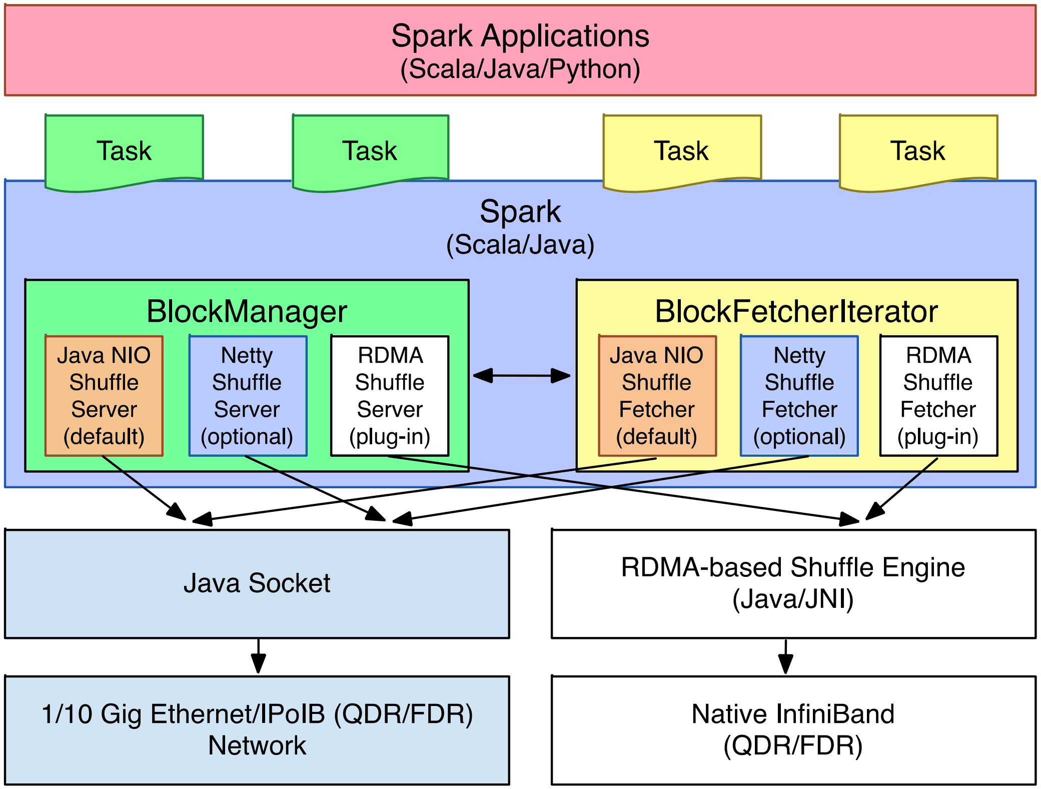


- 50% improvement in Self Join over IPoIB (QDR) for 80 GB data size
- 49% improvement in Sequence Count over IPoIB (QDR) for 30 GB data size

Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark

Design Overview of Spark with RDMA



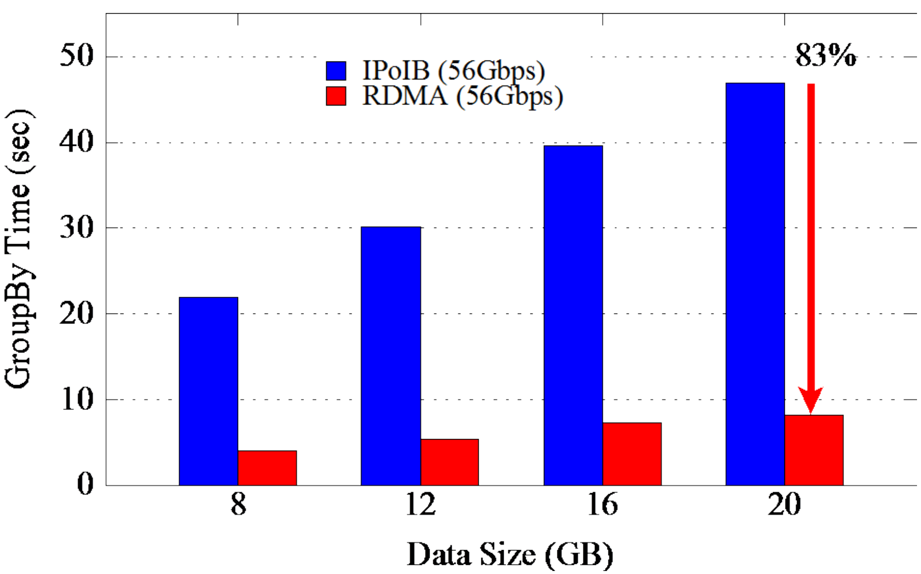
Spark Applications
(Scala/Java/Python)
Task
Task
Task
Task
Spark
(Scala/Java)
BlockManager
Java NIO Shuffle Server (default)
Netty Shuffle Server (optional)
RDMA Shuffle Server (plug-in)
BlockFetcherIterator
Java NIO Shuffle Fetcher (default)
Netty Shuffle Fetcher (optional)
RDMA Shuffle Fetcher (plug-in)
Java Socket
RDMA-based Shuffle Engine
(Java/JNI)
1/10 Gig Ethernet/IPoIB (QDR/FDR)
Network
Native InfiniBand
(QDR/FDR)

- Design Features
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking and out-of-order data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

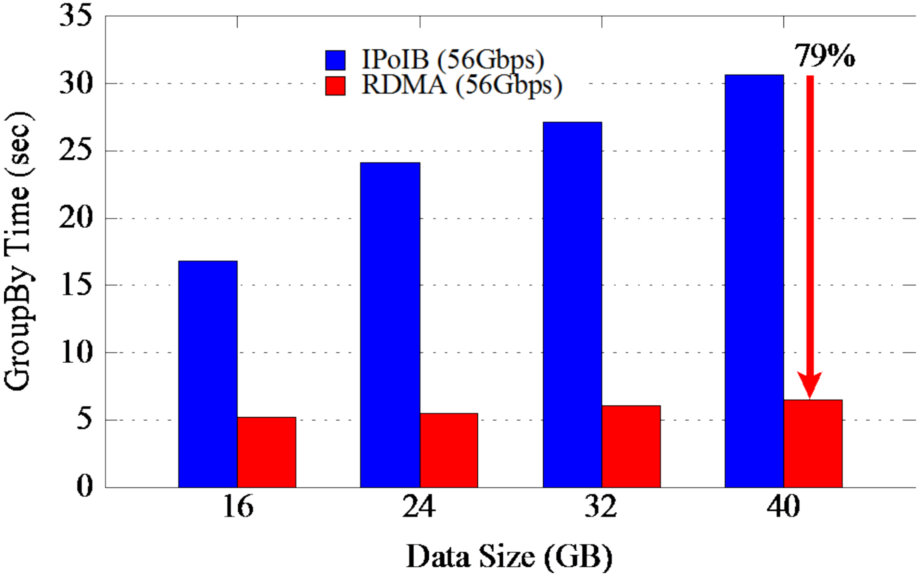
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

Performance Evaluation on TACC Stampede - GroupByTest



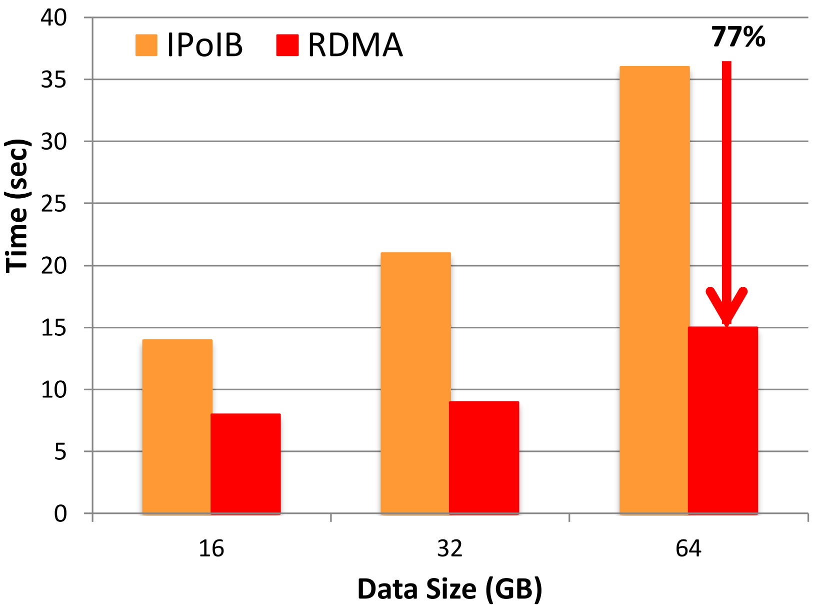
8 Worker Nodes, 128 Cores, (128M 128R)



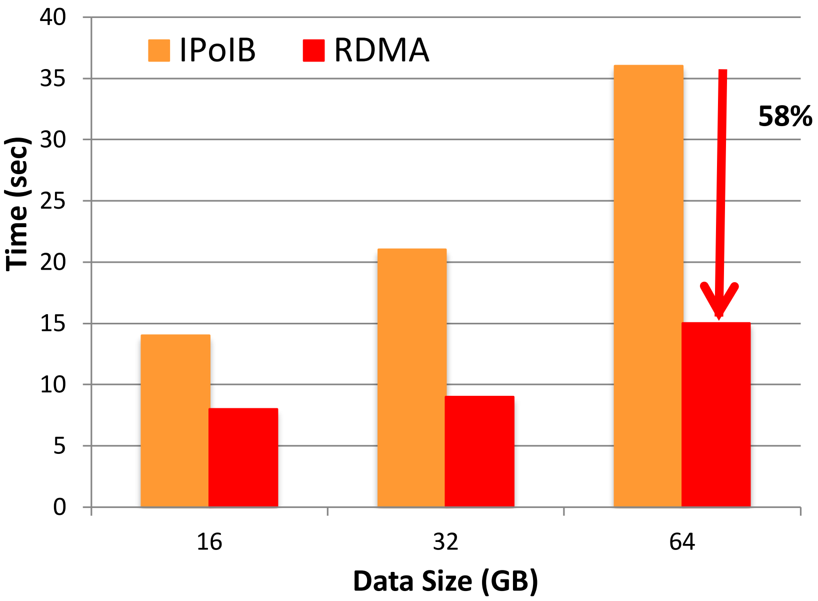
16 Worker Nodes, 256 Cores, (256M 256R)

- Intel SandyBridge + FDR
- Cluster with 8 HDD Nodes, single disk per node, 128 concurrent tasks
  - up to 83% over IPoIB (56Gbps)
- Cluster with 16 HDD Nodes, single disk per node, 256 concurrent tasks
  - up to 79% over IPoIB (56Gbps)

Performance Evaluation on TACC Stampede - SortByTest



16 Worker Nodes, SortByTest Shuffle Time



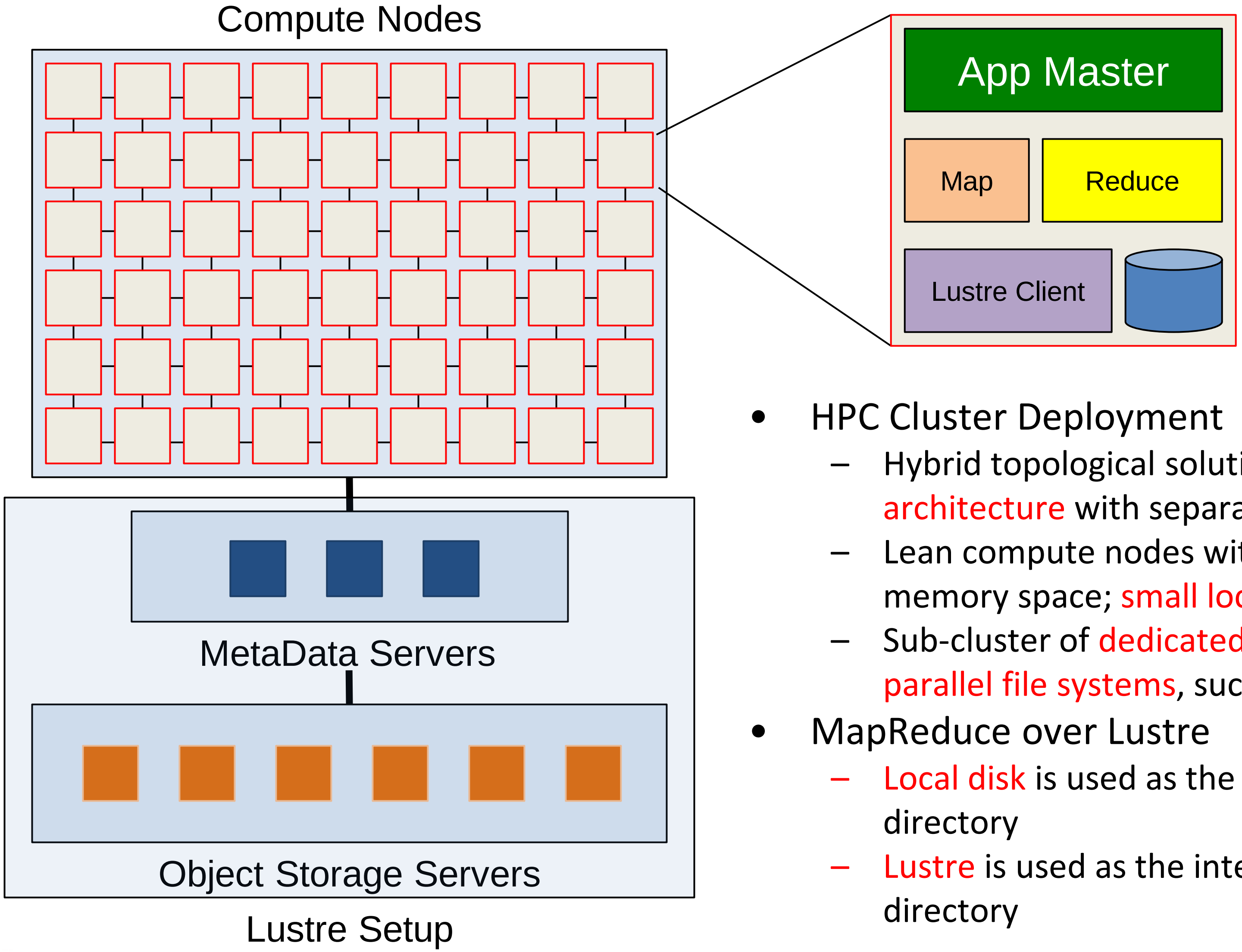
16 Worker Nodes, SortByTest Total Time

- Intel SandyBridge + FDR, 16 Worker Nodes, 256 Cores, (256M 256R)
- RDMA-based design for Spark 1.4.0
- RDMA vs. IPoIB with 256 concurrent tasks, single disk per node and RAMDisk. For SortByKey Test:
  - Shuffle time reduced by up to 77% over IPoIB (56Gbps)
  - Total time reduced by up to 58% over IPoIB (56Gbps)

Presentation Outline

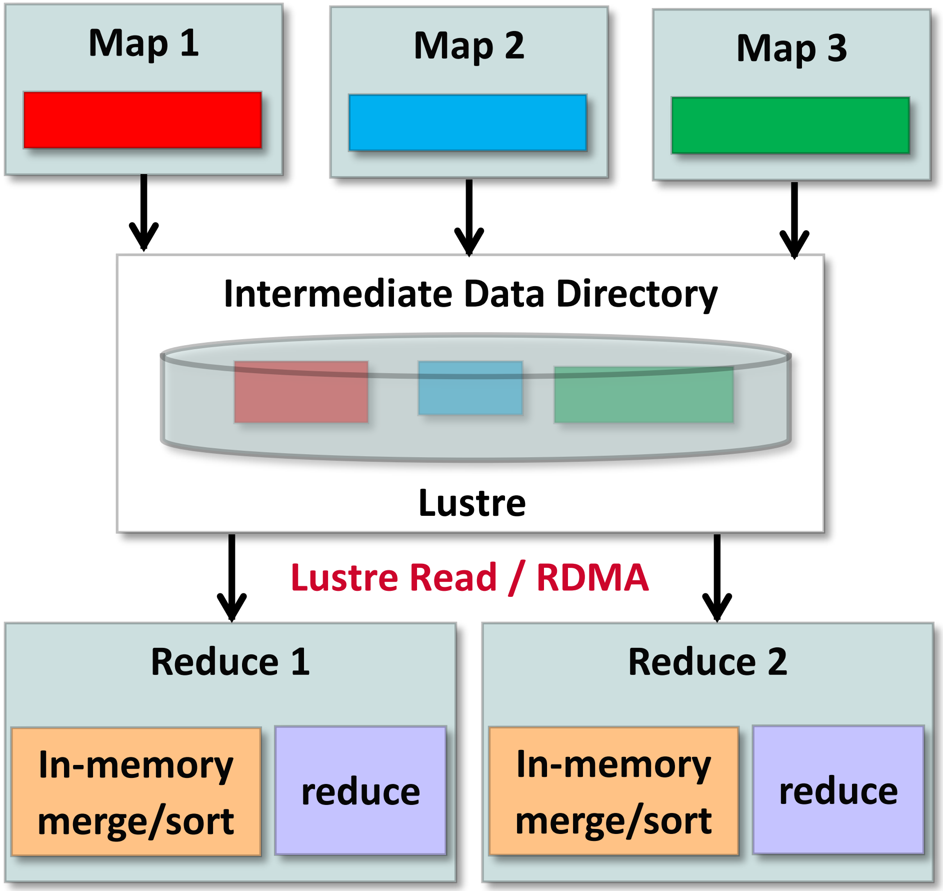
- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Optimize Hadoop YARN MapReduce over Parallel File Systems



- HPC Cluster Deployment
  - Hybrid topological solution of Beowulf architecture with separate I/O nodes
  - Lean compute nodes with light OS; more memory space; small local storage
  - Sub-cluster of dedicated I/O nodes with parallel file systems, such as Lustre
- MapReduce over Lustre
  - Local disk is used as the intermediate data directory
  - Lustre is used as the intermediate data directory

Design Overview of Shuffle Strategies for MapReduce over Lustre

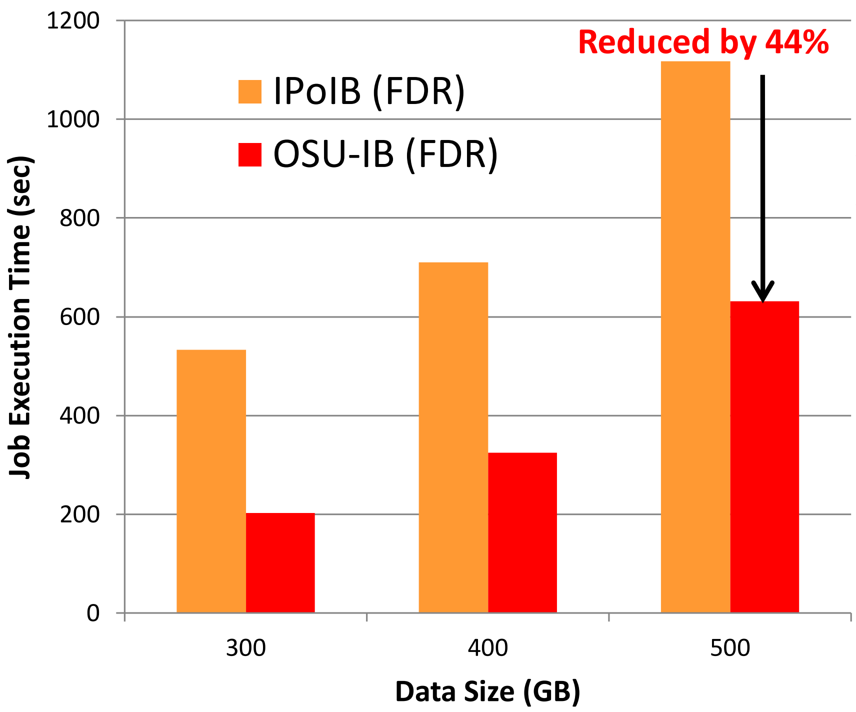


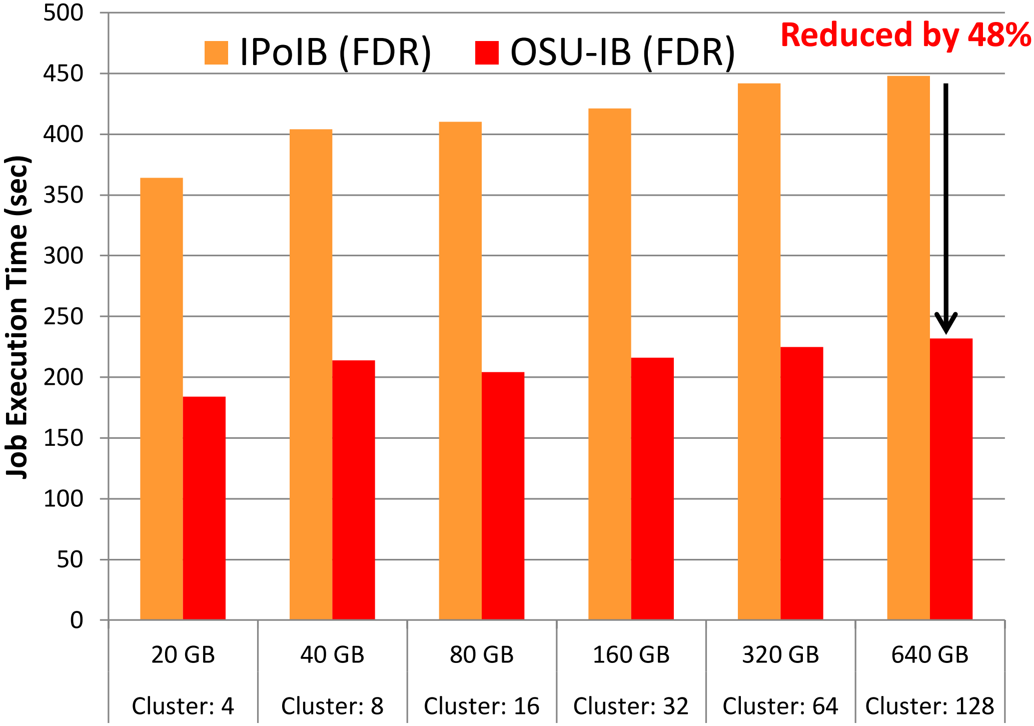
- Design Features
  - Two shuffle approaches
    - Lustre read based shuffle
    - RDMA based shuffle
  - Hybrid shuffle algorithm to take benefit from both shuffle approaches
  - Dynamically adapts to the better shuffle approach for each shuffle request based on profiling values for each Lustre read operation
  - In-memory merge and overlapping of different phases are kept similar to RDMA-enhanced MapReduce design

M. W. Rahman, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, High Performance Design of YARN MapReduce on Modern HPC Clusters with Lustre and RDMA, IPDPS, May 2015.

Performance Improvement of MapReduce over Lustre on TACC-Stampede

- Local disk is used as the intermediate data directory



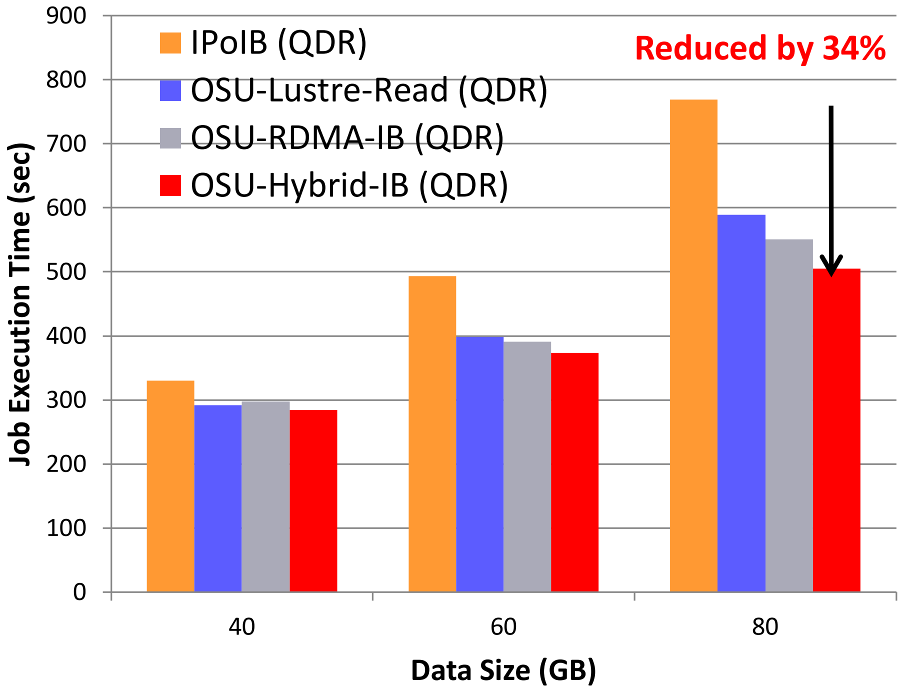


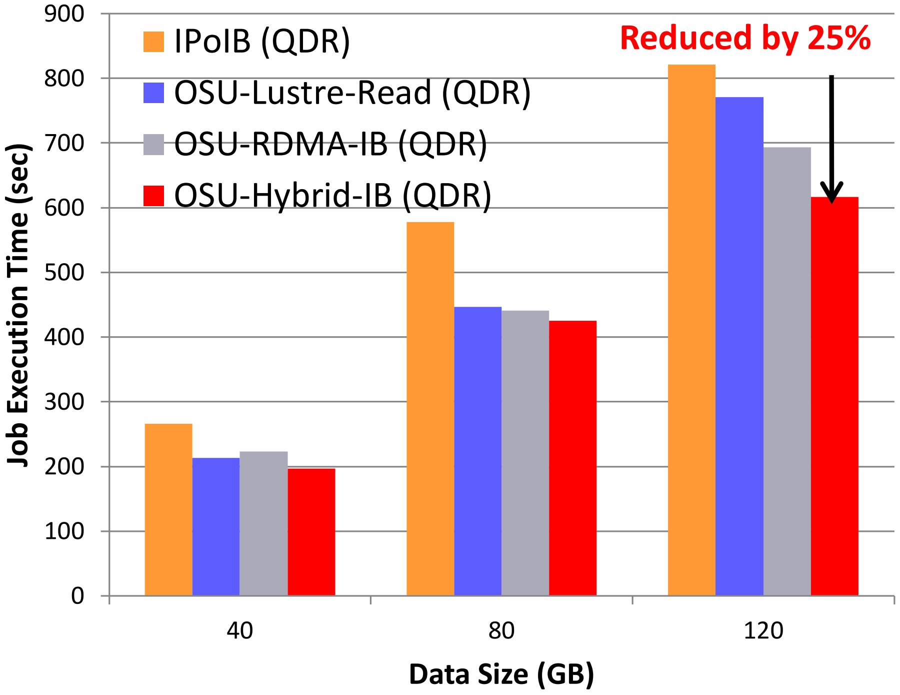
- For 500GB Sort in 64 nodes
  - 44% improvement over IPoIB (FDR)
- For 640GB Sort in 128 nodes
  - 48% improvement over IPoIB (FDR)

M. W. Rahman, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, MapReduce over Lustre: Can RDMA-based Approach Benefit?, Euro-Par, August 2014.

Case Study - Performance Improvement of MapReduce over Lustre on SDSC-Gordon

- Lustre is used as the intermediate data directory





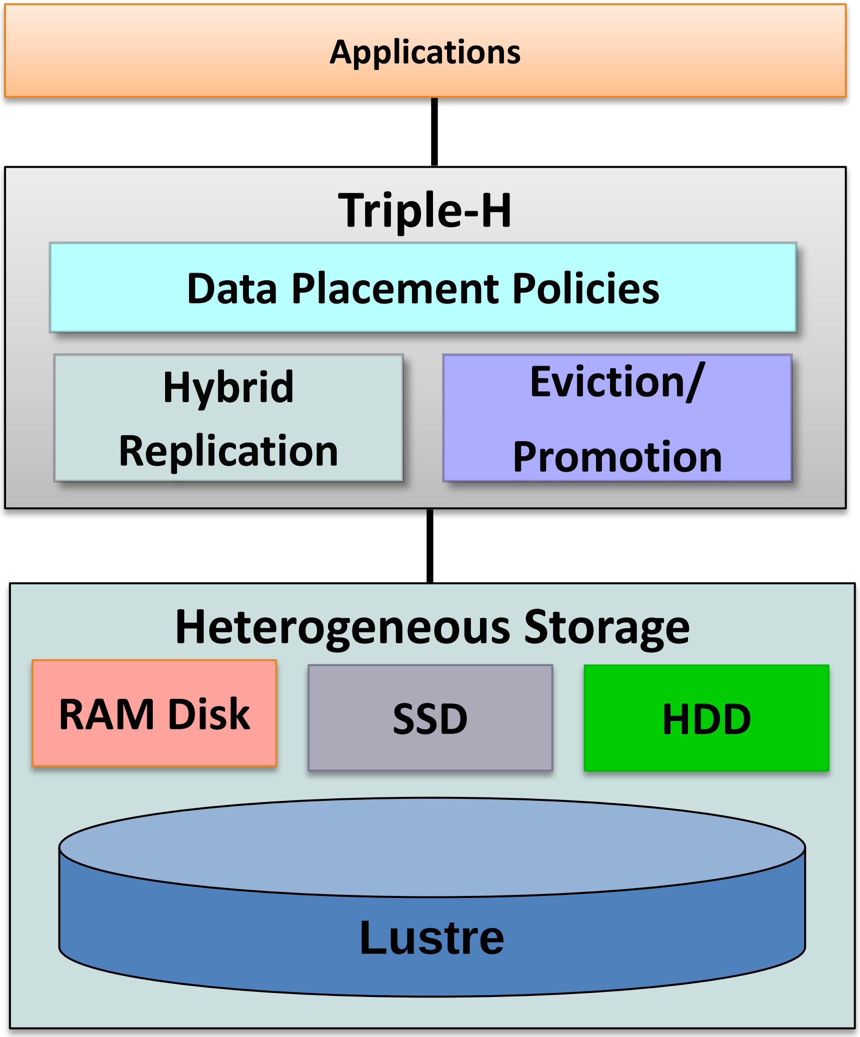
- For 80GB Sort in 8 nodes
  - 34% improvement over IPoIB (QDR)

- For 120GB TeraSort in 16 nodes
  - 25% improvement over IPoIB (QDR)

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

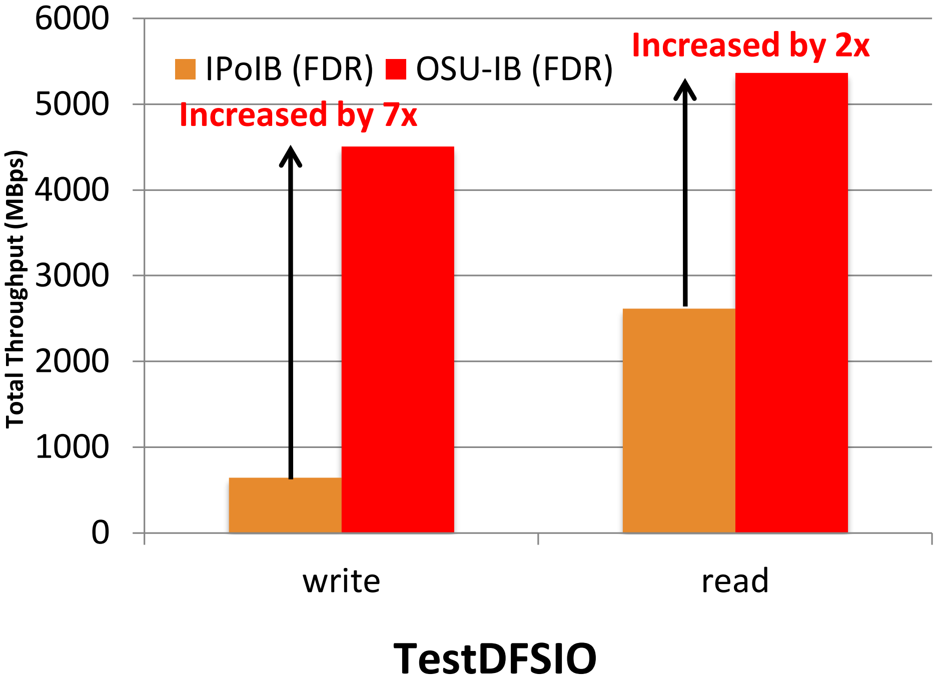
Enhanced HDFS with In-memory and Heterogeneous Storage

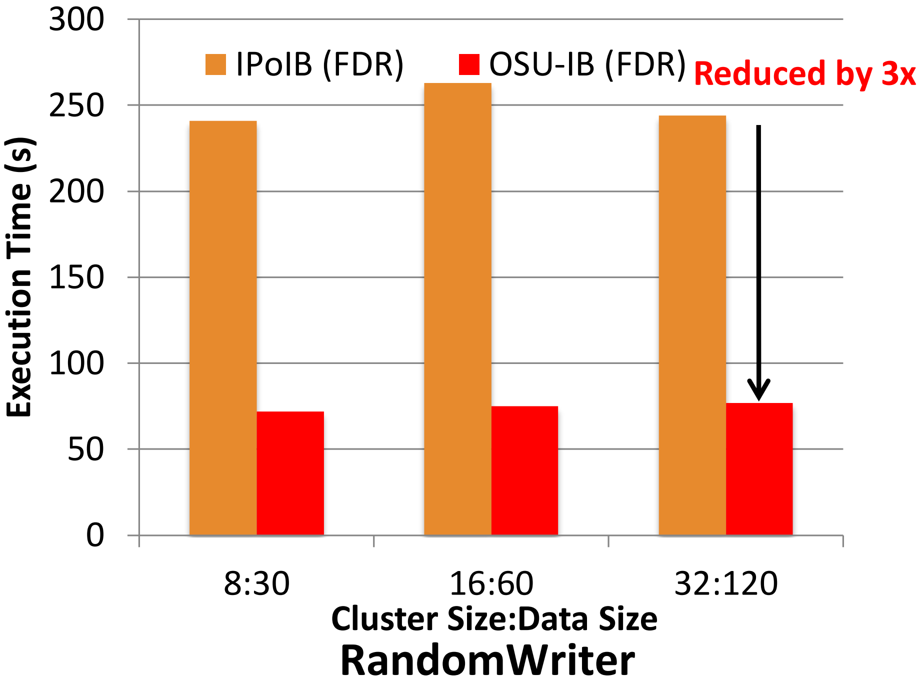


- Design Features
  - Two modes
    - Standalone
    - Lustre-Integrated
  - Policies to efficiently utilize the heterogeneous storage devices
    - RAM, SSD, HDD, Lustre
  - Eviction/Promotion based on data usage pattern
  - Hybrid Replication
  - Lustre-Integrated mode:
    - Lustre-based fault-tolerance

N. Islam, X. Lu, M. W. Rahman, D. Shankar, and D. K. Panda, Triple-H: A Hybrid Approach to Accelerate HDFS on HPC Clusters with Heterogeneous Storage Architecture, CCGrid '15, May 2015

Performance Improvement on TACC Stampede (HHH)

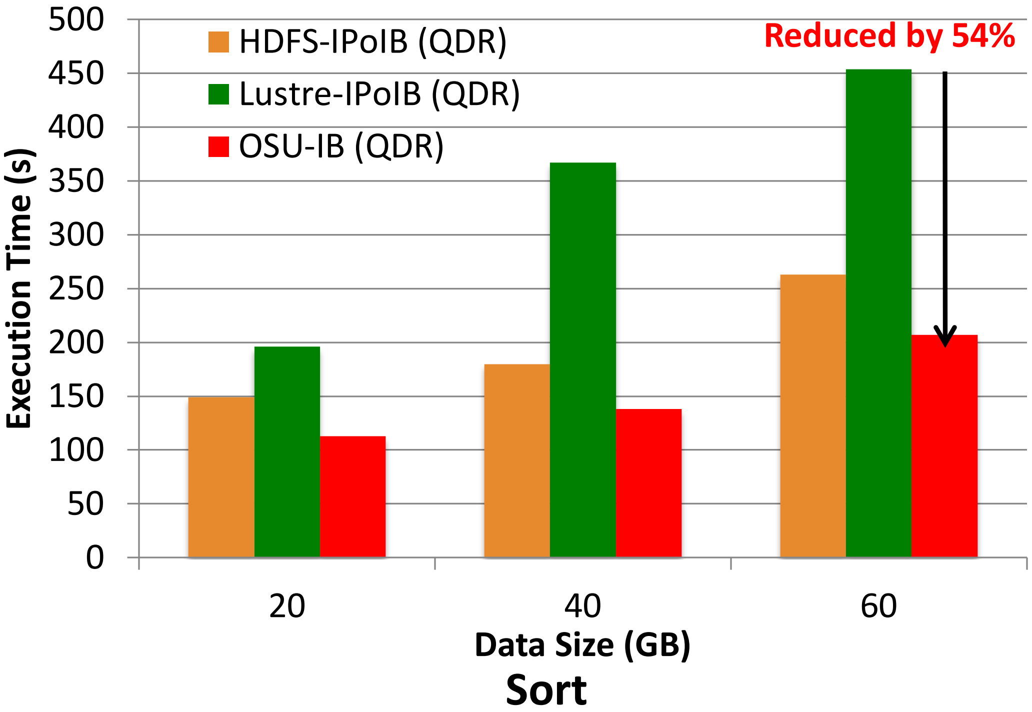




- For 160GB TestDFSIO in 32 nodes
  - Write Throughput: 7x improvement over IPoIB (FDR)
  - Read Throughput: 2x improvement over IPoIB (FDR)

- For 120GB RandomWriter in 32 nodes
  - 3x improvement over IPoIB (QDR)

Performance Improvement on SDSC Gordon (HHH-L)



Sort

Storage Used (GB)	
HDFS-IPoIB (QDR)	360
Lustre-IPoIB (QDR)	120
OSU-IB (QDR)	240

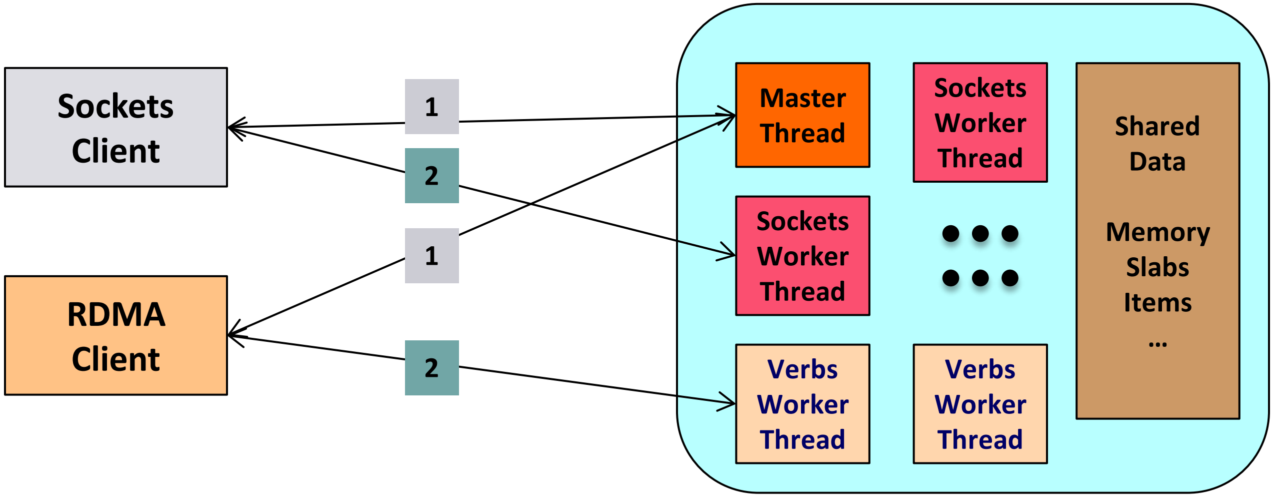
Storage space for 60GB Sort

- For 60GB Sort in 8 nodes
  - 24% improvement over default HDFS
  - 54% improvement over Lustre
  - 33% storage space saving compared to default HDFS

Presentation Outline

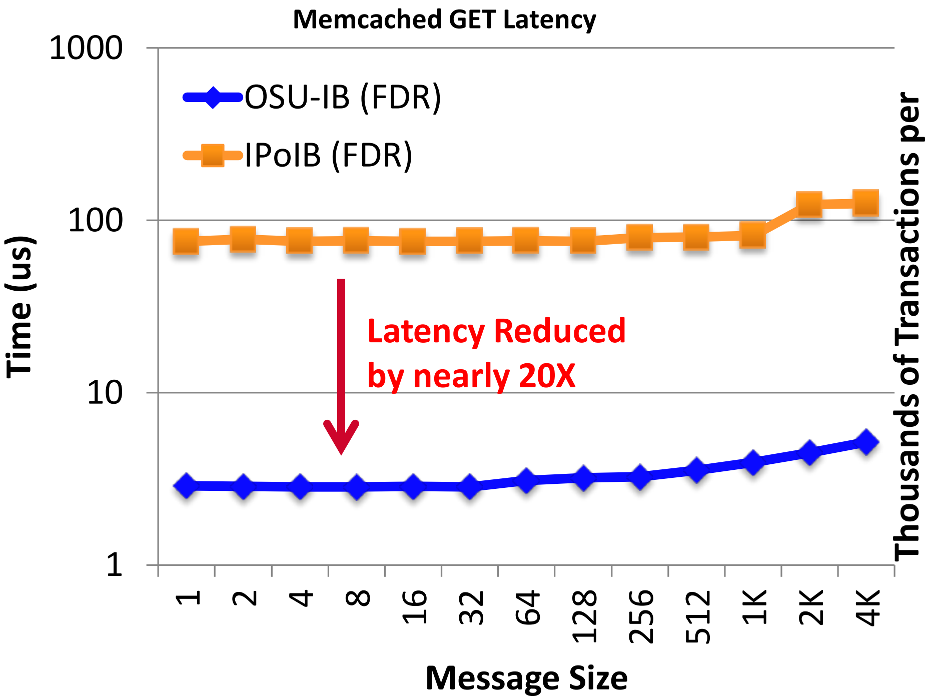
- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

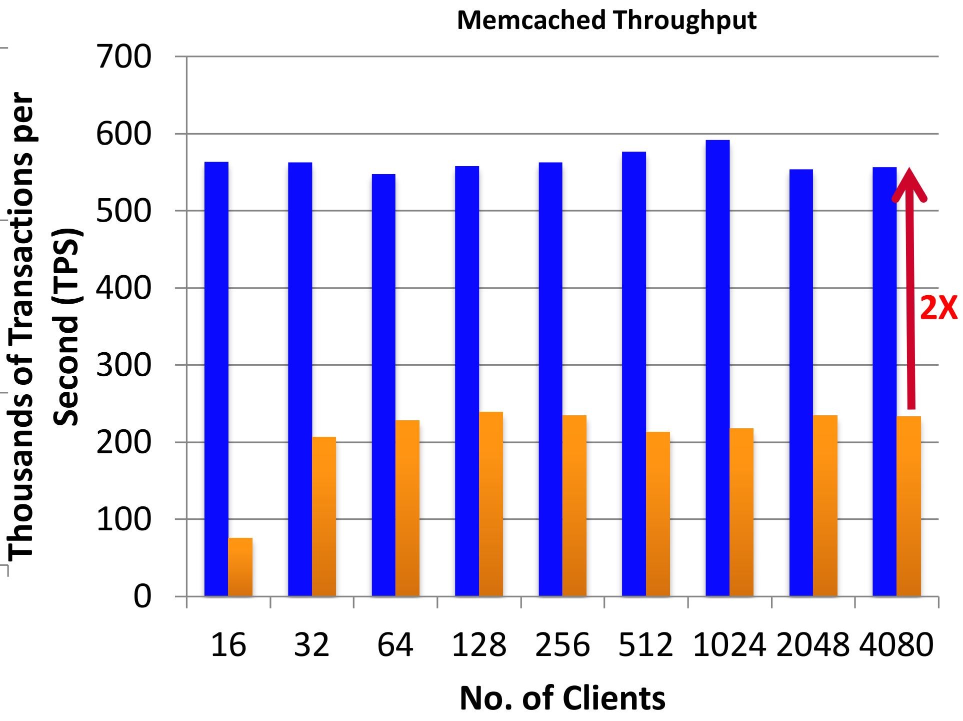
Memcached-RDMA Design



- Server and client perform a negotiation protocol
  - Master thread assigns clients to appropriate worker thread
- Once a client is assigned a verbs worker thread, it can communicate directly and is "bound" to that thread
- All other Memcached data structures are shared among RDMA and Sockets worker threads
- Native IB-verbs-level Design and evaluation with
  - Server : Memcached (http://memcached.org)
  - Client : libmemcached (http://libmemcached.org)
  - Different networks and protocols: 10GigE, IPoIB, native IB (RC, UD)

Memcached Performance (FDR Interconnect)

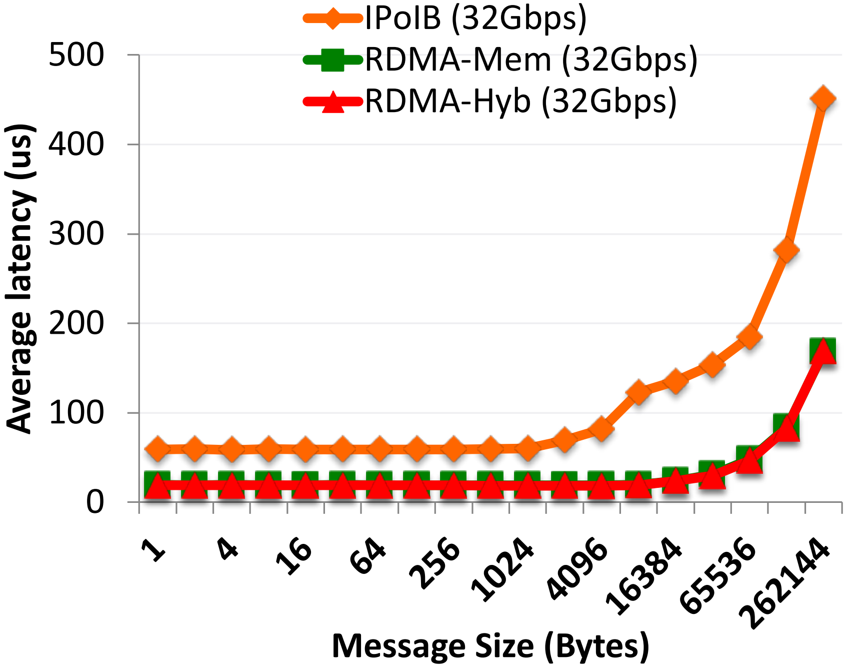


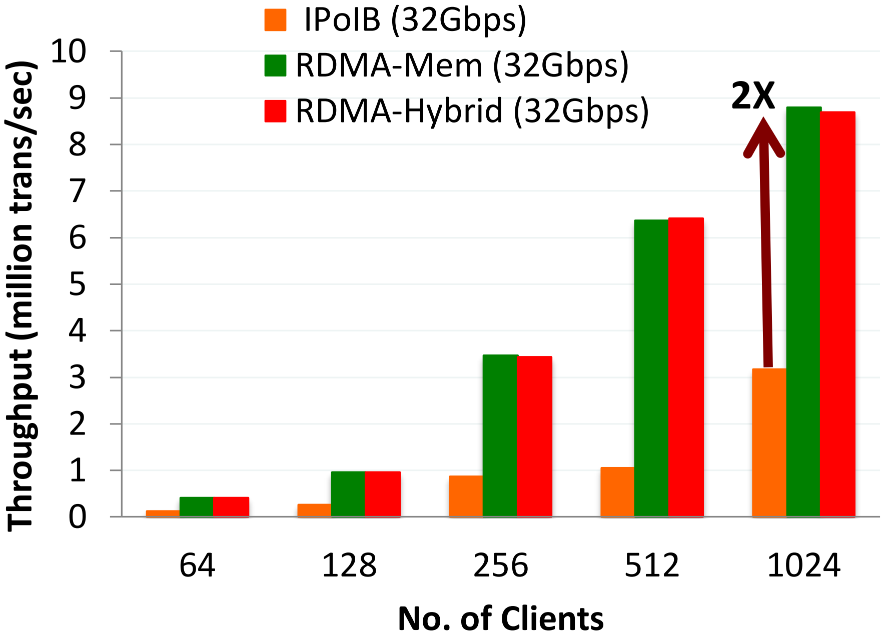


Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- Memcached Get latency
  - 4 bytes OSU-IB: 2.84 us; IPoIB: 75.53 us
  - 2K bytes OSU-IB: 4.49 us; IPoIB: 123.42 us
- Memcached Throughput (4bytes)
  - 4080 clients OSU-IB: 556 Kops/sec, IPoIB: 233 Kops/s
  - Nearly 2X improvement in throughput

Performance Benefits on SDSC-Gordon – OHB Latency & Throughput Micro-Benchmarks



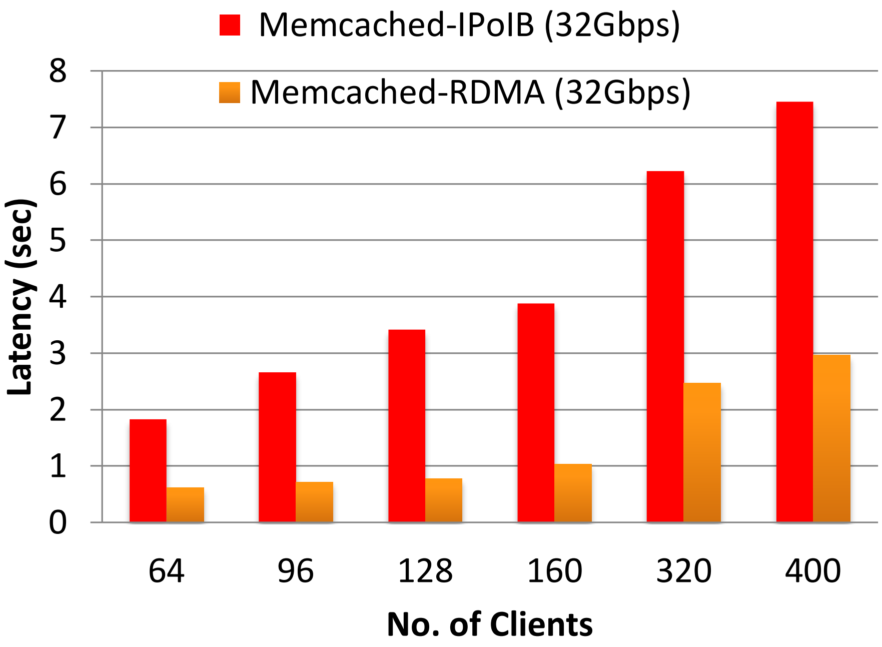


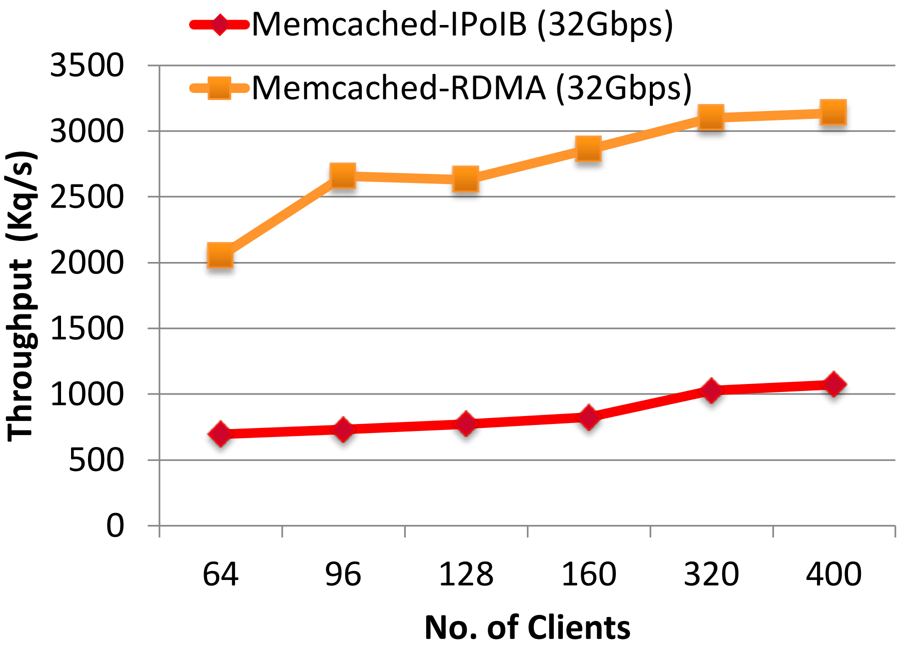
- ohb_memlat & ohb_memthr latency & throughput micro-benchmarks
- Memcached-RDMA can
  - improve query latency by up to 70% over IPoIB (32Gbps)
  - improve throughput by up to 2X over IPoIB (32Gbps)
  - No overhead in using hybrid mode when all data can fit in memory

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- On-going and Future Activities
- Conclusion and Q&A

Micro-benchmark Evaluation for OLDP workloads

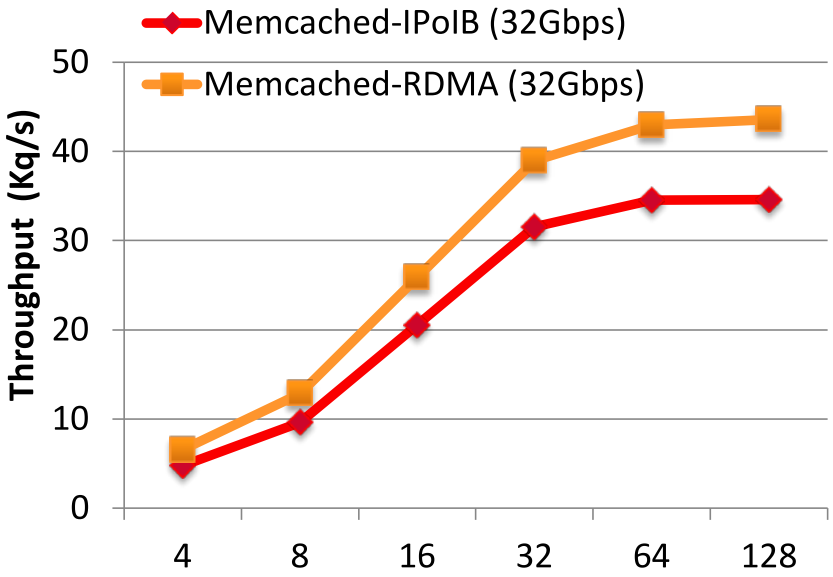




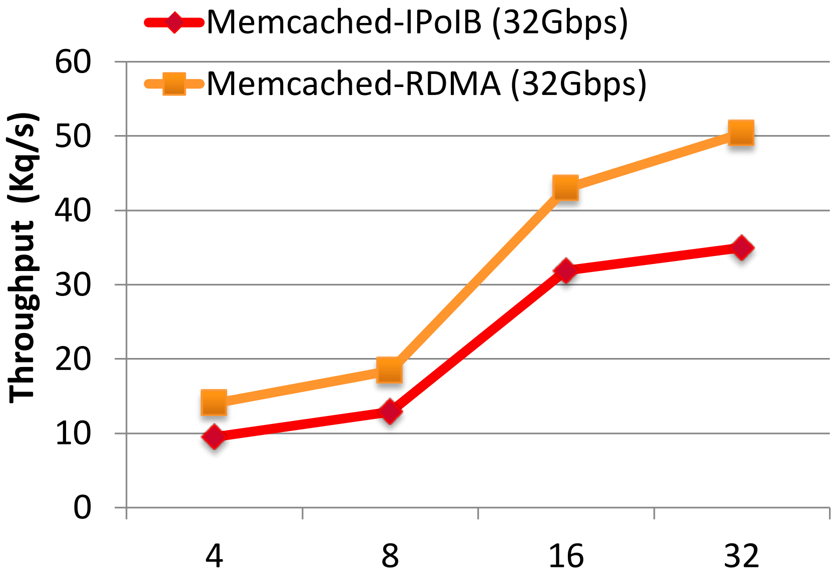
- Illustration with Read-Cache-Read access pattern using modified mysqlslap load testing tool
- Memcached-RDMA can
  - improve query latency by up to 66% over IPoIB (32Gbps)
  - throughput by up to 69% over IPoIB (32Gbps)

D. Shankar, X. Lu, J. Jose, M. W. Rahman, N. Islam, and D. K. Panda, Can RDMA Benefit On-Line Data Processing Workloads with Memcached and MySQL, ISPASS'15

Evaluation with Transactional and Web-oriented Workloads



Evaluation with TATP workload using modified OLTP-Bench



Evaluation with Twitter Workload using modified OLTP-Bench

Transactional workloads. Example: TATP

- Up to 29% improvement in overall throughput as compared to default Memcached running over IPoIB

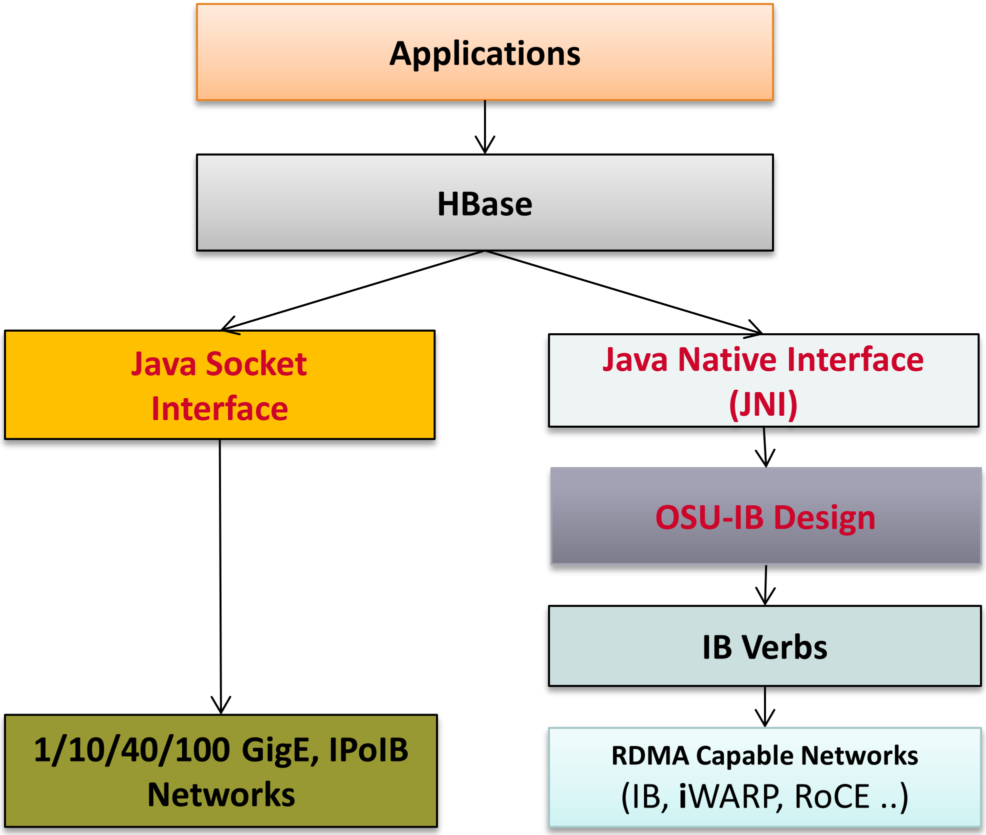
Web-Oriented workloads. Example: Twitter workload

- Up to 42% improvement in overall throughput compared to default Memcached running over IPoIB

Presentation Outline

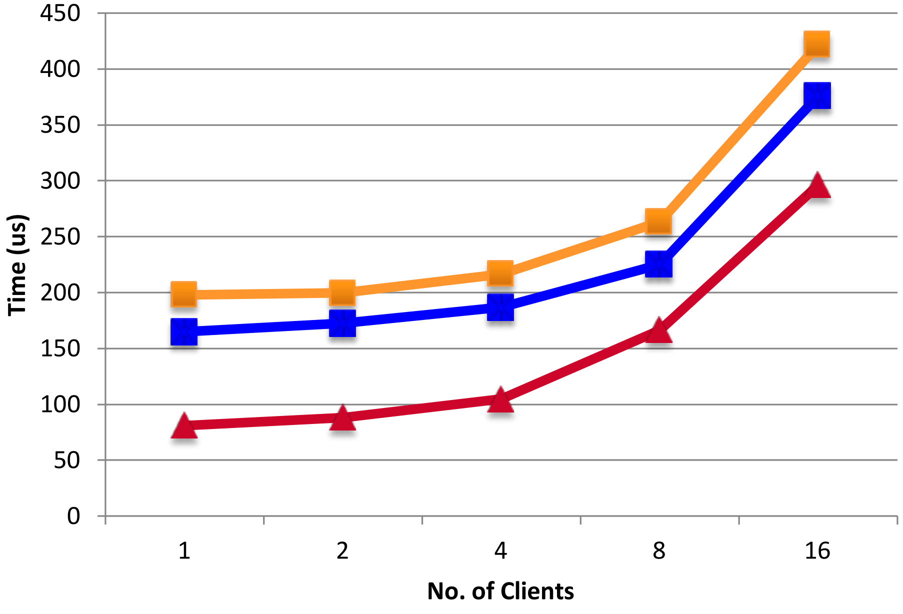
- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

HBase-RDMA Design Overview

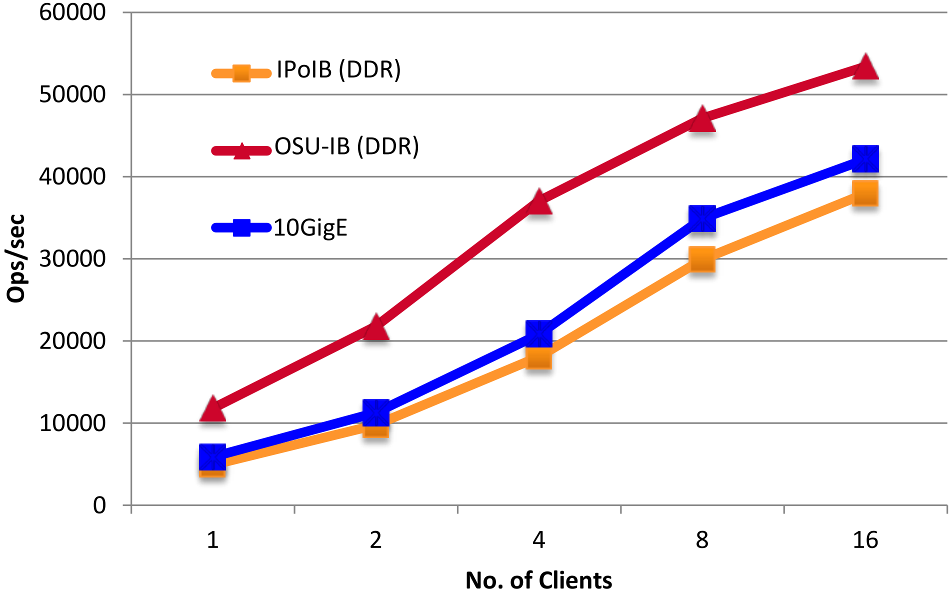


- JNI Layer bridges Java based HBase with communication library written in native code
- Enables high performance RDMA communication, while supporting traditional socket interface

HBase Micro-benchmark (Single-Server-Multi-Client) Results



Latency

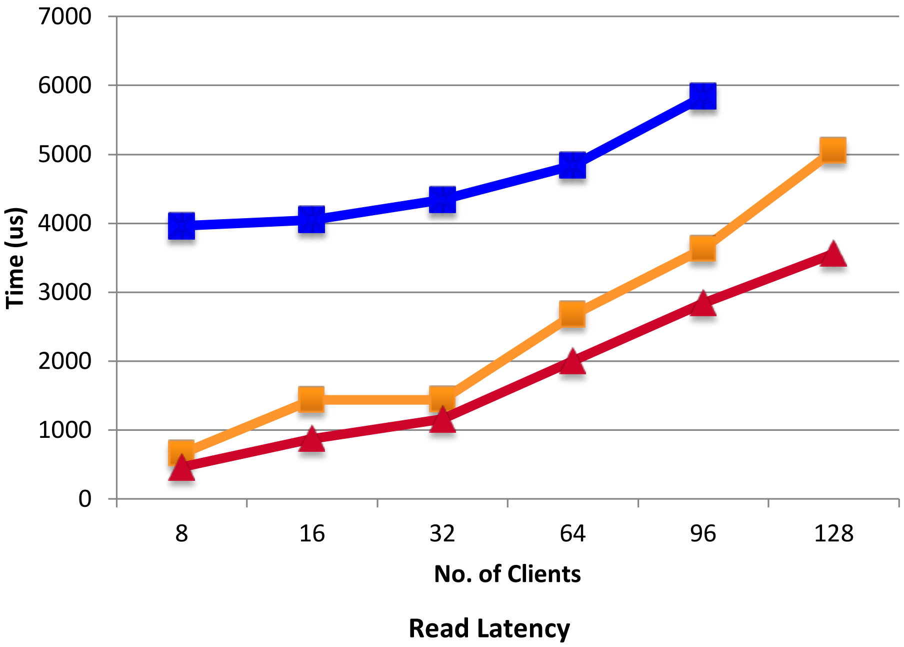


Throughput

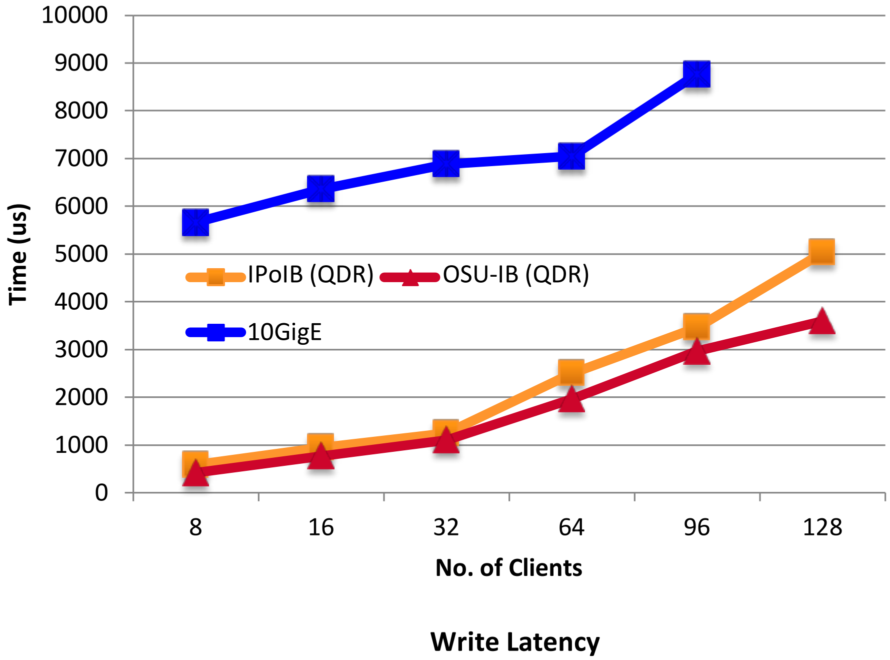
- HBase Get latency
  - 4 clients: 104.5 us; 16 clients: 296.1 us
- HBase Get throughput
  - 4 clients: 37.01 Kops/sec; 16 clients: 53.4 Kops/sec
- 27% improvement in throughput for 16 clients over 10GE

J. Huang, X. Ouyang, J. Jose, M. W. Rahman, H. Wang, M. Luo, H. Subramoni, Chet Murthy, and D. K. Panda, High-Performance Design of HBase with RDMA over InfiniBand, IPDPS’12

HBase – YCSB Read-Write Workload



Read Latency



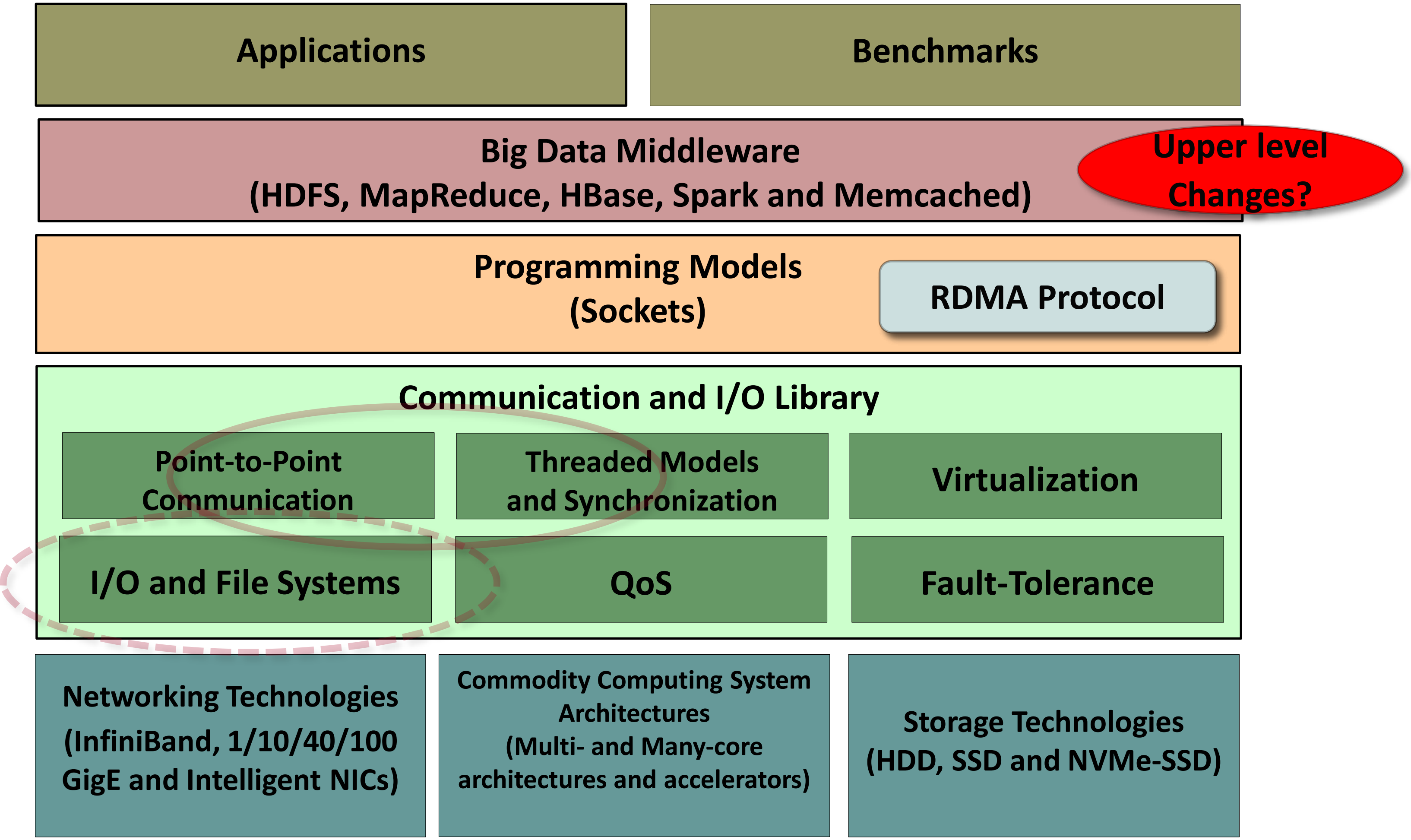
Write Latency

- HBase Get latency (Yahoo! Cloud Service Benchmark)
  - 64 clients: 2.0 ms; 128 Clients: 3.5 ms
  - 42% improvement over IPoIB for 128 clients
- HBase Put latency
  - 64 clients: 1.9 ms; 128 Clients: 3.5 ms
  - 40% improvement over IPoIB for 128 clients

Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
  - Enhanced HDFS with In-memory and Heterogeneous Storage
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with Hybrid Memory
  - Case study with OLDP
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Designing Communication and I/O Libraries for Big Data Systems: Solved a Few Initial Challenges



Applications

Benchmarks

Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)

Upper level Changes?

Programming Models
(Sockets)

RDMA Protocol

Communication and I/O Library

Point-to-Point Communication

Threaded Models and Synchronization

Virtualization

I/O and File Systems

QoS

Fault-Tolerance

Networking Technologies
(InfiniBand, 1/10/40/100 GigE and Intelligent NICs)

Commodity Computing System Architectures
(Multi- and Many-core architectures and accelerators)

Storage Technologies
(HDD, SSD and NVMe-SSD)

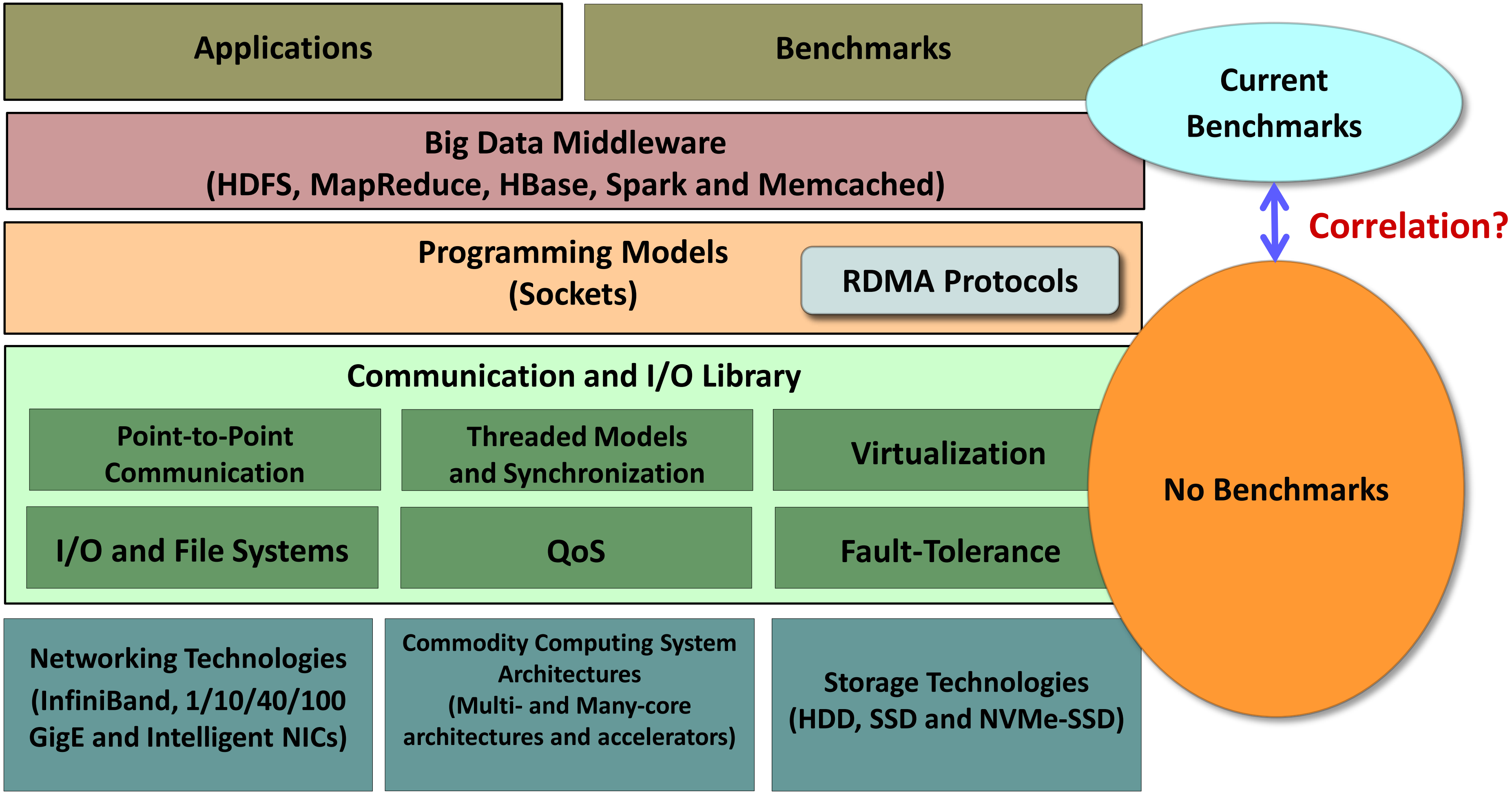
Are the Current Benchmarks Sufficient for Big Data Management and Processing?

- The current benchmarks provide some performance behavior
- However, do not provide any information to the designer/developer on:
  - What is happening at the lower-layer?
  - Where the benefits are coming from?
  - Which design is leading to benefits or bottlenecks?
  - Which component in the design needs to be changed and what will be its impact?
  - Can performance gain/loss at the lower-layer be correlated to the performance gain/loss observed at the upper layer?

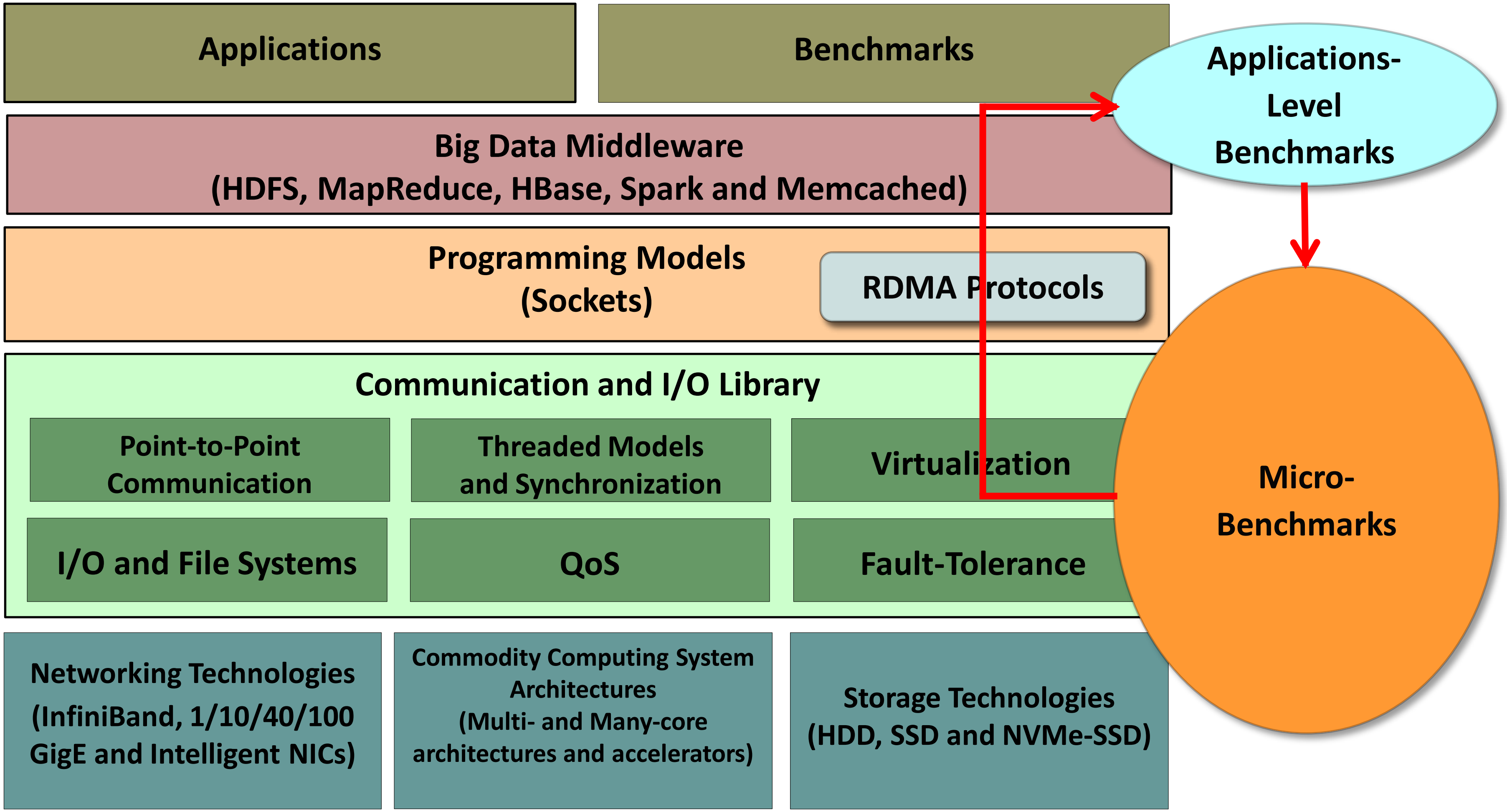
OSU MPI Micro-Benchmarks (OMB) Suite

- A comprehensive suite of benchmarks to
  - Compare performance of different MPI libraries on various networks and systems
  - Validate low-level functionalities
  - Provide insights to the underlying MPI-level designs
- Started with basic send-recv (MPI-1) micro-benchmarks for latency, bandwidth and bi-directional bandwidth
- Extended later to
  - MPI-2 one-sided
  - Collectives
  - GPU-aware data movement
  - OpenSHMEM (point-to-point and collectives)
  - UPC
- Has become an industry standard
- Extensively used for design/development of MPI libraries, performance comparison of MPI libraries and even in procurement of large-scale systems
- Available from http://mvapich.cse.ohio-state.edu/benchmarks
- Available in an integrated manner with MVAPICH2 stack

Challenges in Benchmarking of RDMA-based Designs


Applications
Benchmarks
Current Benchmarks
Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)
Correlation?
Programming Models
(Sockets)
RDMA Protocols
Communication and I/O Library
Point-to-Point Communication
Threaded Models and Synchronization
Virtualization
No Benchmarks
I/O and File Systems
QoS
Fault-Tolerance
Networking Technologies (InfiniBand, 1/10/40/100 GigE and Intelligent NICs)
Commodity Computing System Architectures (Multi- and Many-core architectures and accelerators)
Storage Technologies (HDD, SSD and NVMe-SSD)

Iterative Process – Requires Deeper Investigation and Design for Benchmarking Next Generation Big Data Systems and Applications



Applications
Benchmarks
Applications-Level Benchmarks
Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)
Programming Models
(Sockets)
RDMA Protocols
Communication and I/O Library
Point-to-Point Communication
Threaded Models and Synchronization
Virtualization
I/O and File Systems
QoS
Fault-Tolerance
Micro-Benchmarks
Networking Technologies (InfiniBand, 1/10/40/100 GigE and Intelligent NICs)
Commodity Computing System Architectures (Multi- and Many-core architectures and accelerators)
Storage Technologies (HDD, SSD and NVMe-SSD)

OSU HiBD Micro-Benchmark (OHB) Suite - HDFS

- Evaluate the performance of standalone HDFS
- Five different benchmarks
  - Sequential Write Latency (**SWL**)
  - Sequential or Random Read Latency (**SRL** or **RRL**)
  - Sequential Write Throughput (**SWT**)
  - Sequential Read Throughput (**SRT**)
  - Sequential Read-Write Throughput (**SRWT**)

N. S. Islam, X. Lu, M. W. Rahman, J. Jose, and D. K. Panda, A Micro-benchmark Suite for Evaluating HDFS Operations on Modern Clusters, Int'l Workshop on Big Data Benchmarking (WBDB '12), December 2012.

Benchmark	File Name	File Size	HDFS Parameter	Readers	Writers	Random/ Sequential Read	Seek Interval
SWL	√	√	√				
SRL/RRL	√	√	√			√	√ (RRL)
SWT		√	√		√		
SRT		√	√	√			
SRWT		√	√	√	√		

OSU HiBD Micro-Benchmark (OHB) Suite - RPC

- Two different micro-benchmarks to evaluate the performance of standalone Hadoop RPC
  - Latency: Single Server, Single Client
  - Throughput: Single Server, Multiple Clients
- A simple script framework for job launching and resource monitoring
- Calculates statistics like Min, Max, Average
- Network configuration, Tunable parameters, DataType, CPU Utilization

Component	Network Address	Port	Data Type	Min Msg Size	Max Msg Size	No. of Iterations	Handlers	Verbose
lat_client	√	√	√	√	√	√		√
lat_server	√	√					√	√

Component	Network Address	Port	Data Type	Min Msg Size	Max Msg Size	No. of Iterations	No. of Clients	Handlers	Verbose
thr_client	√	√	√	√	√	√			√
thr_server	√	√			√		√	√	√

X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, A Micro-Benchmark Suite for Evaluating Hadoop RPC on High-Performance Networks, Int'l Workshop on Big Data Benchmarking (WBDB '13), July 2013.

OSU HiBD Micro-Benchmark (OHB) Suite - MapReduce

- Evaluate the performance of stand-alone MapReduce
- Does not require or involve HDFS or any other distributed file system
- Considers various factors that influence the data shuffling phase
  - underlying network configuration, number of map and reduce tasks, intermediate shuffle data pattern, shuffle data size etc.
- Three different micro-benchmarks based on intermediate shuffle data patterns
  - **MR-AVG micro-benchmark:** intermediate data is evenly distributed among reduce tasks.
  - **MR-RAND micro-benchmark:** intermediate data is pseudo-randomly distributed among reduce tasks.
  - **MR-SKEW micro-benchmark:** intermediate data is unevenly distributed among reduce tasks.

D. Shankar, X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, A Micro-Benchmark Suite for Evaluating Hadoop MapReduce on High-Performance Networks, BPOE-5 (2014).

Upcoming HiBD Releases and Future Activities

- Upcoming Releases of RDMA-enhanced Packages will support
  - CDH plugin
  - Spark
  - HBase
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  - MapReduce, RPC
- Exploration of other components (Threading models, QoS, Virtualization, Accelerators, etc.)
- Advanced designs with upper-level changes and optimizations

Concluding Remarks

- Presented an overview of Big Data processing middleware
- Provided an overview of cluster technologies
- Discussed challenges in accelerating Big Data middleware
- Presented initial designs to take advantage of InfiniBand/RDMA for Hadoop, Spark, Memcached, and HBase
- Presented challenges in designing benchmarks
- Results are promising
- Many other open issues need to be solved
- Will enable Big processing community to take advantage of modern HPC technologies to carry out their analytics in a fast and scalable manner

Personnel Acknowledgments

Current Students

- A. Augustine (M.S.)
- A. Awan (Ph.D.)
- A. Bhat (M.S.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- N. Islam (Ph.D.)
- K. Kulkarni (M.S.)
- M. Li (Ph.D.)
- M. Rahman (Ph.D.)
- D. Shankar (Ph.D.)
- A. Venkatesh (Ph.D.)
- J. Zhang (Ph.D.)

Current Senior Research Associates

- K. Hamidouche
- X. Lu
- H. Subramoni

Current Post-Doc

- J. Lin
- D. Shankar

Current Programmer

- J. Perkins

Current Research Specialist

- M. Arnold

Past Students

- P. Balaji (Ph.D.)
- D. Buntinas (Ph.D.)
- S. Bhagvat (M.S.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)

Past Post-Docs

- H. Wang
- X. Besseron
- H.-W. Jin
- M. Luo
- E. Mancini
- S. Marcarelli
- J. Vienne

Past Research Scientist

- S. Sur

Past Programmers

- D. Bureddy

International Workshop on High-Performance Big Data Computing (HPBDC)

HPBDC 2015 was held with Int'l Conference on Distributed Computing Systems (ICDCS '15), Columbus, Ohio, USA, Monday, June 29th, 2015

Two Keynote Talks: Dan Stanzione (TACC) and Zhiwei Xu (ICT/CAS)
Two Invited Talks: Jianfeng Zhan (ICT/CAS), Raghunath Nambiar (Cisco)
Panel: Jianfeng Zhan (ICT/CAS)
Four Research Papers

http://web.cse.ohio-state.edu/~luxi/hpbdc2015

HPBDC 2016 will be held in conjunction with IPDPS '16

http://web.cse.ohio-state.edu/~luxi/hpbdc2016

Submission: January 2016

Thank You!

panda@cse.ohio-state.edu

Network Based Computing Laboratory

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

MVAPICH

HiBD
High-Performance
Big Data

The MVAPICH2/MVAPICH2-X Project
http://mvapich.cse.ohio-state.edu/

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/