Accelerating Big Data Processing with Hadoop, Spark and Memcached

Talk at HPC Advisory Council Switzerland Conference (Mar '15)

by

Dhabaleswar K. (DK) Panda

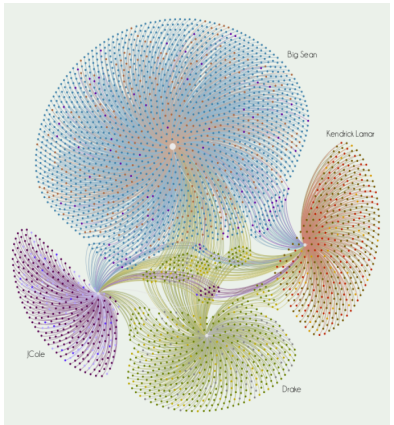
The Ohio State University

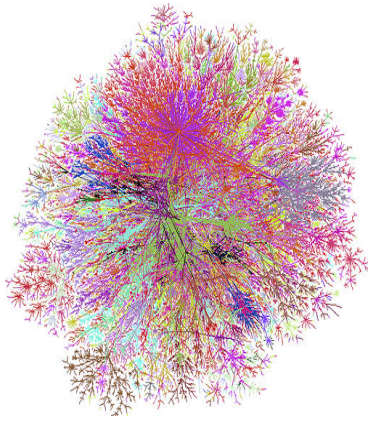
E-mail: panda@cse.ohio-state.edu

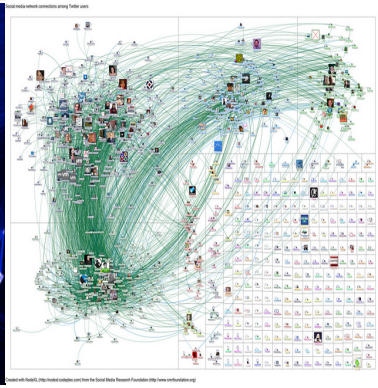
http://www.cse.ohio-state.edu/~panda

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





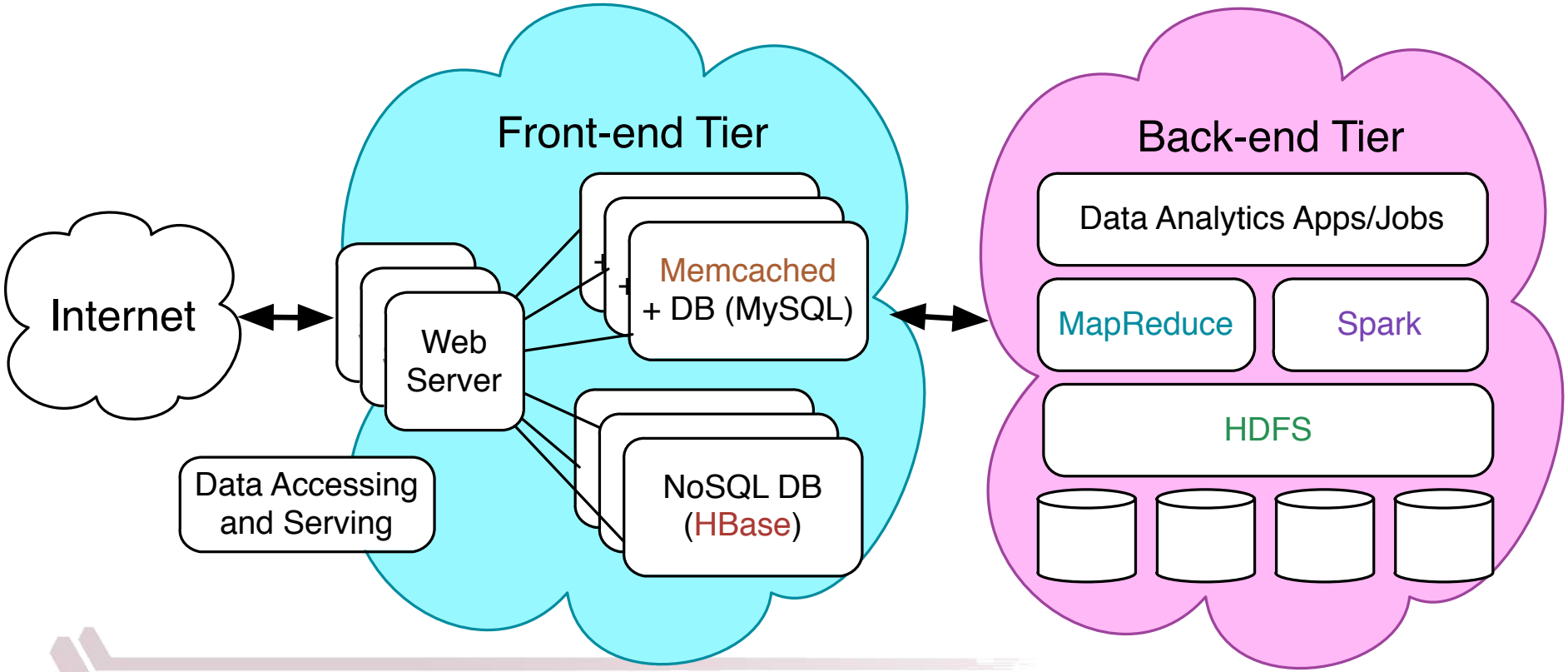


- Commonly accepted **3V**'s of Big Data
  - **Volume, Velocity, Variety**

  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**

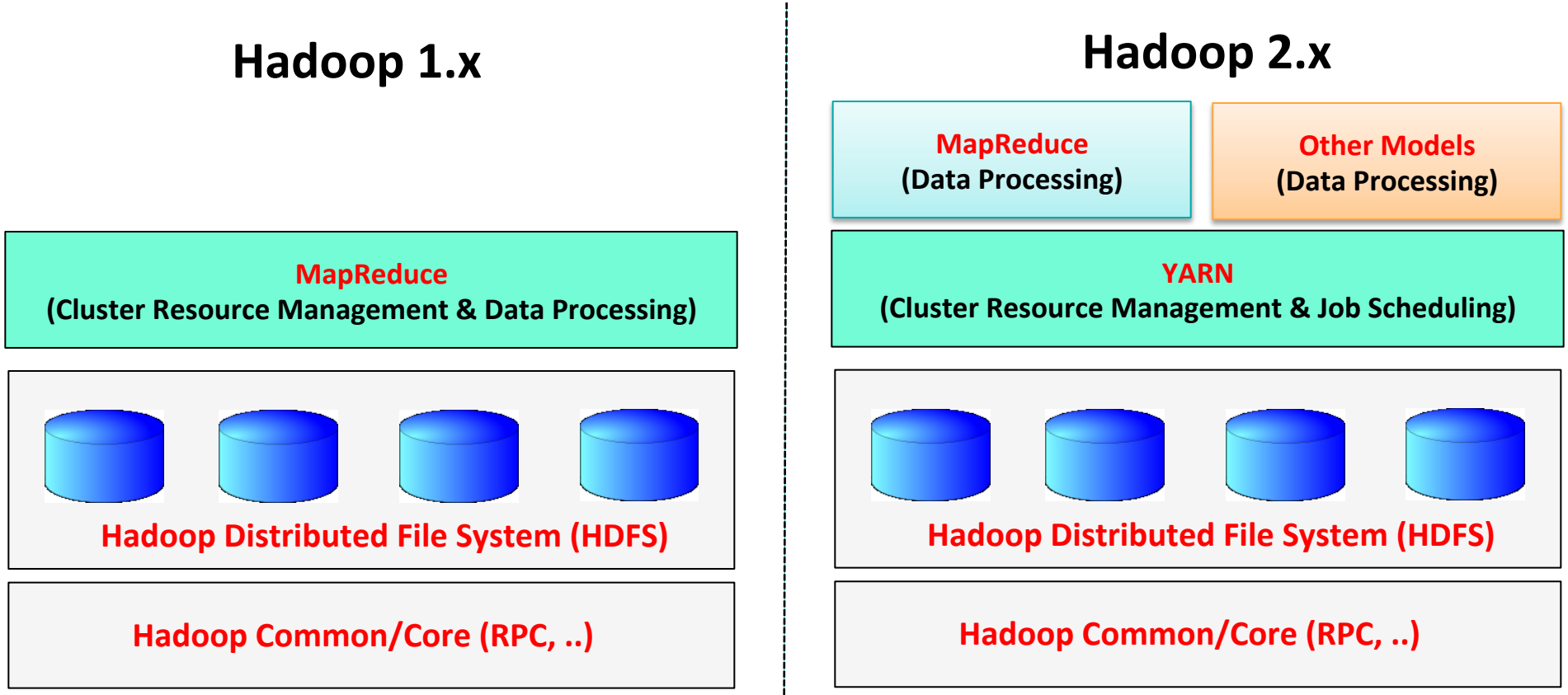
Data Management and Processing on Modern Clusters

- 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



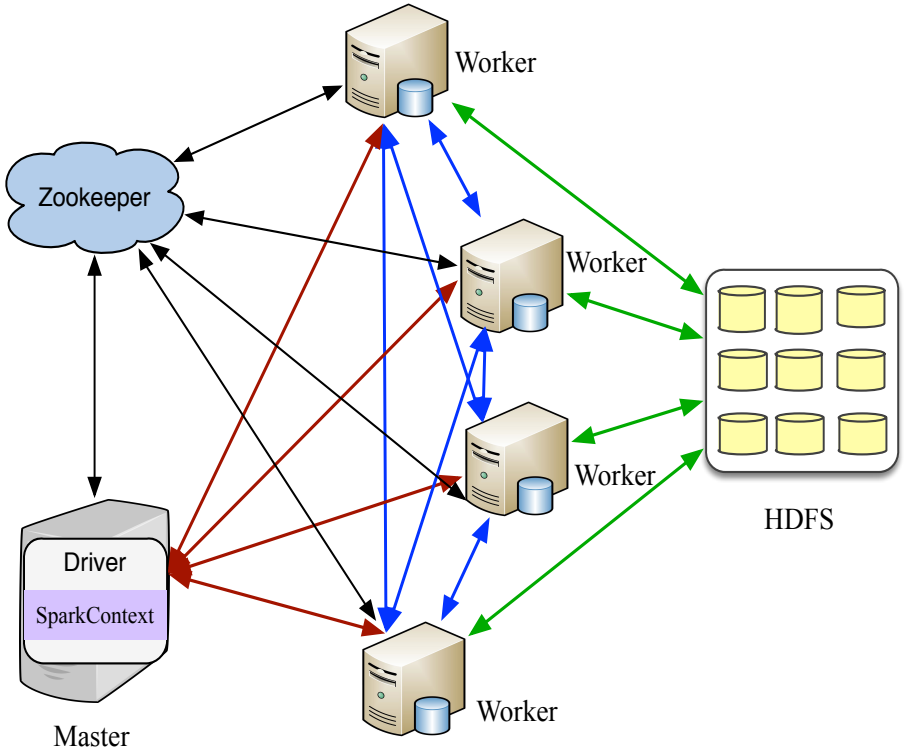
Overview of Apache Hadoop Architecture

- 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



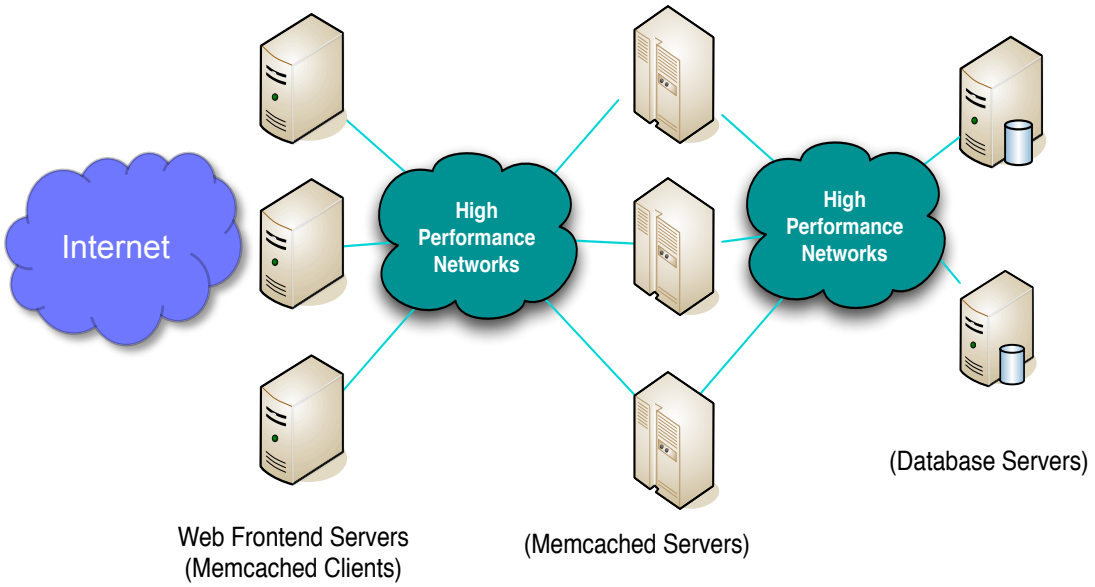
Spark Architecture Overview

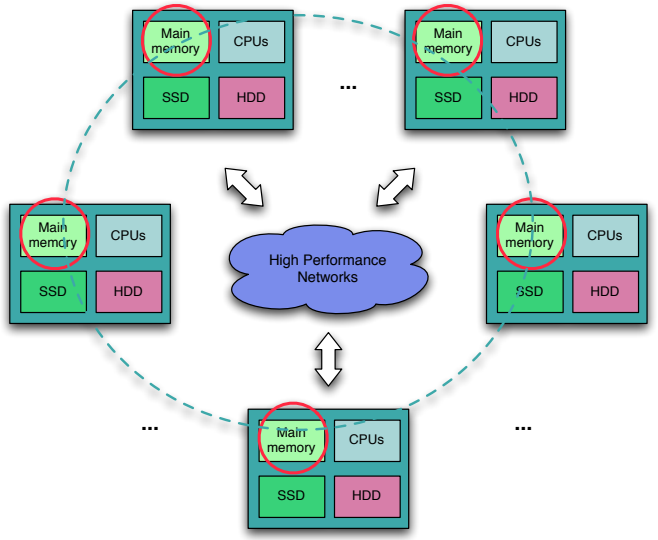
- An in-memory data-processing framework
  - Iterative machine learning jobs
  - Interactive data analytics
  - Scala based Implementation
  - Standalone, YARN, Mesos
- Scalable and communication intensive
  - Wide dependencies between Resilient Distributed Datasets (RDDs)
  - MapReduce-like shuffle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



http://spark.apache.org

Memcached Architecture





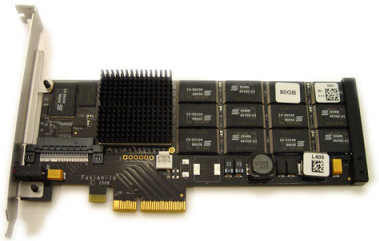
- Three-layer architecture of Web 2.0
  - Web Servers, Memcached Servers, Database Servers
- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

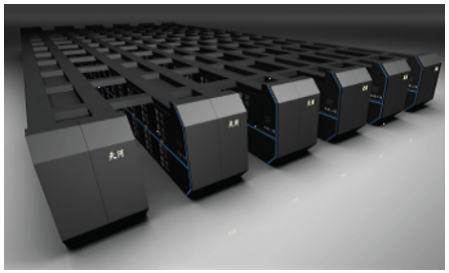
Presentation Outline

- 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
  - Sample Performance Numbers for RDMA-Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

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)



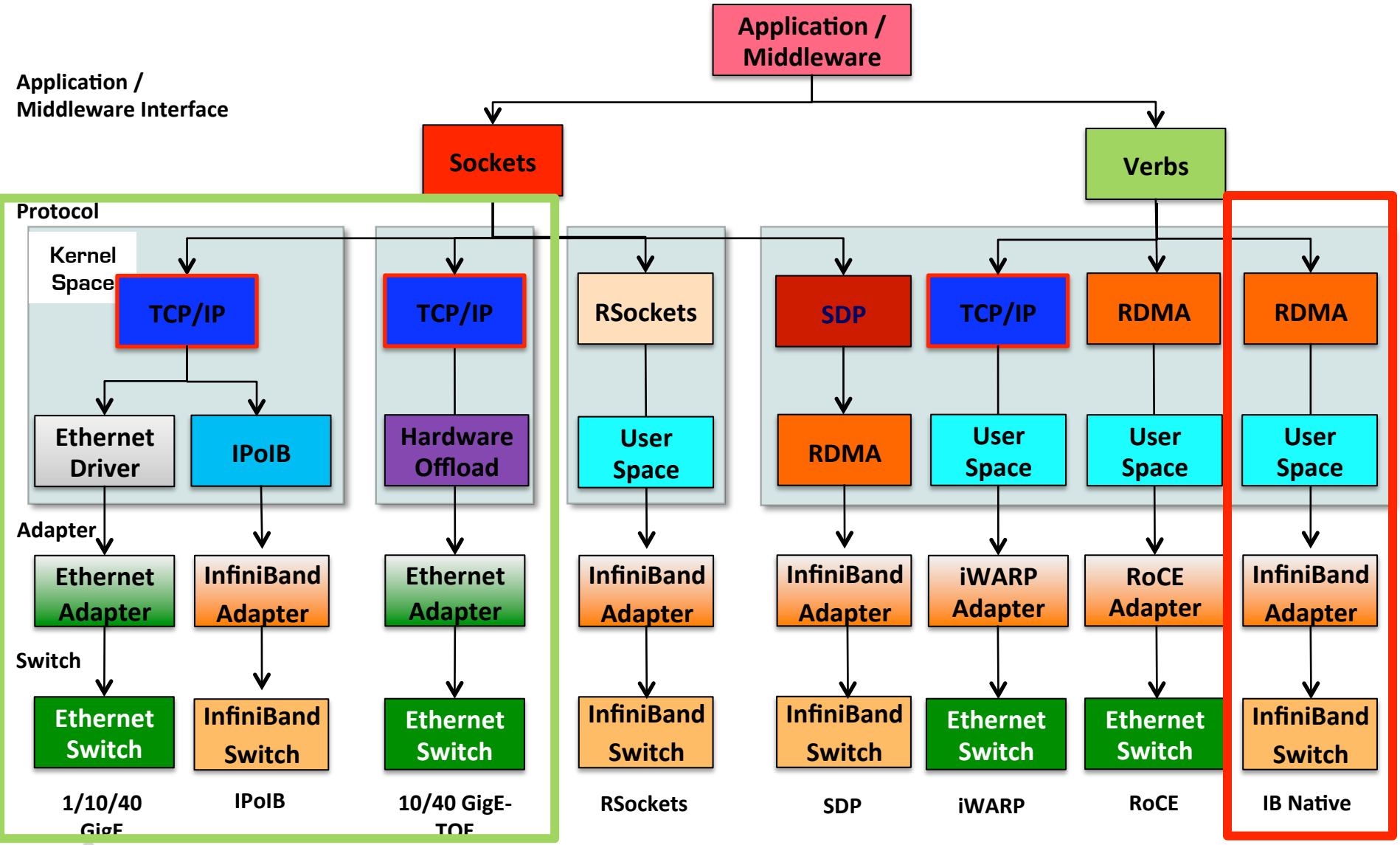


Tianhe – 2


Titan


Stampede


Tianhe – 1A

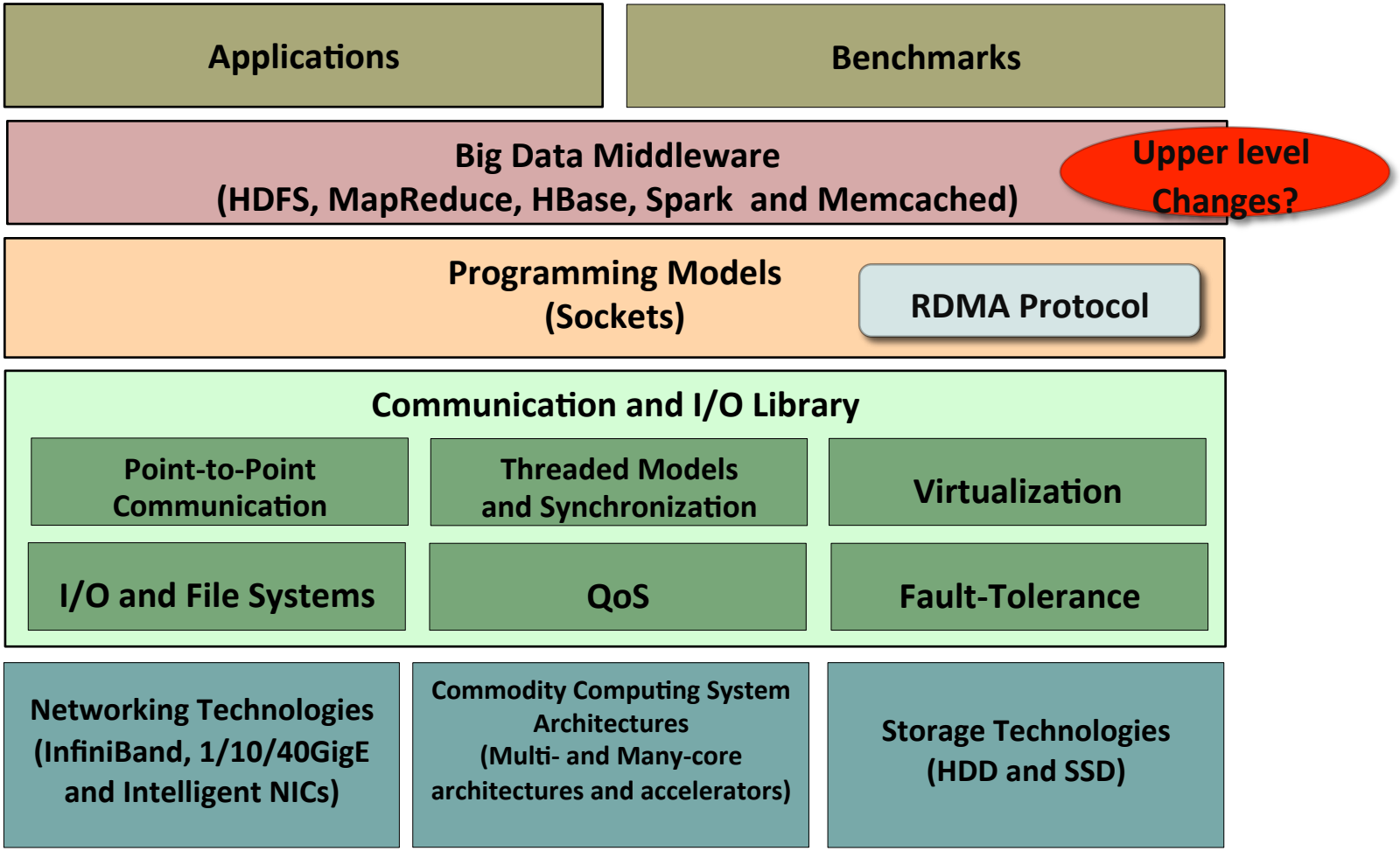
Interconnects and Protocols in the Open Fabrics Stack



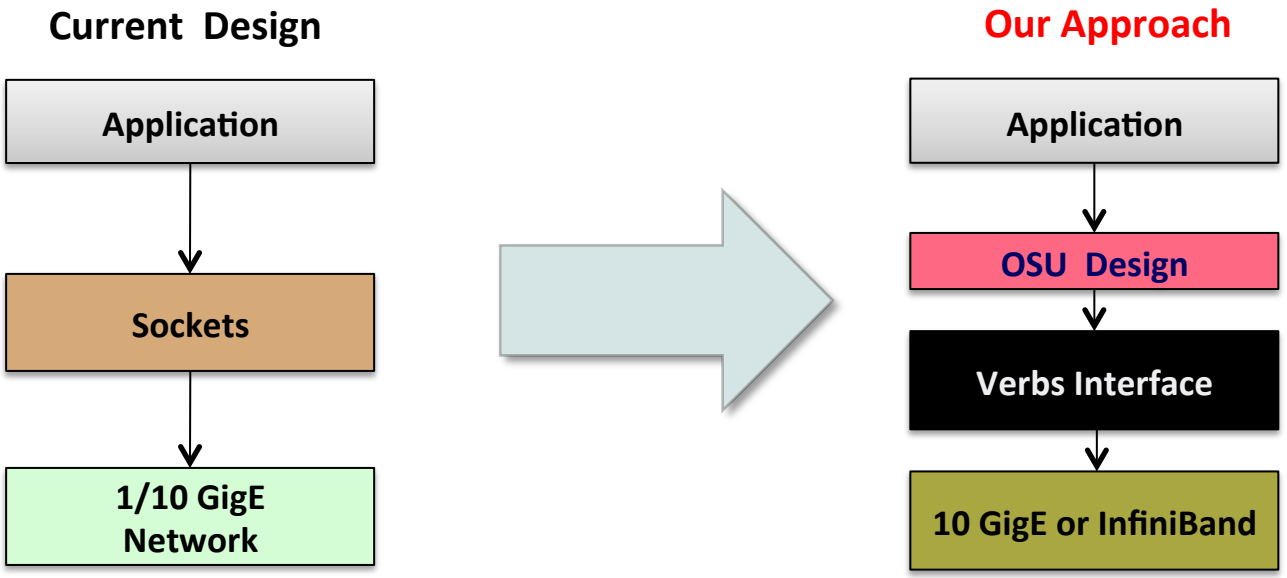
Wide Adoption of RDMA Technology

- 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

Challenges in Designing Communication and I/O Libraries for Big Data Systems



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



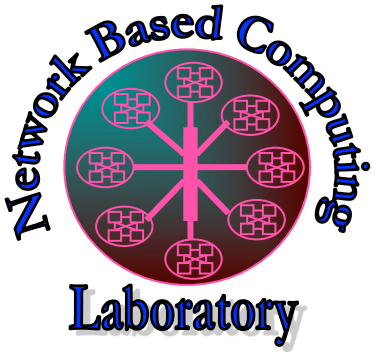
- Sockets not designed for high-performance
  - Stream semantics often mismatch for upper layers
  - Zero-copy not available for non-blocking sockets

Presentation Outline

- 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
  - Sample Performance Numbers for RDMA-Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Overview of the HiBD Project and Releases

- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- RDMA for Memcached (RDMA-Memcached)
- OSU HiBD-Benchmarks (OHB)
- **http://hibd.cse.ohio-state.edu**
- Users Base: 95 organizations from 18 countries
- More than 2,900 downloads
- RDMA for Apache HBase and Spark will be available in near future



THE OHIO STATE UNIVERSITY

RDMA for Apache Hadoop 2.x Distribution

- High-Performance Design of Hadoop over RDMA-enabled Interconnects
  - High performance 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
  - 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: 0.9.6
  - Based on Apache Hadoop 2.6.0
  - Compliant with Apache Hadoop 2.6.0 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 design with native InfiniBand and RoCE support at the verbs-level for Memcached and libMemcached components
  - Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)
  - High performance design of SSD-Assisted Hybrid Memory
- 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 - Memcached

- Released in OHB 0.7.1 (**ohb_memlat**)
- Evaluates the performance of stand-alone Memcached
- Three different micro-benchmarks
  - **SET Micro-benchmark**: Micro-benchmark for memcached set operations
  - **GET Micro-benchmark**: Micro-benchmark for memcached get operations
  - **MIX Micro-benchmark**: Micro-benchmark for a mix of memcached set/get operations (Read:Write ratio is 90:10)
- Calculates average latency of Memcached operations
- Can measure throughput in Transactions Per Second

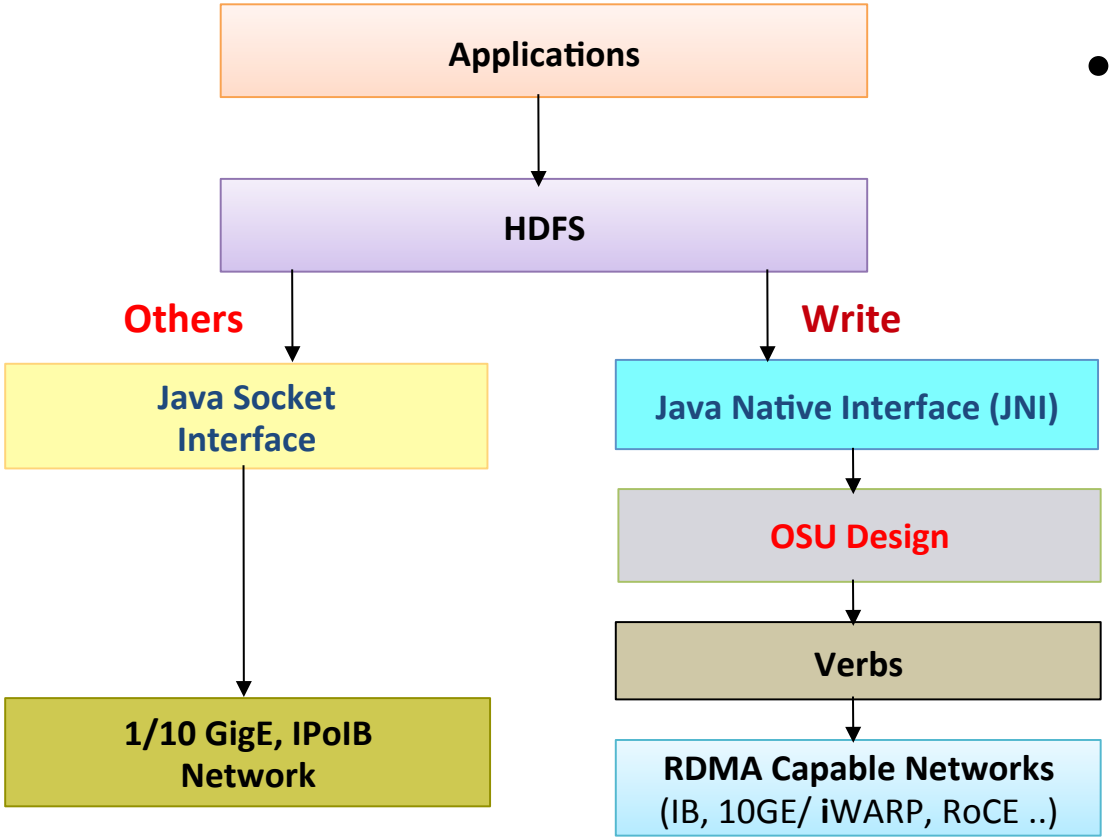
Presentation Outline

- 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
  - Sample Performance Numbers for Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - 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

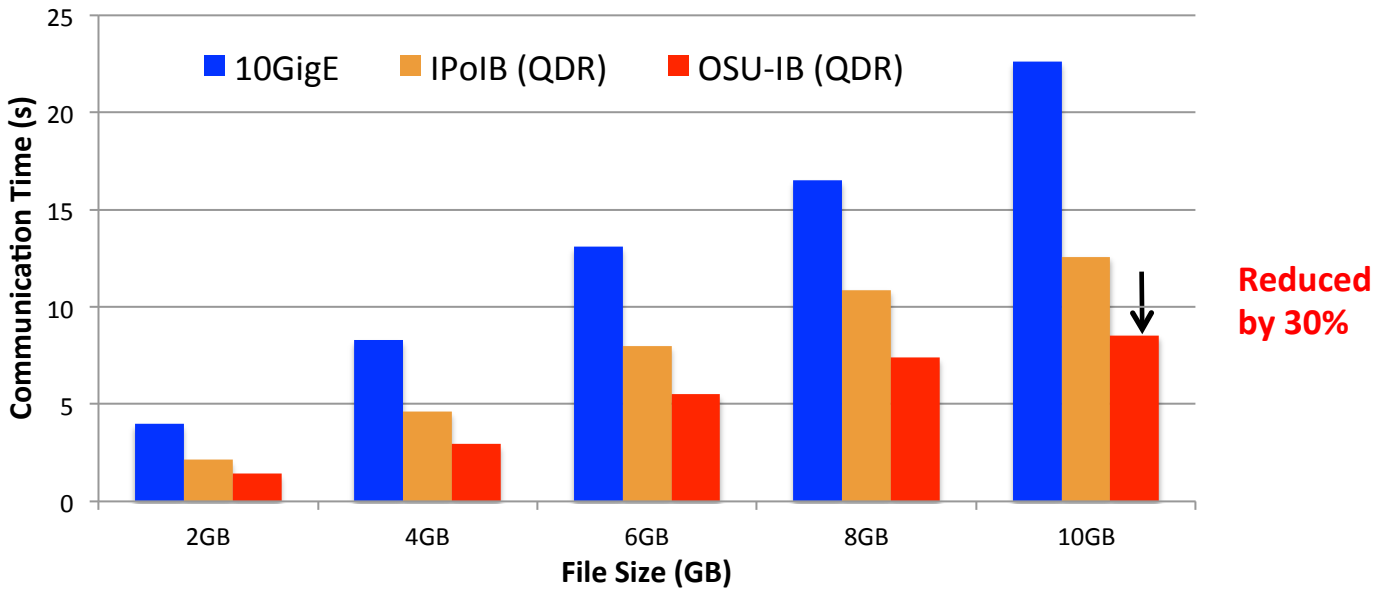
Design Overview of HDFS with RDMA



- 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

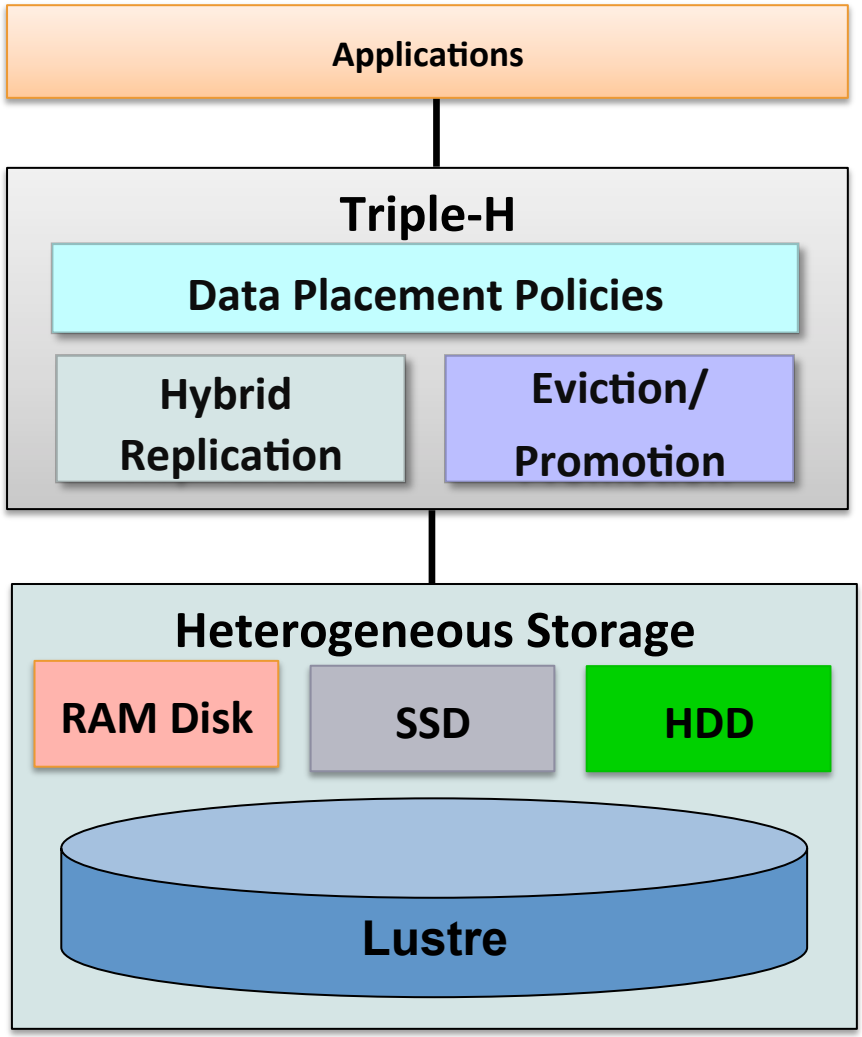


- Cluster with HDD DataNodes
  - **30%** improvement in communication time over IPoIB (QDR)
  - **56%** improvement in communication time over 10GigE
- Similar improvements are obtained for SSD DataNodes

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

Enhanced HDFS with In-memory and Heterogeneous Storage



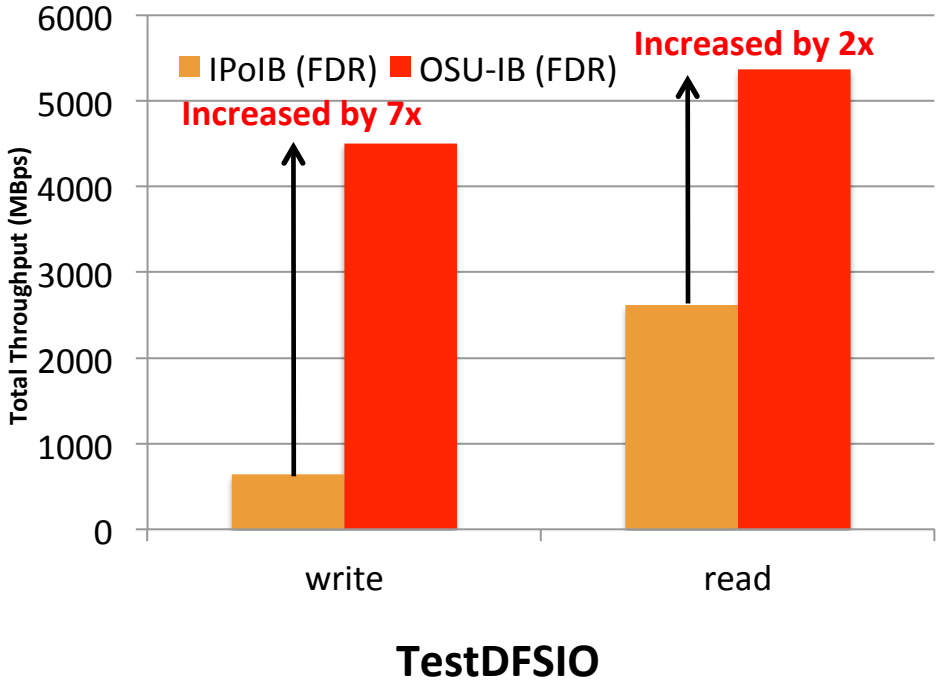
- Design Features
  - Three modes
    - Default (HHH)
    - In-Memory (HHH-M)
    - Lustre-Integrated (HHH-L)
  - Policies to efficiently utilize the heterogeneous storage devices
    - RAM, SSD, HDD, Lustre
  - Eviction/Promotion based on data usage pattern
  - Hybrid Replication
  - Lustre-Integrated 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

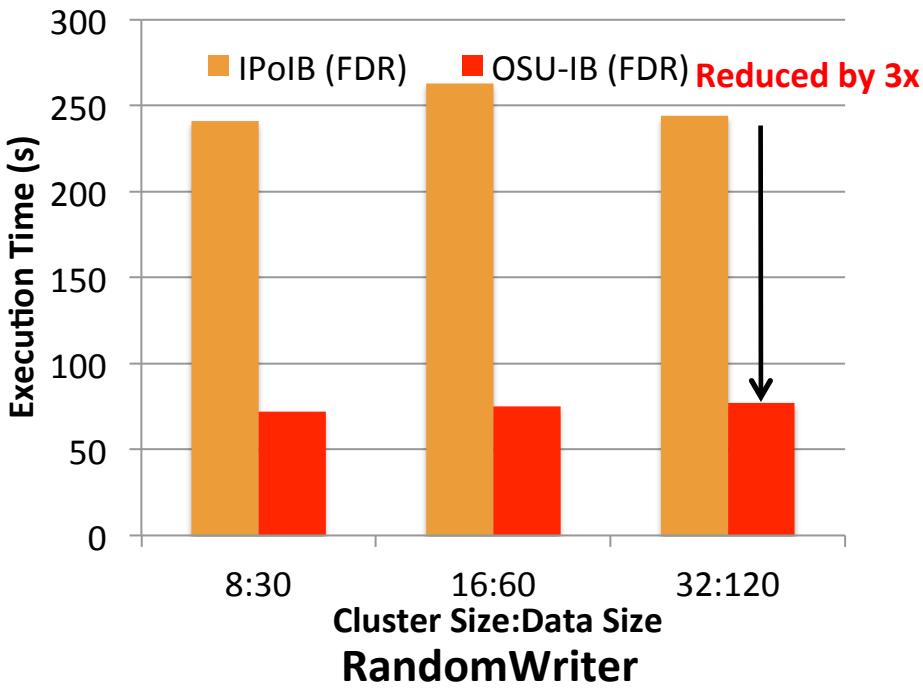
Enhanced HDFS with In-memory and Heterogeneous Storage – Three Modes

- **HHH (default)**: Heterogeneous storage devices with hybrid replication schemes
  - I/O operations over RAM disk, SSD, and HDD
  - Hybrid replication (in-memory and persistent storage)
  - Better fault-tolerance as well as performance
- **HHH-M**: High-performance in-memory I/O operations
  - Memory replication (in-memory only with lazy persistence)
  - As much performance benefit as possible
- **HHH-L**: Lustre integrated
  - Take advantage of the Lustre available in HPC clusters
  - Lustre-based fault-tolerance (No HDFS replication)
  - Reduced local storage space usage

Performance Improvement on TACC Stampede (HHH)



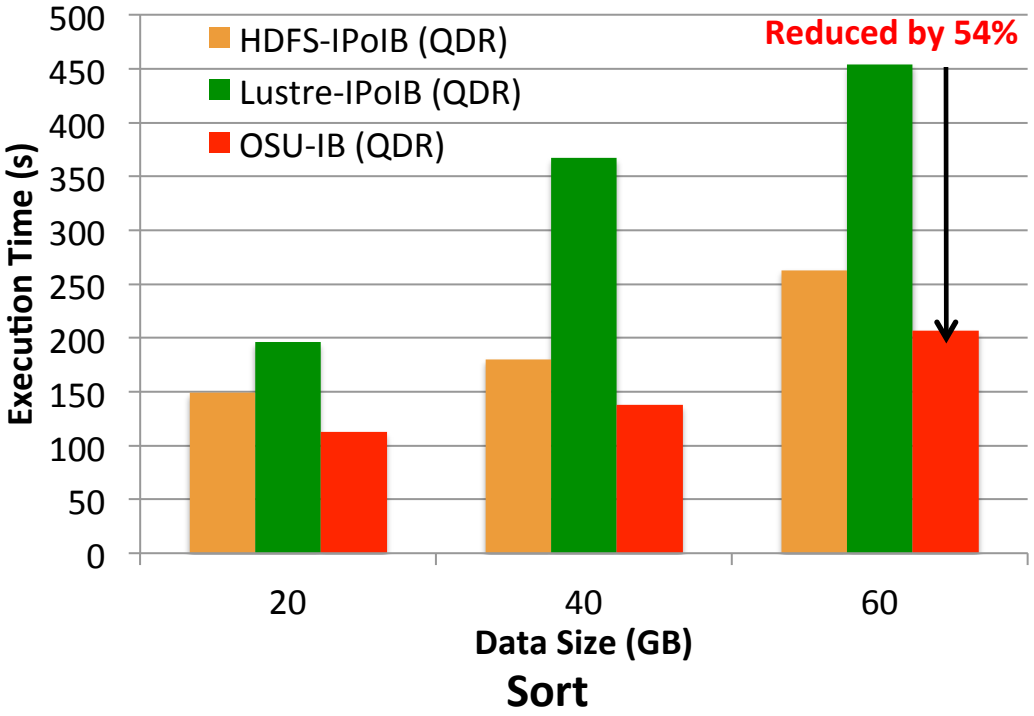
TestDFSIO



RandomWriter

- 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)

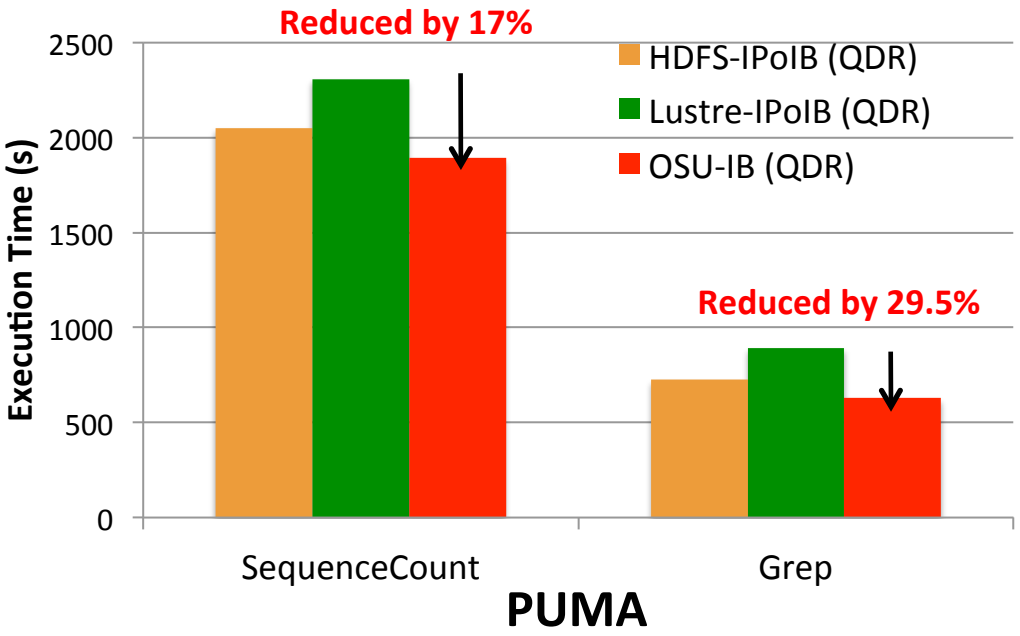


Storage Used (GB)	
HDFS-IPoIB (QDR)	360
Lustre-IPoIB (QDR)	120
OSU-IB (QDR)	240

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

Evaluation with PUMA and CloudBurst (HHH-L/HHH)



HDFS-IPoIB (FDR)	OSU-IB (FDR)
60.24 s	48.3 s

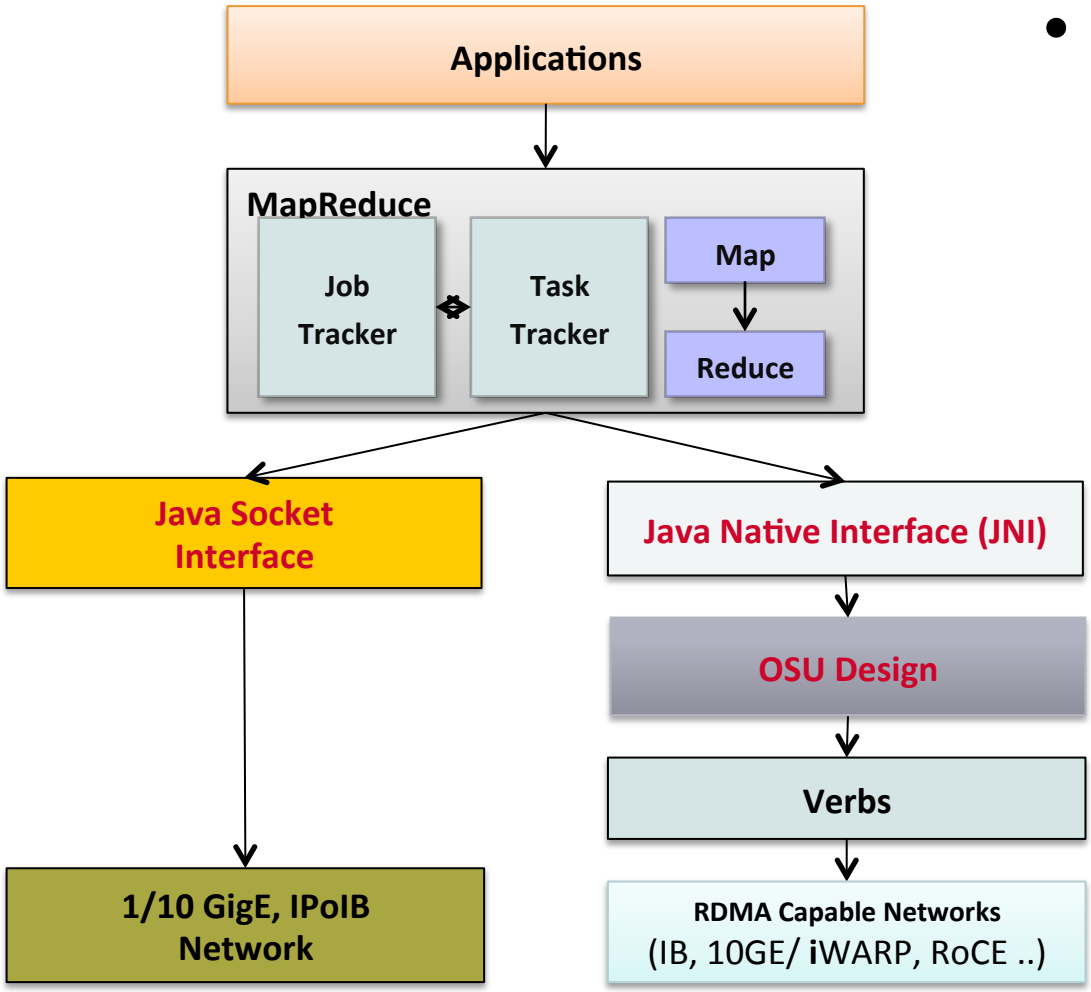
CloudBurst

- PUMA on OSU RI
  - SequenceCount with **HHH-L**: **17%** benefit over Lustre, **8%** over HDFS
  - Grep with **HHH**: **29.5%** benefit over Lustre, **13.2%** over HDFS
- CloudBurst on TACC Stampede
  - With **HHH**: **19%** improvement over HDFS

Acceleration Case Studies and In-Depth Performance Evaluation

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

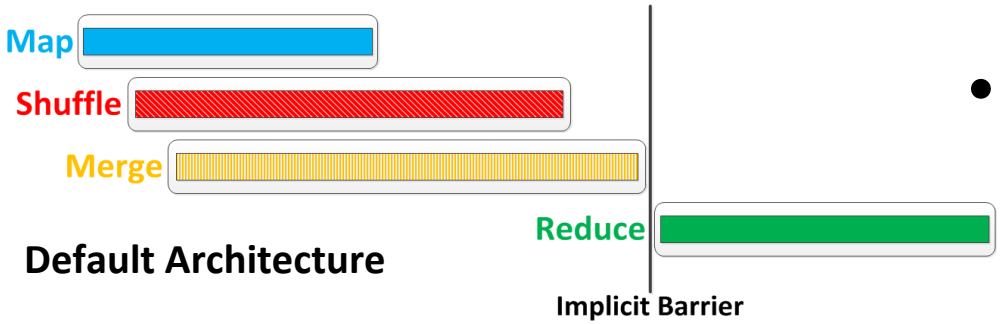
Design Overview of MapReduce with RDMA



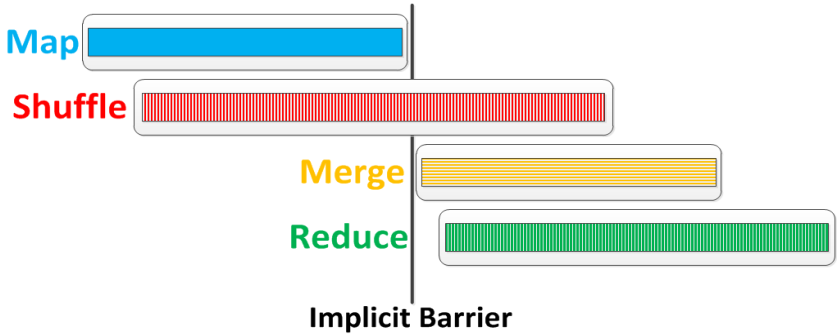
- Design Features
  - RDMA-based shuffle
  - Prefetching and caching map output
  - Efficient Shuffle Algorithms
  - In-memory merge
  - On-demand Shuffle Adjustment
  - Advanced overlapping
    - map, shuffle, and merge
    - shuffle, merge, and reduce
  - On-demand connection setup
  - InfiniBand/RoCE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

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

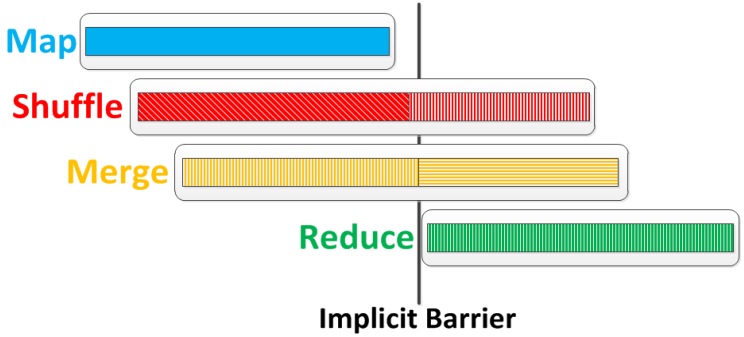
Advanced Overlapping among different phases



Default Architecture



Enhanced Overlapping

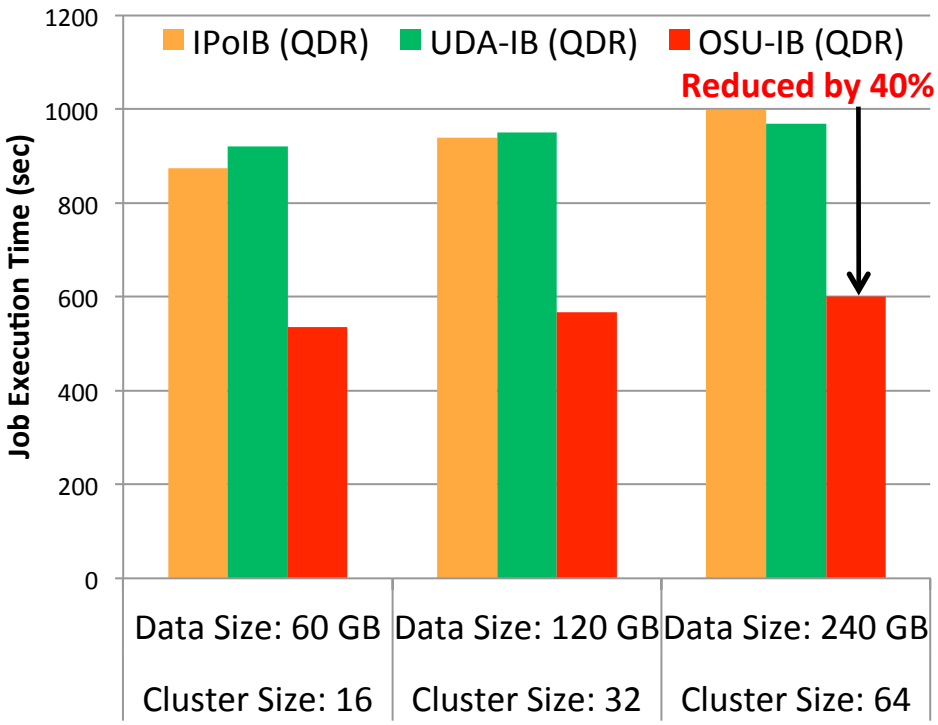


Advanced Overlapping

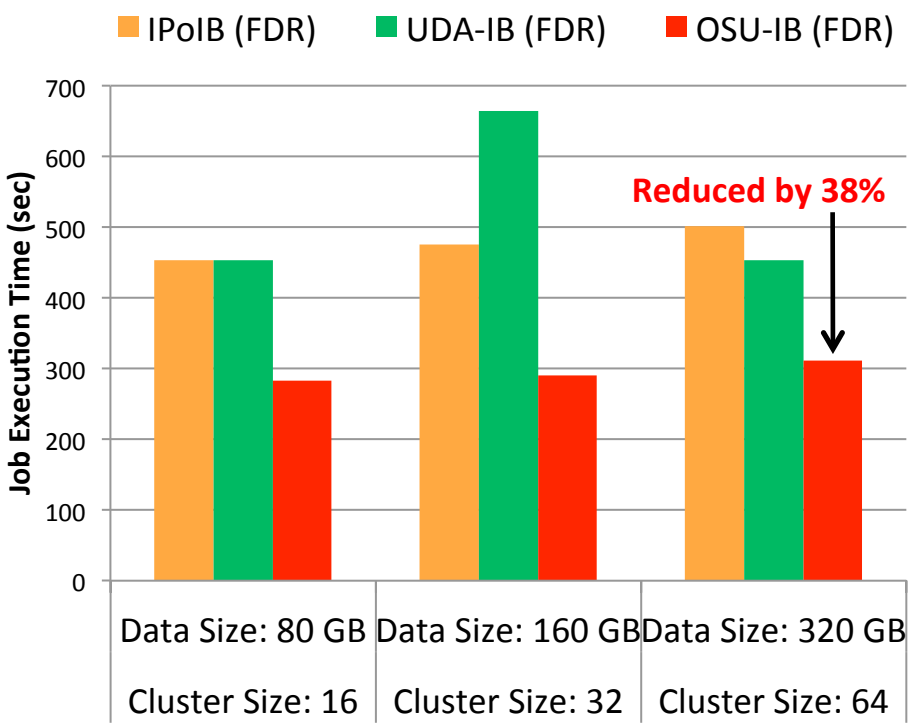
- A hybrid approach to achieve maximum possible overlapping in MapReduce across all phases compared to other approaches
  - Efficient Shuffle Algorithms
  - Dynamic and Efficient Switching
  - On-demand Shuffle Adjustment

M. W. Rahman, X. Lu, N. S. Islam, and D. K. Panda, HOMR: A Hybrid Approach to Exploit Maximum Overlapping in MapReduce over High Performance Interconnects, ICS, June 2014.

Performance Evaluation of Sort and TeraSort



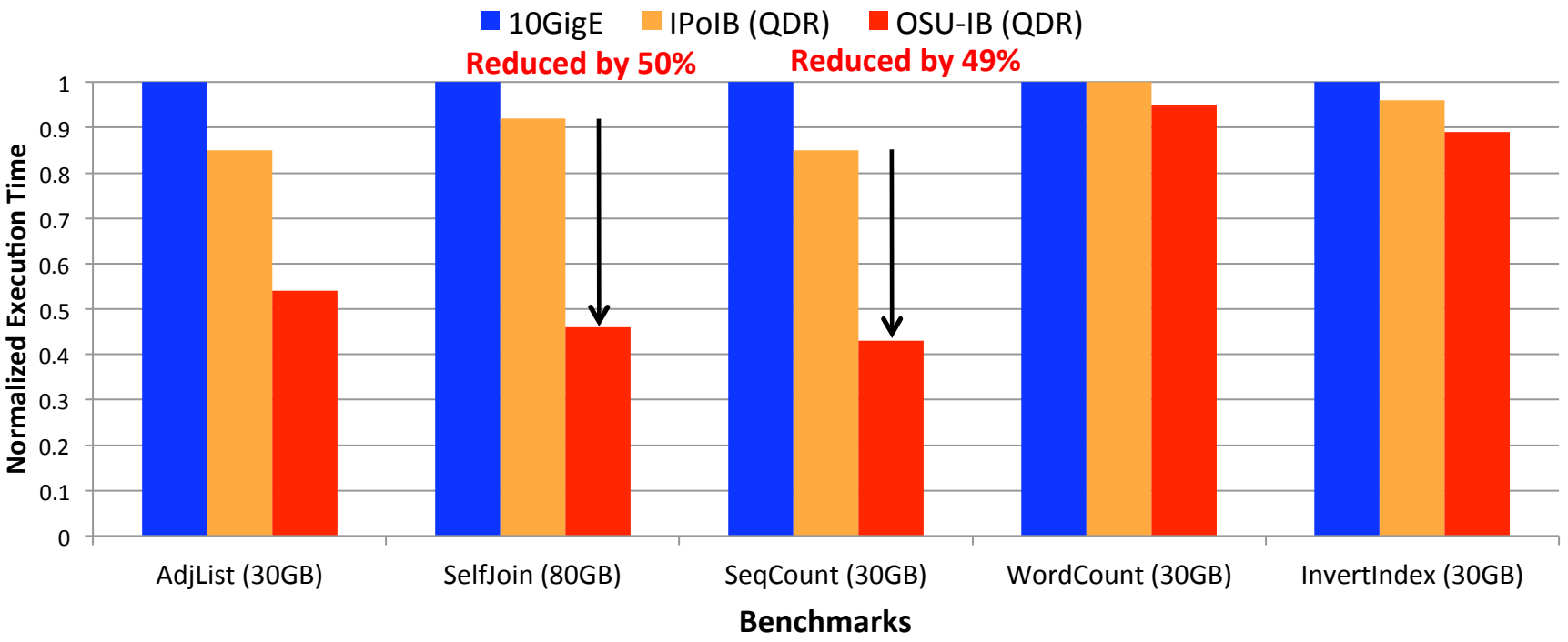
Sort in OSU Cluster



TeraSort in TACC Stampede

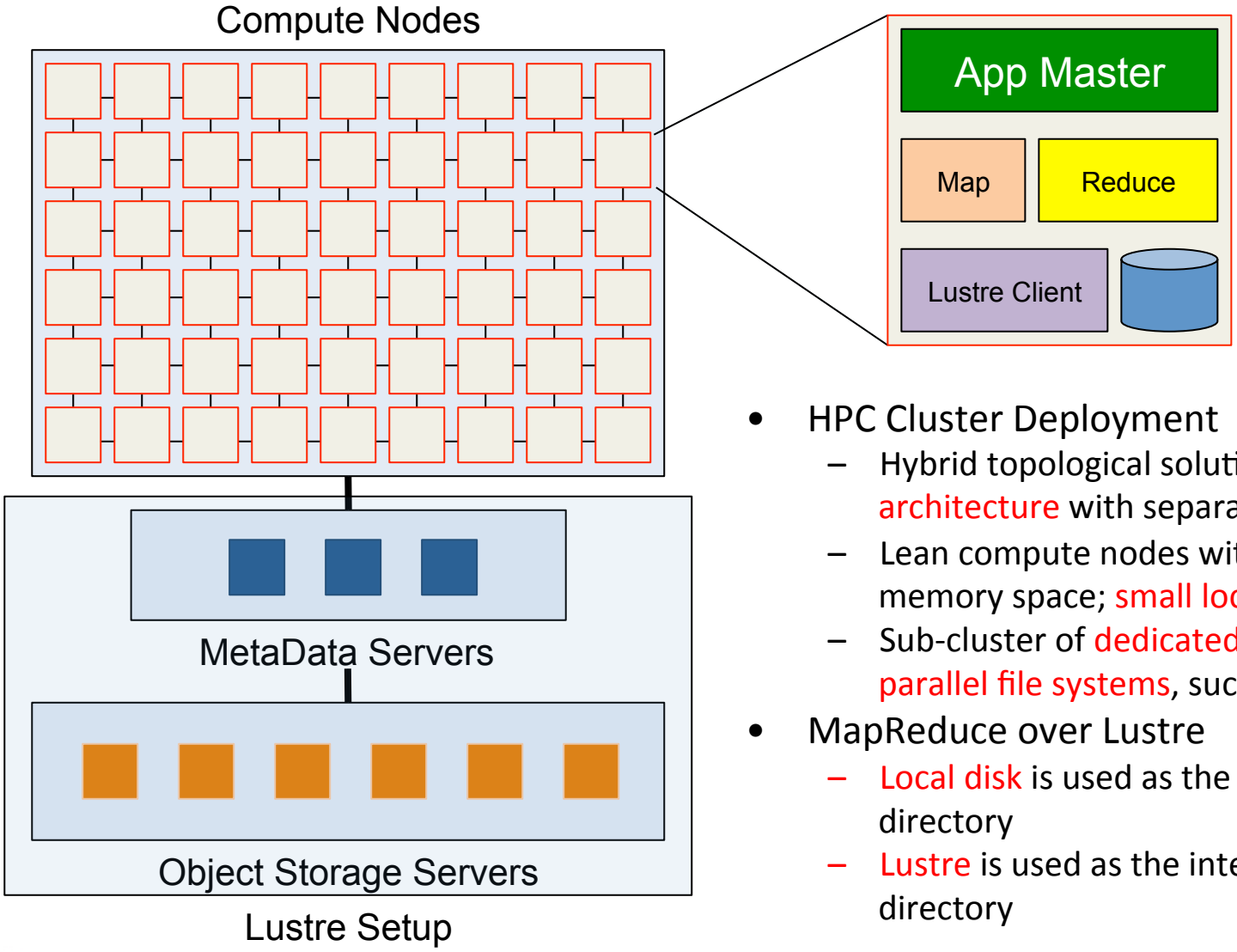
- For 240GB Sort in 64 nodes (512 cores)
  - 40% improvement over IPoIB (QDR) with HDD used for HDFS
- For 320GB TeraSort in 64 nodes (1K cores)
  - 38% improvement over IPoIB (FDR) with HDD used for HDFS

Evaluations using PUMA Workload



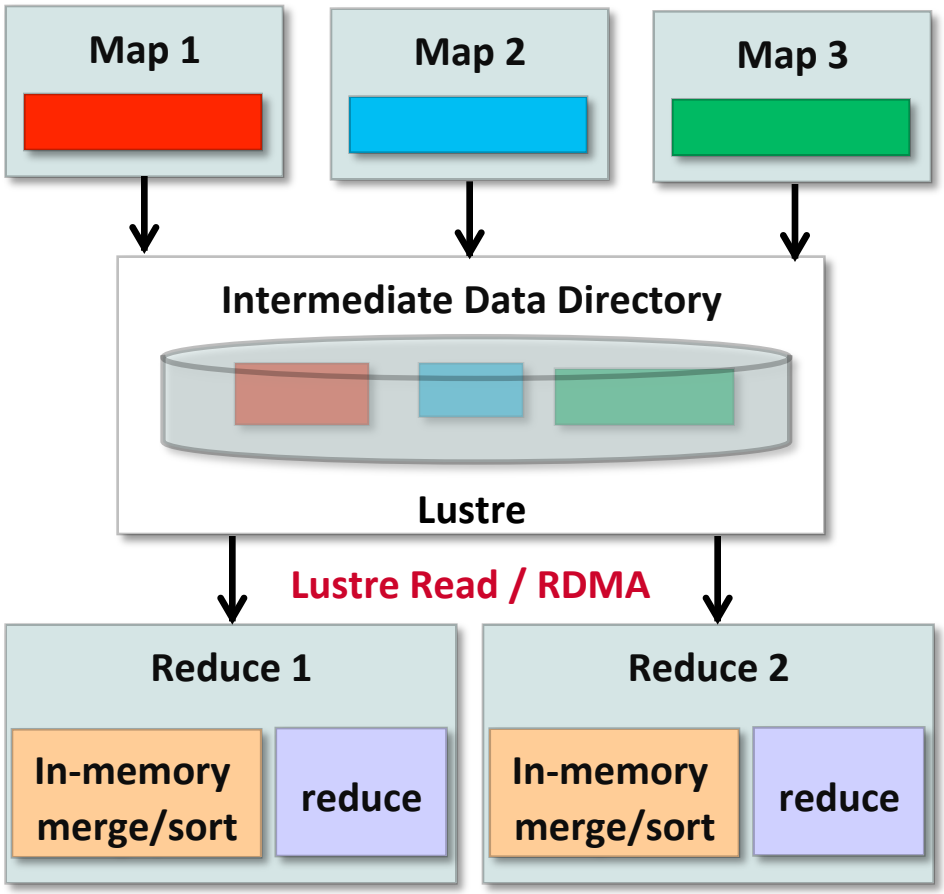
- 50% improvement in Self Join over IPoIB (QDR) for 80 GB data size
- 49% improvement in Sequence Count over IPoIB (QDR) for 30 GB data size

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

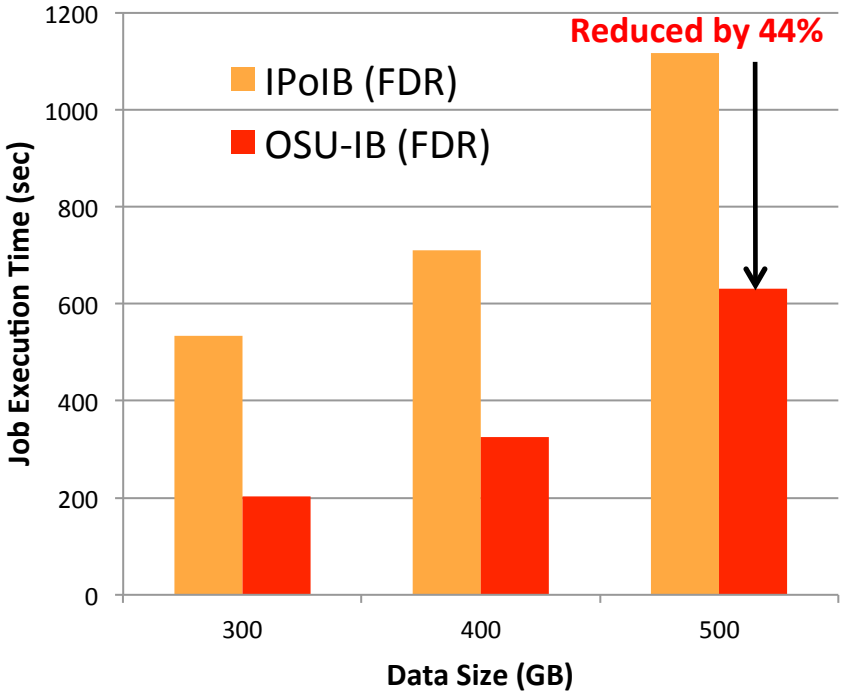


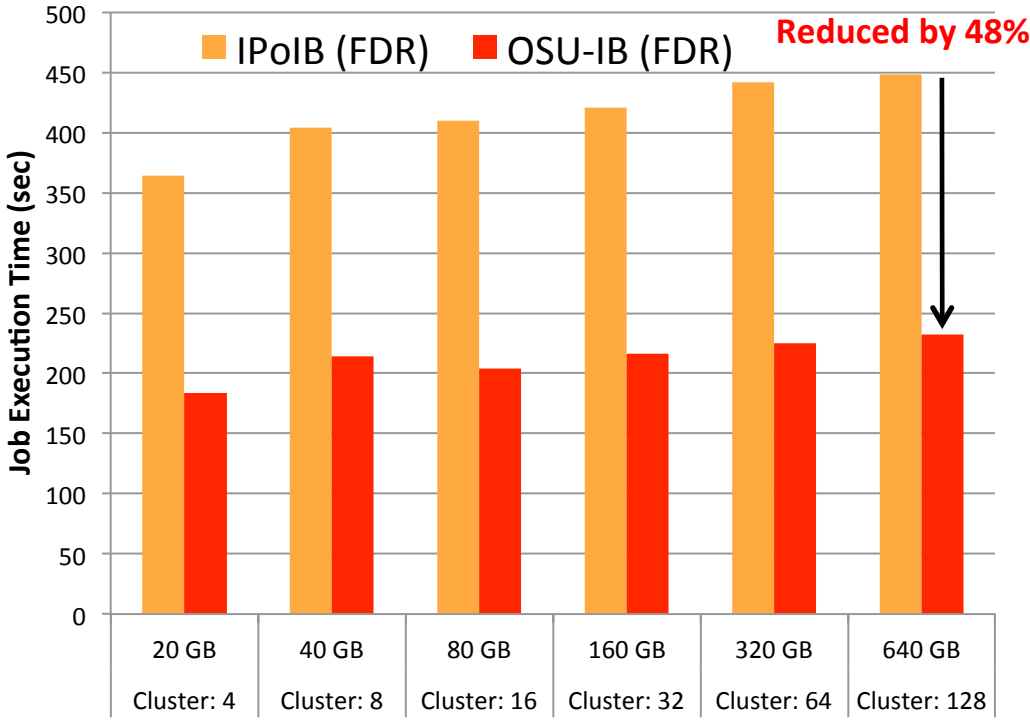
- Design Features
  - Two shuffle approaches
    - Lustre read based shuffle
    - RDMA based shuffle
  - Hybrid shuffle algorithm to take benefit from both shuffle approaches
  - Dynamically adapts to the better shuffle approach for each shuffle request based on profiling values for each Lustre read 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



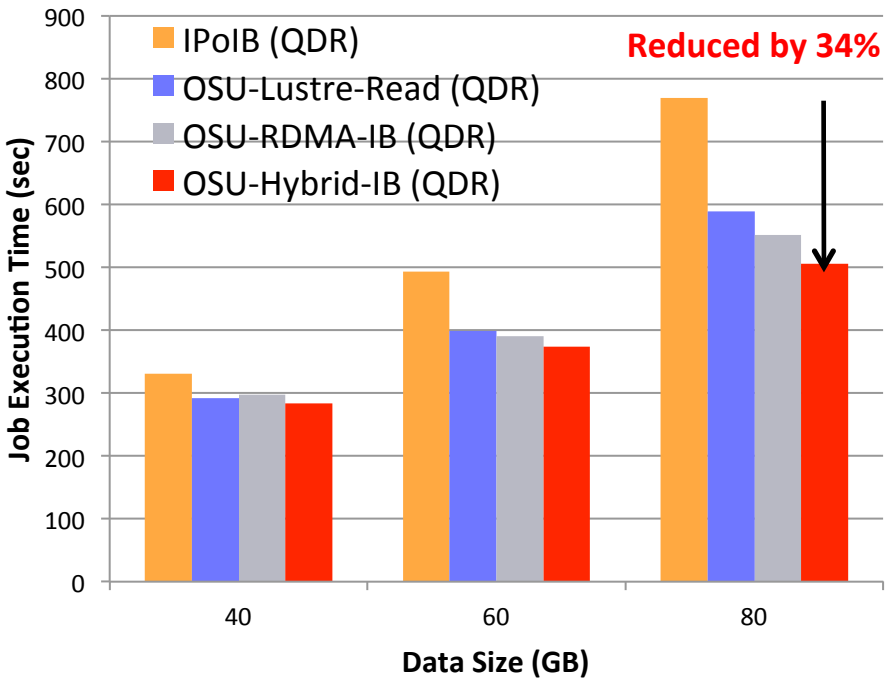


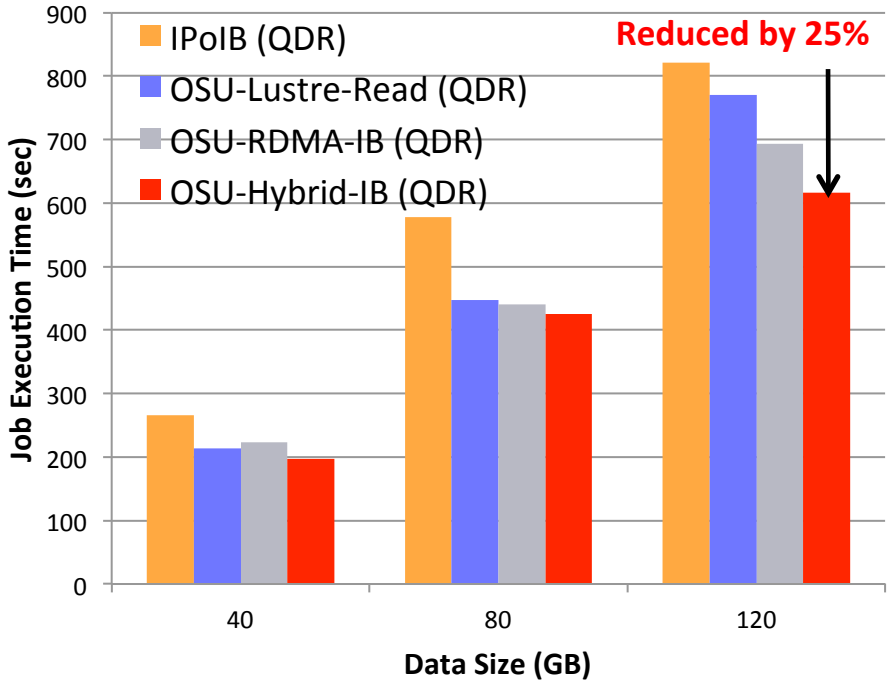
- For 500GB Sort in 64 nodes
  - 44% improvement over IPoIB (FDR)
- For 640GB Sort in 128 nodes
  - 48% improvement over IPoIB (FDR)

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



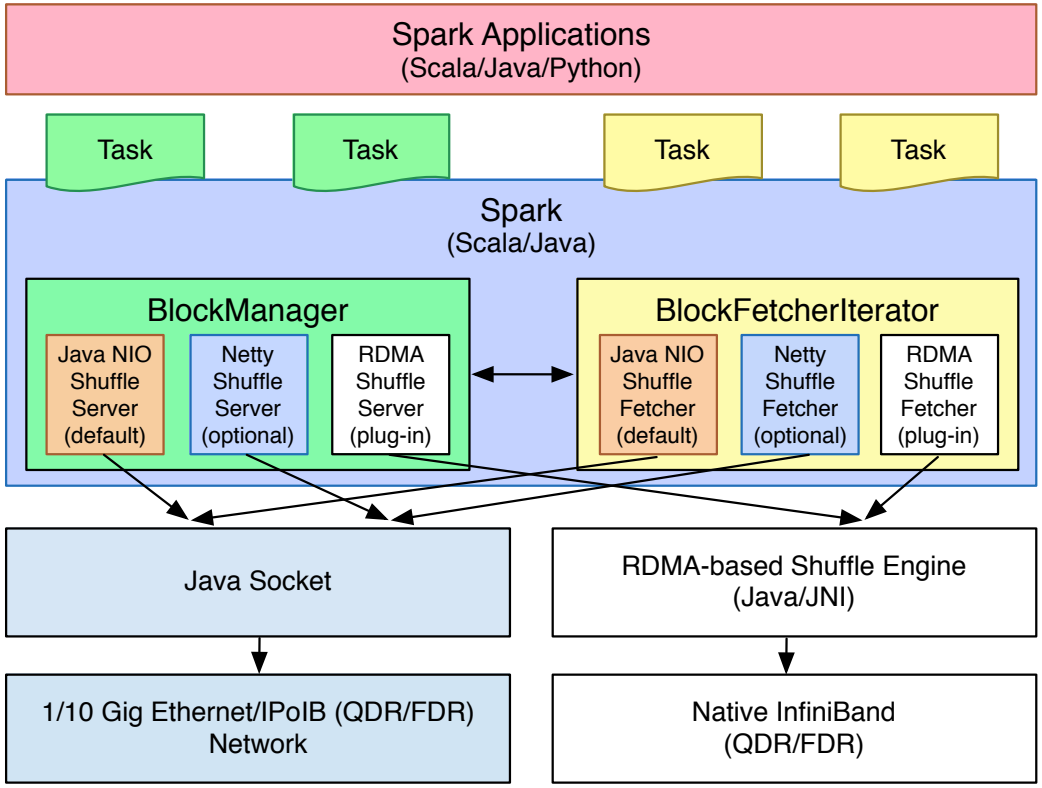


- For 80GB Sort in 8 nodes
  - 34% improvement over IPoIB (QDR)
- For 120GB TeraSort in 16 nodes
  - 25% improvement over IPoIB (QDR)

Acceleration Case Studies and In-Depth Performance Evaluation

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

Design Overview of Spark with RDMA

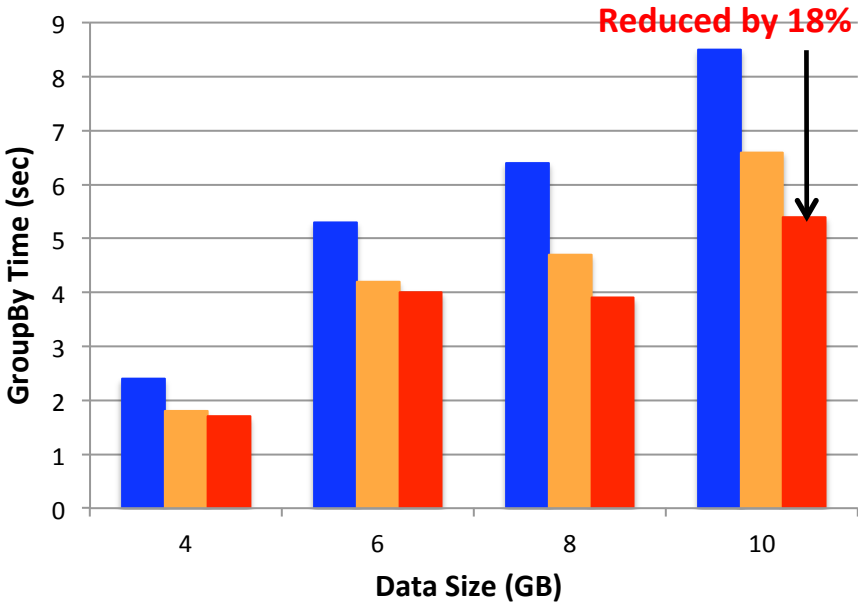


- Design Features
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking and out-of-order data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

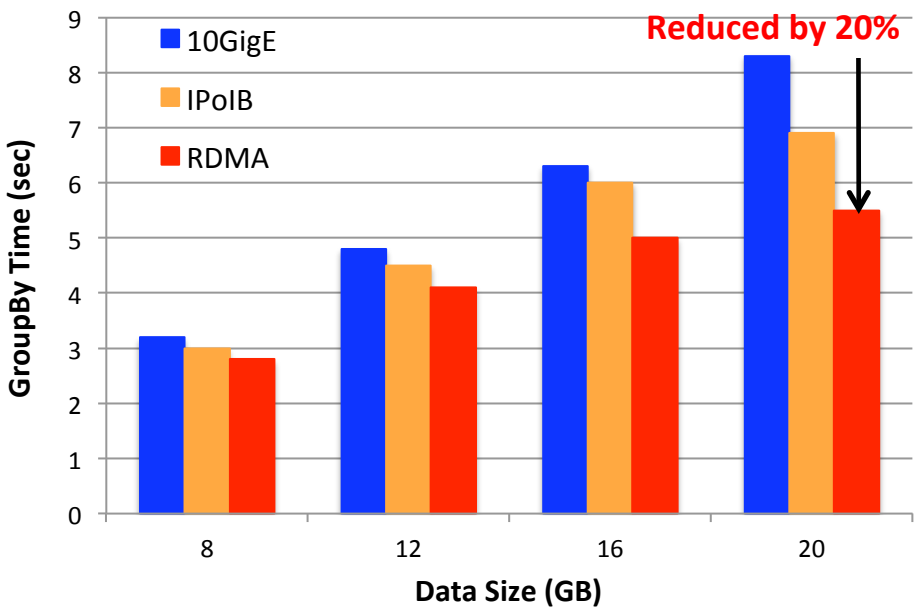
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

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

Preliminary Results of Spark-RDMA Design - GroupBy



Cluster with 4 HDD Nodes, GroupBy with 32 cores



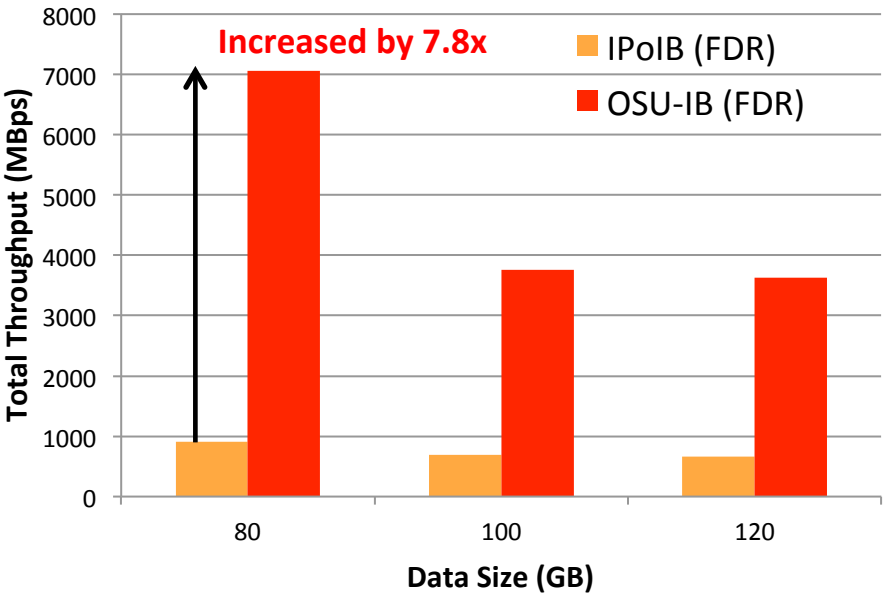
Cluster with 8 HDD Nodes, GroupBy with 64 cores

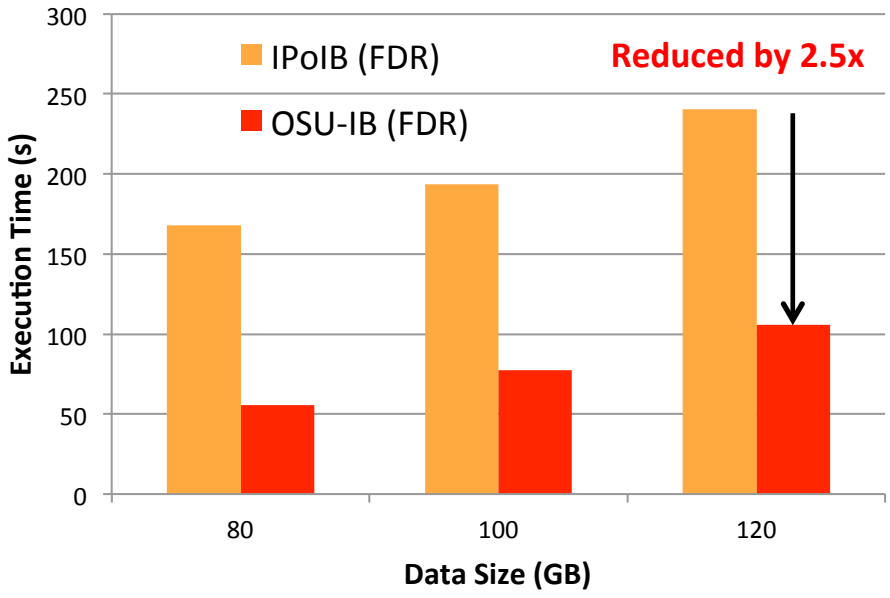
- Cluster with 4 HDD Nodes, single disk per node, 32 concurrent tasks
  - **18%** improvement over IPoIB (QDR) for 10GB data size
- Cluster with 8 HDD Nodes, single disk per node, 64 concurrent tasks
  - **20%** improvement over IPoIB (QDR) for 20GB data size

Presentation Outline

- 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
  - Sample Performance Numbers for Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Performance Benefits – TestDFSIO in TACC Stampede (HHH)

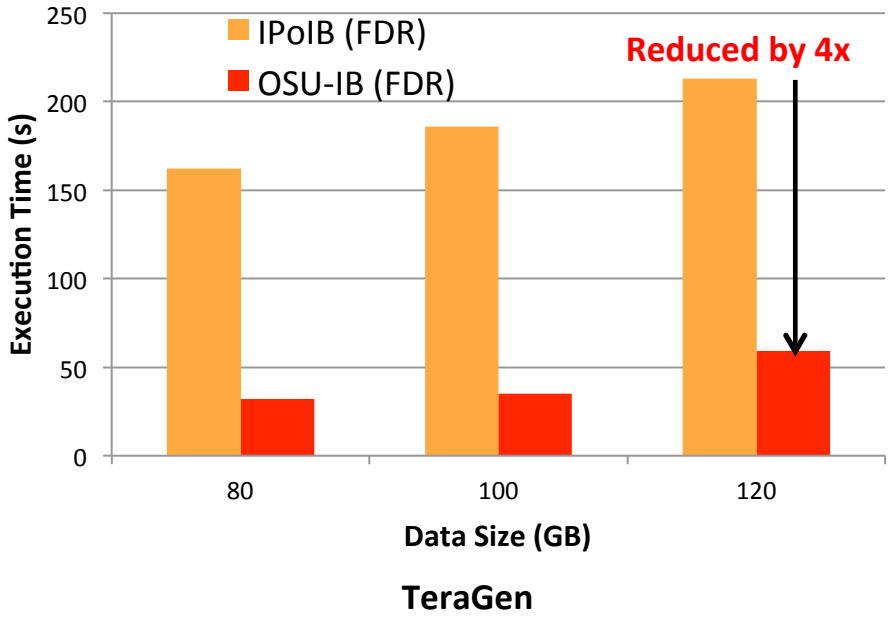




Cluster with 32 Nodes with HDD, TestDFSIO with a total 128 maps

- For Throughput,
  - **6-7.8x** improvement over IPoIB for 80-120 GB file size
- For Latency,
  - **2.5-3x** improvement over IPoIB for 80-120 GB file size

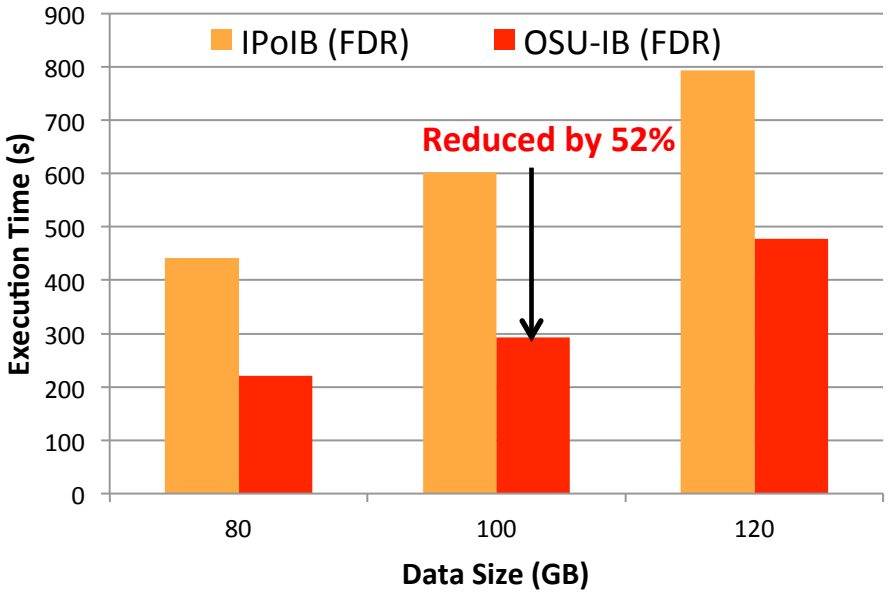
Performance Benefits – RandomWriter & TeraGen in TACC-Stampede (HHH)



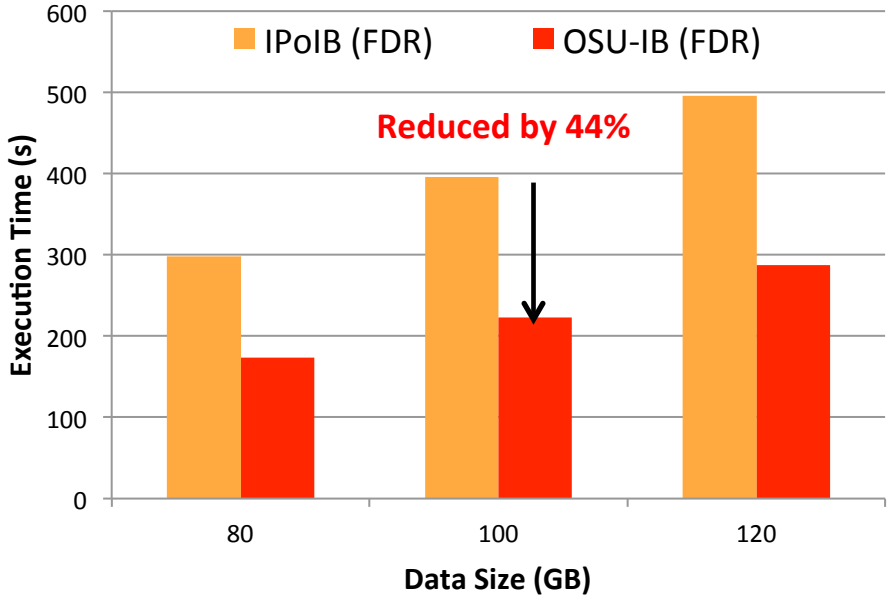
Cluster with 32 Nodes with a total of 128 maps

- RandomWriter
  - **3-4x** improvement over IPoIB for 80-120 GB file size
- TeraGen
  - **4-5x** improvement over IPoIB for 80-120 GB file size

Performance Benefits – Sort & TeraSort in TACC-Stampede (HHH)



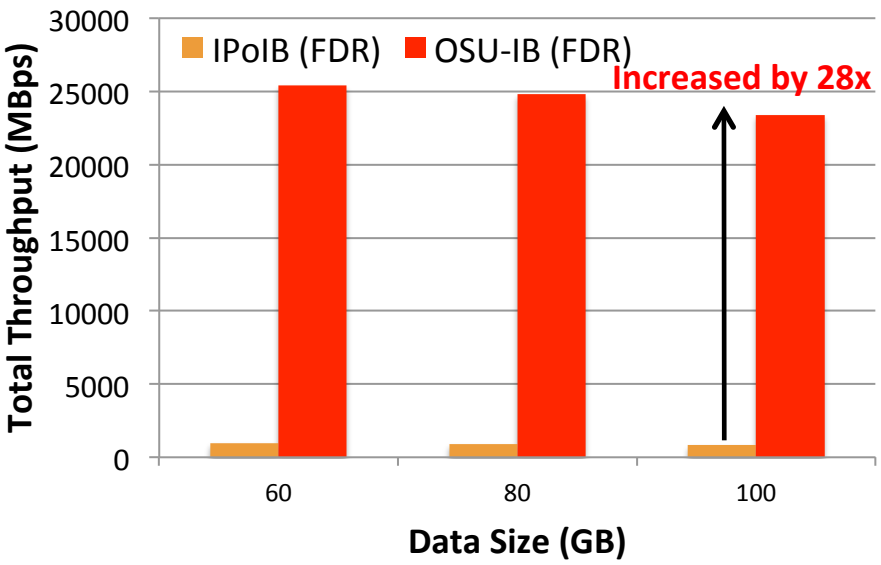
Cluster with 32 Nodes with a total of 128 maps and 57 reduces



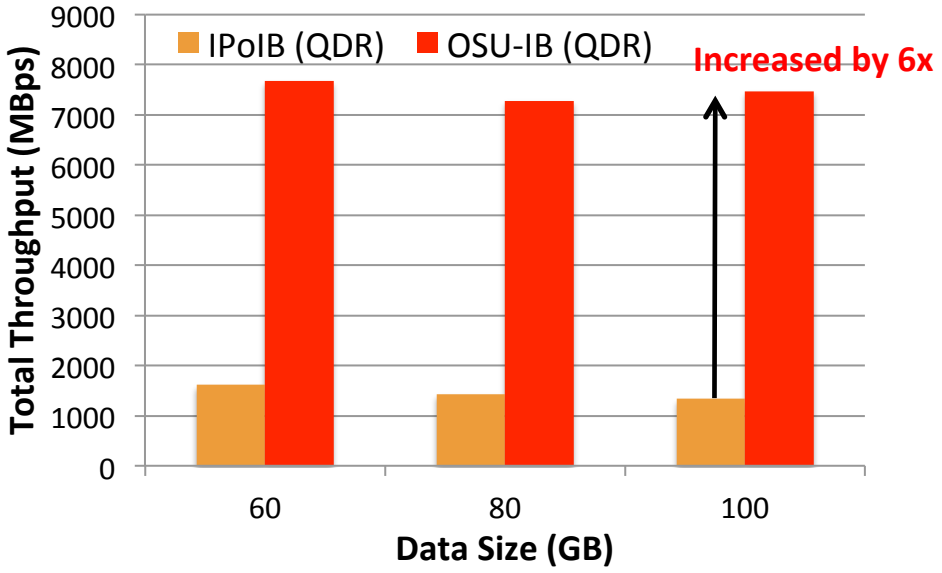
Cluster with 32 Nodes with a total of 128 maps and 64 reduces

- Sort with single HDD per node
  - **40-52%** improvement over IPoIB for 80-120 GB data
- TeraSort with single HDD per node
  - **42-44%** improvement over IPoIB for 80-120 GB data

Performance Benefits – TestDFSIO on TACC Stampede and SDSC Gordon (HHH-M)



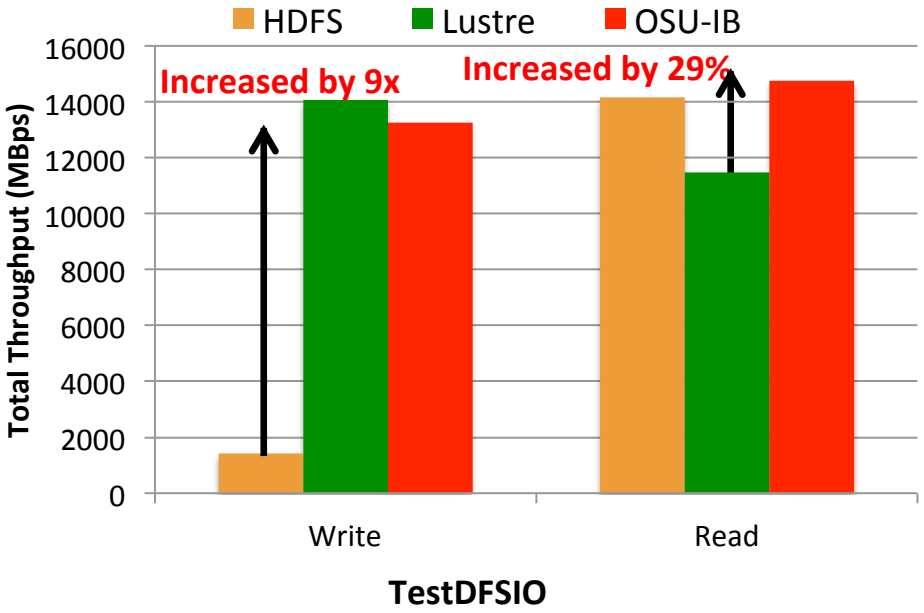
TACC Stampede Cluster with 32 Nodes with HDD, TestDFSIO with a total 128 maps

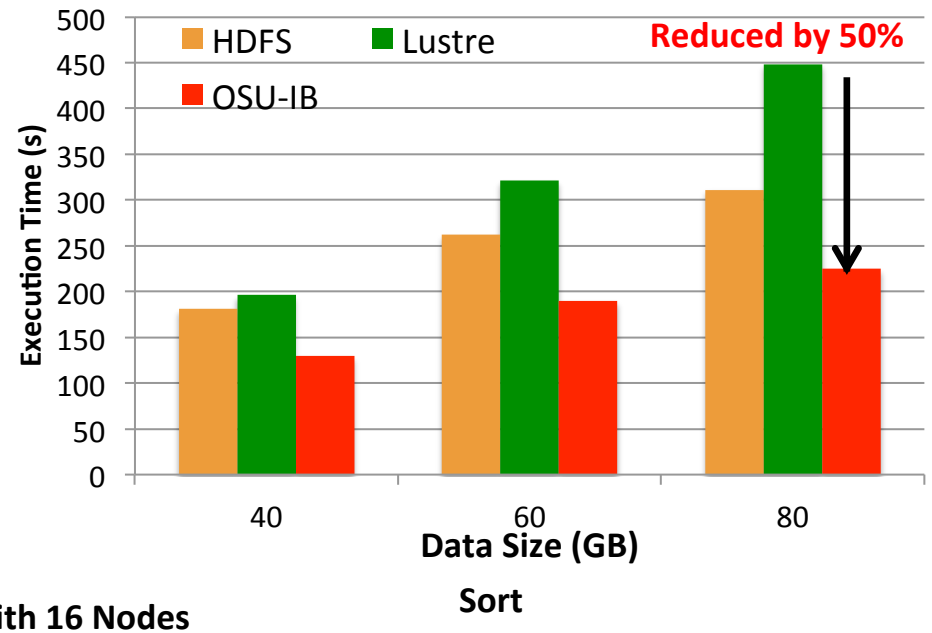


SDSC Gordon Cluster with 16 Nodes with SSD, TestDFSIO with a total 64 maps

- TestDFSIO Write on TACC Stampede
  - **28x** improvement over IPoIB for 60-100 GB file size
- TestDFSIO Write on SDSC Gordon
  - **6x** improvement over IPoIB for 60-100 GB file size

Performance Benefits – TestDFSIO and Sort in SDSC-Gordon (HHH-L)





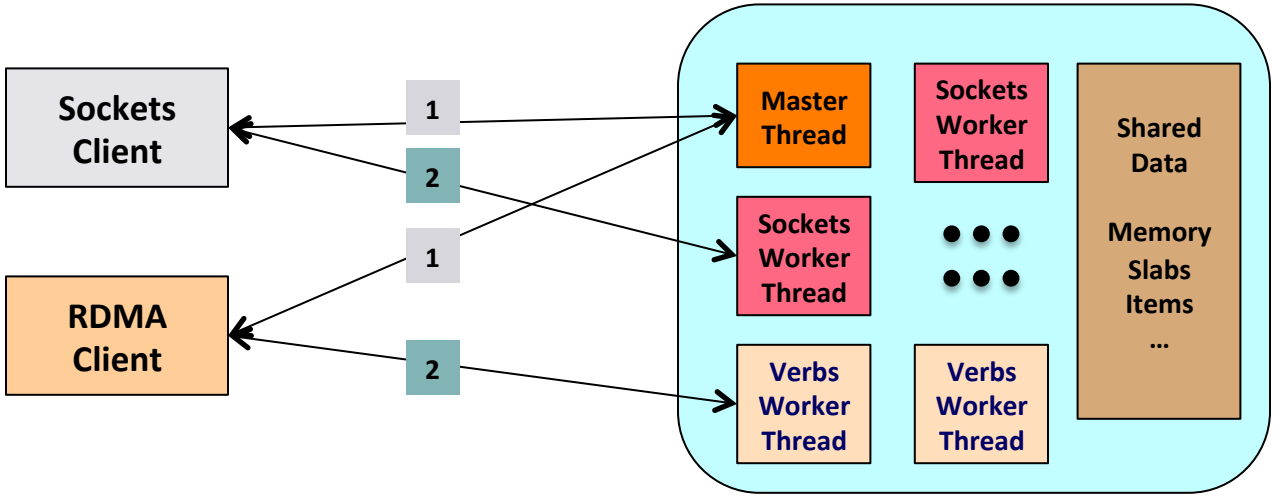
Cluster with 16 Nodes

- TestDFSIO for 80GB data size
  - Write: **9x** improvement over HDFS
  - Read: **29%** improvement over Lustre
- Sort
  - up to **28%** improvement over HDFS
  - up to **50%** improvement over Lustre

Presentation Outline

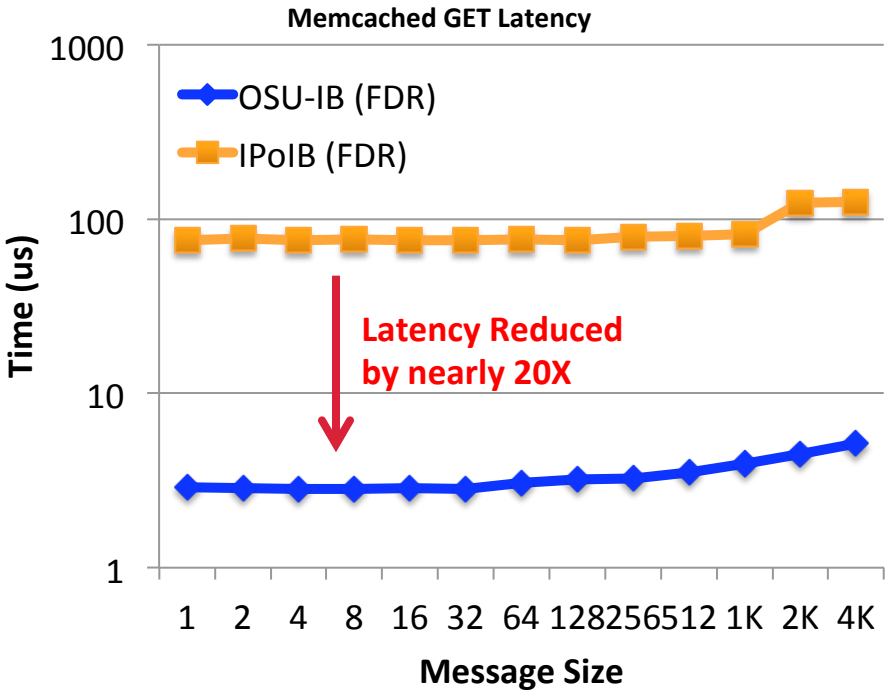
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - Sample Performance Numbers for Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

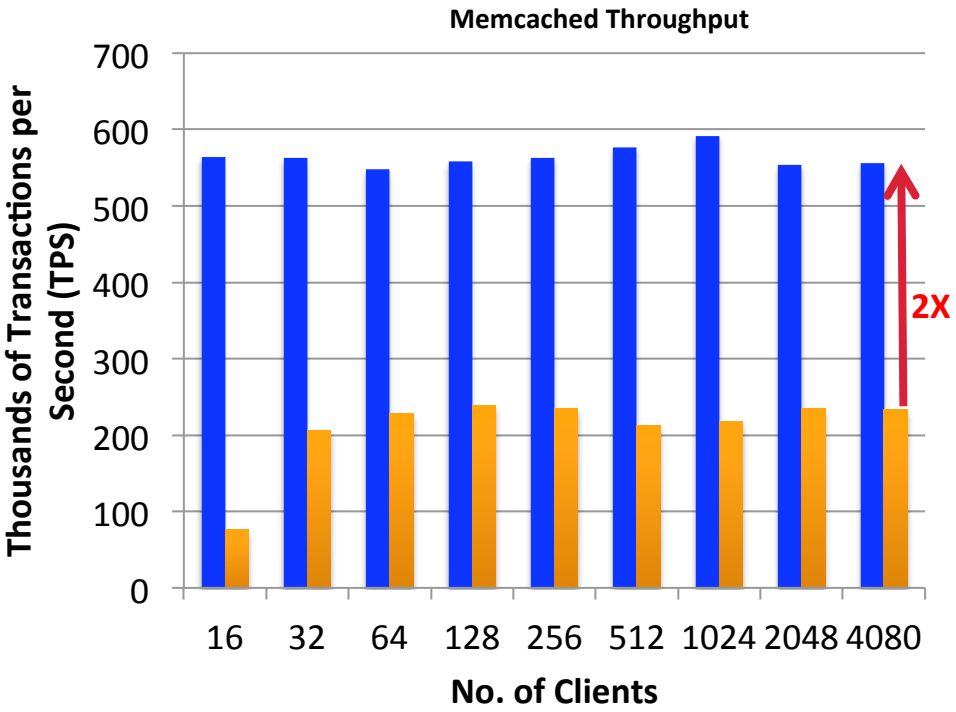
Memcached-RDMA Design



- 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)

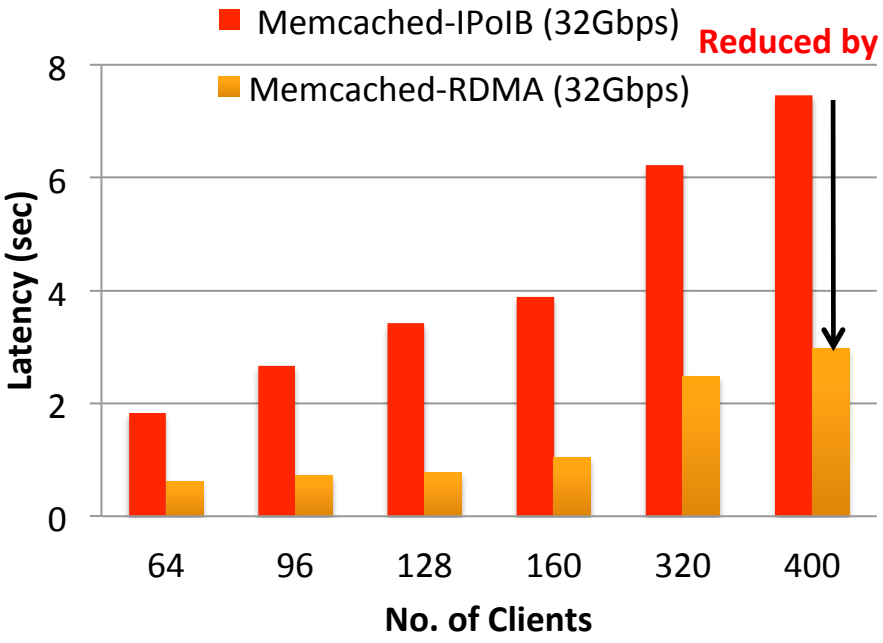


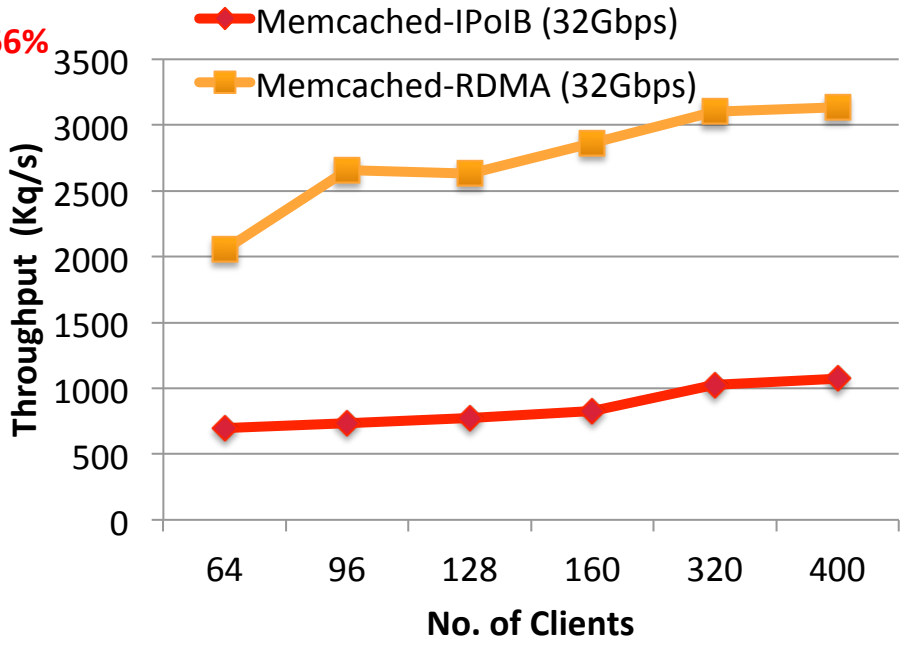


Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- 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

Micro-benchmark Evaluation for OLDP workloads

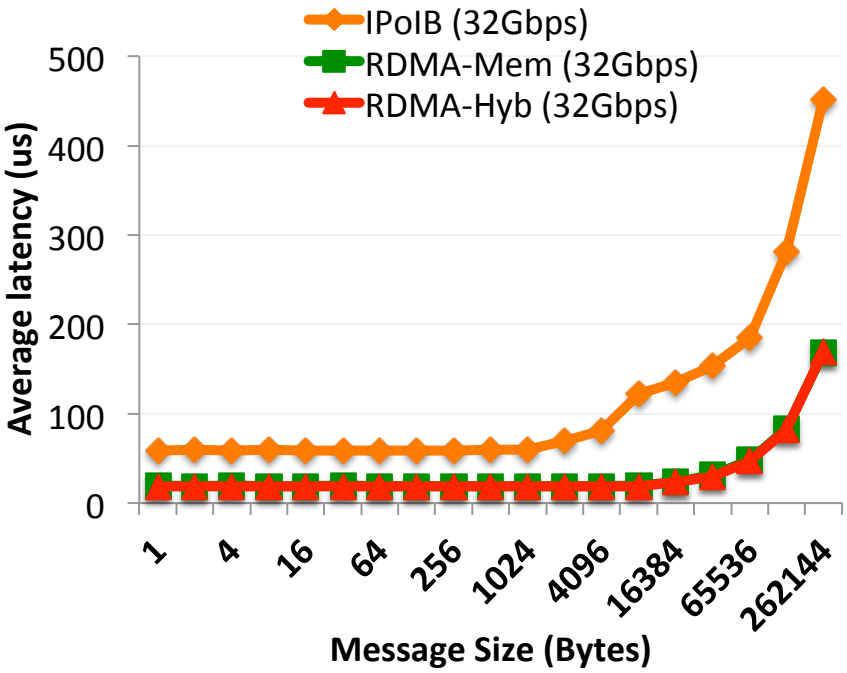


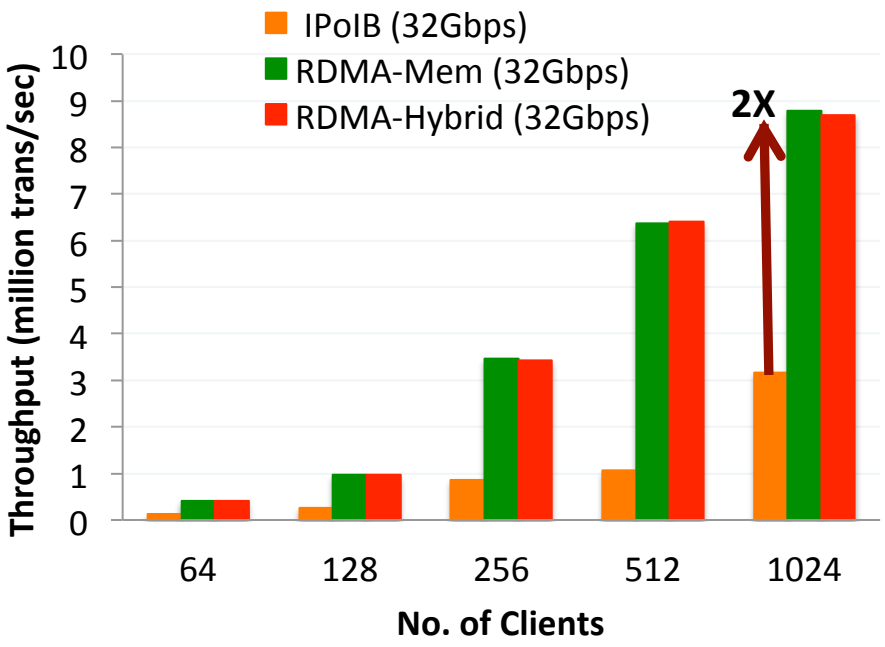


- Illustration with Read-Cache-Read access pattern using modified mysqlslap load testing tool
- Memcached-RDMA can
  - improve query latency by up to 66% over IPoIB (32Gbps)
  - throughput by up to 69% over IPoIB (32Gbps)

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

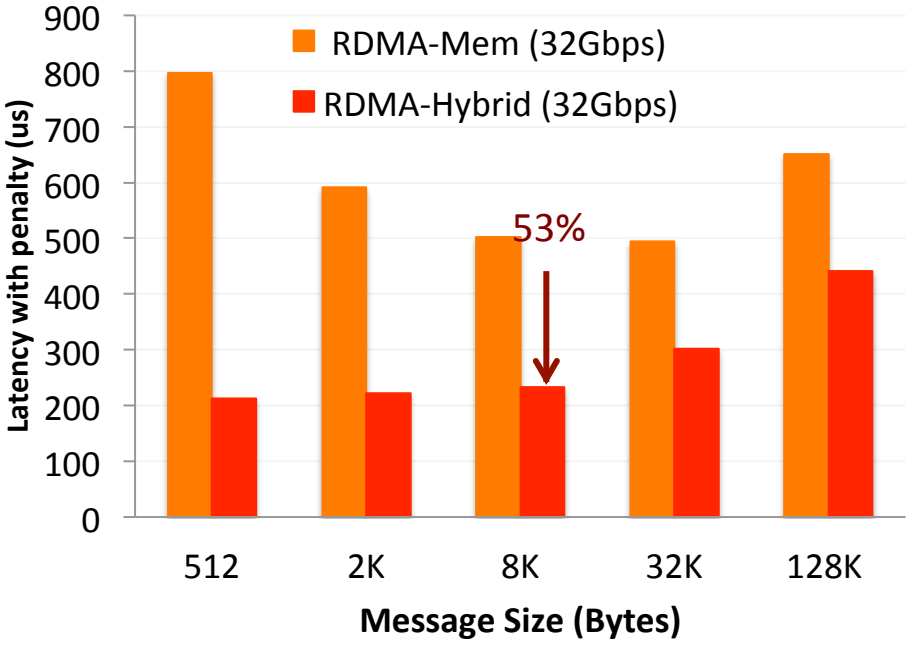
Performance Benefits on SDSC-Gordon – OHB Latency & Throughput Micro-Benchmarks

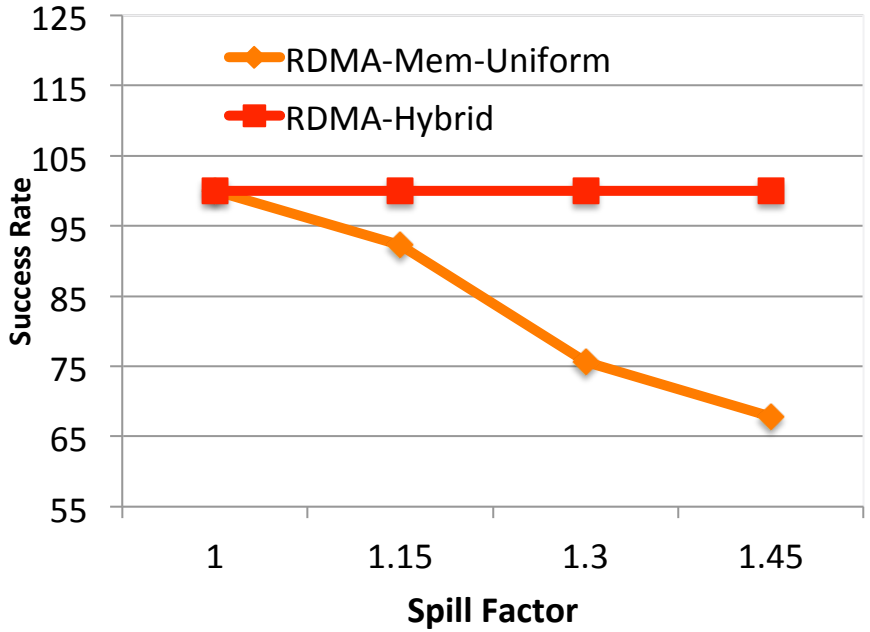




- ohb_memlat & ohb_memthr latency & throughput micro-benchmarks
- Memcached-RDMA can
  - improve query latency by up to 70% over IPoIB (32Gbps)
  - improve throughput by up to 2X over IPoIB (32Gbps)
  - No overhead in using hybrid mode when all data can fit in memory

Performance Benefits on OSU-RI-SSD – OHB Micro-benchmark for Hybrid Memcached





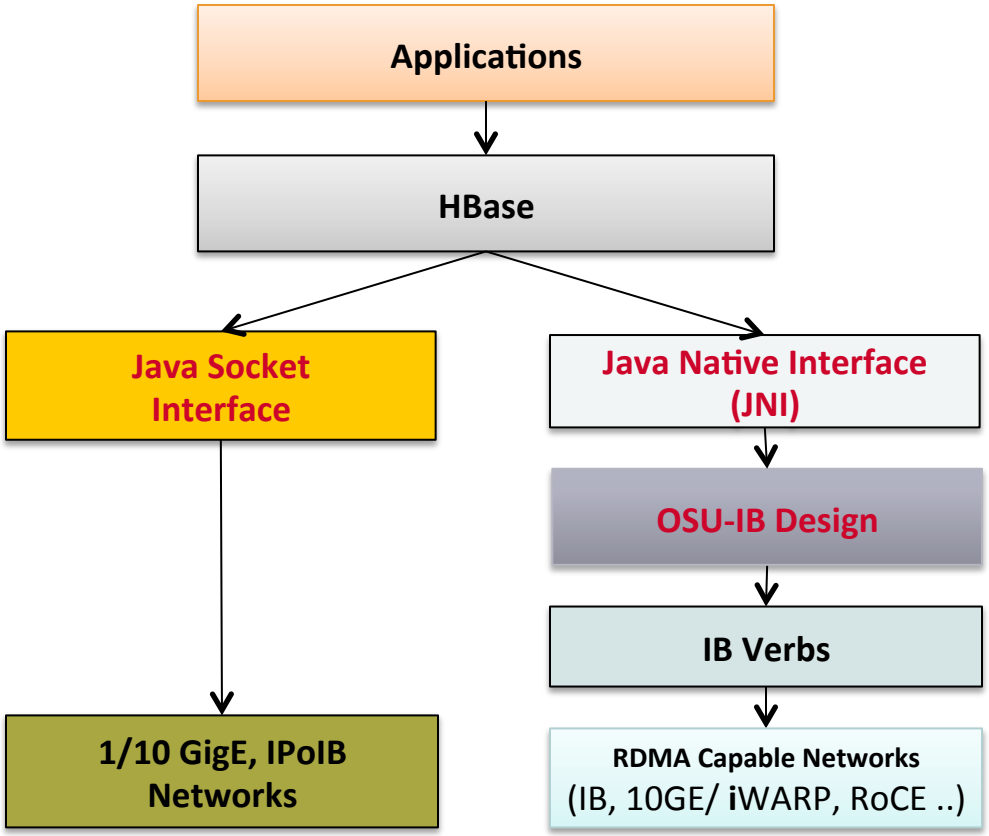
ohb_memhybrid – Uniform Access Pattern, single client and single server with 64MB

- Success Rate of In-Memory Vs. Hybrid SSD-Memory for different spill factors
  - 100% success rate for Hybrid design while that of pure In-memory degrades
- Average Latency with penalty for In-Memory Vs. Hybrid SSD-Assisted mode for spill factor 1.5.
  - up to 53% improvement over In-memory with server miss penalty as low as 1.5 ms

Presentation Outline

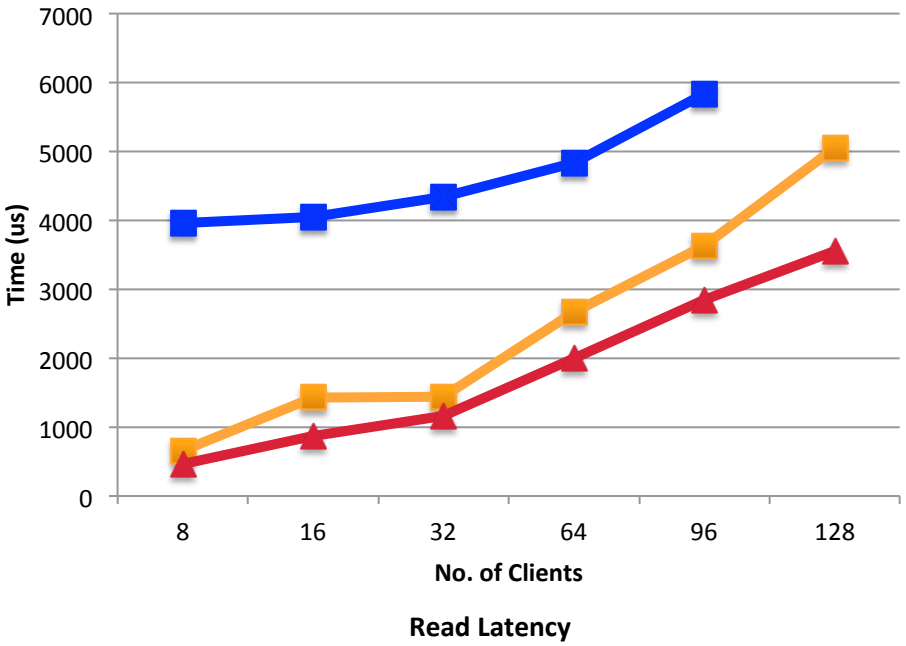
- Challenges for Accelerating Big Data Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - Sample Performance Numbers for Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - RDMA-based HBase
- Challenges in Designing Benchmarks for Big Data Processing
  - OSU HiBD Benchmarks
- Conclusion and Q&A

HBase-RDMA Design Overview

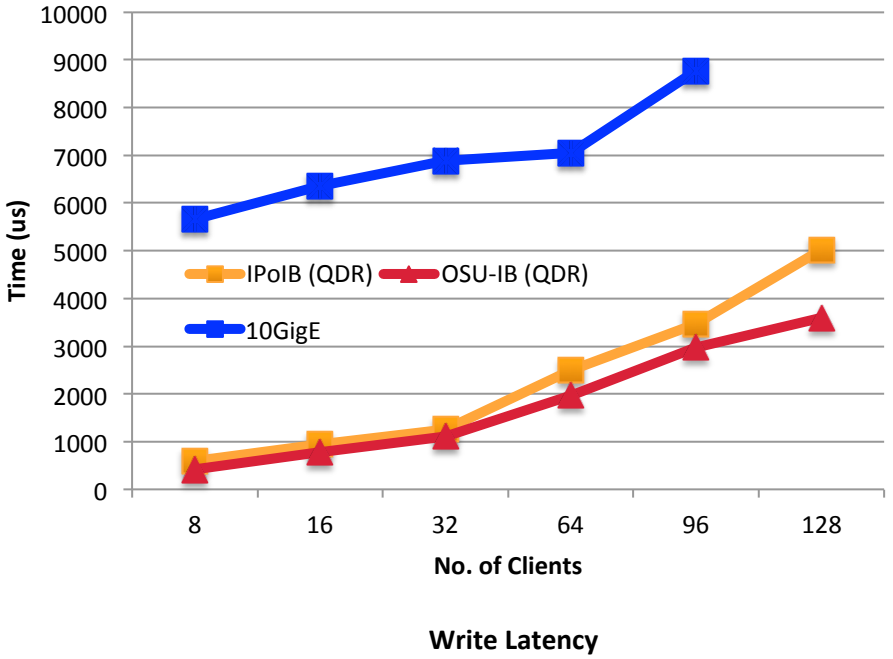


- JNI Layer bridges Java based HBase with communication library written in native code
- Enables high performance RDMA communication, while supporting traditional socket interface

HBase – YCSB Read-Write Workload



Read Latency



Write Latency

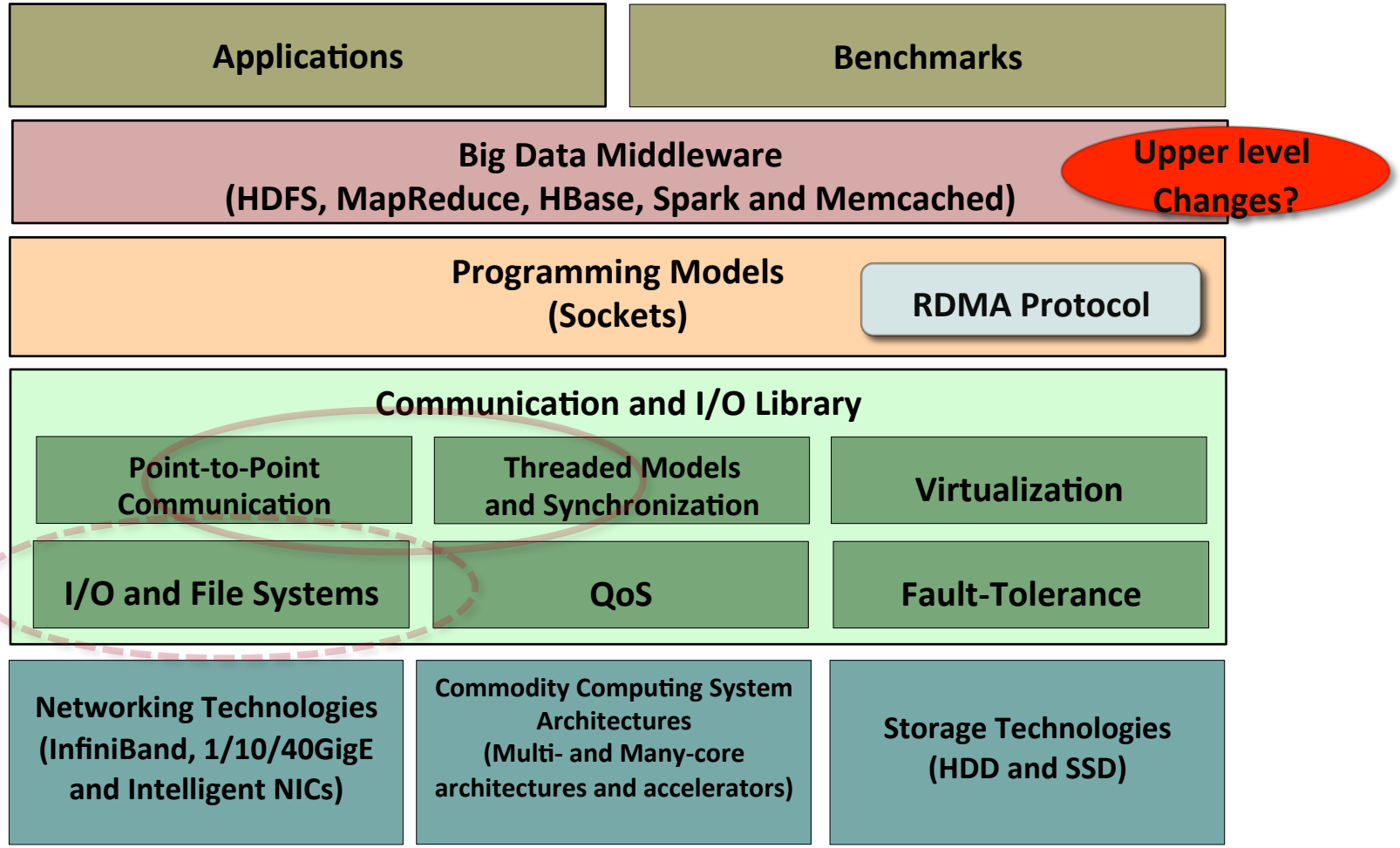
- HBase Get latency (Yahoo! Cloud Service Benchmark)
  - 64 clients: 2.0 ms; 128 Clients: 3.5 ms
  - 42% improvement over IPoIB for 128 clients
- HBase Put latency
  - 64 clients: 1.9 ms; 128 Clients: 3.5 ms
  - 40% improvement over IPoIB for 128 clients

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

Presentation Outline

- 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
  - Sample Performance Numbers for Hadoop 2.x 0.9.6 Release
- RDMA-based designs for Memcached and HBase
  - RDMA-based Memcached with SSD-assisted Hybrid Memory
  - 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



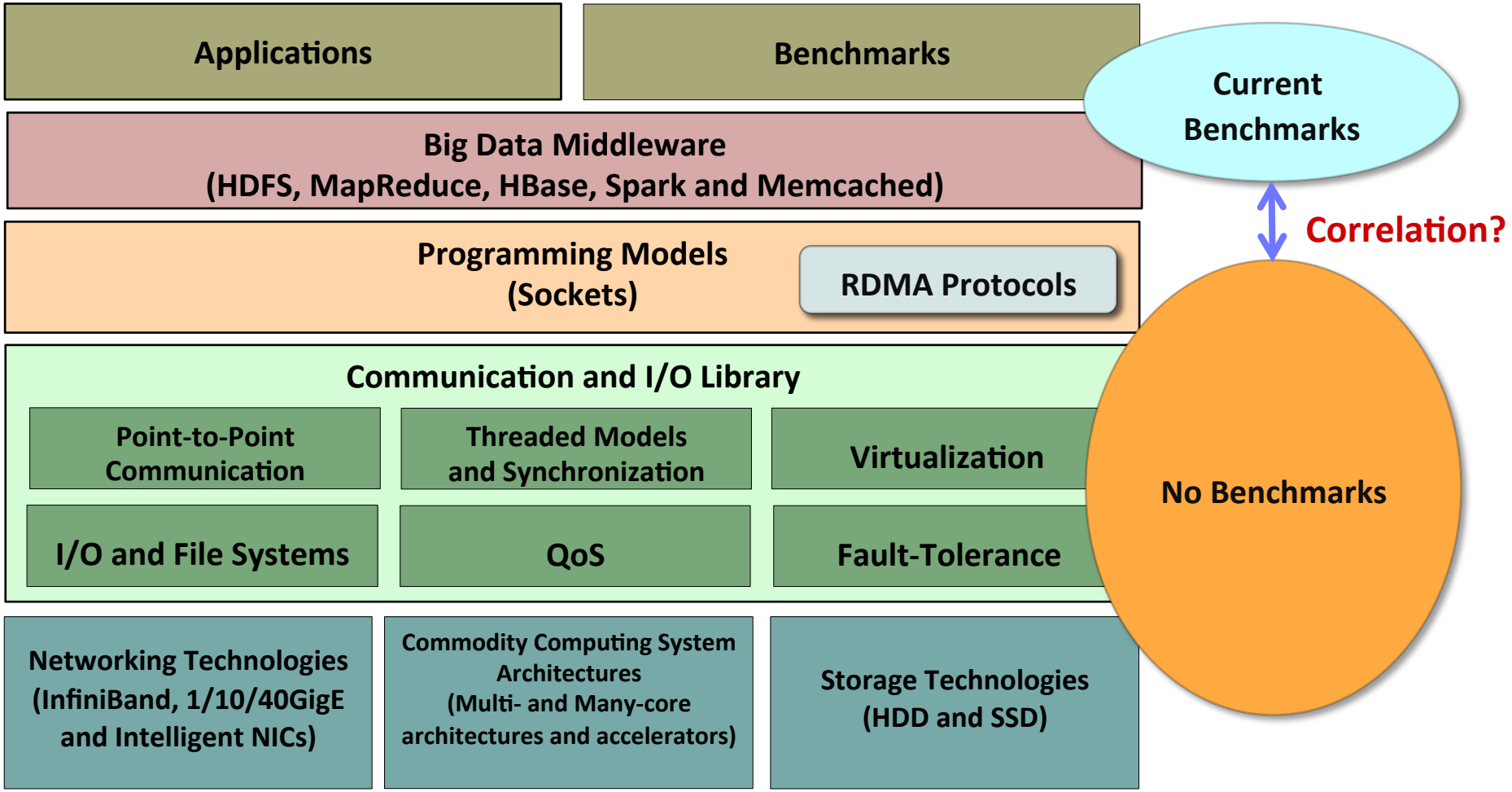
Are the Current Benchmarks Sufficient for Big Data Management and Processing?

- 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?

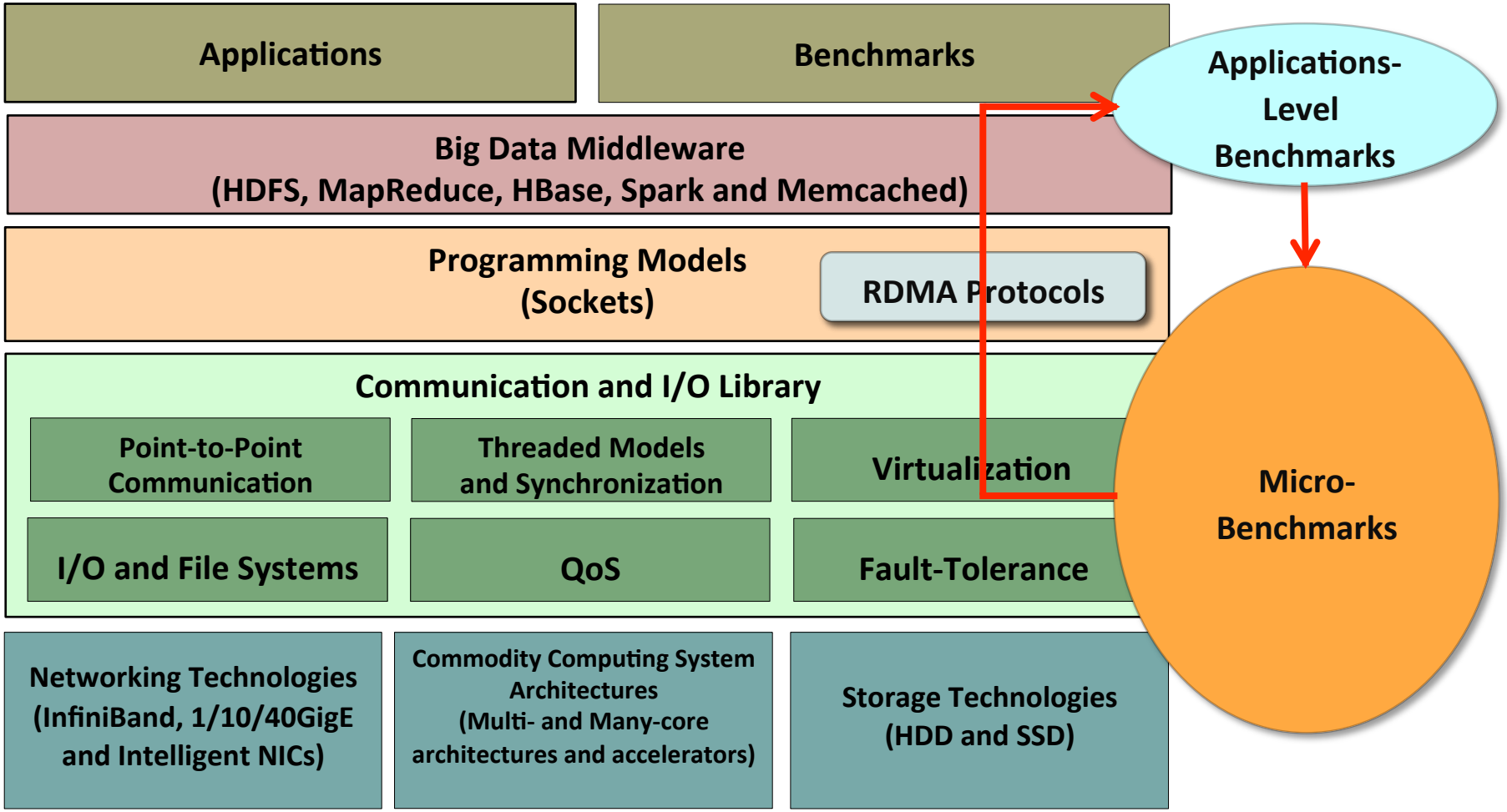
OSU MPI Micro-Benchmarks (OMB) Suite

- 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



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



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).

Future Plans of OSU High Performance Big Data Project

- Upcoming Releases of RDMA-enhanced Packages will support
  - Spark
  - HBase
  - Plugin-based designs
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  - HDFS
  - 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
- 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. Awan (Ph.D.)
- A. Bhat (M.S.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- N. Islam (Ph.D.)
- M. Li (Ph.D.)
- M. Rahman (Ph.D.)
- D. Shankar (Ph.D.)
- A. Venkatesh (Ph.D.)
- J. Zhang (Ph.D.)

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.)

Past Post-Docs

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

Current Senior Research Associates

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

Current Post-Doc

- J. Lin

Current Programmer

- J. Perkins

Current Research Specialist

- M. Arnold

Past Research Scientist

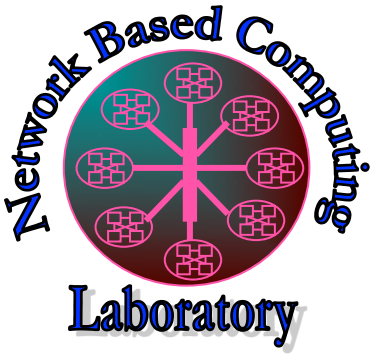
- S. Sur

Past Programmers

- D. Bureddy
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (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.)

Thank You!

panda@cse.ohio-state.edu



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

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

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

Call For Participation

International Workshop on High-Performance Big Data Computing (HPBDC 2015)

In conjunction with International Conference on Distributed Computing Systems (ICDCS 2015)

In Hilton Downtown, Columbus, Ohio, USA, Monday, June 29th, 2015

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

Multiple Positions Available in My Group

- Looking for Bright and Enthusiastic Personnel to join as
  - Post-Doctoral Researchers
  - PhD Students
  - Hadoop/Big Data Programmer/Software Engineer
  - MPI Programmer/Software Engineer
- If interested, please contact me at this conference and/or send an e-mail to panda@cse.ohio-state.edu