High Performance Big Data (HiBD): Accelerating Hadoop, Spark and Memcached on Modern Clusters

Talk at Mellanox Theatre (SC '15)

by

Dhabaleswar K. (DK) Panda

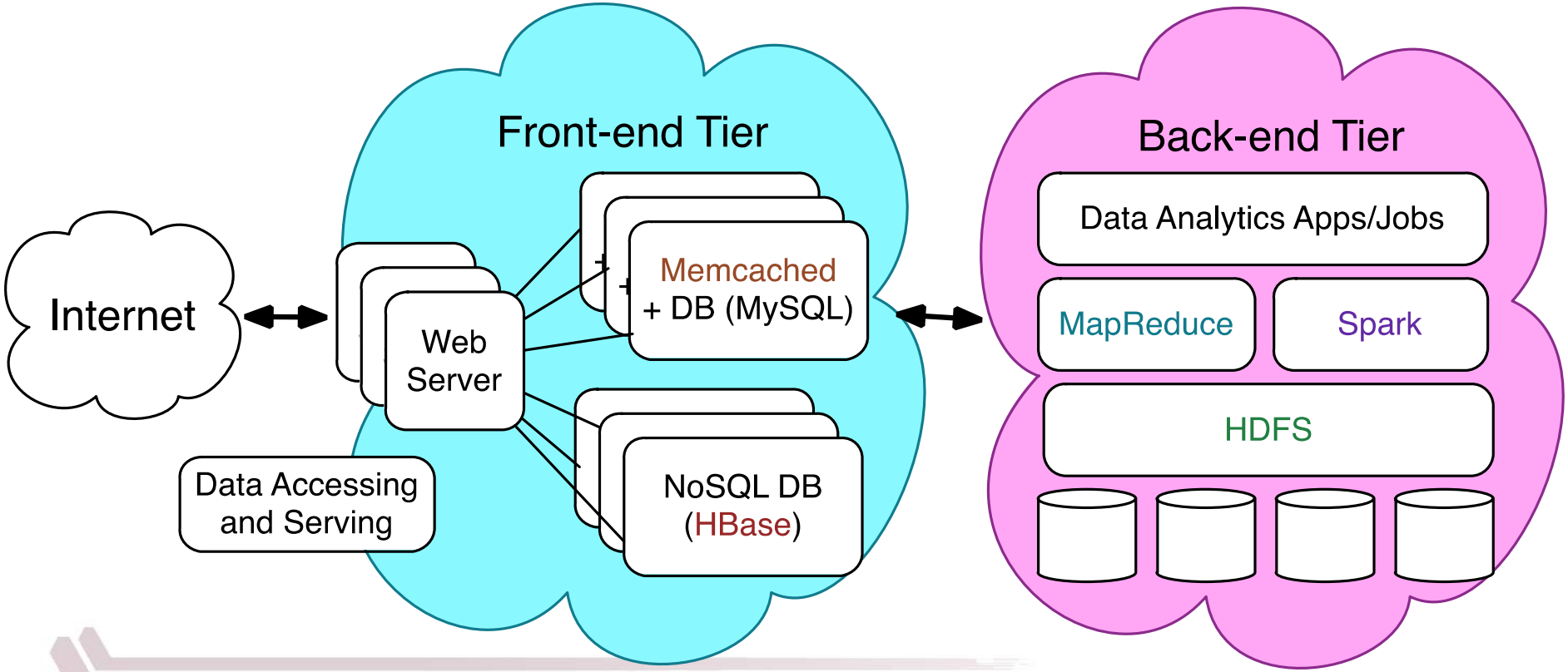
The Ohio State University

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

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

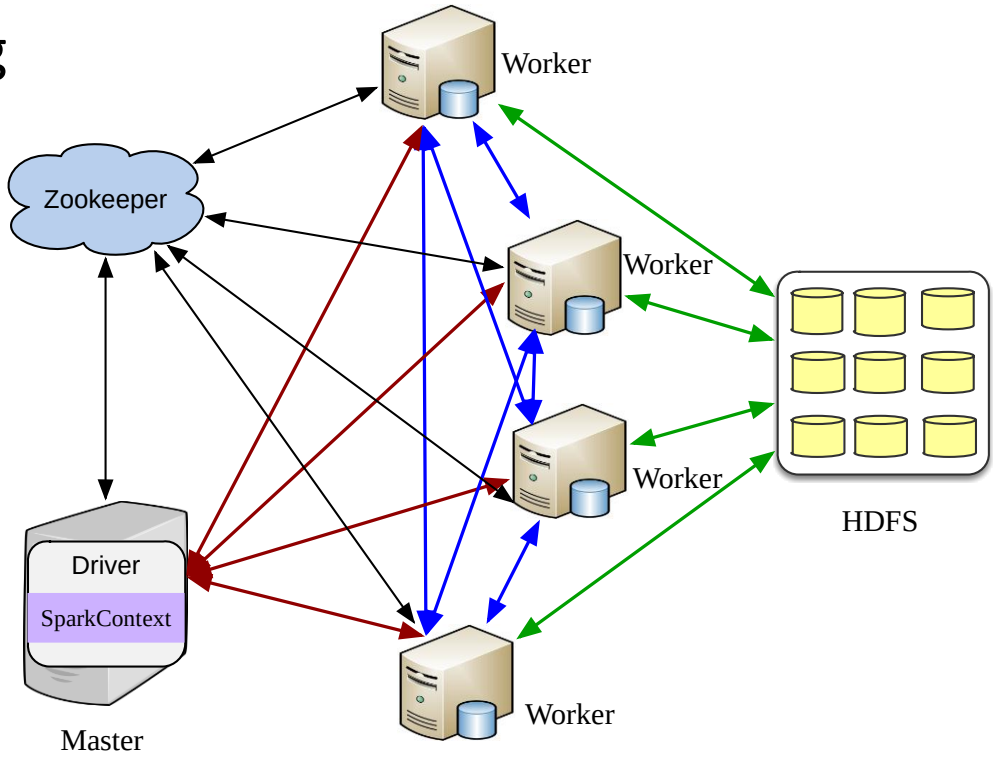
Data Processing and Management on Modern Clusters

- 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



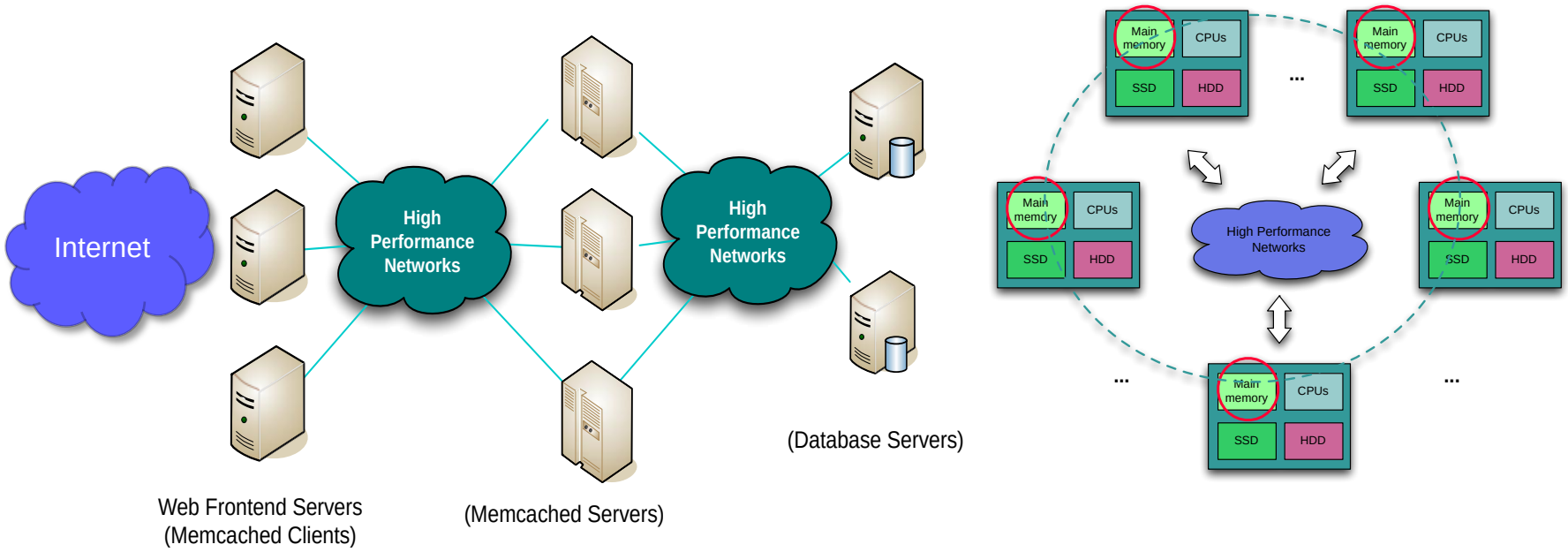
Spark - An Example of Back-end Data Processing Middleware

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



http://spark.apache.org

Memcached - An Example of Front-end Data Management Middleware



- 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

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
  - Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
  - Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), NVMe-SSD
  - Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)





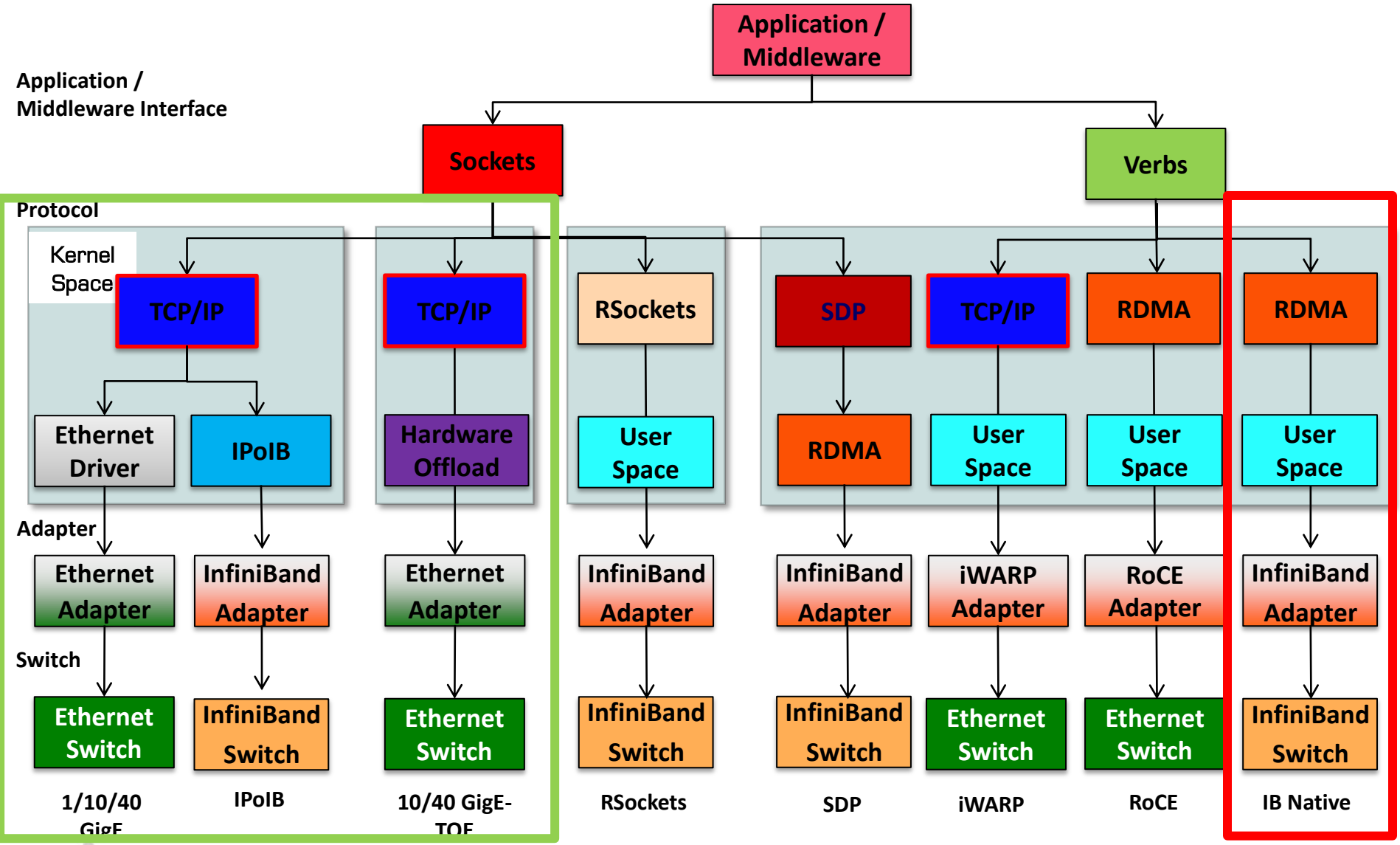
Tianhe – 2


Titan


Stampede


Tianhe – 1A

Interconnects and Protocols in the Open Fabrics Stack



Presentation Outline

- Challenges for Accelerating Big Data Processing
- Accelerating Big Data Processing on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for HDFS, MapReduce, Spark, and Memcached
- Accelerating Big Data Processing on High-Performance Storage
  - Enhanced Designs for HDFS and MapReduce over Lustre
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- The High-Performance Big Data (HiBD) Project
- Conclusion and Q&A

How Can HPC Clusters with High-Performance Interconnect and Storage Architectures Benefit Big Data Applications?

Can the bottlenecks be alleviated with new designs by taking advantage of HPC technologies?

Can RDMA-enabled high-performance interconnects benefit Big Data processing?

Can HPC Clusters with high-performance storage systems (e.g. SSD, parallel file systems) benefit Big Data applications?

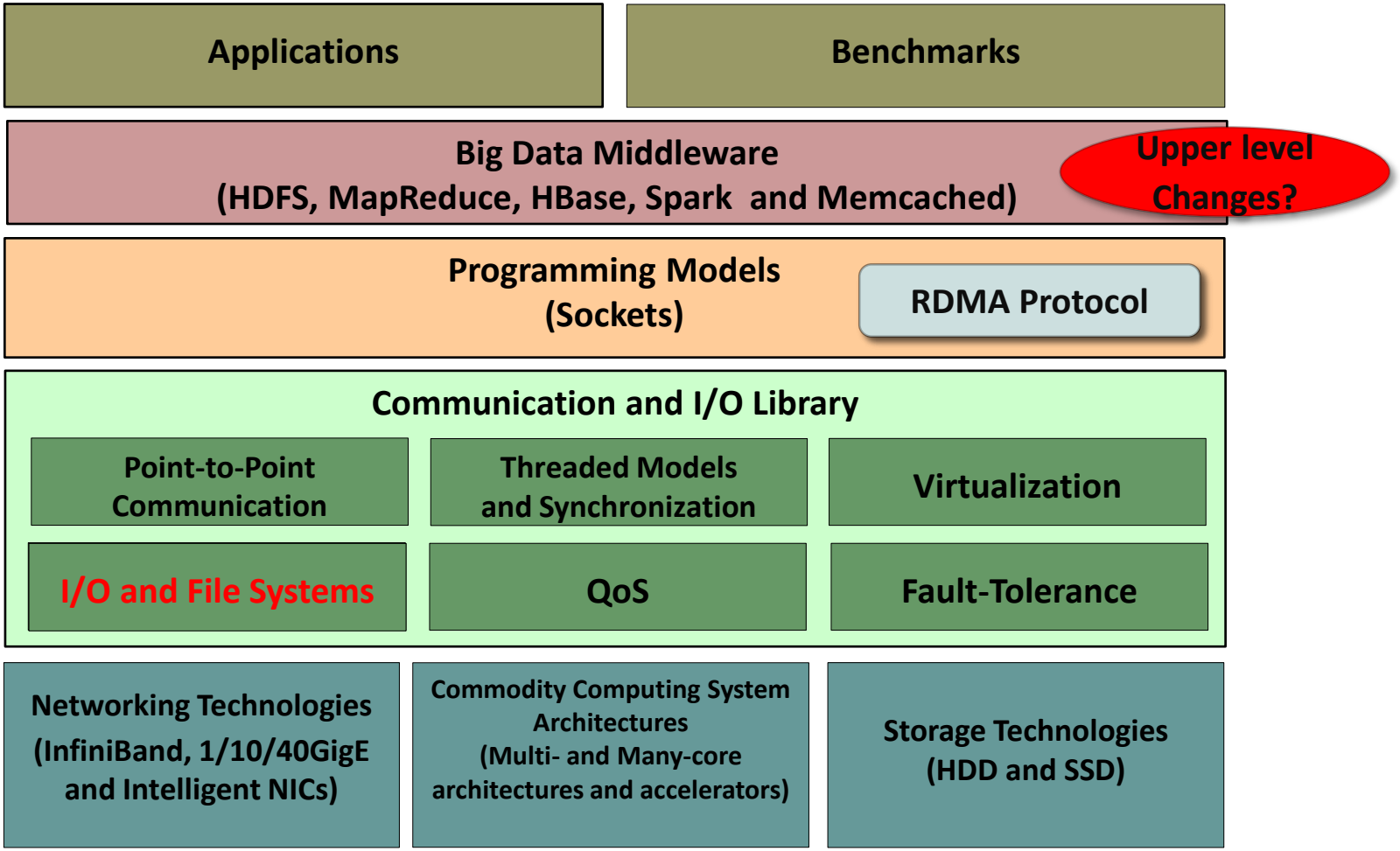
How much performance benefits can be achieved through enhanced designs?

What are the major bottlenecks in current Big Data processing middleware (e.g. Hadoop, Spark, and Memcached)?

How to design benchmarks for evaluating the performance of Big Data middleware on HPC clusters?

Bring HPC and Big Data processing into a "convergent trajectory"!

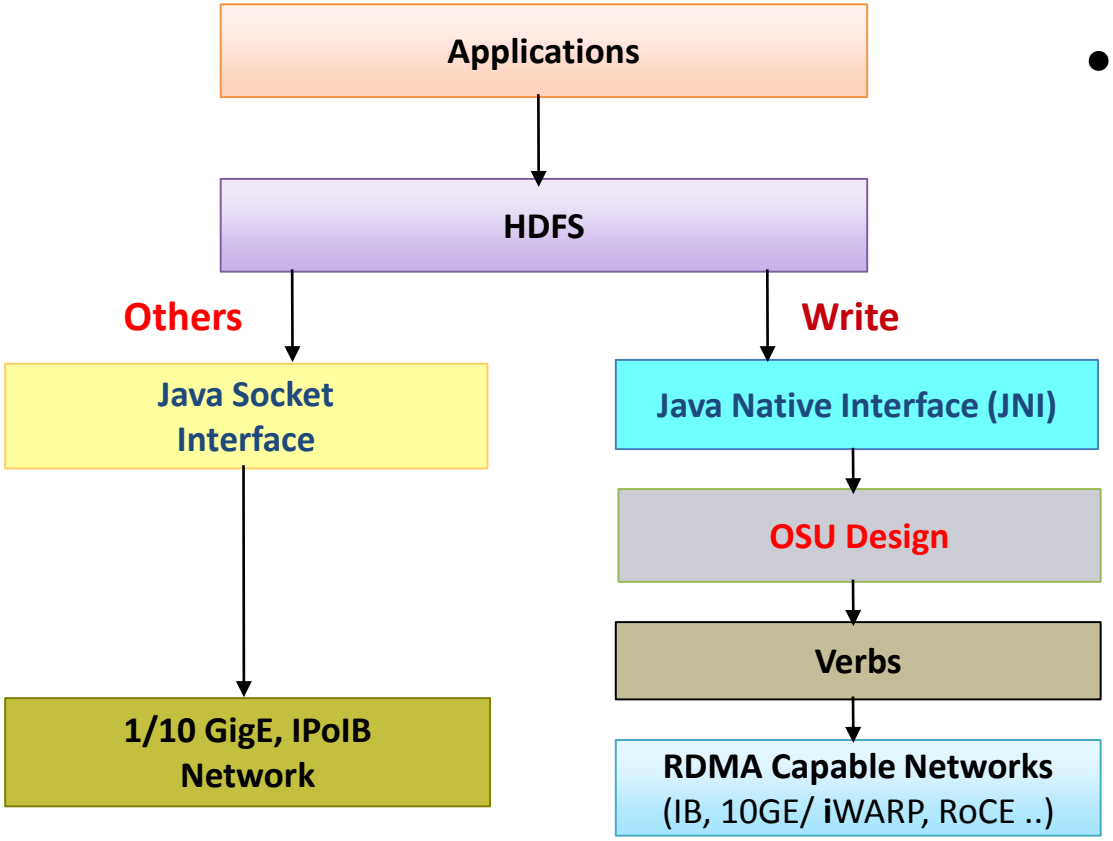
Challenges in Designing Communication and I/O Libraries for Big Data Systems



Presentation Outline

- Challenges for Accelerating Big Data Processing
- Accelerating Big Data Processing on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for HDFS, MapReduce, Spark, and Memcached
- Accelerating Big Data Processing on High-Performance Storage
  - Enhanced Designs for HDFS and MapReduce over Lustre
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- The High-Performance Big Data (HiBD) Project
- Conclusion and Q&A

Design Overview of HDFS with RDMA

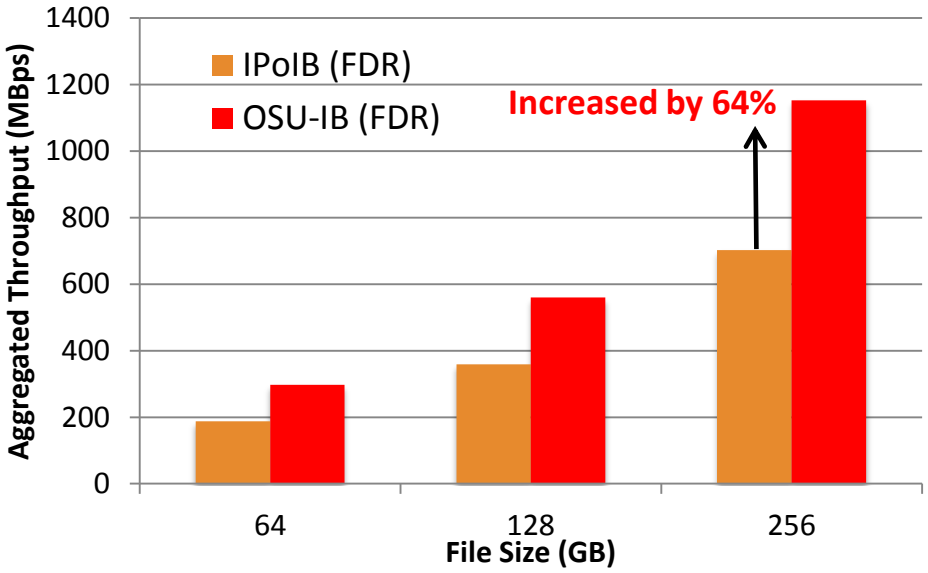


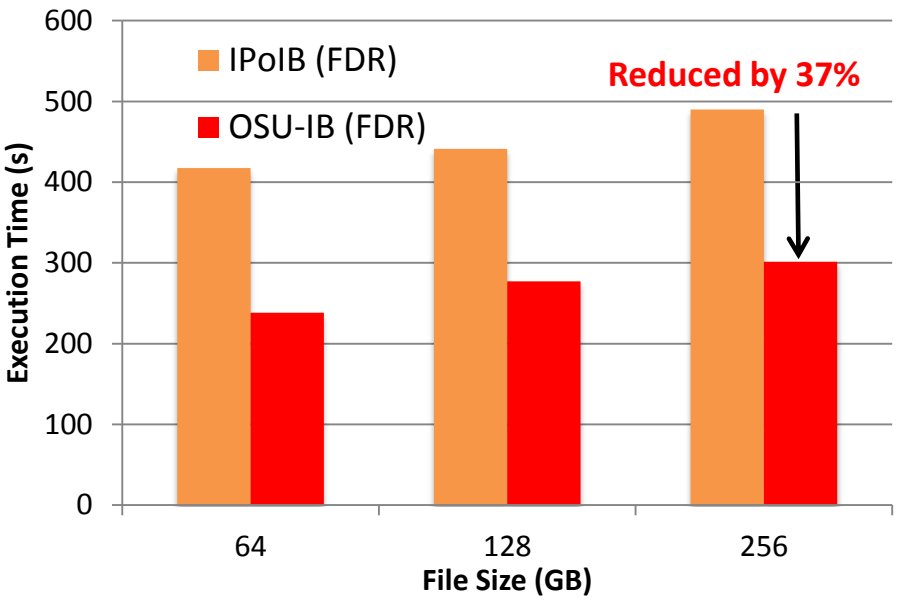
- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - Parallel replication support
  - On-demand connection setup
  - InfiniBand/RoCE support

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

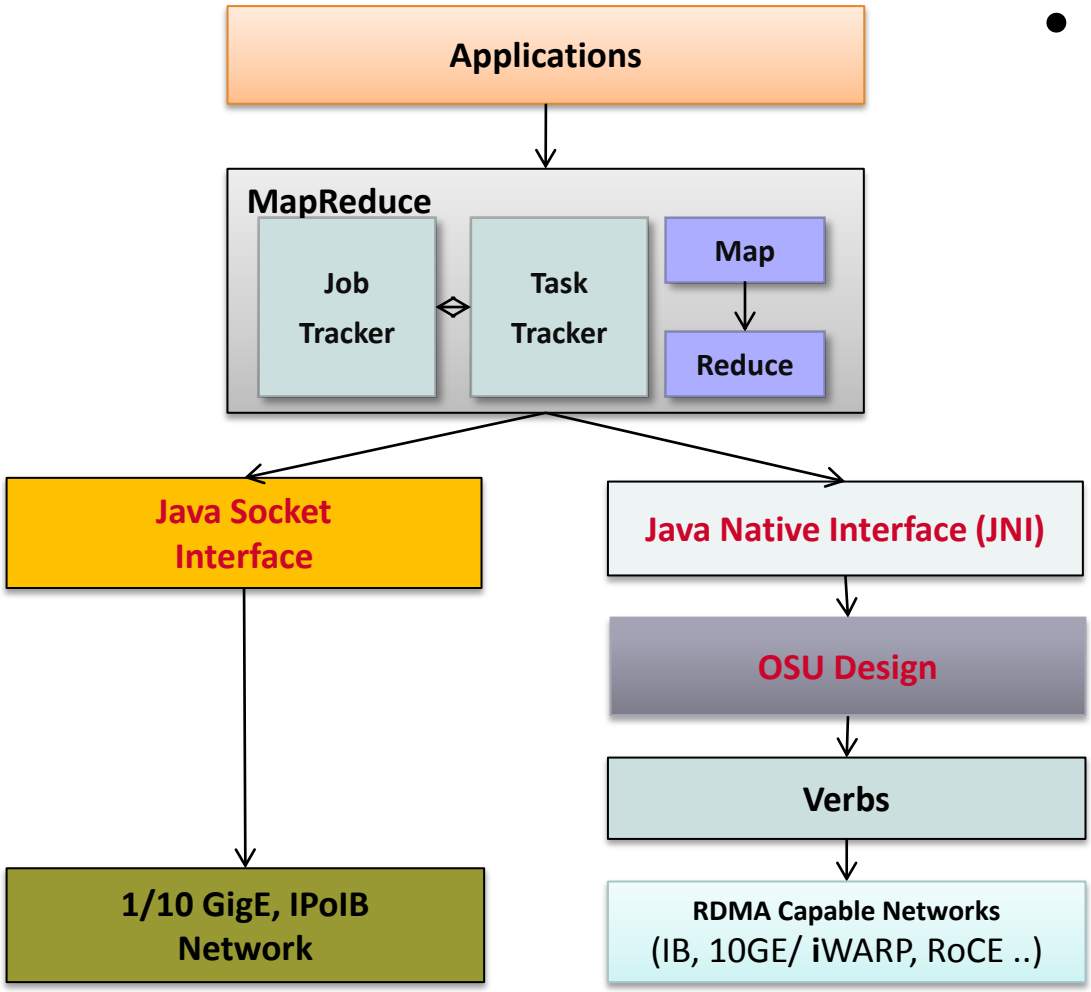
Evaluations using Enhanced DFSIO of Intel HiBench on TACC-Stampede





- Cluster with 64 DataNodes, 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

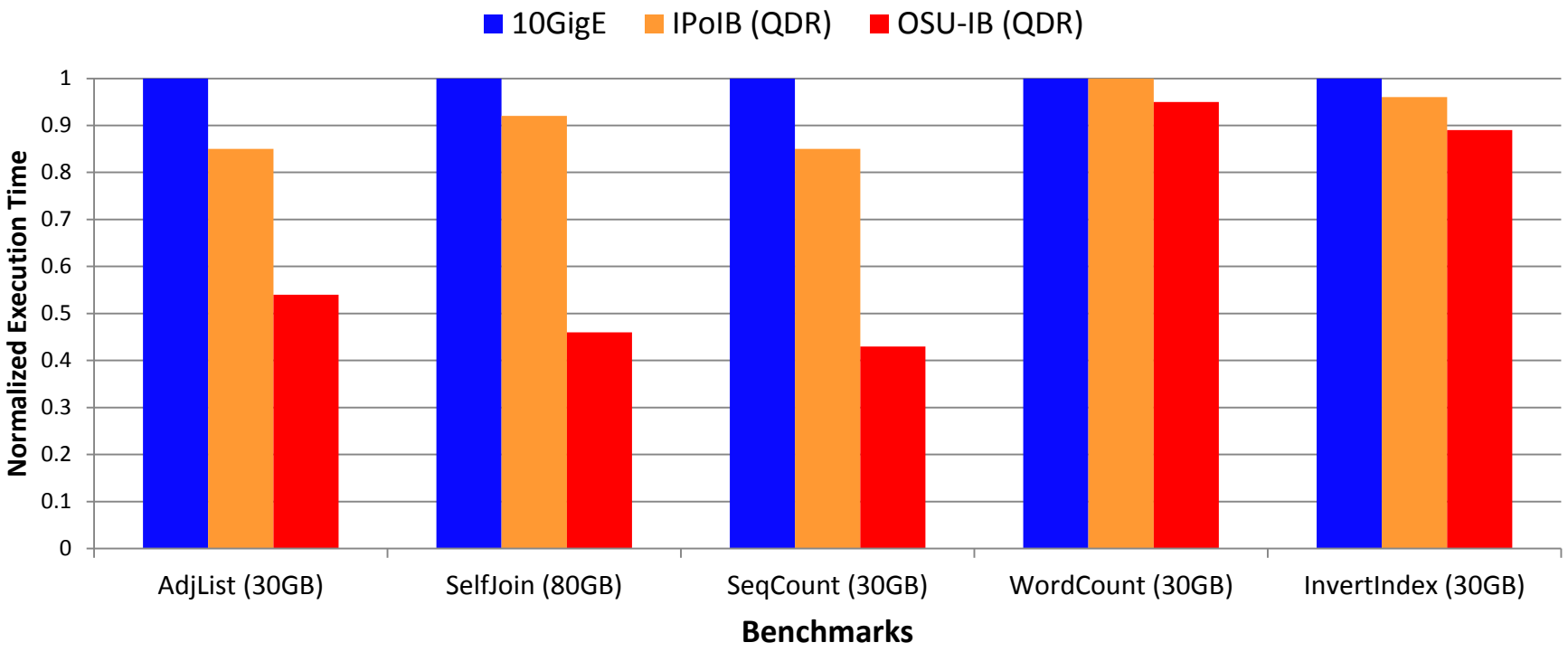
Design Overview of MapReduce with RDMA



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

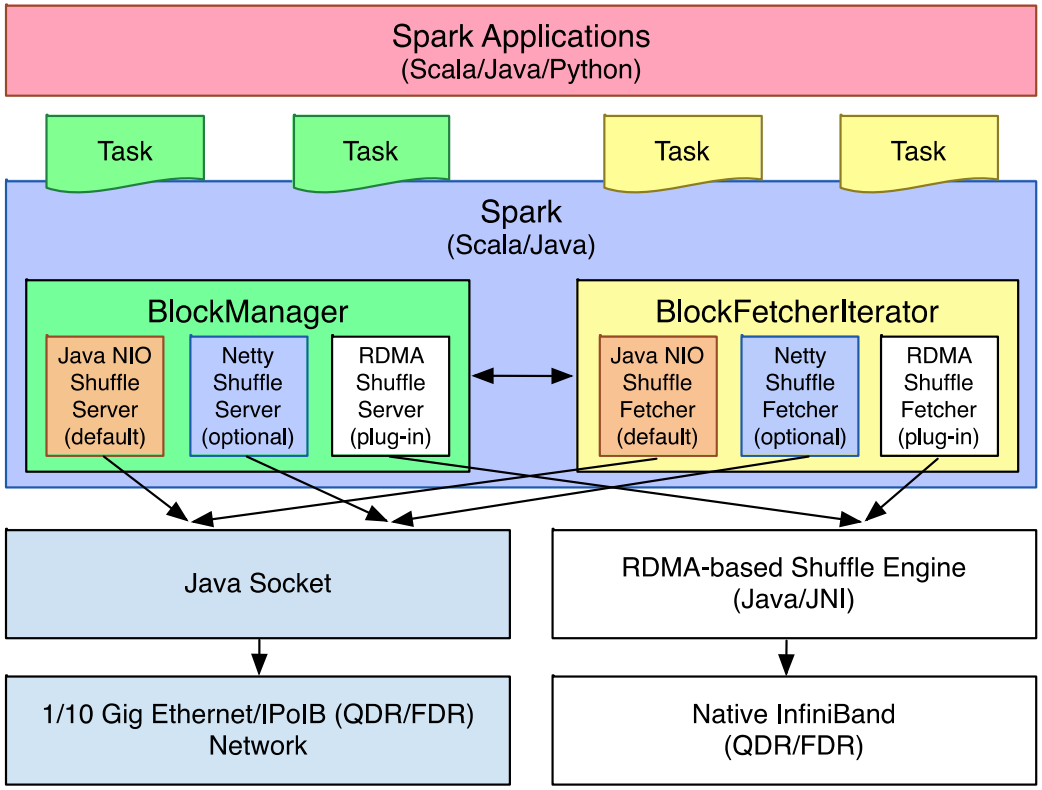
M. W. Rahman, X. Lu, N. S. Islam, and D. K. Panda, HOMR: A Hybrid Approach to Exploit Maximum Overlapping in MapReduce over High Performance Interconnects, ICS, June 2014.

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

Design Overview of Spark with RDMA

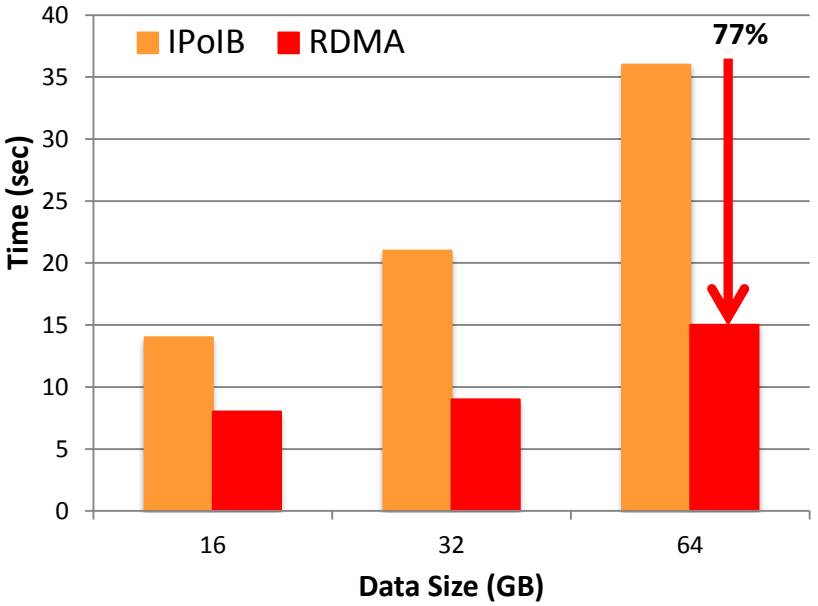


- Design Features
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking and out-of-order data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

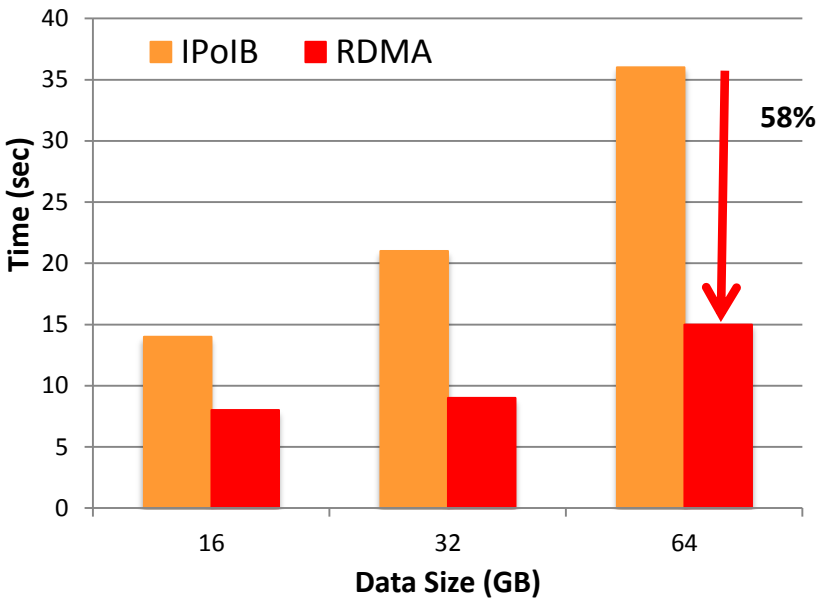
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

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

Performance Evaluation on TACC Stampede - SortByTest



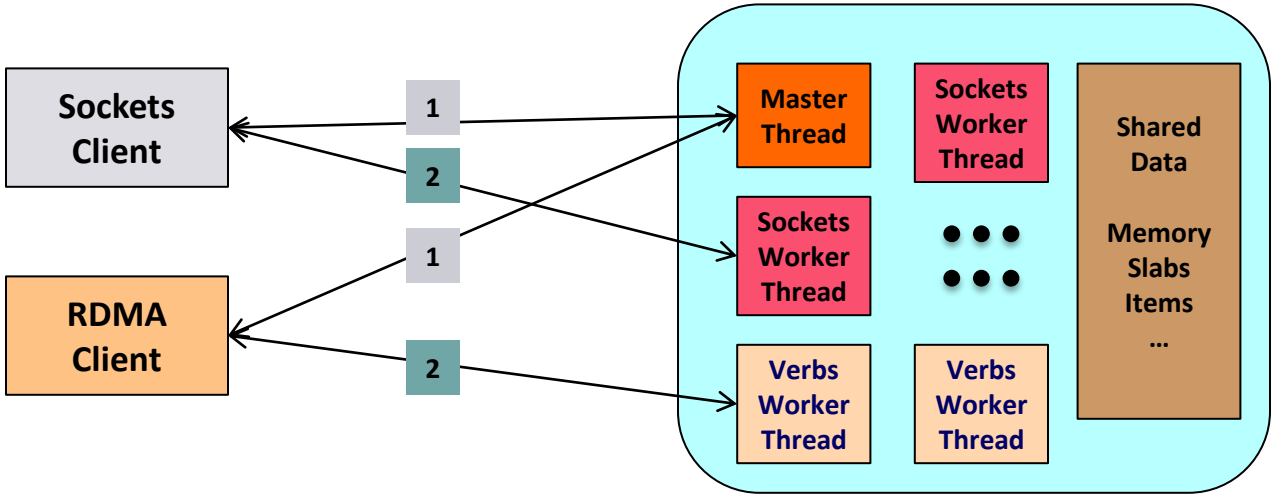
16 Worker Nodes, SortByTest Shuffle Time



16 Worker Nodes, SortByTest Total Time

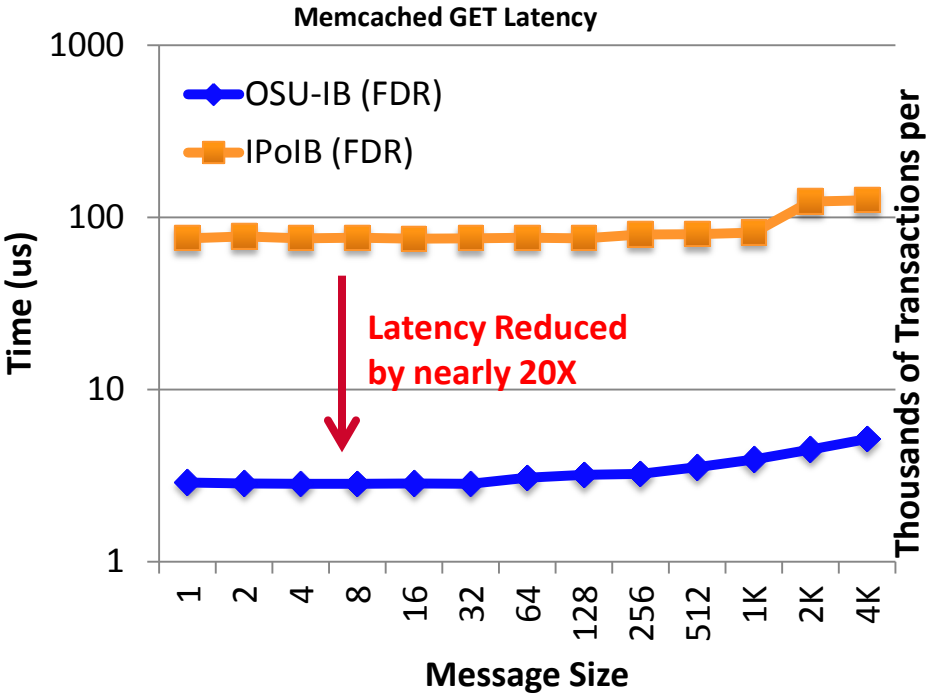
- Intel SandyBridge + FDR, 16 Worker Nodes, 256 Cores, (256M 256R)
- RDMA-based design for Spark 1.4.0
- RDMA vs. IPoIB with 256 concurrent tasks, single disk per node and RamDisk. For SortByKey Test:
  - Shuffle time reduced by up to 77% over IPoIB (56Gbps)
  - Total time reduced by up to 58% over IPoIB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

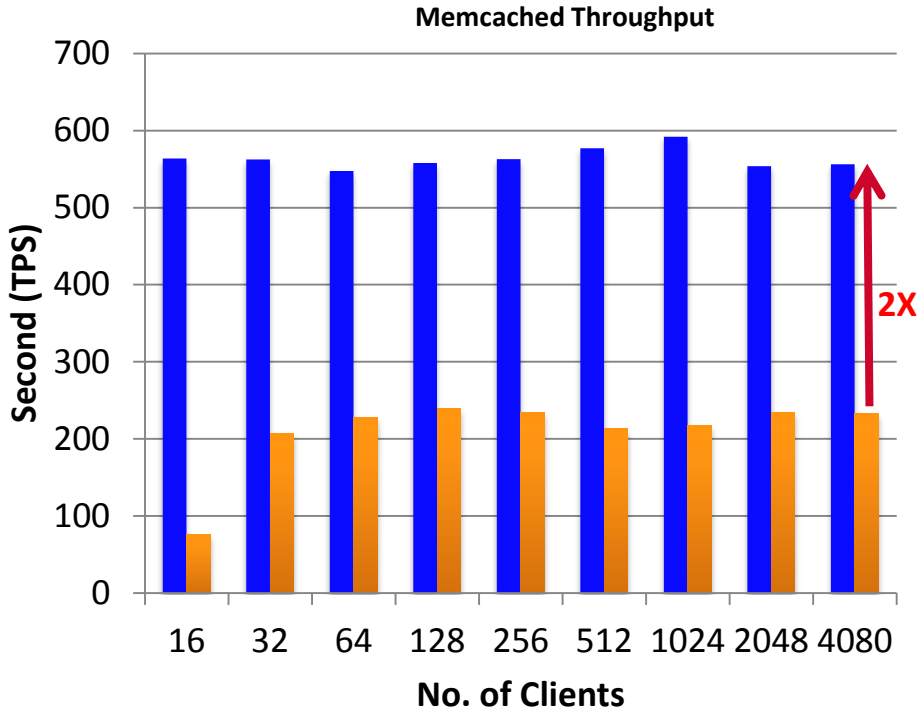
Memcached-RDMA Design



- 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 (TACC Stampede)





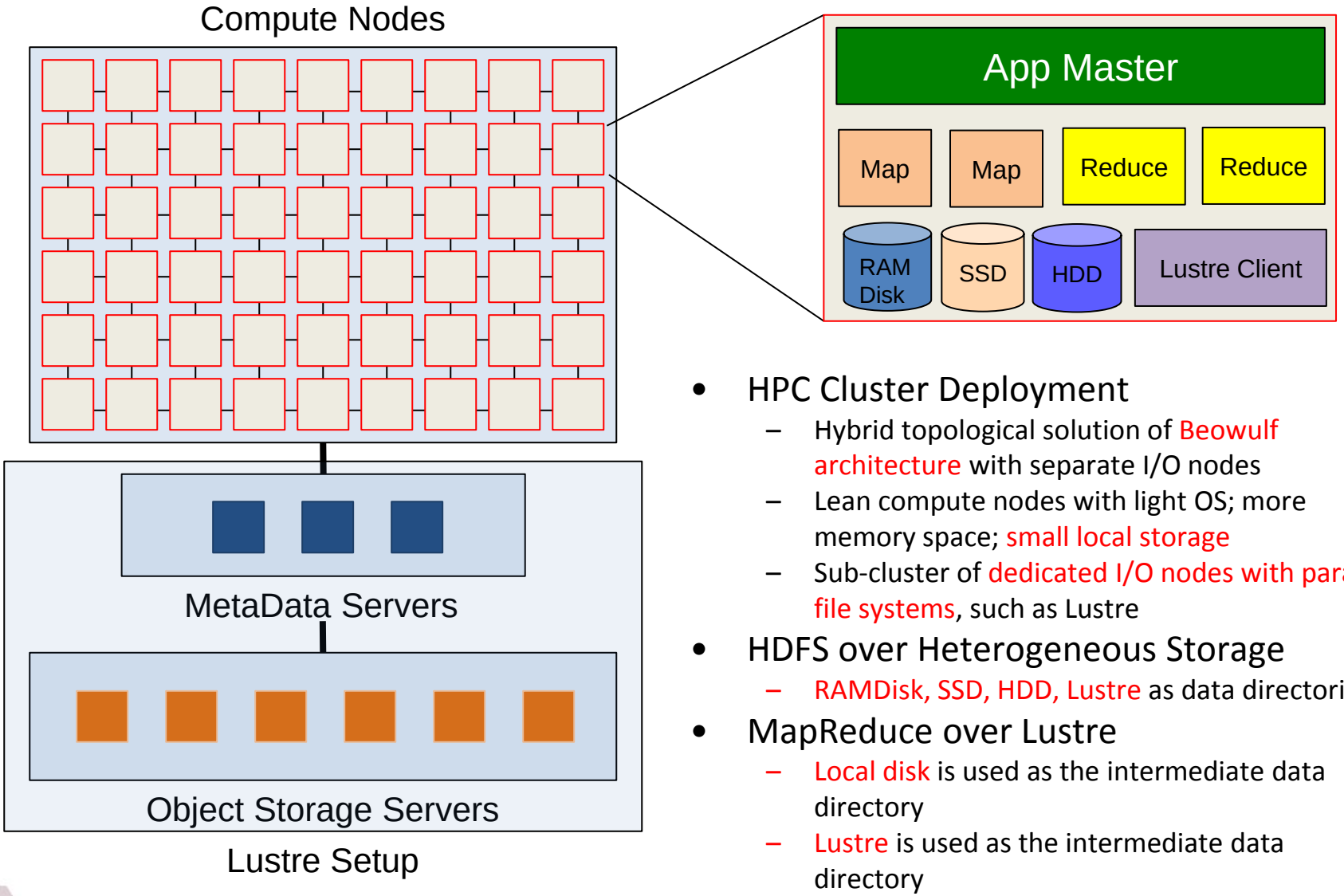
Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- 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

Presentation Outline

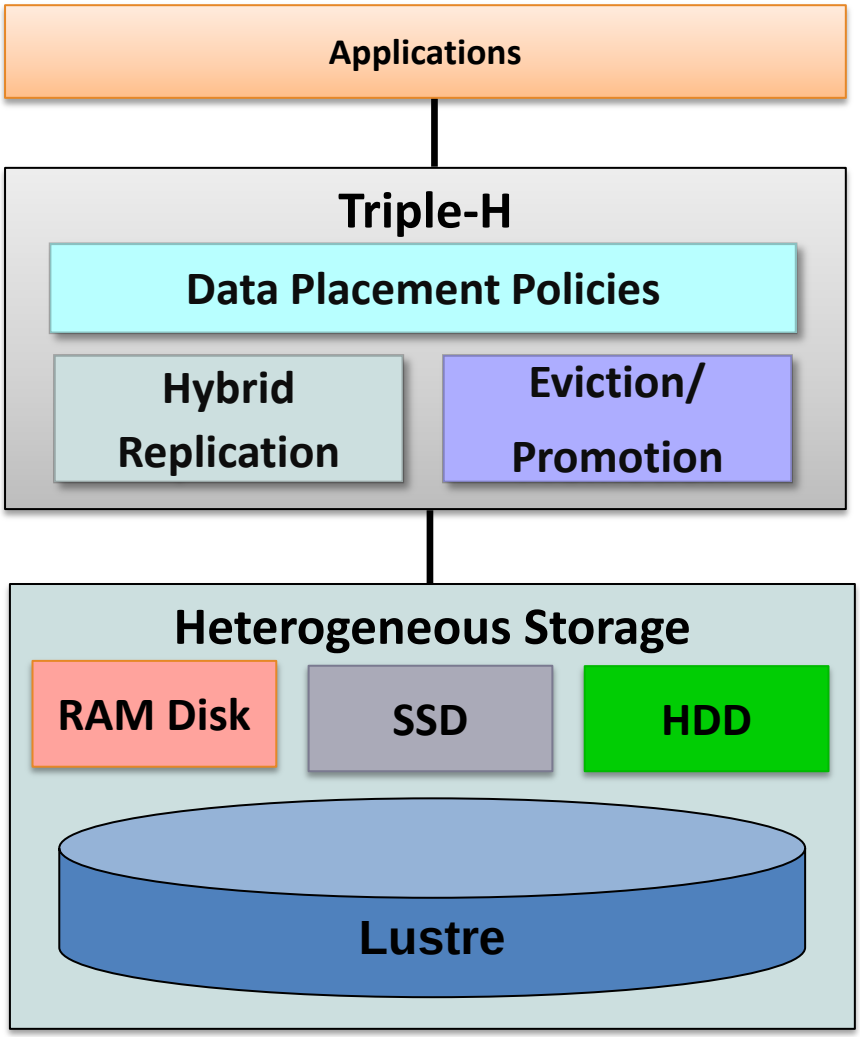
- Challenges for Accelerating Big Data Processing
- Accelerating Big Data Processing on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for HDFS, MapReduce, Spark, and Memcached
- Accelerating Big Data Processing on High-Performance Storage
  - Enhanced Designs for HDFS and MapReduce over Lustre
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- The High-Performance Big Data (HiBD) Project
- Conclusion and Q&A

Optimize I/O Performance for Big Data Applications with High-Performance Storage on HPC Clusters



- 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
- HDFS over Heterogeneous Storage
  - RAMDisk, SSD, HDD, Lustre as data directories
- MapReduce over Lustre
  - Local disk is used as the intermediate data directory
  - Lustre is used as the intermediate data directory
- Hybrid Memcached with RAM + SSD

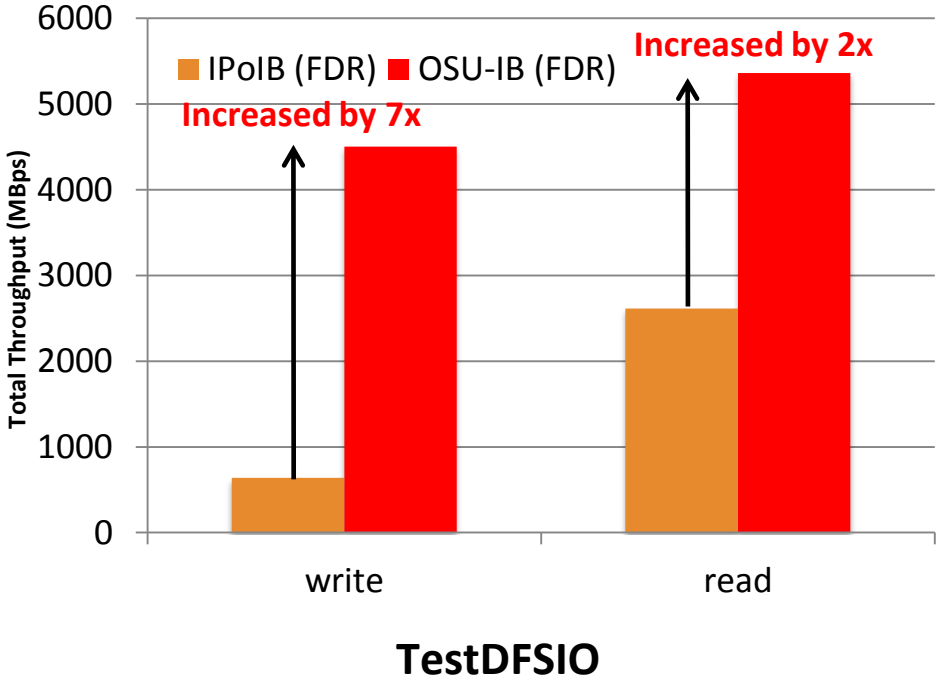
Enhanced HDFS with In-memory and Heterogeneous Storage

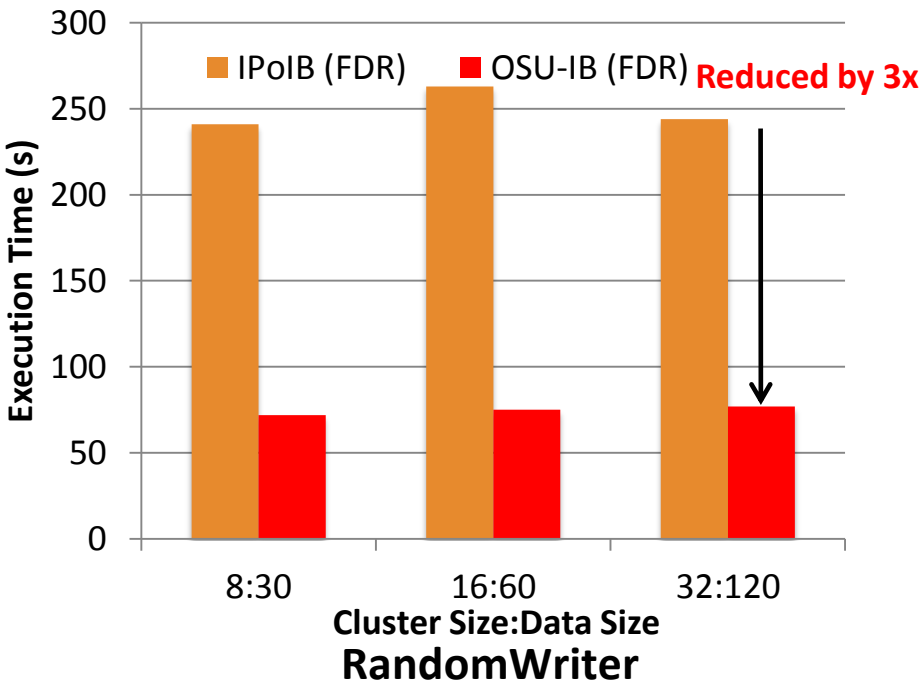


- Design Features
  - Three modes
    - Default (HHH)
    - In-Memory (HHH-M)
    - Lustre-Integrated (HHH-L)
  - Policies to efficiently utilize the heterogeneous storage devices
    - RAM, SSD, HDD, Lustre
  - Eviction/Promotion based on data usage pattern
  - Hybrid Replication
  - Lustre-Integrated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

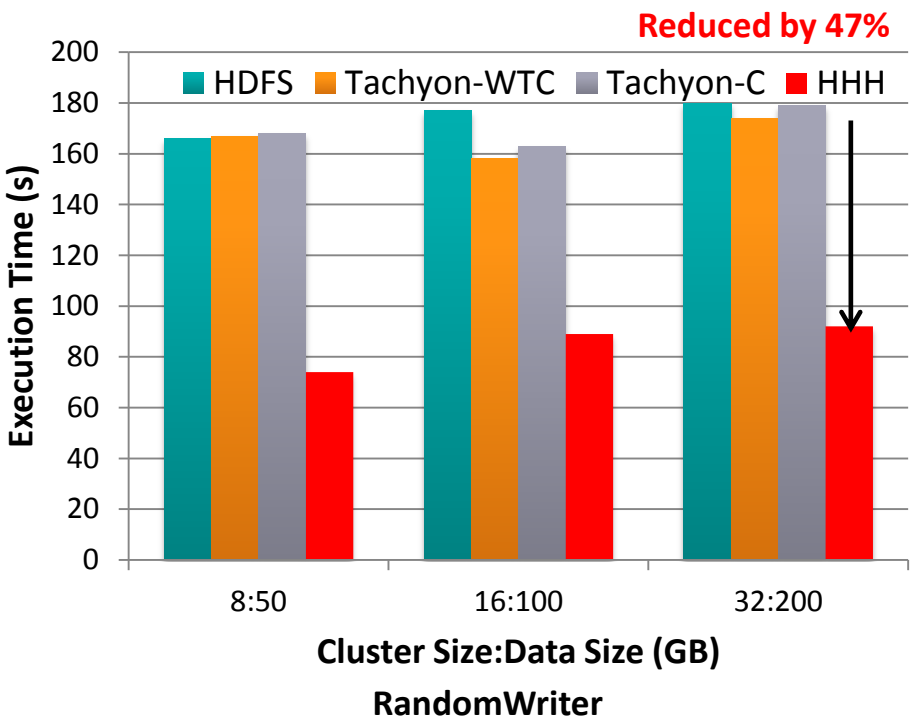
Performance Improvement on TACC Stampede (HHH)





- 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)

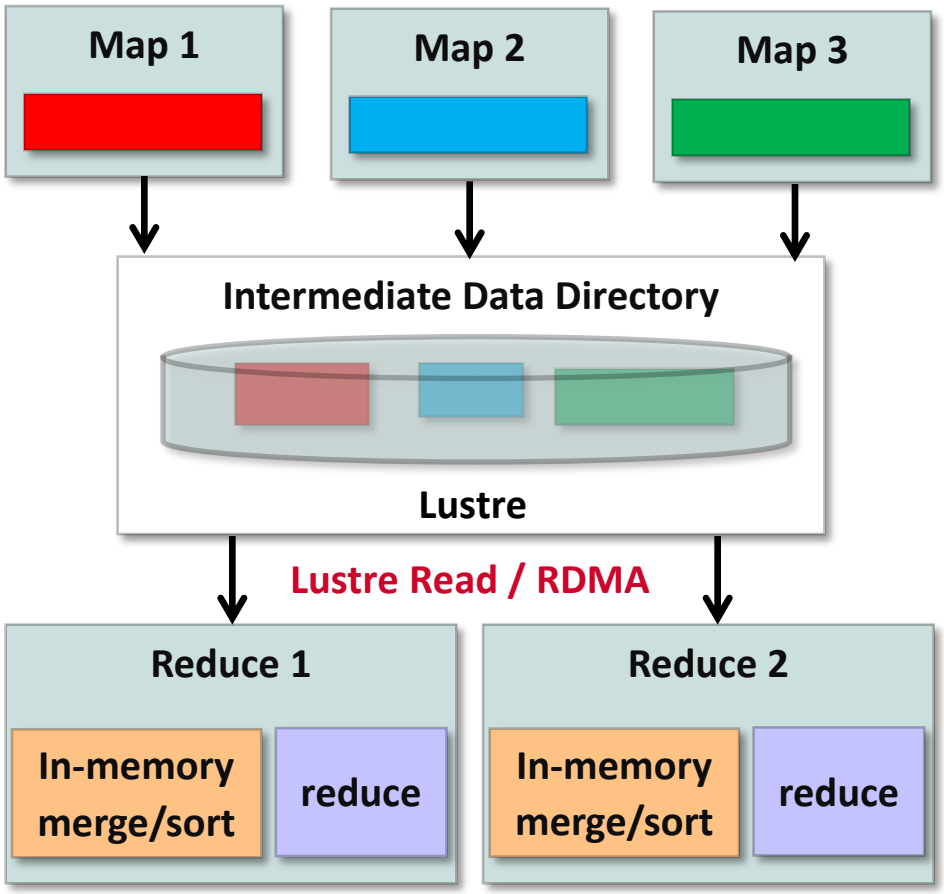
Performance Improvement on SDSC Gordon (HHH vs. Tachyon)



- RandomWriter: 200GB on 32 SSD Nodes (QDR)
  - HHH reduces the execution time by **47%** over Tachyon-WTC (IPoIB)), **56%** over HDFS (IPoIB)
- Sort: 200GB on 32 SSD Nodes (QDR)
  - HHH reduces the execution time by **19%** over Tachyon-WTC (IPoIB), **31%** over HDFS (IPoIB)

N. Islam, M. W. Rahman, X. Lu, D. Shankar, and D. K. Panda, Performance Characterization and Acceleration of In-Memory File Systems for Hadoop and Spark Applications on HPC Clusters, IEEE BigData '15, October 2015

Design Overview of Shuffle Strategies for MapReduce over Lustre

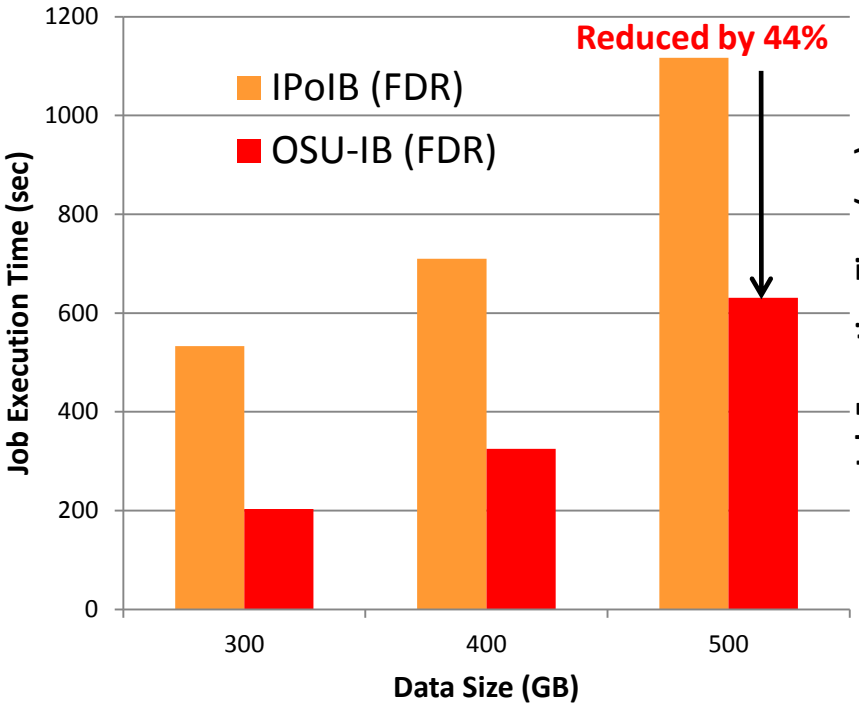


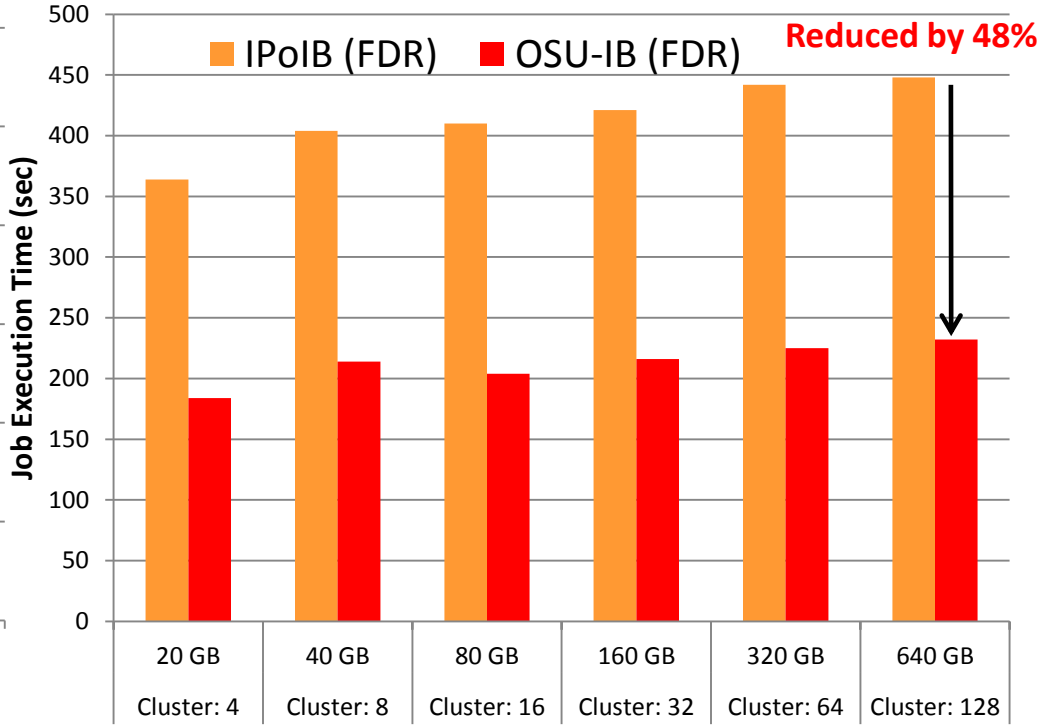
- Design Features
  - Two shuffle approaches
    - Lustre read based shuffle
    - RDMA based shuffle
  - Hybrid shuffle algorithm to take benefit from both shuffle approaches
  - Dynamically adapts to the better shuffle approach for each shuffle request based on profiling values for each Lustre read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



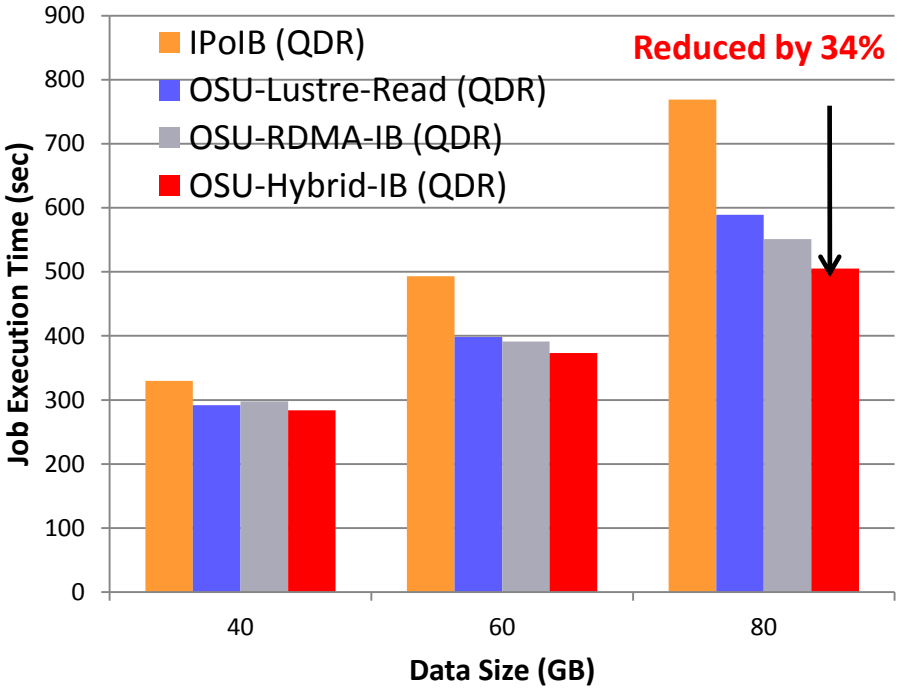


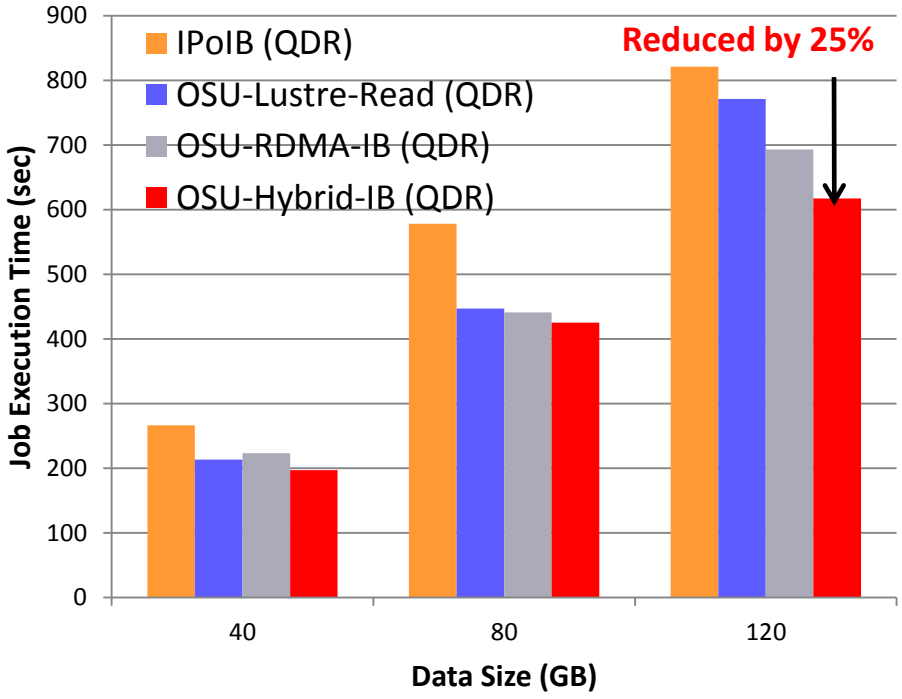
- For 500GB Sort in 64 nodes
  - 44% improvement over IPoIB (FDR)
- For 640GB Sort in 128 nodes
  - 48% improvement over IPoIB (FDR)

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

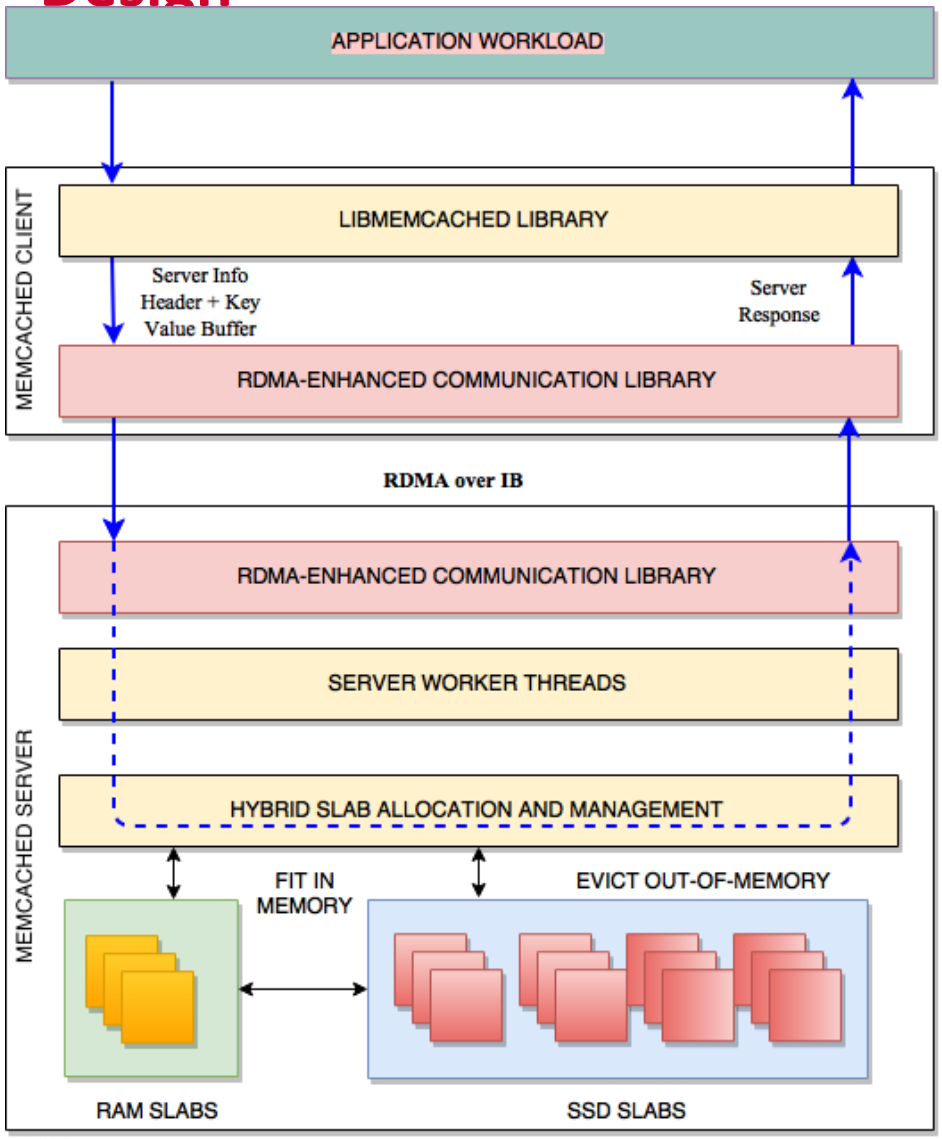
- Lustre is used as the intermediate data directory





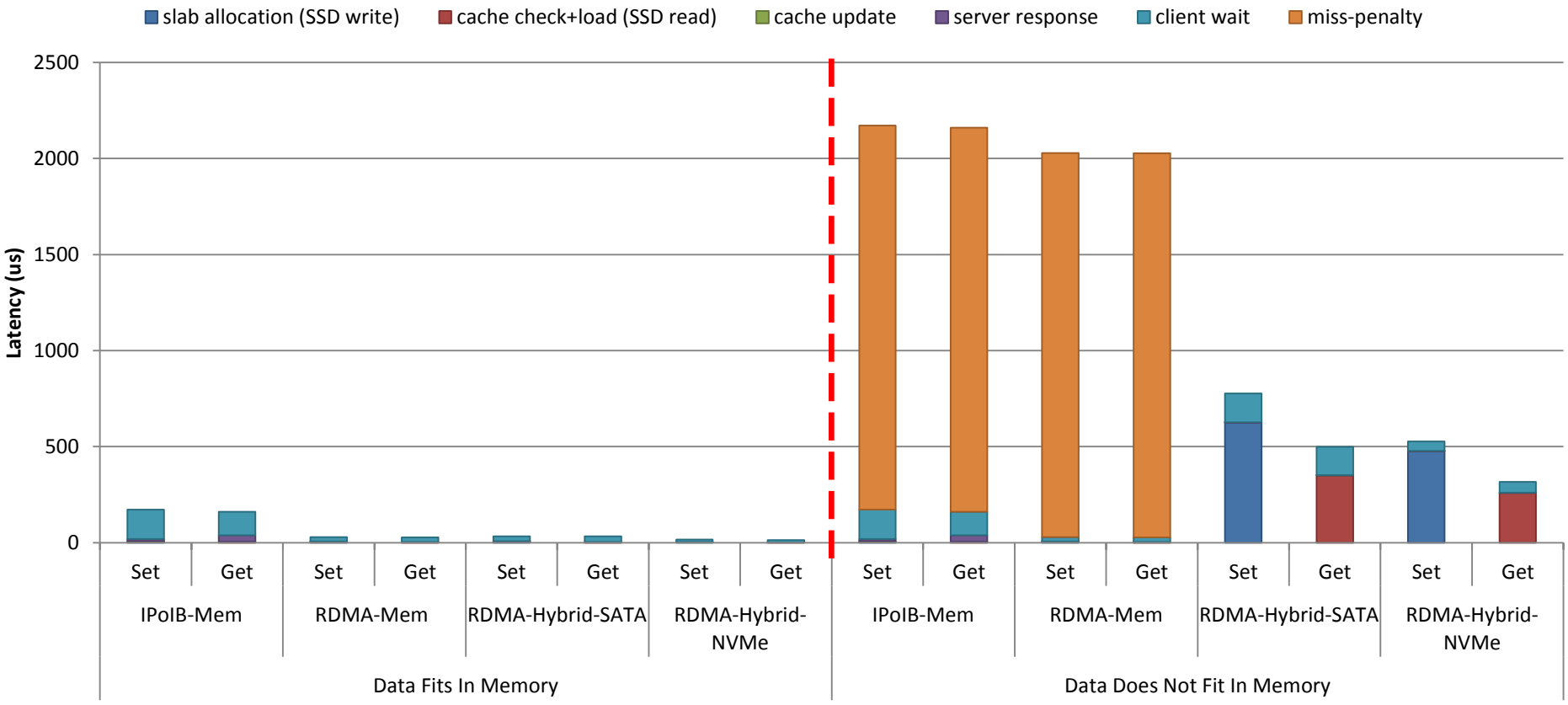
- For 80GB Sort in 8 nodes
  - 34% improvement over IPoIB (QDR)
- For 120GB TeraSort in 16 nodes
  - 25% improvement over IPoIB (QDR)

Overview of SSD-Assisted Hybrid RDMA-Memcached Design



- Design Features
  - Hybrid slab allocation and management for higher data retention
  - Log-structured sequence of blocks flushed to SSD
  - SSD fast random read to achieve low latency object access
  - Uses LRU to evict data to SSD

Performance Evaluation on SDSC Comet (IB FDR + SATA/NVMe SSDs)



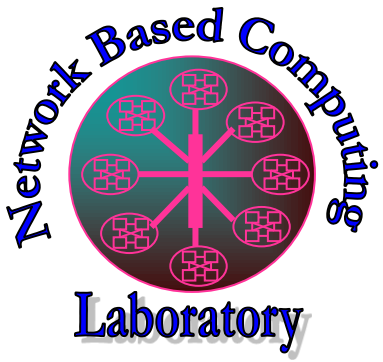
- Memcached latency test with Zipf distribution, server with 1 GB memory, 32 KB key-value pair size, total size of data accessed is 1 GB (when data fits in memory) and 1.5 GB (when data does not fit in memory)
- When data fits in memory: RDMA-Mem/Hybrid gives 5x improvement over IPoIB-Mem
- When data does not fix in memory: RDMA-Hybrid gives 2x-2.5x over IPoIB/RDMA-Mem

Presentation Outline

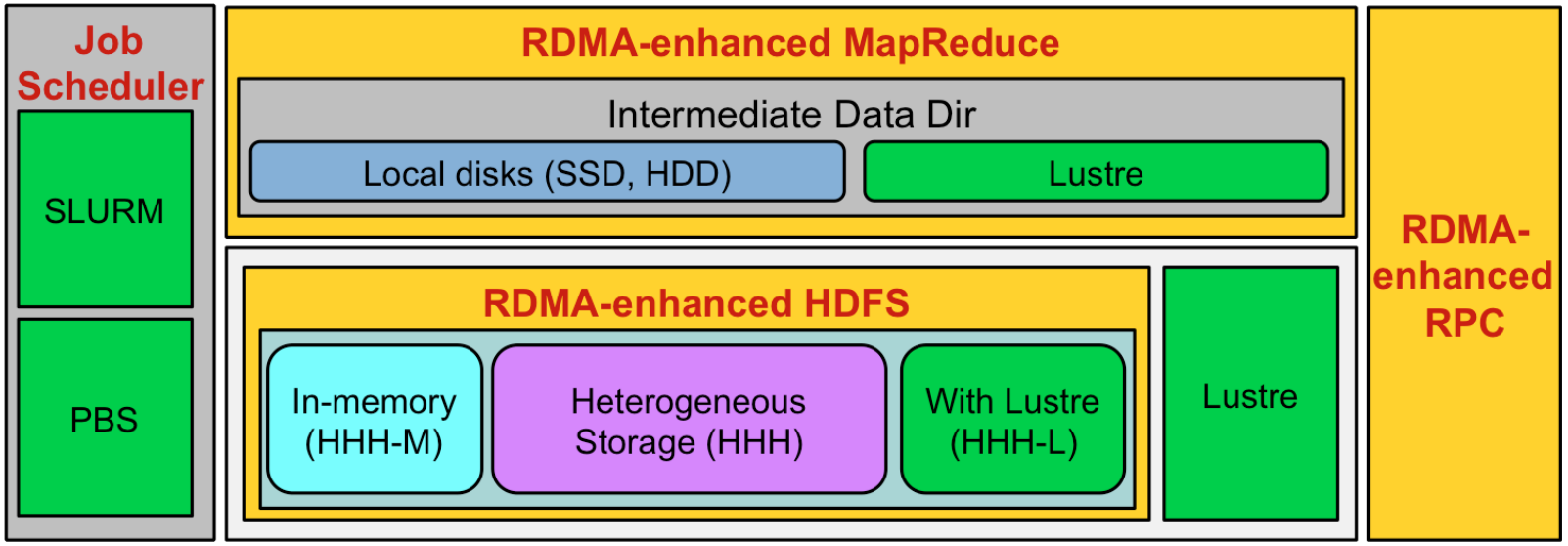
- Challenges for Accelerating Big Data Processing
- Accelerating Big Data Processing on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for HDFS, MapReduce, Spark, and Memcached
- Accelerating Big Data Processing on High-Performance Storage
  - Enhanced Designs for HDFS and MapReduce over Lustre
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- The High-Performance Big Data (HiBD) Project
- 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: 135 organizations from 20 countries
- More than 13,300 downloads from the project site



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).

Future Plans of OSU High Performance Big Data Project

- 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

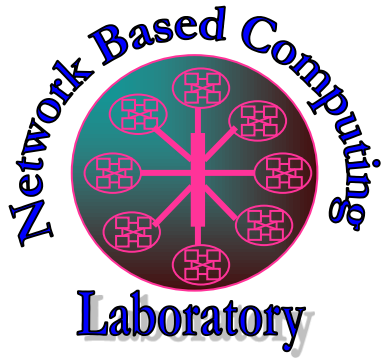
- Discussed communication and I/O challenges in accelerating Big Data middleware
- Presented initial designs to take advantage of InfiniBand/RDMA and high-performance storage architectures for Hadoop, Spark, and Memcached
- Presented challenges in designing benchmarks
- Results are promising
- Many other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

An Additional Talk

- **Tomorrow, Thursday (11:00-11:30am)**
  - **Exploiting Full Potential of GPU Clusters with InfiniBand using MVAPICH2-GDR**

Thank You!

panda@cse.ohio-state.edu



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

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