

# HPC Meets Big Data: Accelerating Hadoop, Spark, and Memcached with HPC Technologies

Talk at OpenFabrics Alliance Workshop (OFAW '17)

by

**Dhabaleswar K. (DK) Panda**

The Ohio State University

E-mail: [panda@cse.ohio-state.edu](mailto:panda@cse.ohio-state.edu)

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

**Xiaoyi Lu**

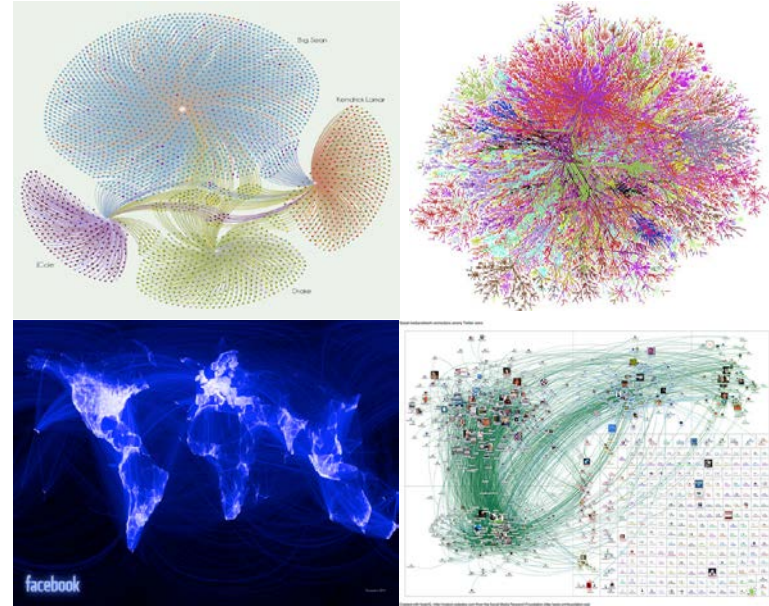
The Ohio State University

E-mail: [luxi@cse.ohio-state.edu](mailto:luxi@cse.ohio-state.edu)

<http://www.cse.ohio-state.edu/~luxi>

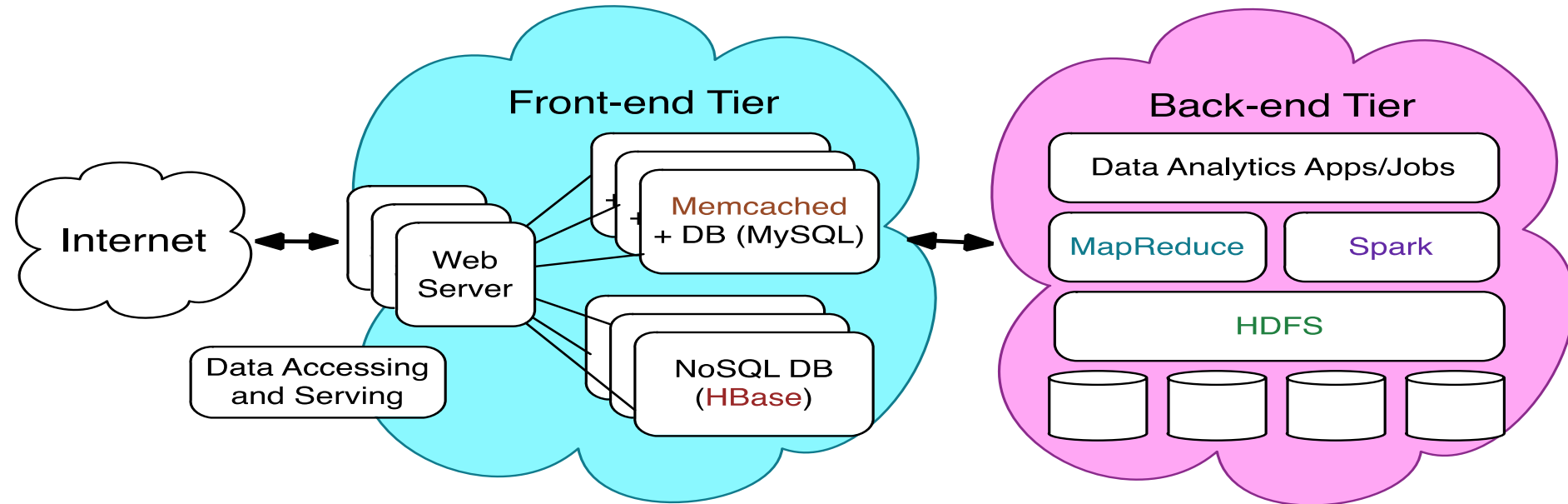
# Introduction to Big Data Applications and Analytics

- **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



# Data Management and Processing on Modern Clusters

- Substantial impact on designing and utilizing 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



# Drivers of Modern HPC Cluster Architectures



Multi-core Processors



High Performance Interconnects -  
InfiniBand

<1usec latency, 100Gbps Bandwidth>

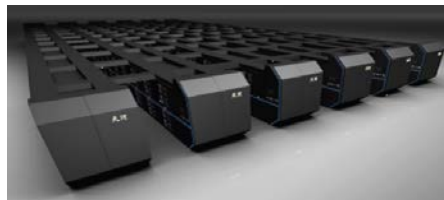


Accelerators / Coprocessors  
high compute density, high  
performance/watt  
>1 TFlop DP on a chip



SSD, NVMe-SSD, NVRAM

- 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*

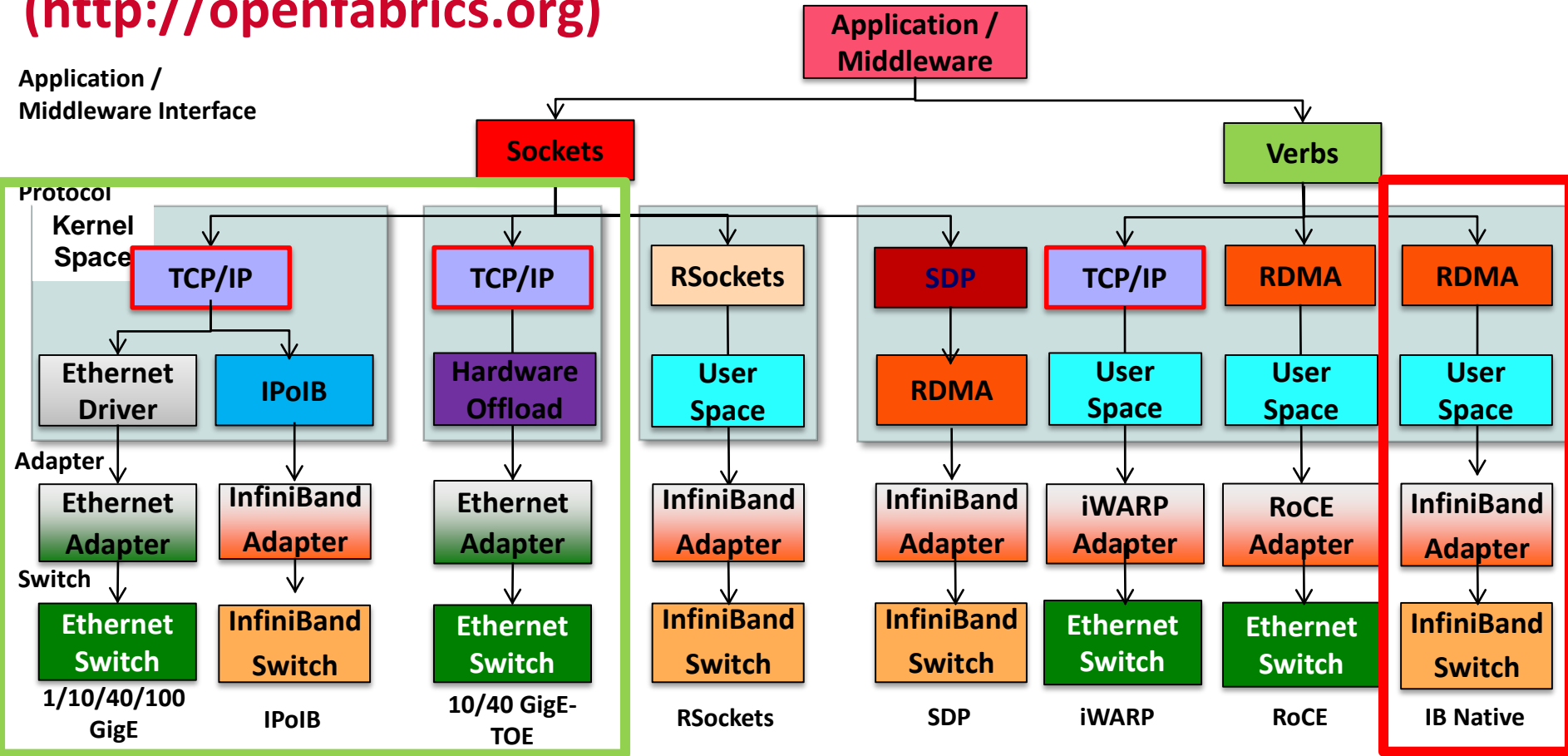


*Tianhe – 1A*

# Interconnects and Protocols in OpenFabrics Stack for HPC

(<http://openfabrics.org>)

Application /  
Middleware Interface



# 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?

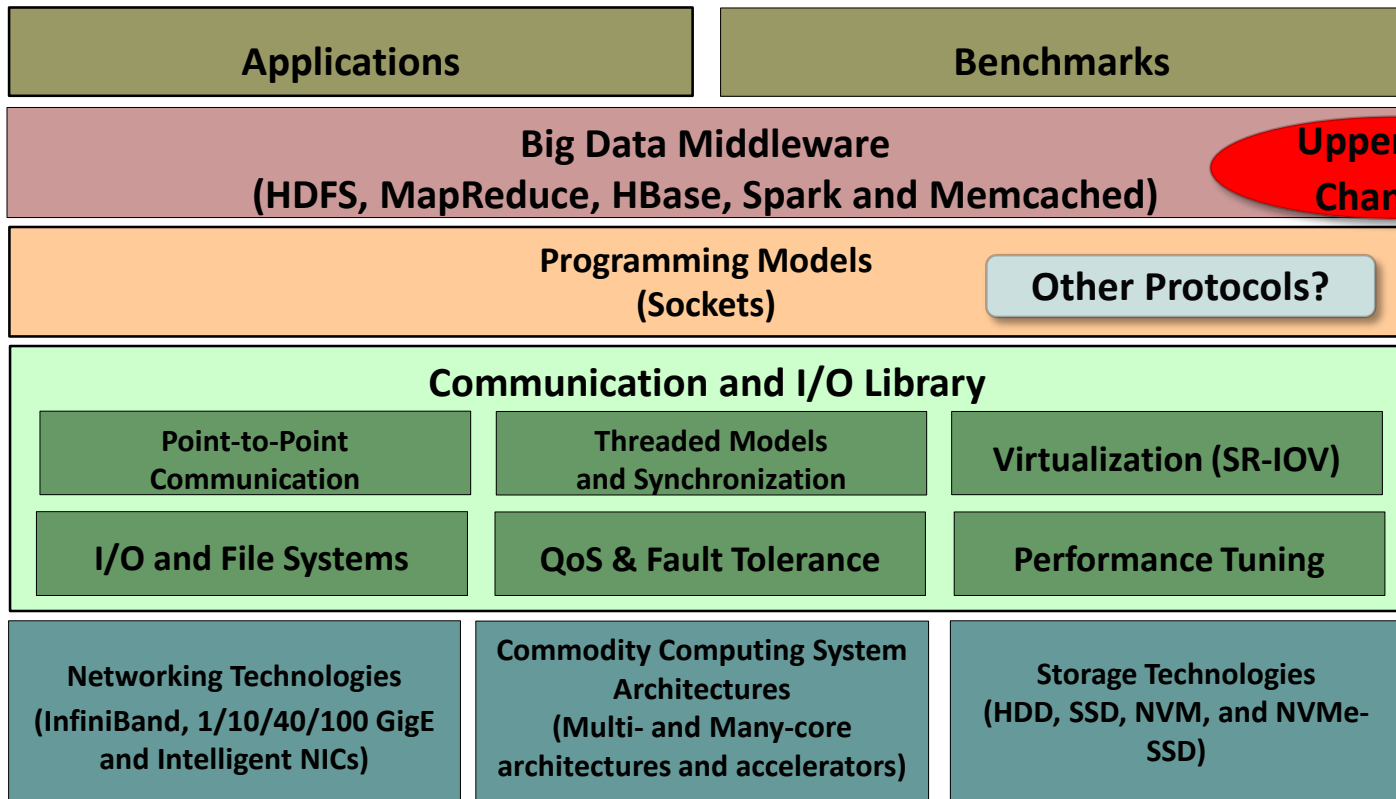
What are the major **bottlenecks** in current Big Data processing middleware (e.g. Hadoop, Spark, and Memcached)?

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

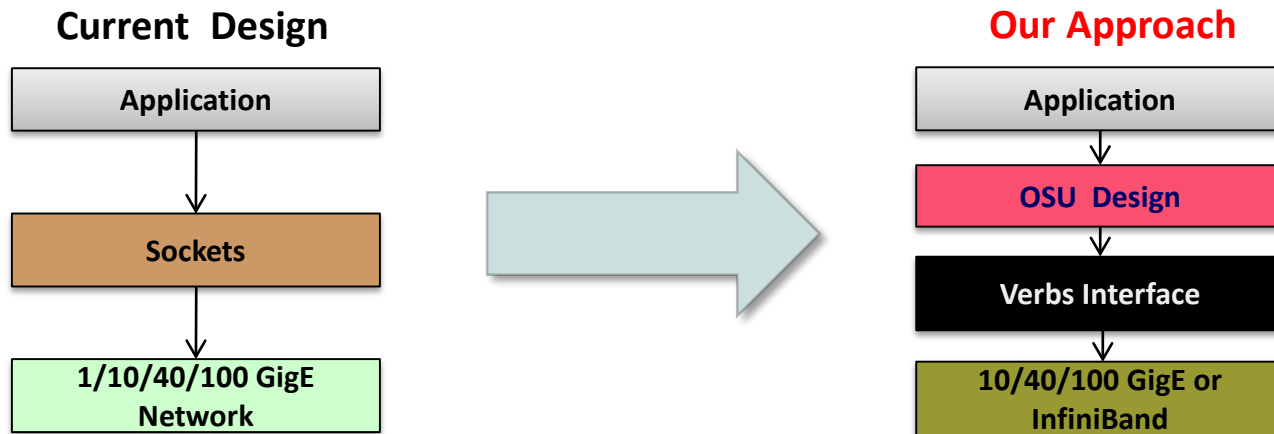


Bring HPC and Big Data processing into a “convergent trajectory”!

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



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



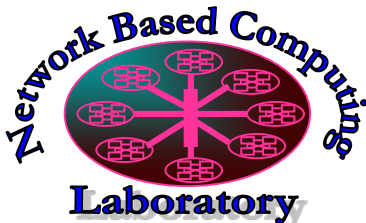
- Sockets not designed for high-performance
  - Stream semantics often mismatch for upper layers
  - Zero-copy not available for non-blocking sockets



# The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
  - HDFS, Memcached, HBase, and Spark Micro-benchmarks
- <http://hibd.cse.ohio-state.edu>
- Users Base: 215 organizations from 29 countries
- More than 20,950 downloads from the project site

Available for InfiniBand and RoCE

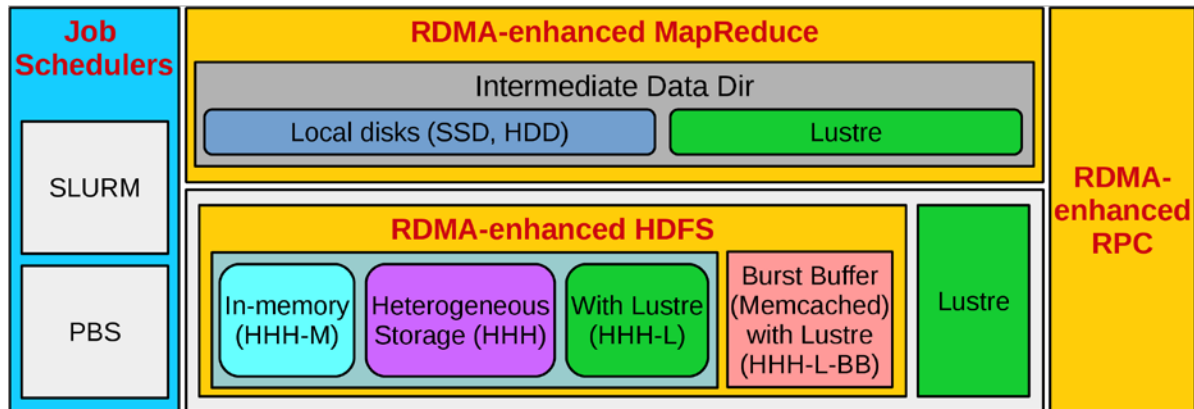


# RDMA for Apache Hadoop 2.x Distribution

- 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
  - Memcached-based burst buffer for MapReduce over Lustre-integrated HDFS (HHH-L-BB mode)
  - Plugin-based architecture supporting RDMA-based designs for Apache Hadoop, CDH and HDP
  - Easily configurable for different running modes (HHH, HHH-M, HHH-L, HHH-L-BB, and MapReduce over Lustre) and different protocols (native InfiniBand, RoCE, and IPoIB)
- Current release: **1.1.0**
  - Based on Apache Hadoop **2.7.3**
  - Compliant with Apache Hadoop 2.7.1, HDP 2.5.0.3 and CDH 5.8.2 APIs and applications
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR, FDR, and EDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - Different file systems with disks and SSDs and Lustre

<http://hibd.cse.ohio-state.edu>

# 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.
- **HHH-L-BB:** This mode deploys a Memcached-based burst buffer system to reduce the bandwidth bottleneck of shared file system access. The burst buffer design is hosted by Memcached servers, each of which has a local SSD.
- **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 Spark Distribution

- High-Performance Design of Spark over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for Spark
  - RDMA-based data shuffle and SEDA-based shuffle architecture
  - Support pre-connection, on-demand connection, and connection sharing
  - Non-blocking and chunk-based data transfer
  - Off-JVM-heap buffer management
  - Easily configurable for different protocols (native InfiniBand, RoCE, and IPoIB)
- Current release: **0.9.4**
  - Based on Apache Spark **2.1.0**
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR, FDR, and EDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - RAM disks, SSDs, and HDD
  - <http://hibd.cse.ohio-state.edu>

# HiBD Packages on SDSC Comet and Chameleon Cloud

- RDMA for Apache Hadoop 2.x and RDMA for Apache Spark are installed and available on SDSC Comet.
  - Examples for various modes of usage are available in:
    - RDMA for Apache Hadoop 2.x: /share/apps/examples/HADOOP
    - RDMA for Apache Spark: /share/apps/examples/SPARK/
  - Please email [help@xsede.org](mailto:help@xsede.org) (reference Comet as the machine, and SDSC as the site) if you have any further questions about usage and configuration.
- RDMA for Apache Hadoop is also available on Chameleon Cloud as an appliance
  - <https://www.chameleoncloud.org/appliances/17/>

M. Tatineni, X. Lu, D. J. Choi, A. Majumdar, and D. K. Panda, Experiences and Benefits of Running RDMA Hadoop and Spark on SDSC Comet, XSEDE'16, July 2016

# RDMA for Apache HBase Distribution

- High-Performance Design of HBase over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for HBase
  - Compliant with Apache HBase 1.1.2 APIs and applications
  - On-demand connection setup
  - Easily configurable for different protocols (native InfiniBand, RoCE, and IPoIB)
- Current release: **0.9.1**
  - Based on Apache HBase **1.1.2**
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR, FDR, and EDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
  - <http://hibd.cse.ohio-state.edu>

# RDMA for Memcached Distribution

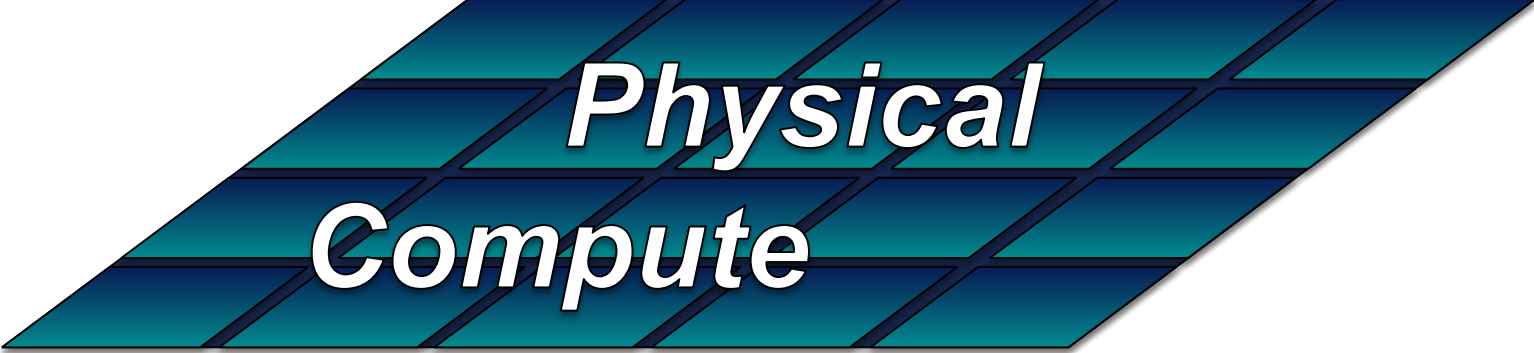
- High-Performance Design of Memcached over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for Memcached and libMemcached components
  - High performance design of SSD-Assisted Hybrid Memory
  - Non-Blocking Libmemcached Set/Get API extensions
  - Support for burst-buffer mode in Lustre-integrated design of HDFS in RDMA for Apache Hadoop-2.x
  - Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)
- Current release: 0.9.5
  - Based on Memcached 1.4.24 and libMemcached 1.0.18
  - Compliant with libMemcached APIs and applications
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR, FDR, and EDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - SSD
  - <http://hibd.cse.ohio-state.edu>

## OSU HiBD Micro-Benchmark (OHB) Suite – HDFS, Memcached, and HBase

- 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, **Non-Blocking API** Latency Benchmark, **Hybrid Memory** Latency Benchmark
- Micro-benchmarks for HBase
  - **Get** Latency Benchmark, **Put** Latency Benchmark
- Current release: **0.9.1**
- <http://hibd.cse.ohio-state.edu>



# Using HiBD Packages on Existing HPC Infrastructure



*Physical  
Compute*

# Using HiBD Packages on Existing HPC Infrastructure



*Resource Manager*  
(Torque, SLURM, etc.)

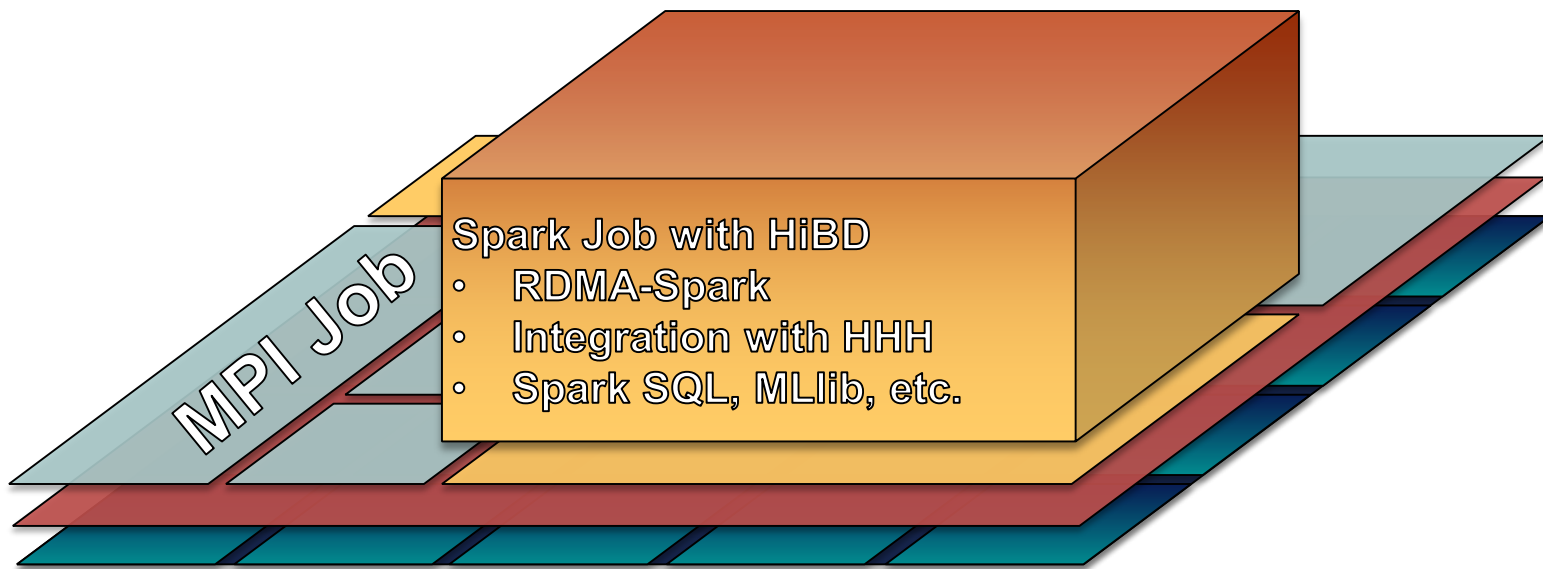
# Using HiBD Packages on Existing HPC Infrastructure



# Using HiBD Packages on Existing HPC Infrastructure



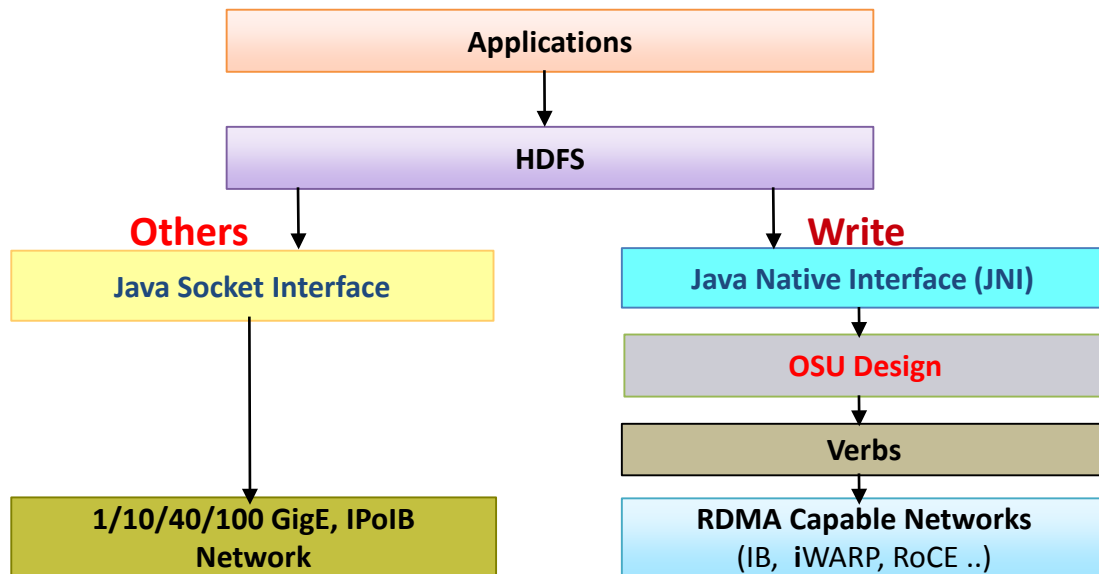
# Using HiBD Packages on Existing HPC Infrastructure



# Acceleration Case Studies and Performance Evaluation

- Basic Designs for HiBD Packages
  - HDFS, MapReduce, and RPC
  - HBase
  - Spark
  - Memcached
  - OSU HiBD Benchmarks (OHB)
- Advanced Designs
  - Memcached with Hybrid Memory and Non-blocking APIs
  - Accelerating Big Data I/O (Lustre + Burst-Buffer)

# Design Overview of HDFS with RDMA



- Design Features

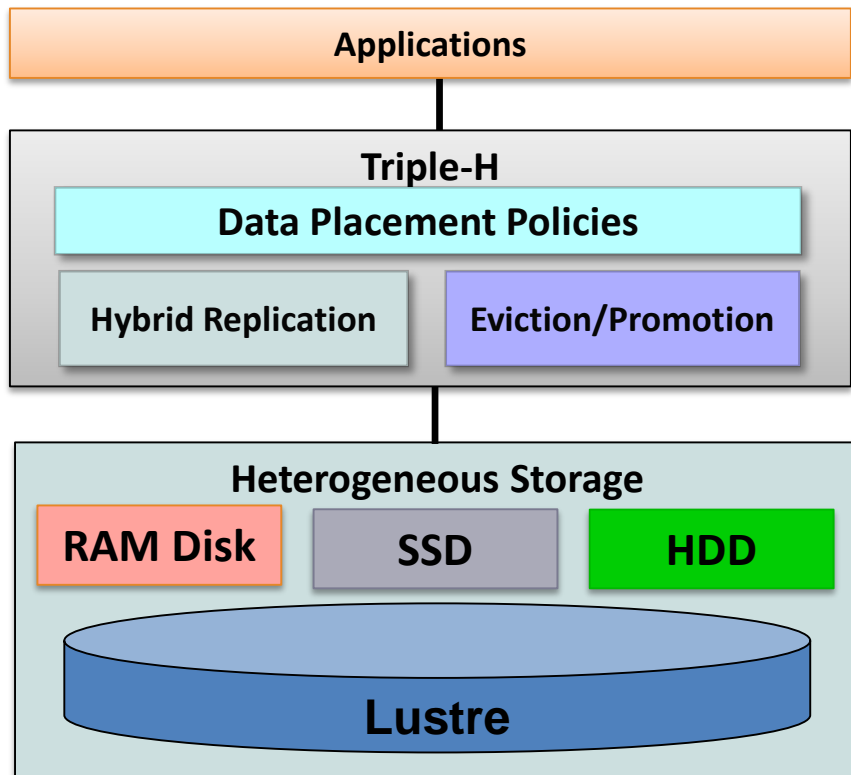
- RDMA-based HDFS write
- RDMA-based HDFS replication
- Parallel replication support
- On-demand connection setup
- InfiniBand/RoCE support

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

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

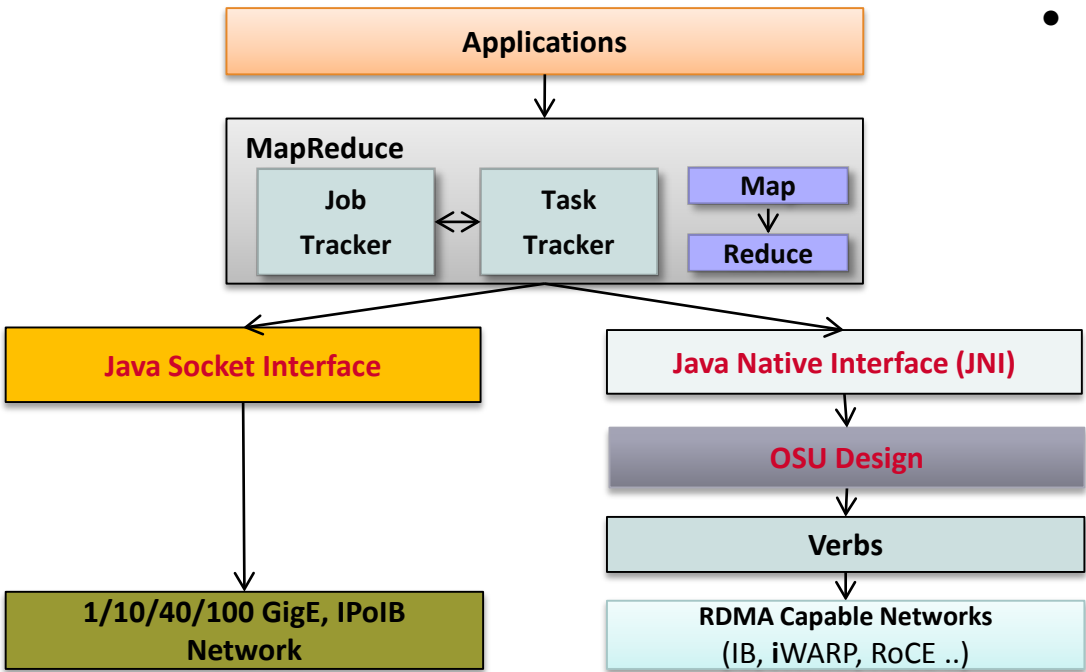


- Design Features
  - Three modes
    - Default (HHH)
    - In-Memory (HHH-M)
    - Lustre-Integrated (HHH-L)
  - Policies to efficiently utilize the heterogeneous storage devices
    - RAM, SSD, HDD, Lustre
  - Eviction/Promotion based on data usage pattern
  - Hybrid Replication
  - Lustre-Integrated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



# Design Overview of MapReduce with RDMA



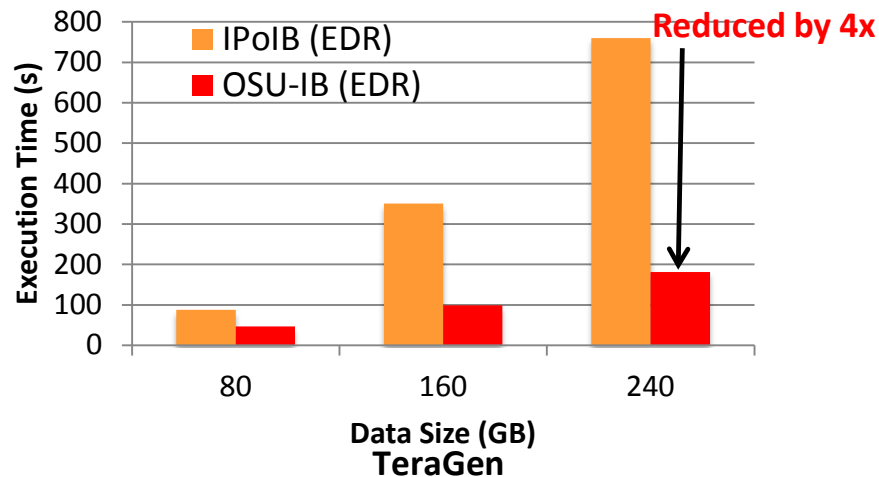
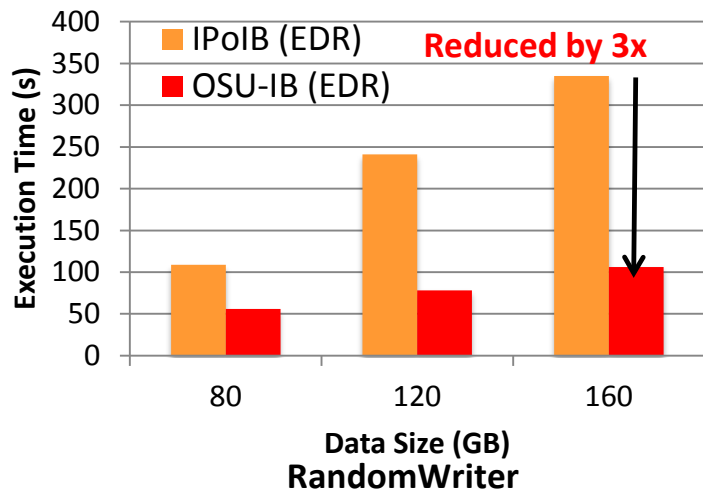
- Design Features

- 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

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 Numbers of RDMA for Apache Hadoop 2.x – RandomWriter & TeraGen in OSU-RI2 (EDR)



Cluster with 8 Nodes with a total of 64 maps

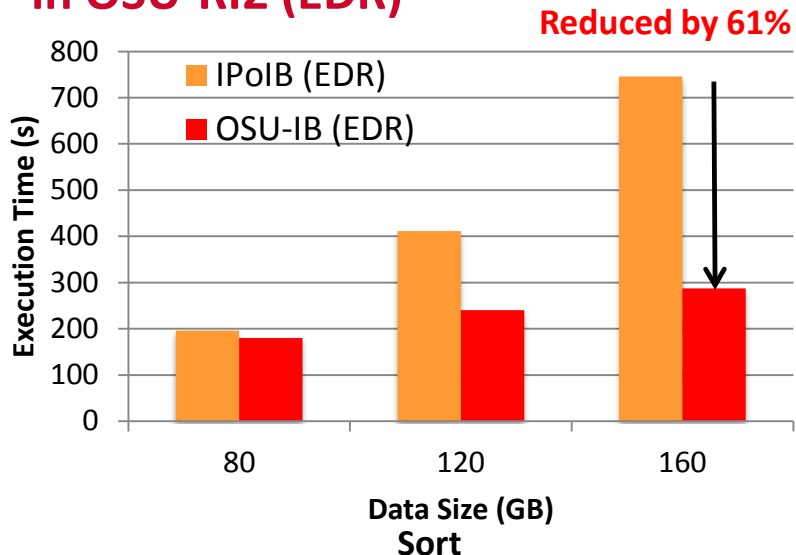
- RandomWriter

- **3x** improvement over IPoIB for 80-160 GB file size

- TeraGen

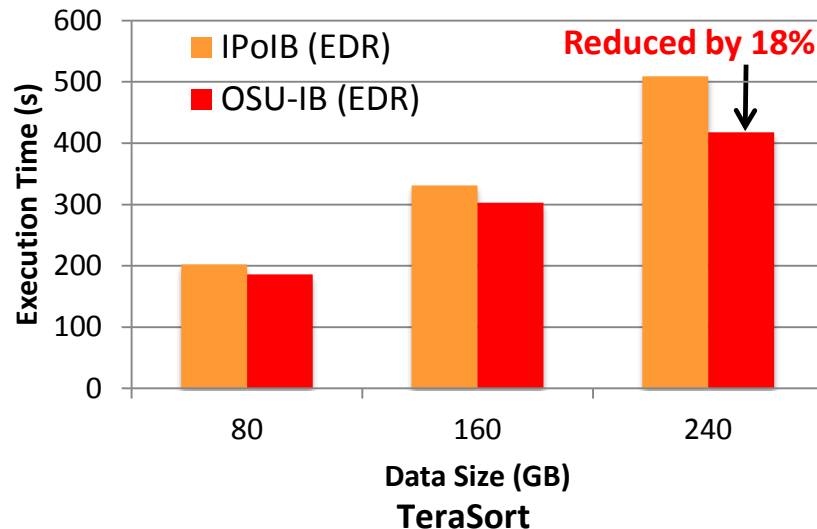
- **4x** improvement over IPoIB for 80-240 GB file size

## Performance Numbers of RDMA for Apache Hadoop 2.x – Sort & TeraSort in OSU-RI2 (EDR)



- Sort

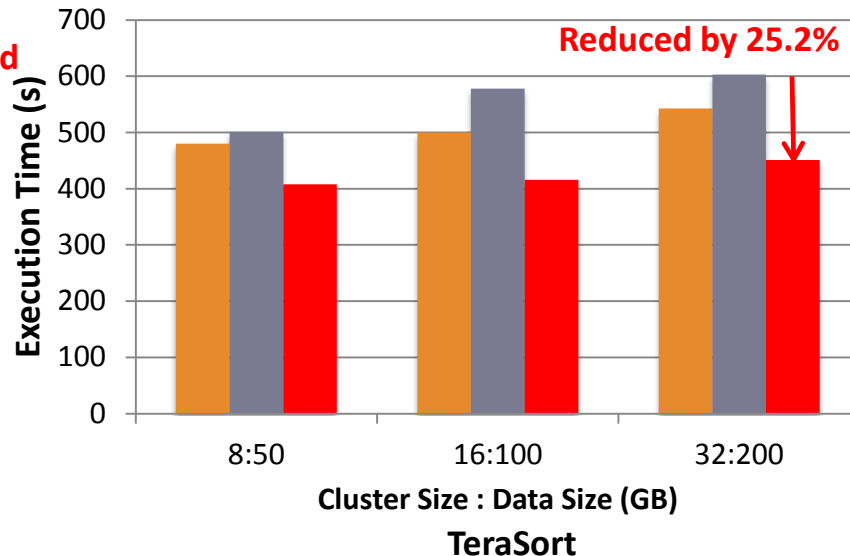
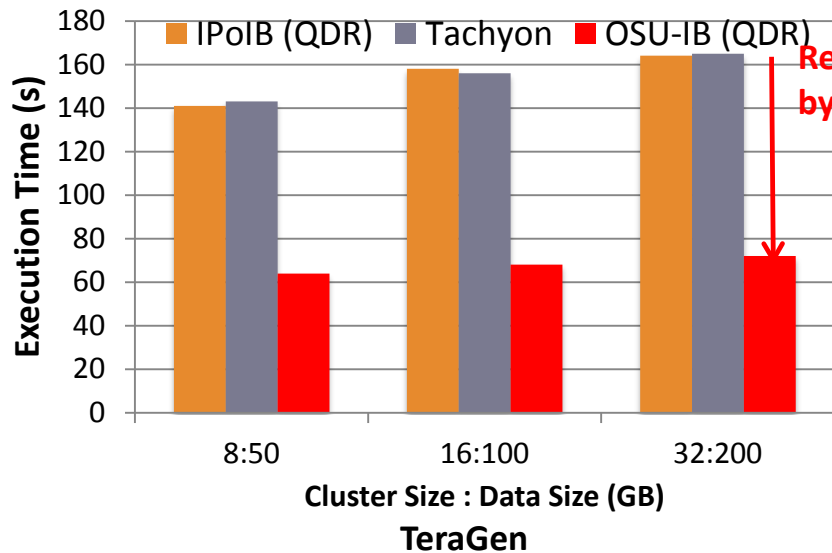
- **61%** improvement over IPoIB for 80-160 GB data



- TeraSort

- **18%** improvement over IPoIB for 80-240 GB data

# Evaluation with Spark on SDSC Gordon (HHH vs. Tachyon/Alluxio)



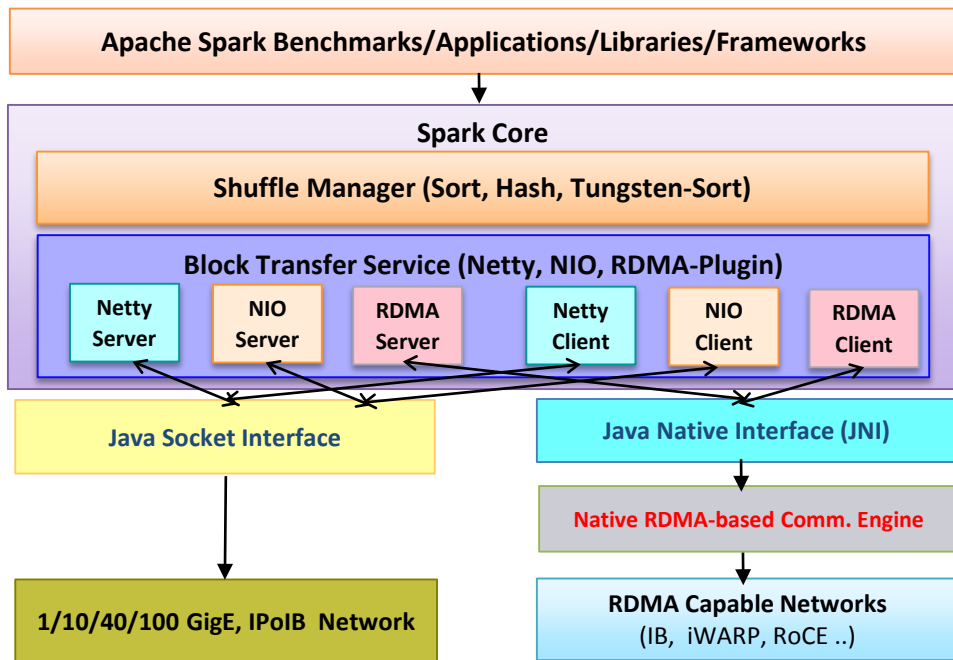
- For 200GB TeraGen on 32 nodes
  - Spark-TeraGen: HHH has 2.4x improvement over Tachyon; 2.3x over HDFS-IPoIB (QDR)
  - Spark-TeraSort: HHH has 25.2% improvement over Tachyon; 17% over HDFS-IPoIB (QDR)

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

# Acceleration Case Studies and Performance Evaluation

- Basic Designs for HiBD Packages
  - HDFS, MapReduce, and RPC
  - HBase
  - Spark
  - Memcached
  - OSU HiBD Benchmarks (OHB)
- Advanced Designs
  - Memcached with Hybrid Memory and Non-blocking APIs
  - Accelerating Big Data I/O (Lustre + Burst-Buffer)

# Design Overview of Spark with RDMA



- Design Features

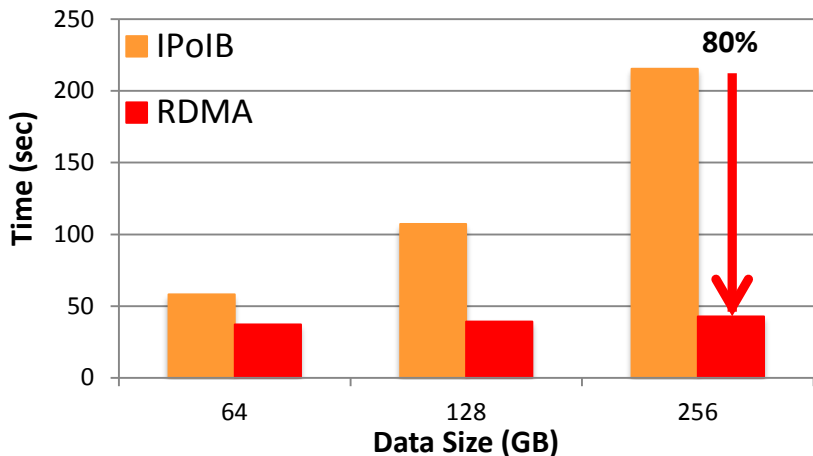
- RDMA based shuffle plugin
- SEDA-based architecture
- Dynamic connection management and sharing
- Non-blocking data transfer
- Off-JVM-heap buffer management
- InfiniBand/RoCE support

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

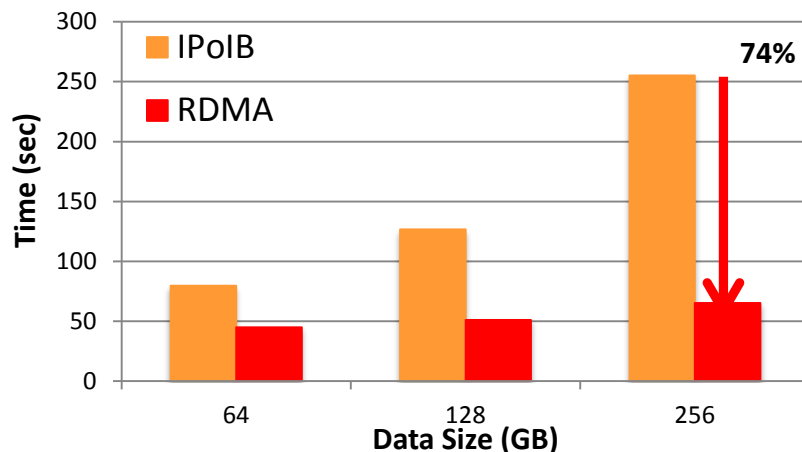
X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, *Accelerating Spark with RDMA for Big Data Processing: Early Experiences*, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

X. Lu, D. Shankar, S. Gugnani, and D. K. Panda, *High-Performance Design of Apache Spark with RDMA and Its Benefits on Various Workloads*, IEEE BigData '16, Dec. 2016.

# Performance Evaluation on SDSC Comet – SortBy/GroupBy



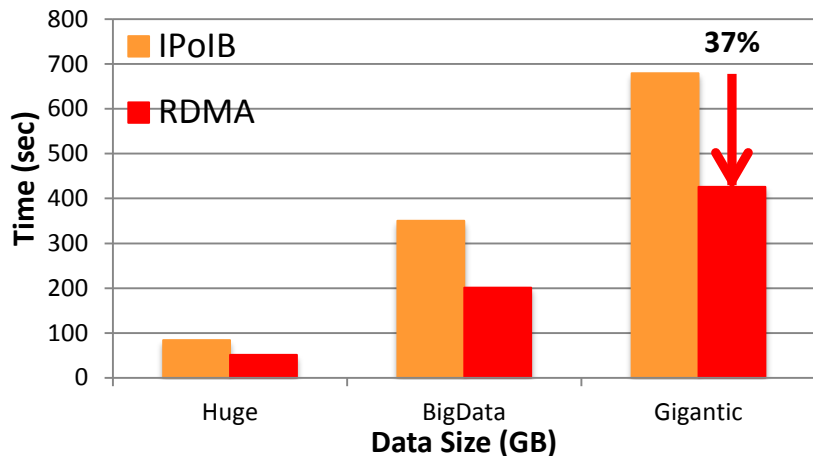
64 Worker Nodes, 1536 cores, **SortByTest** Total Time



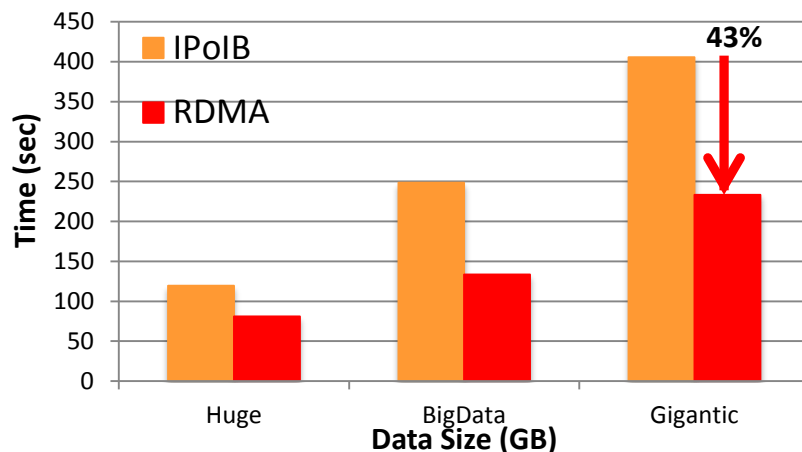
64 Worker Nodes, 1536 cores, **GroupByTest** Total Time

- InfiniBand FDR, SSD, 64 Worker Nodes, 1536 Cores, (1536M 1536R)
- RDMA vs. IPoIB with 1536 concurrent tasks, single SSD per node.
  - SortBy: Total time reduced by up to 80% over IPoIB (56Gbps)
  - GroupBy: Total time reduced by up to 74% over IPoIB (56Gbps)

# Performance Evaluation on SDSC Comet – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time

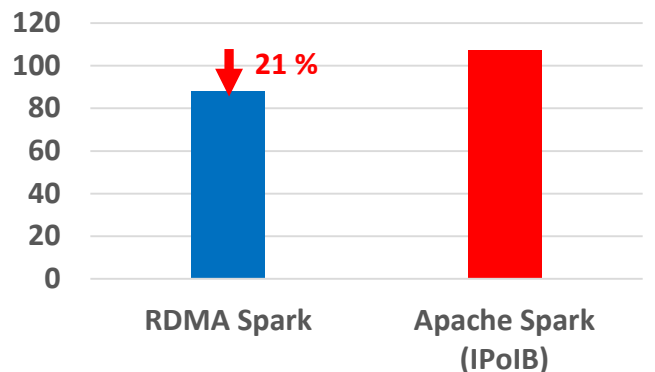


64 Worker Nodes, 1536 cores, PageRank Total Time

- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

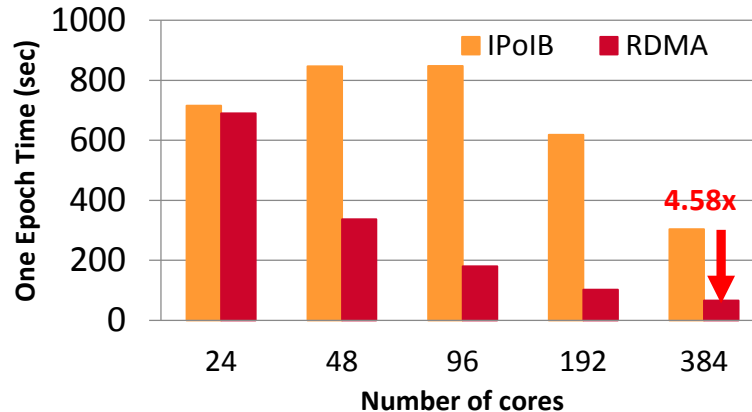


# Application Evaluation on SDSC Comet



Execution times (sec) for Kira SE benchmark using 65 GB dataset, 48 cores.

- **Kira Toolkit:** Distributed astronomy image processing toolkit implemented using Apache Spark
  - <https://github.com/BIDS/Kira>
- Source extractor application, using a 65GB dataset from the SDSS DR2 survey that comprises 11,150 image files.



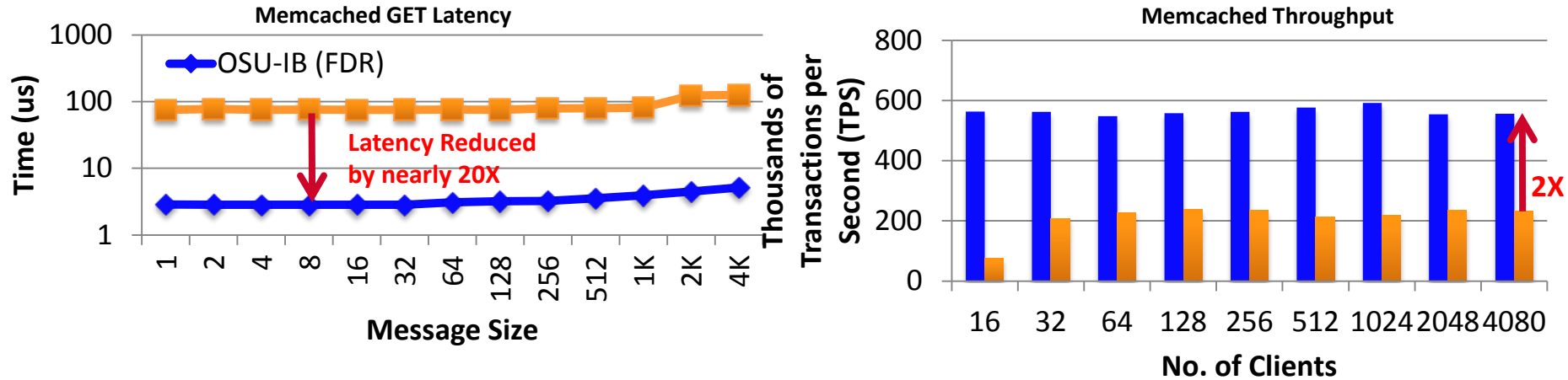
- **BigDL:** Distributed Deep Learning Tool using Apache Spark
  - <https://github.com/intel-analytics/BigDL>
- VGG training model on the CIFAR-10 dataset

M. Tatineni, X. Lu, D. J. Choi, A. Majumdar, and D. K. Panda, Experiences and Benefits of Running RDMA Hadoop and Spark on SDSC Comet, XSEDE'16, July 2016

# Acceleration Case Studies and Performance Evaluation

- Basic Designs for HiBD Packages
  - HDFS, MapReduce, and RPC
  - HBase
  - Spark
  - Memcached
  - OSU HiBD Benchmarks (OHB)
- Advanced Designs and Studies
  - Memcached with Hybrid Memory and Non-blocking APIs
  - Accelerating Big Data I/O (Lustre + Burst-Buffer)

# Memcached Performance (FDR Interconnect)



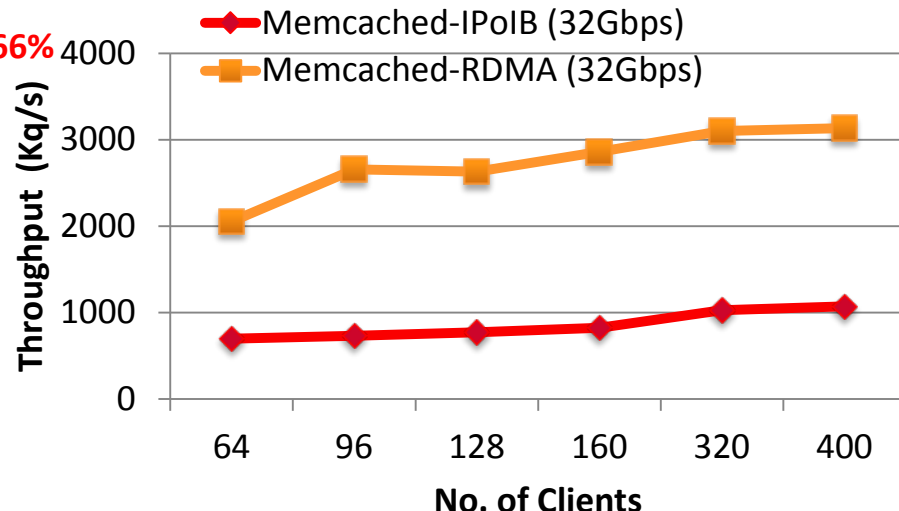
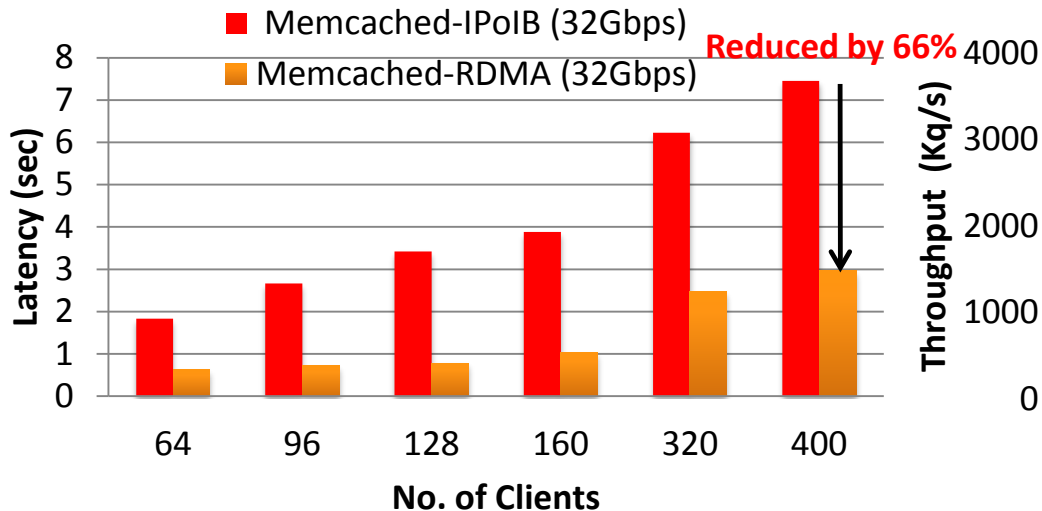
Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- 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

J. Jose, H. Subramoni, M. Luo, M. Zhang, J. Huang, M. W. Rahman, N. Islam, X. Ouyang, H. Wang, S. Sur and D. K. Panda, Memcached Design on High Performance RDMA Capable Interconnects, ICPP'11

J. Jose, H. Subramoni, K. Kandalla, M. W. Rahman, H. Wang, S. Narravula, and D. K. Panda, Scalable Memcached design for InfiniBand Clusters using Hybrid Transport, CCGrid'12

# Micro-benchmark Evaluation for OLDP workloads



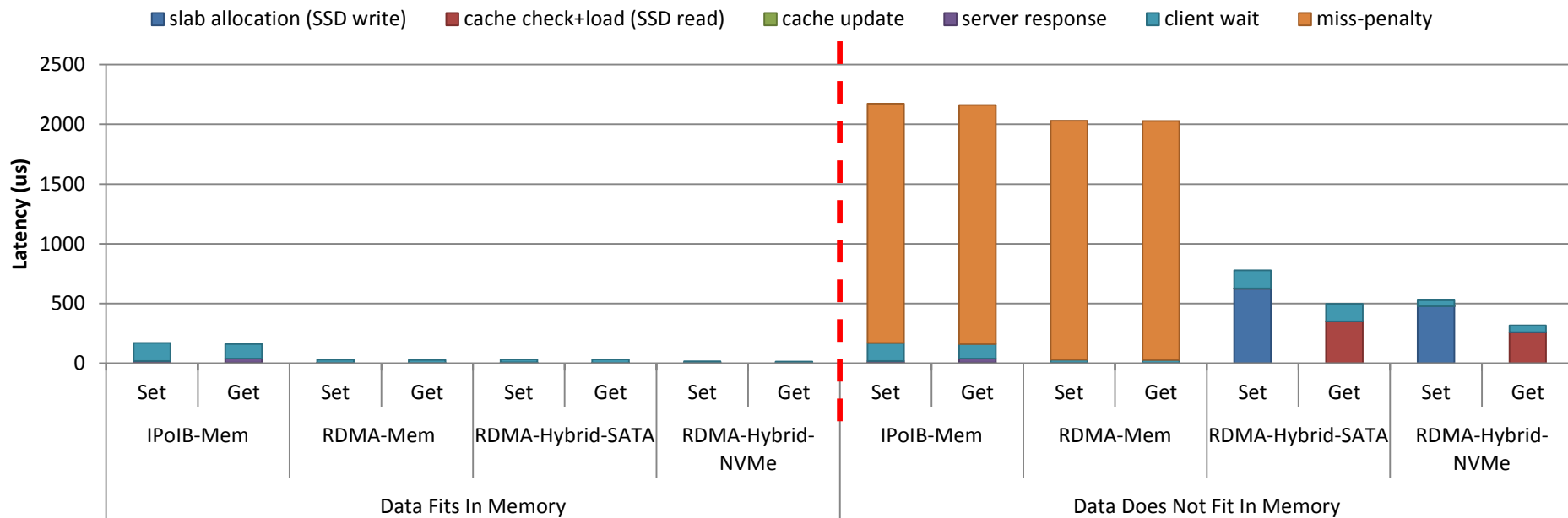
- Illustration with **Read-Cache-Read** access pattern using modified **mysqlslap** load testing tool
- Memcached-RDMA can
  - improve query latency by up to **66%** over IPoIB (32Gbps)
  - throughput by up to **69%** over IPoIB (32Gbps)

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

# Acceleration Case Studies and Performance Evaluation

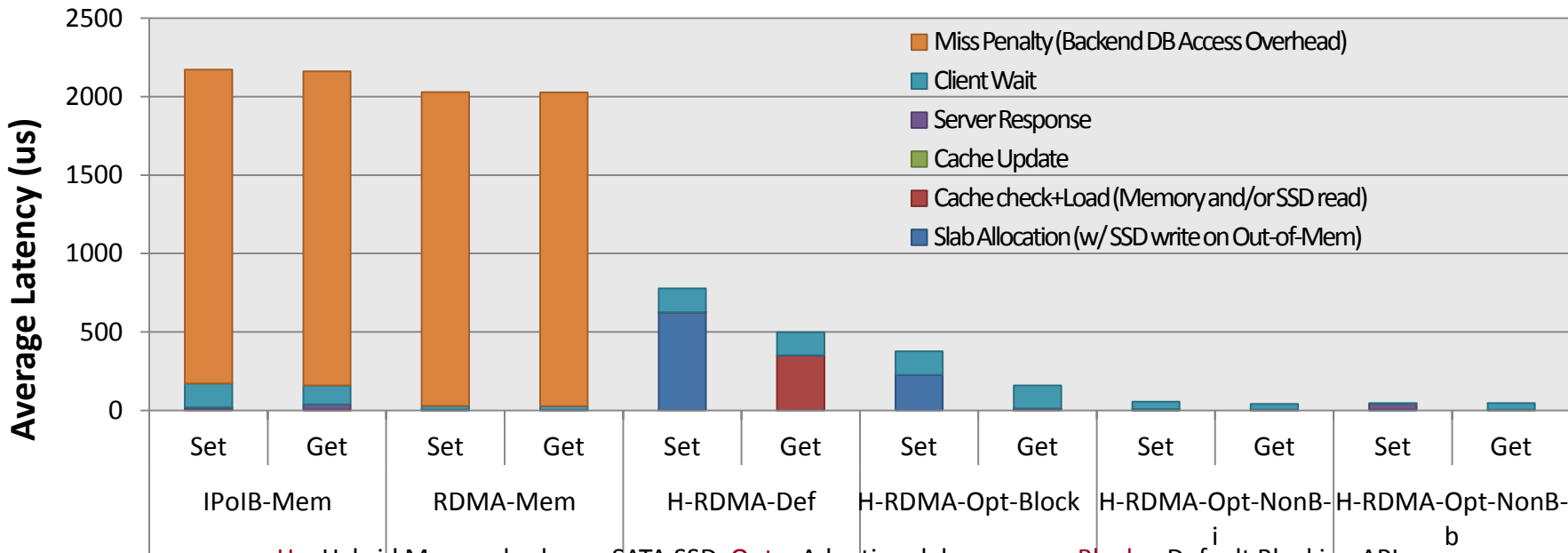
- Basic Designs for HiBD Packages
  - HDFS, MapReduce, and RPC
  - HBase
  - Spark
  - Memcached
  - OSU HiBD Benchmarks (OHB)
- Advanced Designs
  - Memcached with Hybrid Memory and Non-blocking APIs
  - Accelerating Big Data I/O (Lustre + Burst-Buffer)

# Performance Evaluation on IB FDR + SATA/NVMe SSDs (Hybrid Memory)



- 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 fit in memory:** RDMA-Hybrid gives **2x-2.5x** over IPoIB/RDMA-Mem

# Performance Evaluation with Non-Blocking Memcached API



H = Hybrid Memcached over SATA SSD Opt = Adaptive slab manager Block = Default Blocking API  
NonB-i = Non-blocking iset/iget API NonB-b = Non-blocking bset/bget API w/ buffer re-use guarantee

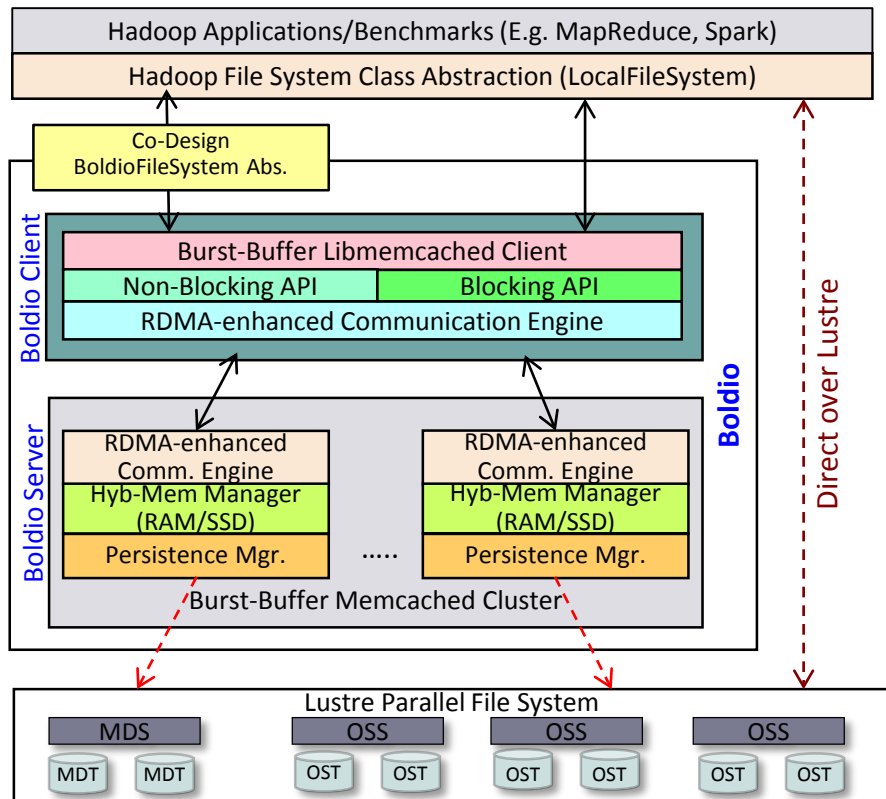
- **Data does not fit in memory:** Non-blocking Memcached Set/Get API Extensions can achieve
  - >16x latency improvement vs. blocking API over RDMA-Hybrid/RDMA-Mem w/ penalty
  - >2.5x throughput improvement vs. blocking API over default/optimized RDMA-Hybrid
- **Data fits in memory:** Non-blocking Extensions perform similar to RDMA-Mem/RDMA-Hybrid and >3.6x improvement over IPoIB-Mem

# Acceleration Case Studies and Performance Evaluation

- Basic Designs for HiBD Packages
  - HDFS, MapReduce, and RPC
  - HBase
  - Spark
  - Memcached
  - OSU HiBD Benchmarks (OHB)
- **Advanced Designs**
  - Memcached with Hybrid Memory and Non-blocking APIs
  - **Accelerating Big Data I/O (Lustre + Burst-Buffer)**



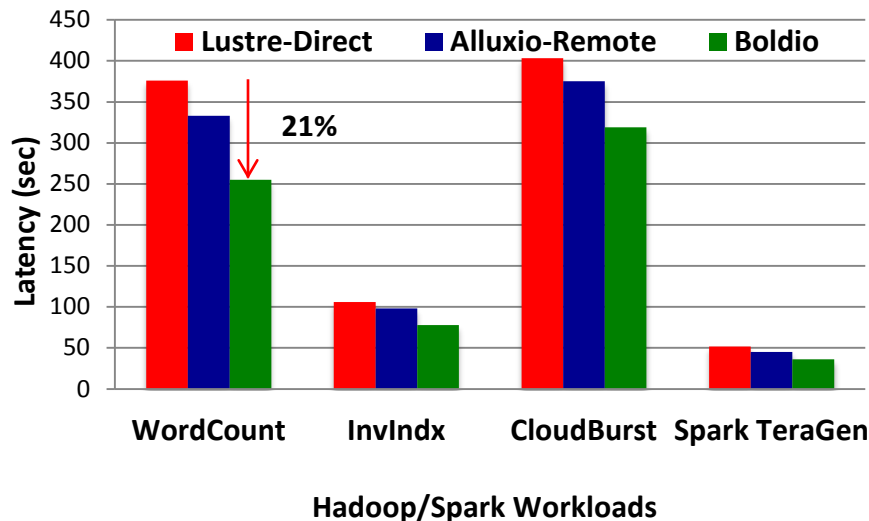
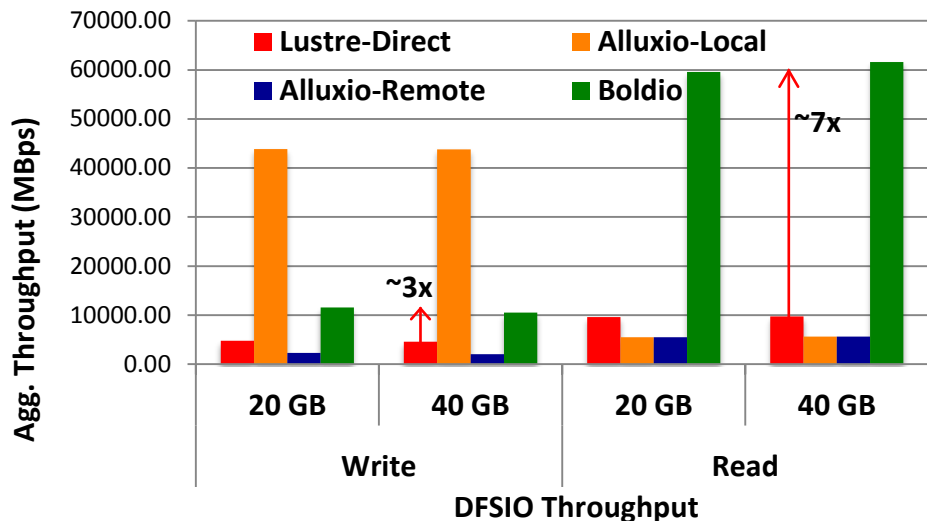
# Burst-Buffer Over Lustre for Accelerating Big Data I/O (Boldio)



- Hybrid and resilient key-value store-based Burst-Buffer system Over Lustre
- Overcome limitations of local storage on HPC cluster nodes
- Light-weight transparent interface to Hadoop/Spark applications
- Accelerating I/O-intensive Big Data workloads
  - Non-blocking Memcached APIs to maximize overlap
  - Client-based replication for resilience
  - Asynchronous persistence to Lustre parallel file system

D. Shankar, X. Lu, D. K. Panda, **Boldio: A Hybrid and Resilient Burst-Buffer over Lustre for Accelerating Big Data I/O**, IEEE Big Data 2016.

# Performance Evaluation with Boldio



- Based on RDMA-based Libmemcached/Memcached 0.9.3, Hadoop-2.6.0
- InfiniBand QDR, 24GB RAM + PCIe-SSDs, 12 nodes, 32/48 Map/Reduce Tasks, 4-node Memcached cluster
- Boldio can improve
  - throughput over Lustre by about **3x** for write throughput and **7x** for read throughput
  - execution time of Hadoop benchmarks over Lustre, e.g. Wordcount, Cloudburst by **>21%**
- Contrasting with Alluxio (formerly Tachyon)
  - Performance degrades about 15x when Alluxio cannot leverage local storage (Alluxio-Local vs. Alluxio-Remote)
  - Boldio can improve throughput over Alluxio with all remote workers by about 3.5x - 8.8x (Alluxio-Remote vs. Boldio)

# On-going and Future Plans of OSU High Performance Big Data (HiBD) Project

- Upcoming Releases of RDMA-enhanced Packages will support
  - Upgrades to the latest versions of Hadoop
  - Streaming
  - MR-Advisor
  - Impala
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  - MapReduce, RPC
- Advanced designs with upper-level changes and optimizations
  - Boldio
  - Efficient Indexing

## Concluding Remarks

- Discussed challenges in accelerating Big Data middleware with HPC technologies
- Presented basic and advanced designs to take advantage of InfiniBand/RDMA for HDFS, MapReduce, RPC, HBase, Memcached, and Spark
- Results are promising
- Many other open issues need to be solved
- Will enable Big Data community to take advantage of modern HPC technologies to carry out their analytics in a fast and scalable manner
- Looking forward to collaboration with the community

# Three More Presentations

- Thursday (03/30/17) at 8:00 am

**Designing MPI and PGAS Libraries for Exascale Systems: The MVAPICH2 Approach**

- Thursday (03/30/17) at 9:00am

**Building Efficient HPC Clouds with MVAPICH2 and RDMA-Hadoop over SR-IOV IB Clusters**

- Friday (03/31/17) at 11:00am

**NVM-aware RDMA-Based Communication and I/O Schemes for High-Perf Big Data Analytics**

# Funding Acknowledgments

## Funding Support by



## Equipment Support by



# Personnel Acknowledgments

## **Current Students**

- A. Awan (Ph.D.)
- R. Biswas (M.S.)
- M. Bayatpour (Ph.D.)
- S. Chakraborty (Ph.D.)
- C.-H. Chu (Ph.D.)
- S. Guganani (Ph.D.)
- J. Hashmi (Ph.D.)
- H. Javed (Ph.D.)
- M. Li (Ph.D.)
- D. Shankar (Ph.D.)
- H. Shi (Ph.D.)
- J. Zhang (Ph.D.)

## **Current Research Scientists**

- X. Lu
- H. Subramoni

## **Current Research Specialist**

- J. Smith

## **Past Students**

- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- 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.)
- K. Kulkarni (M.S.)
- 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 Research Scientist**

- K. Hamidouche
- S. Sur

## **Past Programmers**

- D. Bureddy
- M. Arnold
- J. Perkins

## **Past Post-Docs**

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

# The 3<sup>rd</sup> International Workshop on High-Performance Big Data Computing (HPBDC)

HPBDC 2017 will be held with IEEE International Parallel and Distributed Processing  
Symposium (IPDPS 2017), Orlando, Florida USA, May, 2017

**Keynote Speaker: Prof. Satoshi Matsuoka, Tokyo Institute of Technology, Japan**

**Panel Moderator: Prof. Jianfeng Zhan (ICT/CAS)**

**Panel Topic: Sunrise or Sunset: Exploring the Design Space of Big Data Software Stack**

**Panel Members (Confirmed so far): Prof. Geoffrey C. Fox (Indiana University Bloomington); Dr. Raghunath Nambiar (Cisco); Prof. D. K. Panda (The Ohio State University)**

**Six Regular Research Papers and One Short Research Papers**

**Session I: High-Performance Graph Processing**

**Session II: Benchmarking and Performance Analysis**

<http://web.cse.ohio-state.edu/~luxi/hpbdc2017>

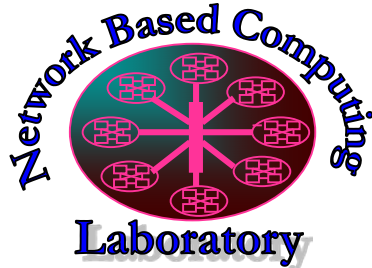


# Thank You!

{panda, luxi}@cse.ohio-state.edu

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

<http://www.cse.ohio-state.edu/~luxi>



Network-Based Computing Laboratory

<http://nowlab.cse.ohio-state.edu/>

The High-Performance Big Data Project

<http://hibd.cse.ohio-state.edu/>