HiBD
High-Performance
Big Data

# *Big Data Meets HPC: Exploiting HPC Technologies for Accelerating Big Data Processing and Management*

SigHPC BigData BoF (SC '17)

by

**Dhabaleswar K. (DK) Panda**

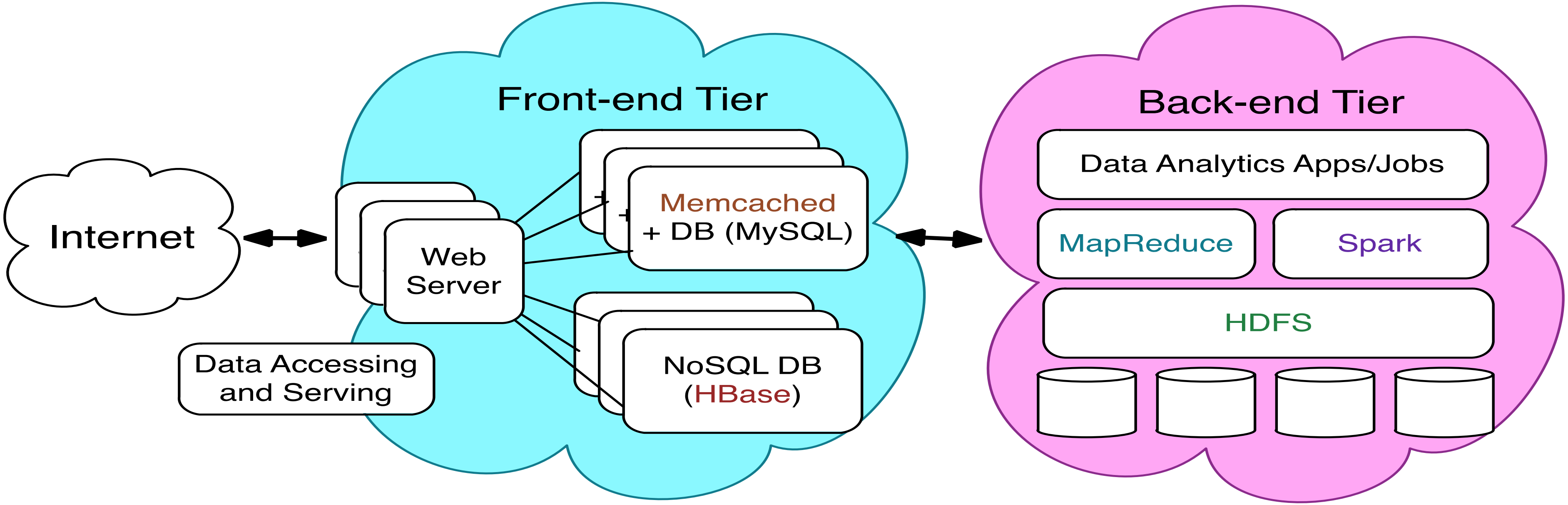
The Ohio State University

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

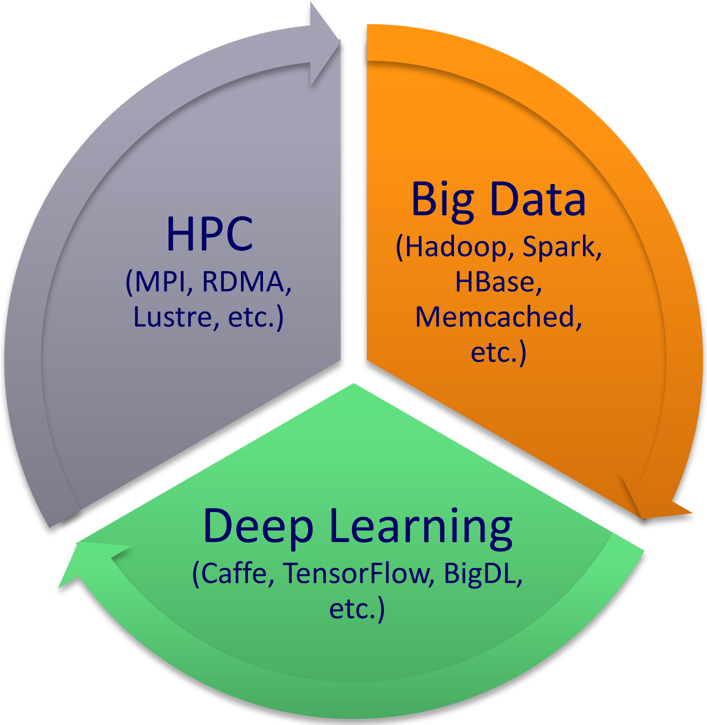
http://www.cse.ohio-state.edu/~panda

# Big Data Processing and Deep Learning on Modern Clusters

- Multiple tiers + Workflow
  - Front-end data accessing and serving (Online)
    - Memcached + DB (e.g. MySQL), HBase, etc.
  - Back-end data analytics and deep learning model training (Offline)
    - HDFS, MapReduce, Spark, TensorFlow, BigDL, Caffe, etc.



# Increasing Usage of HPC, Big Data and Deep Learning



HPC

(MPI, RDMA, Lustre, etc.)

Big Data

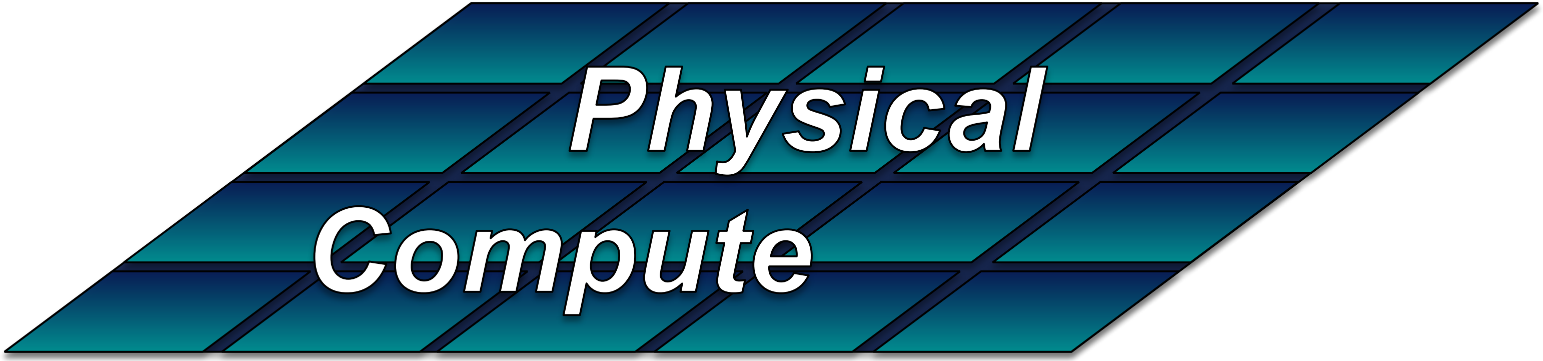
(Hadoop, Spark, HBase, Memcached, etc.)

Deep Learning

(Caffe, TensorFlow, BigDL, etc.)

**Convergence of HPC, Big Data, and Deep Learning!!!**

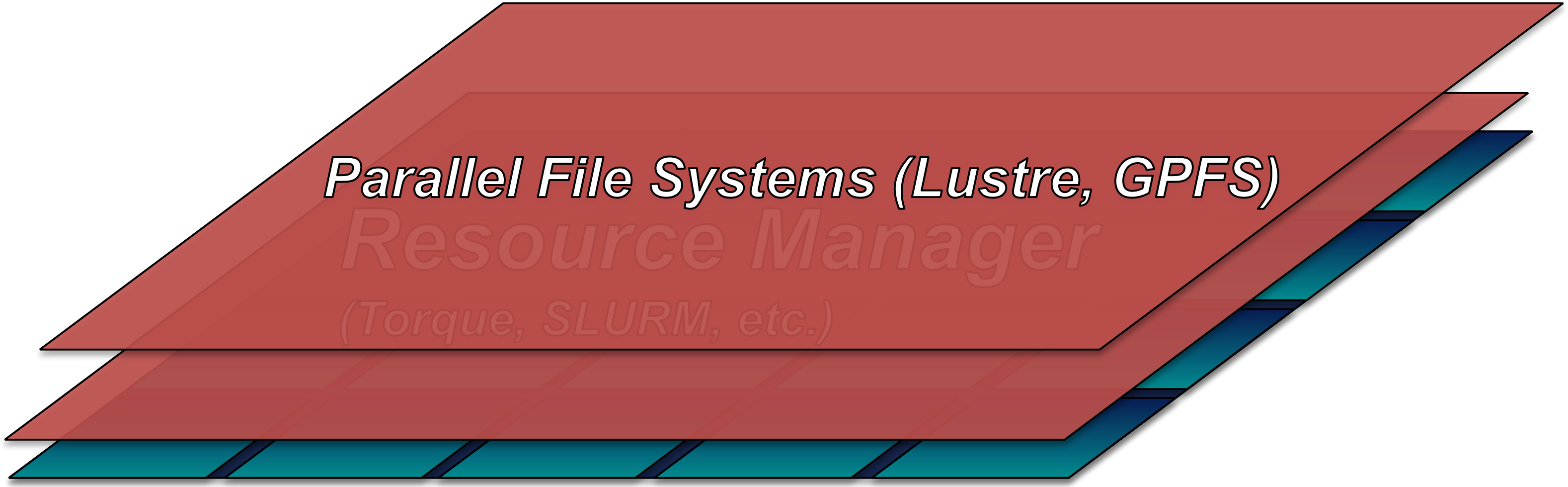
# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



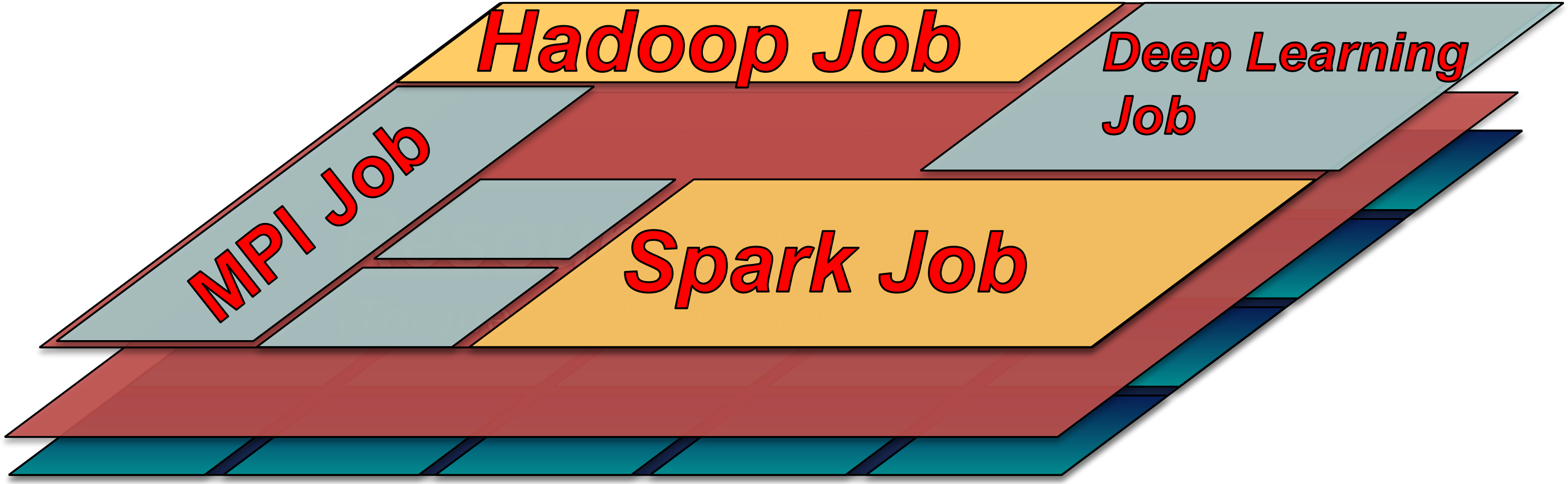
# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?

Resource Manager
(Torque, SLURM, etc.)

# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



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

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

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

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

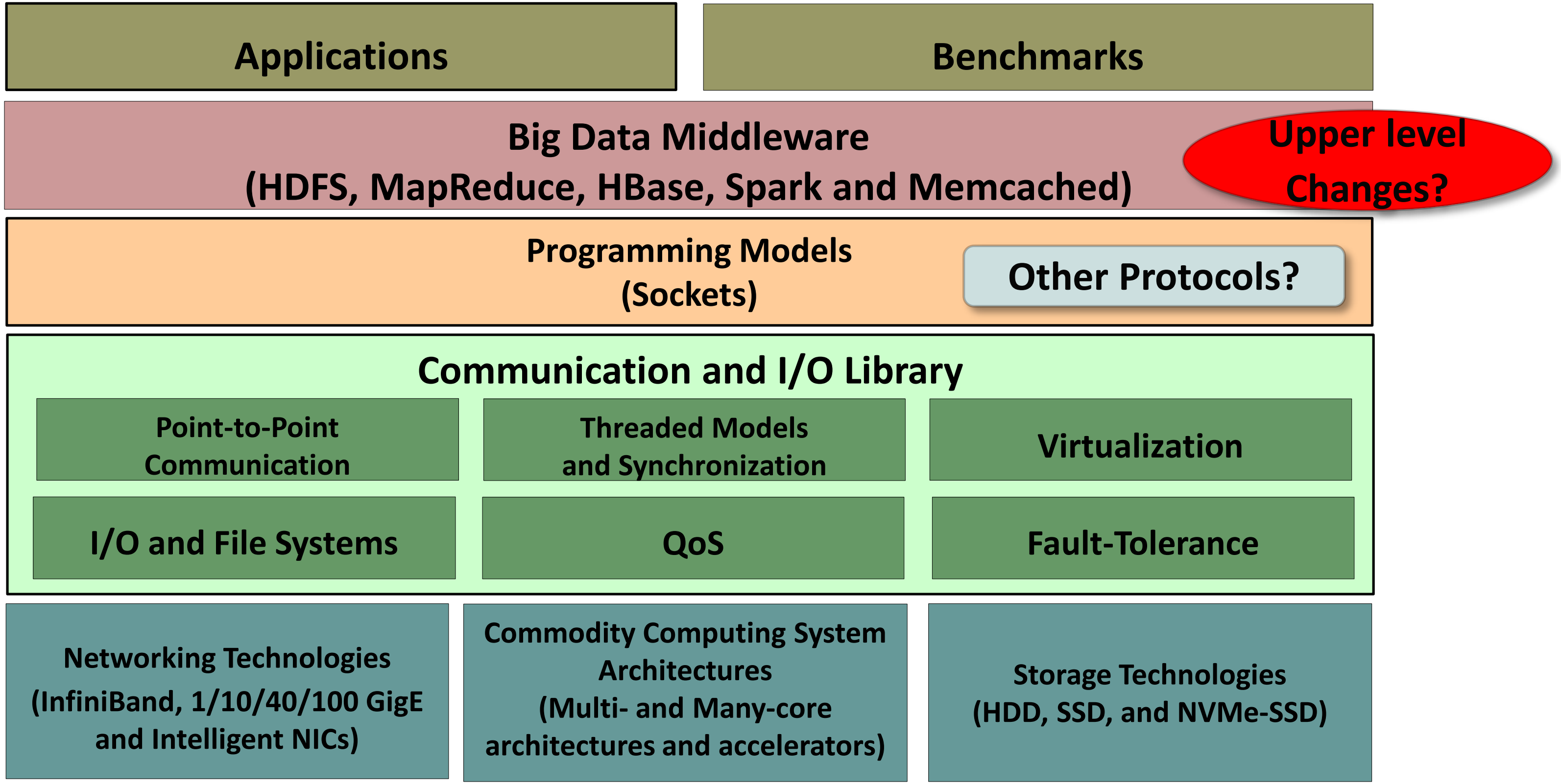
Can HPC Clusters with high-performance storage systems (e.g. SSD, parallel file systems) benefit Big Data and Deep Learning applications?

How much performance benefits can be achieved through enhanced designs?

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

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

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



Applications

Benchmarks

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

Upper level Changes?

Programming Models
(Sockets)

Other Protocols?

Communication and I/O Library

Point-to-Point Communication

Threaded Models and Synchronization

Virtualization

I/O and File Systems

QoS

Fault-Tolerance

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

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

Storage Technologies
(HDD, SSD, and NVMe-SSD)

# 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](http://hibd.cse.ohio-state.edu)
- Users Base: 260 organizations from 31 countries
- More than 23,900 downloads from the project site

**Available for InfiniBand and RoCE**

**Also run on Ethernet**

**Support for OpenPower**

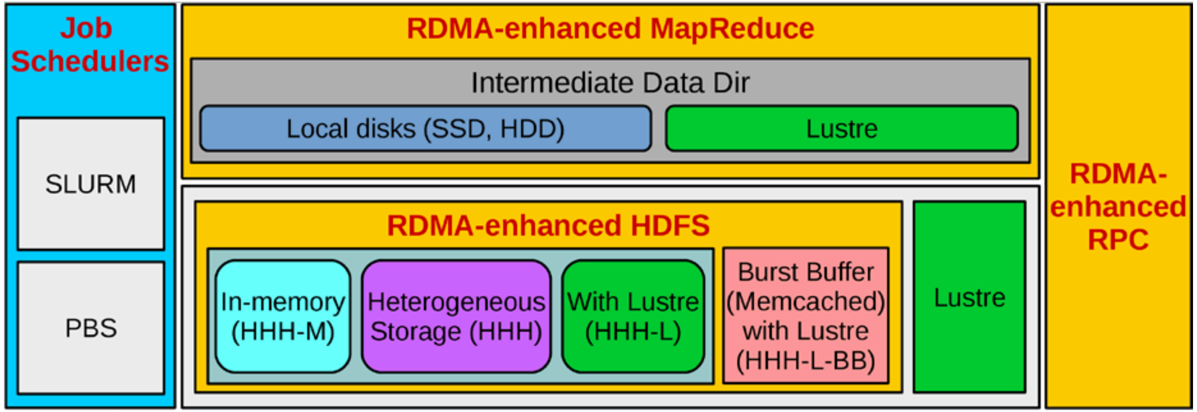
**will be released tonight**

HiBD
High-Performance
Big Data

Network Based Computing
Laboratory

THE OHIO STATE
UNIVERSITY

# Different Modes of RDMA for Apache Hadoop 2.x



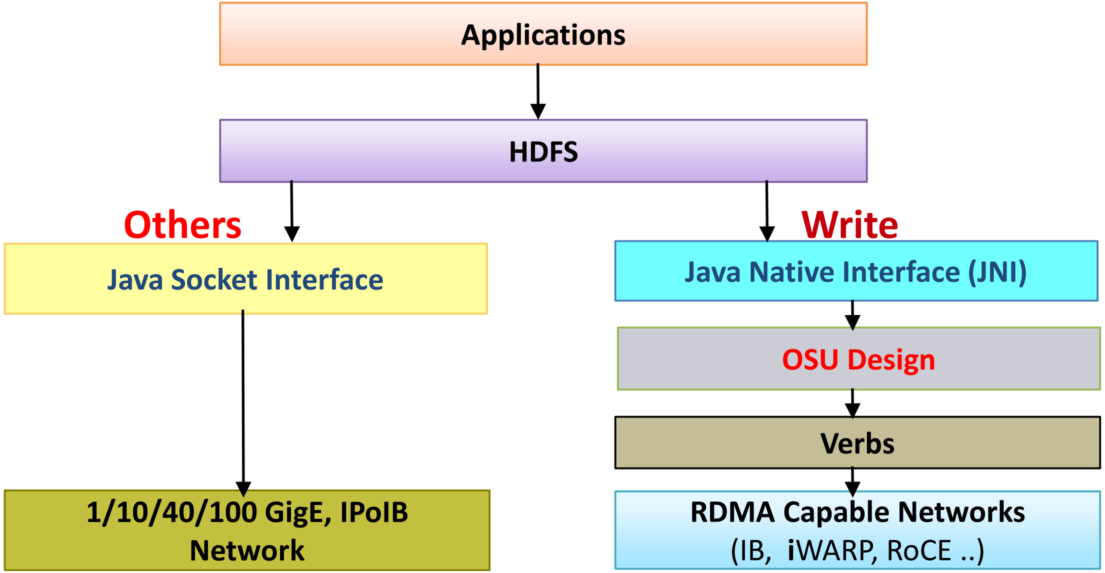
- **HHH**: Heterogeneous storage devices with hybrid replication schemes are supported in this mode of operation to have better fault-tolerance as well as performance. This mode is enabled by **default** in the package.
- **HHH-M**: A high-performance in-memory based setup has been introduced in this package that can be utilized to perform all I/O operations in-memory and obtain as much performance benefit as possible.
- **HHH-L**: With parallel file systems integrated, HHH-L mode can take advantage of the Lustre available in the cluster.
- **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).

# 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 (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**

# Design Overview of HDFS with RDMA


Applications
HDFS
Others
Write
Java Socket Interface
Java Native Interface (JNI)
OSU Design
Verbs
1/10/40/100 GigE, IPoIB Network
RDMA Capable Networks (IB, iWARP, RoCE ..)

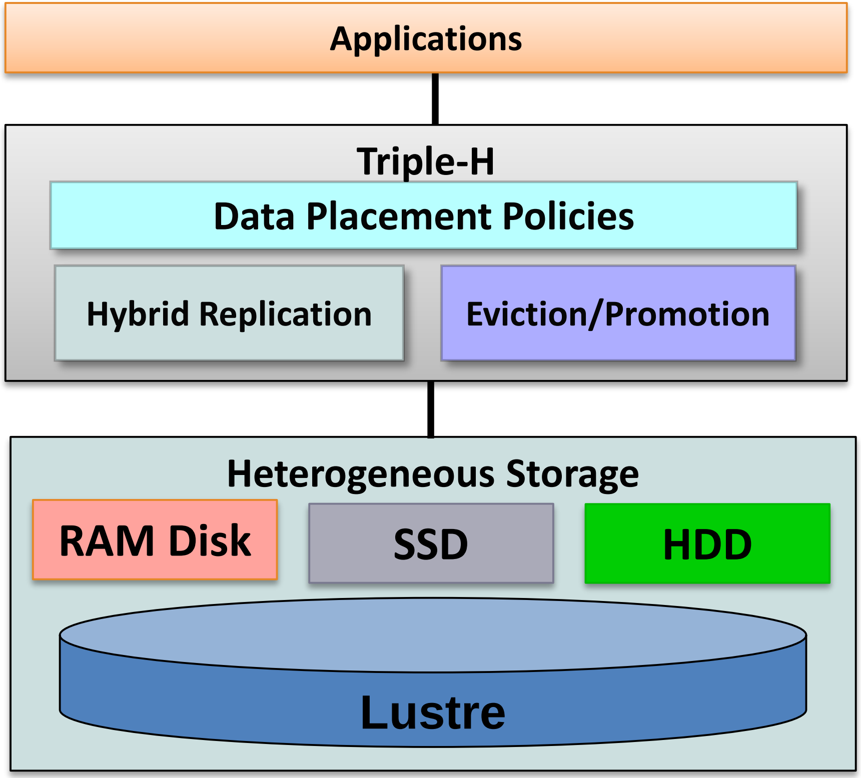
- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - Parallel replication support
  - On-demand connection setup
  - InfiniBand/RoCE support

- 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**

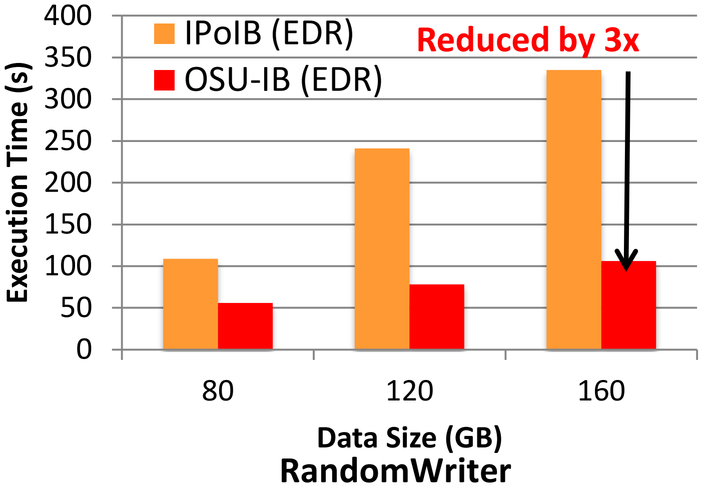
# Enhanced HDFS with In-Memory and Heterogeneous Storage



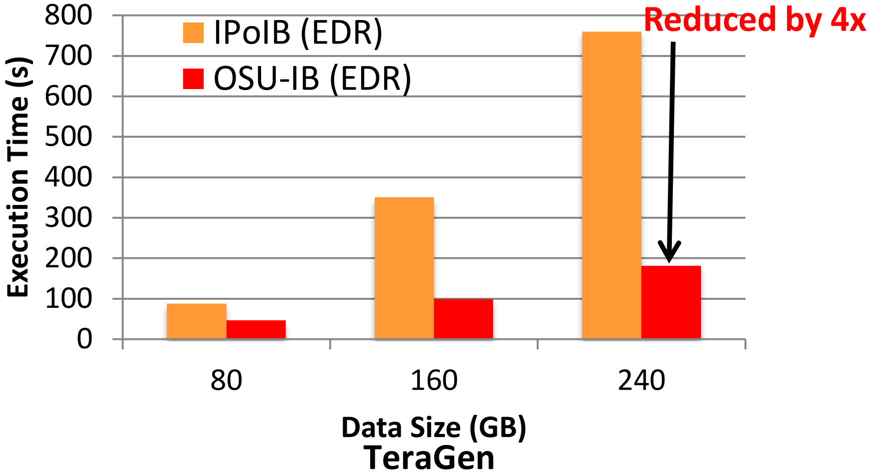
- Design Features
  - Three modes
    - Default (HHH)
    - In-Memory (HHH-M)
    - Lustre-Integrated (HHH-L)
  - Policies to efficiently utilize the heterogeneous storage devices
    - RAM, SSD, HDD, Lustre
  - Eviction/Promotion based on data usage pattern
  - Hybrid Replication
  - Lustre-Integrated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

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



RandomWriter

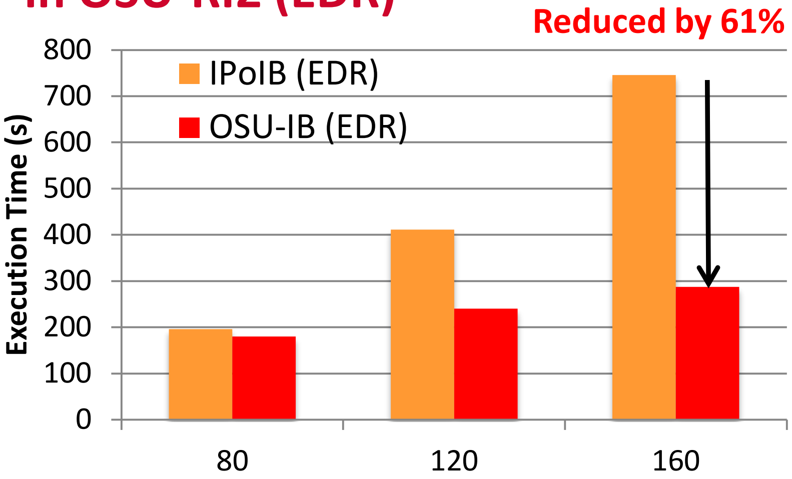


TeraGen

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

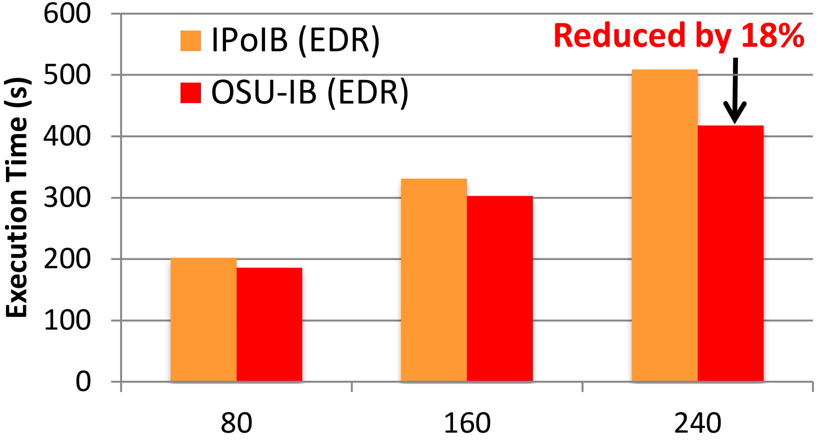
# Performance Numbers of RDMA for Apache Hadoop 2.x – Sort & TeraSort in OSU-RI2 (EDR)



**Data Size (GB)**

**Sort**

**Cluster with 8 Nodes with a total of 64 maps and 14 reduces**



**Data Size (GB)**

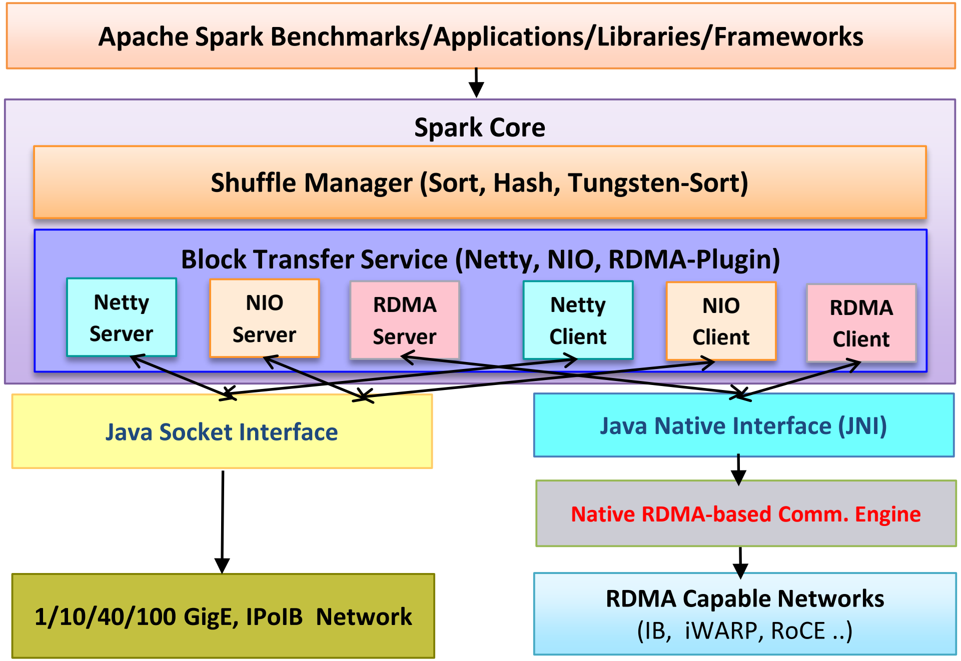
**TeraSort**

**Cluster with 8 Nodes with a total of 64 maps and 32 reduces**

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

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

# Design Overview of Spark with RDMA



Apache Spark Benchmarks/Applications/Libraries/Frameworks
Spark Core
Shuffle Manager (Sort, Hash, Tungsten-Sort)
Block Transfer Service (Netty, NIO, RDMA-Plugin)
Netty Server
NIO Server
RDMA Server
Netty Client
NIO Client
RDMA Client
Java Socket Interface
Java Native Interface (JNI)
Native RDMA-based Comm. Engine
1/10/40/100 GigE, IPoIB Network
RDMA Capable Networks
(IB, iWARP, RoCE ..)

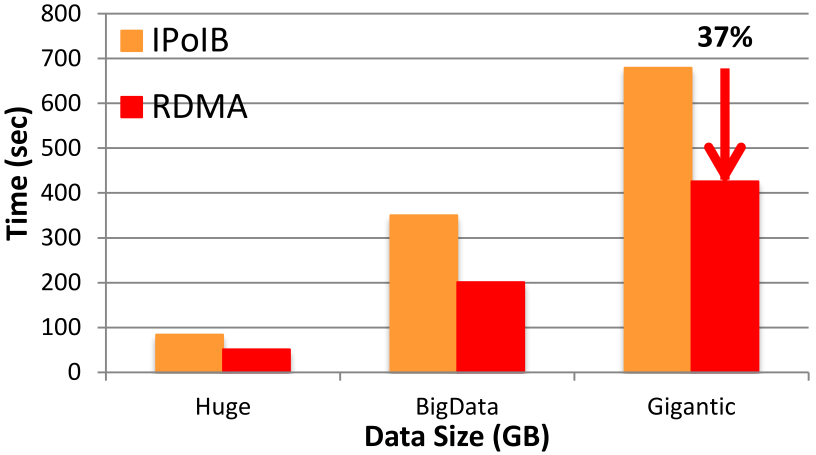
- Design Features
  - RDMA based shuffle plugin
  - SEDA-based architecture
  - Dynamic connection management and sharing
  - Non-blocking data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

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

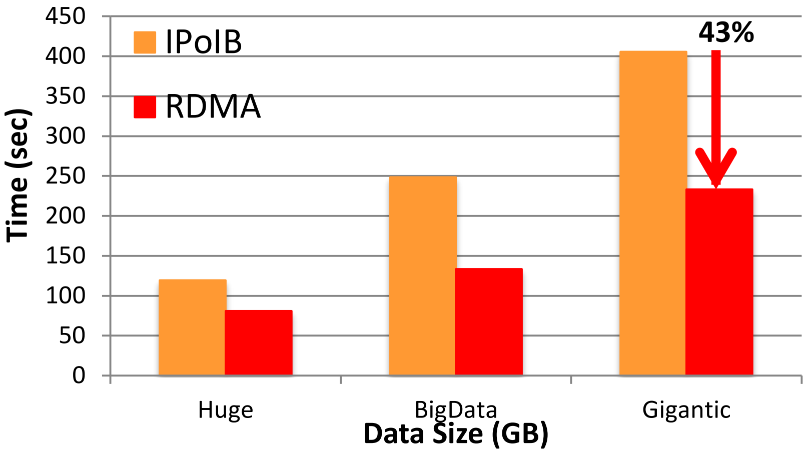
**X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014**

**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 – HiBench PageRank



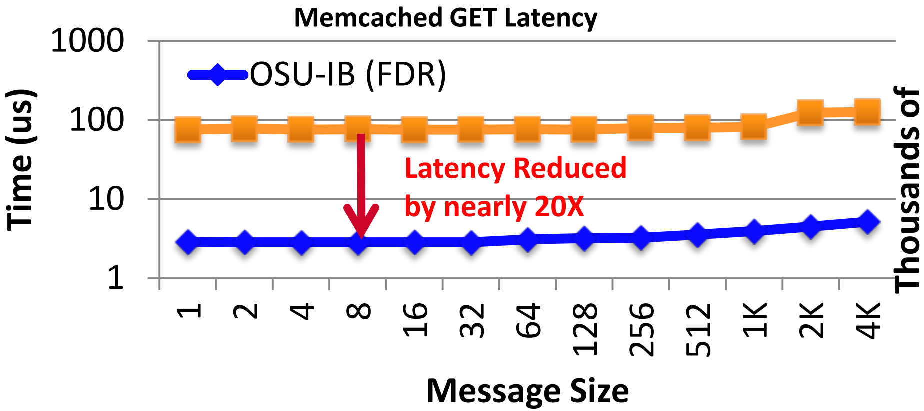
**32 Worker Nodes, 768 cores, PageRank Total Time**

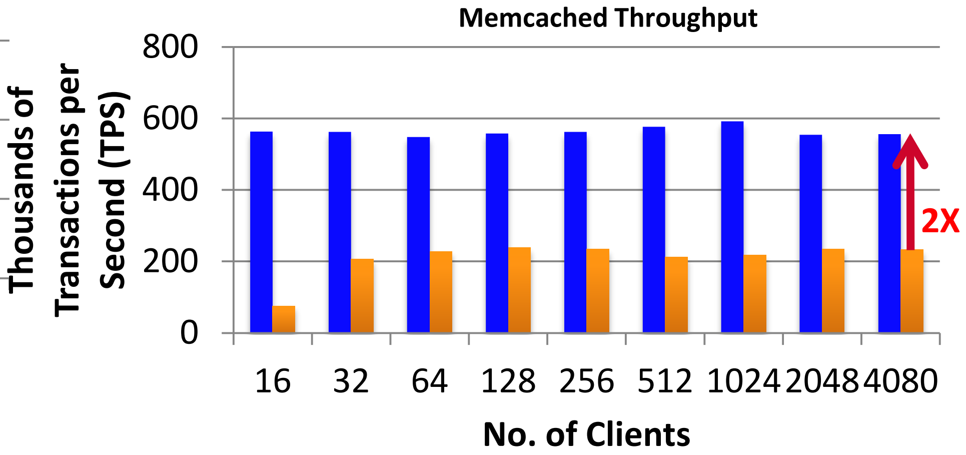


**64 Worker Nodes, 1536 cores, PageRank Total Time**

- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA-based design for Spark 1.5.1
- 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)

# 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

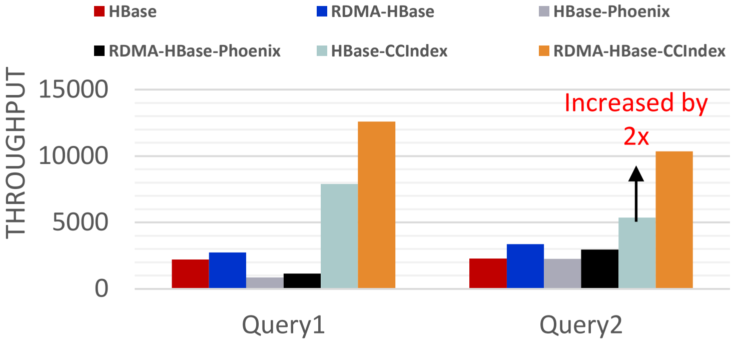
J. Jose, H. Subramoni, K. Kandalla, M. W. Rahman, H. Wang, S. Narravula, and D. K. Panda, Scalable Memcached design for InfiniBand Clusters using Hybrid Transport, CCGrid'12

# Accelerating Indexing Techniques on HBase with RDMA

- Challenges
  - Operations on Distributed Ordered Table (DOT) with indexing techniques are network intensive
  - Additional overhead of creating and maintaining secondary indices
  - Can RDMA benefit indexing techniques (Apache Phoenix and CCIndex) on HBase?
- Results
  - Evaluation with Apache Phoenix and CCIndex
  - Up to 2x improvement in query throughput
  - Up to 35% reduction in application workload execution time

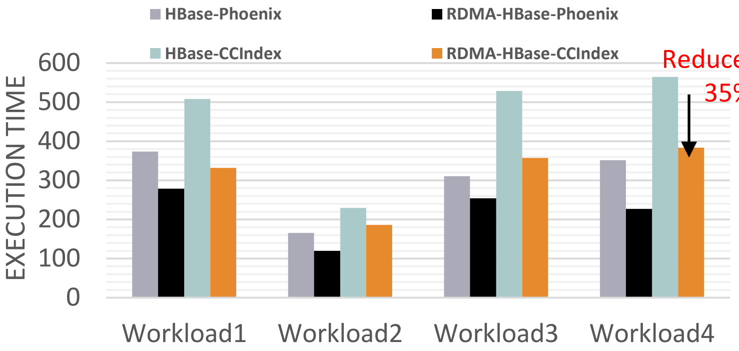
Collaboration with Institute of Computing Technology, Chinese Academy of Sciences

TPC-H Query Benchmarks



Increased by 2x

Ad Master Application Workloads

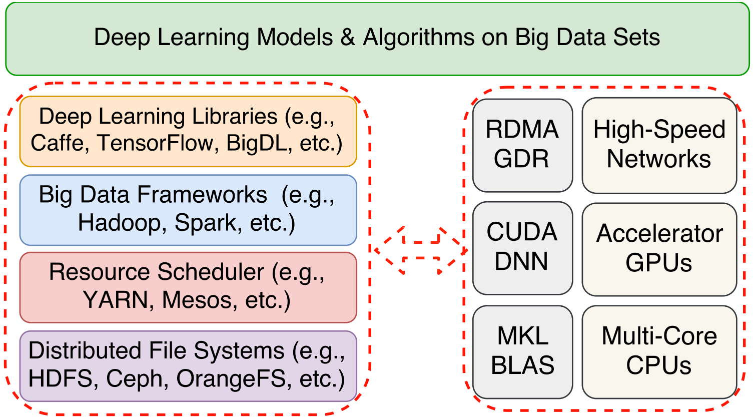


Reduced by 35%

**S. Gugnani, X. Lu, L. Zha, and D. K. Panda, Characterizing and Accelerating Indexing Techniques on Distributed Ordered Tables, IEEE BigData, 2017.**

# High-Performance Deep Learning over Big Data (DLoBD) Stacks

- Challenges of Deep Learning over Big Data (DLoBD)
  - Can RDMA-based designs in DLoBD stacks improve performance, scalability, and resource utilization on high-performance interconnects, GPUs, and multi-core CPUs?
  - What are the performance characteristics of representative DLoBD stacks on RDMA networks?
- Characterization on DLoBD Stacks
  - CaffeOnSpark, TensorFlowOnSpark, and BigDL
  - IPoIB vs. RDMA; In-band communication vs. Out-of-band communication; CPU vs. GPU; etc.
  - Performance, accuracy, scalability, and resource utilization
  - RDMA-based DLoBD stacks (e.g., BigDL over RDMA-Spark) can achieve 2.6x speedup compared to the IPoIB based scheme, while maintain similar accuracy



Deep Learning Models & Algorithms on Big Data Sets

Deep Learning Libraries (e.g., Caffe, TensorFlow, BigDL, etc.)

Big Data Frameworks (e.g., Hadoop, Spark, etc.)

Resource Scheduler (e.g., YARN, Mesos, etc.)

Distributed File Systems (e.g., HDFS, Ceph, OrangeFS, etc.)

RDMA GDR

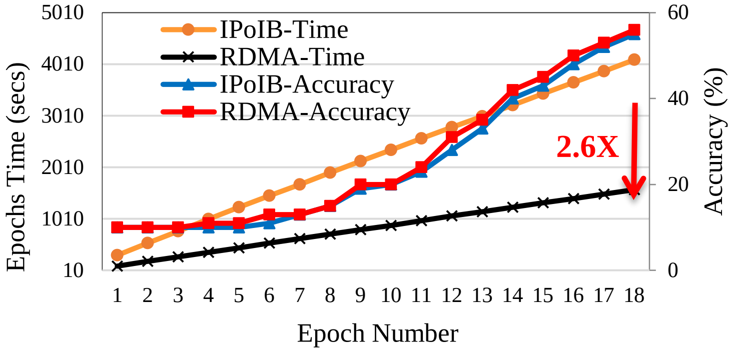
High-Speed Networks

CUDA DNN

Accelerator GPUs

MKL BLAS

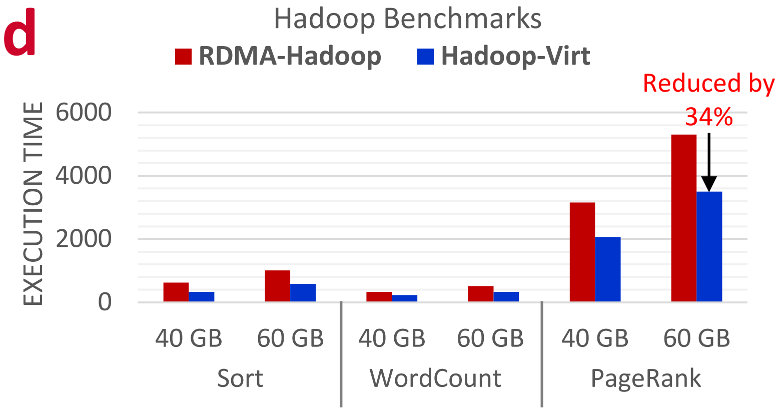
Multi-Core CPUs



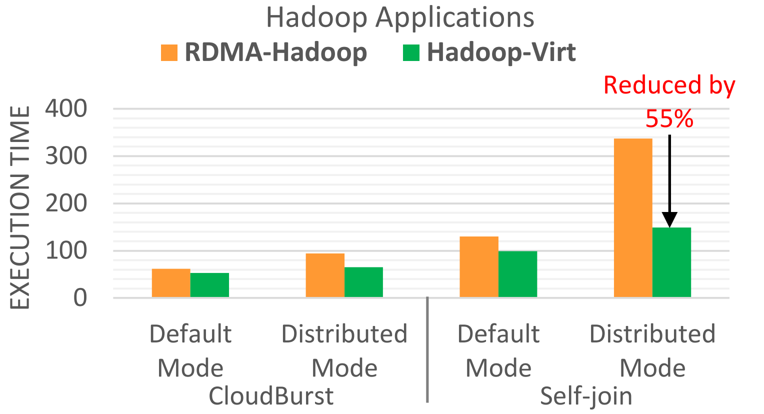
X. Lu, H. Shi, M. H. Javed, R. Biswas, and D. K. Panda, Characterizing Deep Learning over Big Data (DLoBD) Stacks on RDMA-capable Networks, HotI 2017.

# Virtualization-aware and Automatic Topology Detection Schemes in Hadoop on InfiniBand

- Challenges
  - Existing designs in Hadoop not virtualization-aware
  - No support for automatic topology detection
- Design
  - Automatic Topology Detection using MapReduce-based utility
    - Requires no user input
    - Can detect topology changes during runtime without affecting running jobs
  - Virtualization and topology-aware communication through map task scheduling and YARN container allocation policy extensions



Reduced by 34%



Reduced by 55%

**S. Gugnani, X. Lu, and D. K. Panda, Designing Virtualization-aware and Automatic Topology Detection Schemes for Accelerating Hadoop on SR-IOV-enabled Clouds, CloudCom'16, December 2016**

# Concluding Remarks

- Discussed challenges in accelerating Big Data middleware with HPC technologies
- Proposed solutions demonstrate convergence between HPC and BigData
- 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

# Thank You!

**{panda}@cse.ohio-state.edu**

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

Network-Based Computing Laboratory

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

The High-Performance Big Data Project

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