

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

Presentation at Mellanox Theatre (SC '17)

by

Dhabaleswar K. (DK) Panda

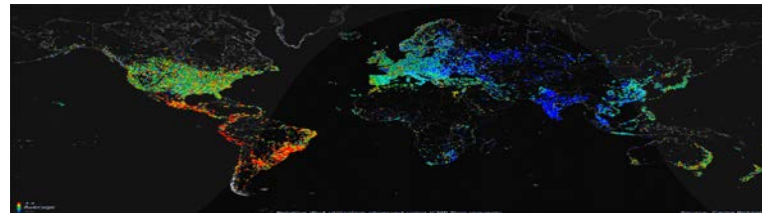
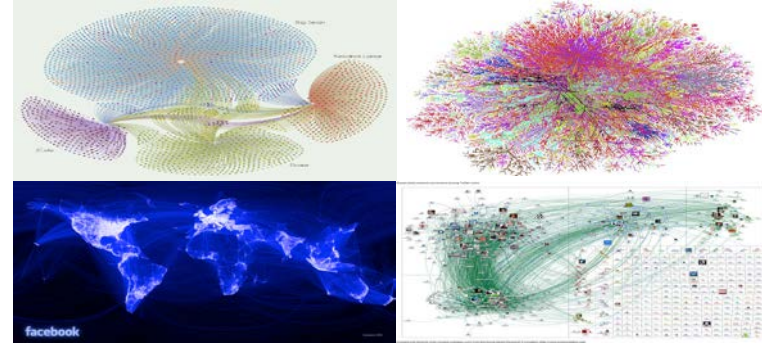
The Ohio State University

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

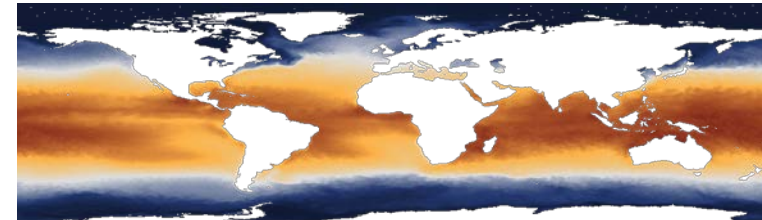
<http://www.cse.ohio-state.edu/~panda>

Introduction to Big Data Analytics and Trends

- **Big Data** has changed the way people understand and harness the power of data, both in the business and research domains
- Big Data has become one of the most important elements in business analytics
- Big Data and High Performance Computing (**HPC**) are **converging** to meet large scale data processing challenges
- Running High Performance Data Analysis (**HPDA**) workloads in the **cloud** is gaining popularity
 - According to the latest OpenStack survey, **27%** of cloud deployments are running HPDA workloads



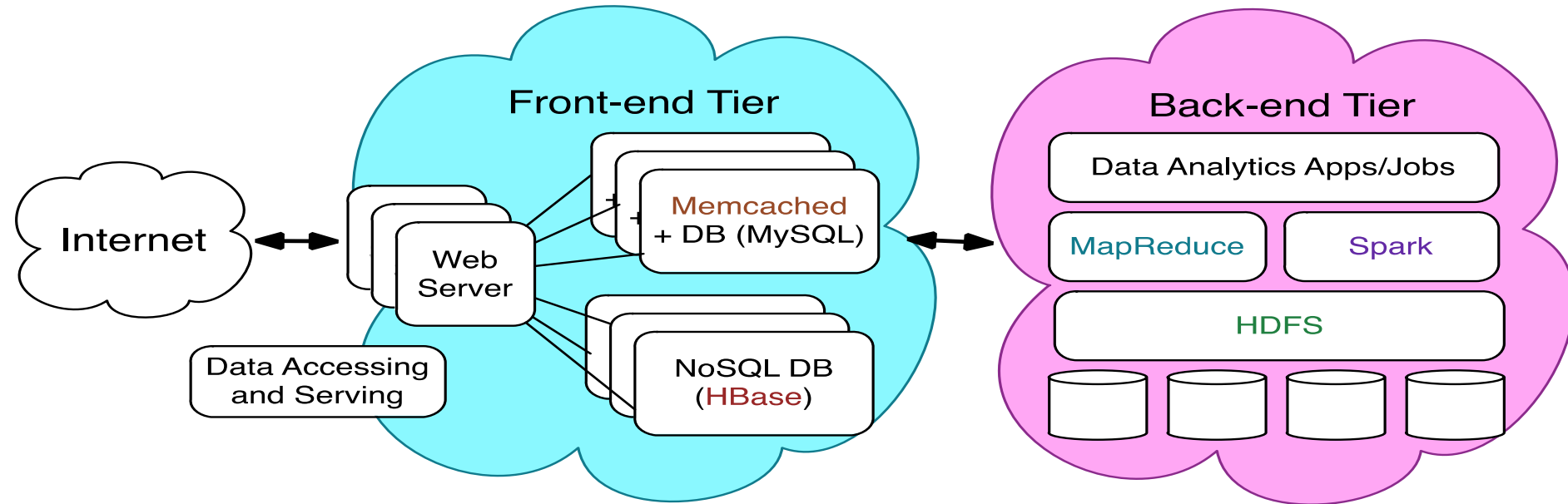
<http://www.coolinfographics.com/blog/tag/data?currentPage=3>



<http://www.climatecentral.org/news/white-house-brings-together-big-data-and-climate-change-17194>

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 and Data Center Architecture



Multi-/Many-core
Processors



High Performance Interconnects –
InfiniBand (with SR-IOV)
<1usec latency, 200Gbps 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)
 - Single Root I/O Virtualization (SR-IOV)
- Solid State Drives (SSDs), NVM, Parallel Filesystems, Object Storage Clusters
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)



SDSC Comet



TACC Stampede



EC2

ORACLE
Cloud

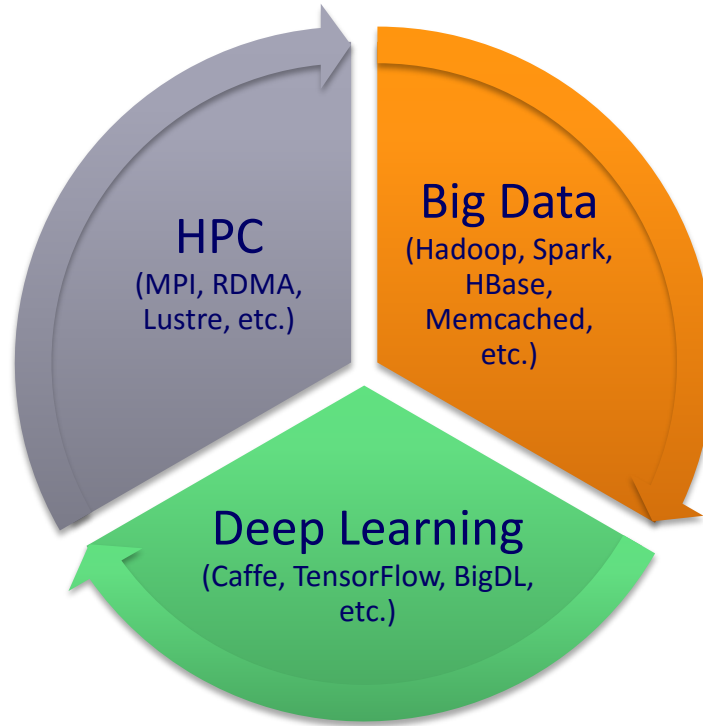


Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
 - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002
 - MVAPICH2-X (MPI + PGAS), Available since 2011
 - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
 - Support for Virtualization (MVAPICH2-Virt), Available since 2015
 - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
 - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
 - **Used by more than 2,825 organizations in 85 countries**
 - **More than 433,000 (> 0.4 million) downloads from the OSU site directly**
 - Empowering many TOP500 clusters (June '17 ranking)
 - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
 - 15th, 241,108-core (Pleiades) at NASA
 - 20th, 462,462-core (Stampede) at TACC
 - 44th, 74,520-core (Tsubame 2.5) at Tokyo Institute of Technology
 - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
 - <http://mvapich.cse.ohio-state.edu>
- Empowering Top500 systems for over a decade
 - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
 - Sunway TaihuLight (1st in Jun'17, 10M cores, 100 PFlops)

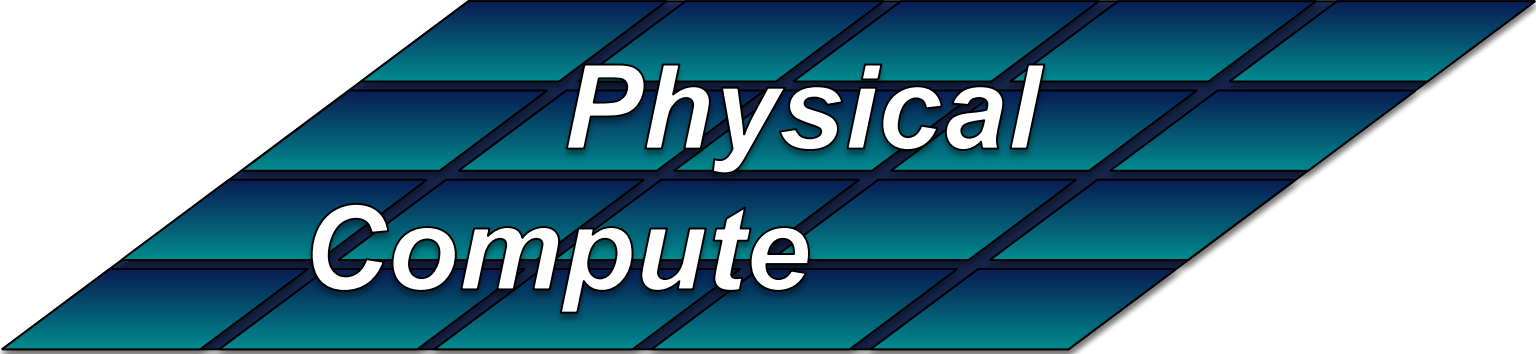


Increasing Usage of HPC, Big Data and Deep Learning



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

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



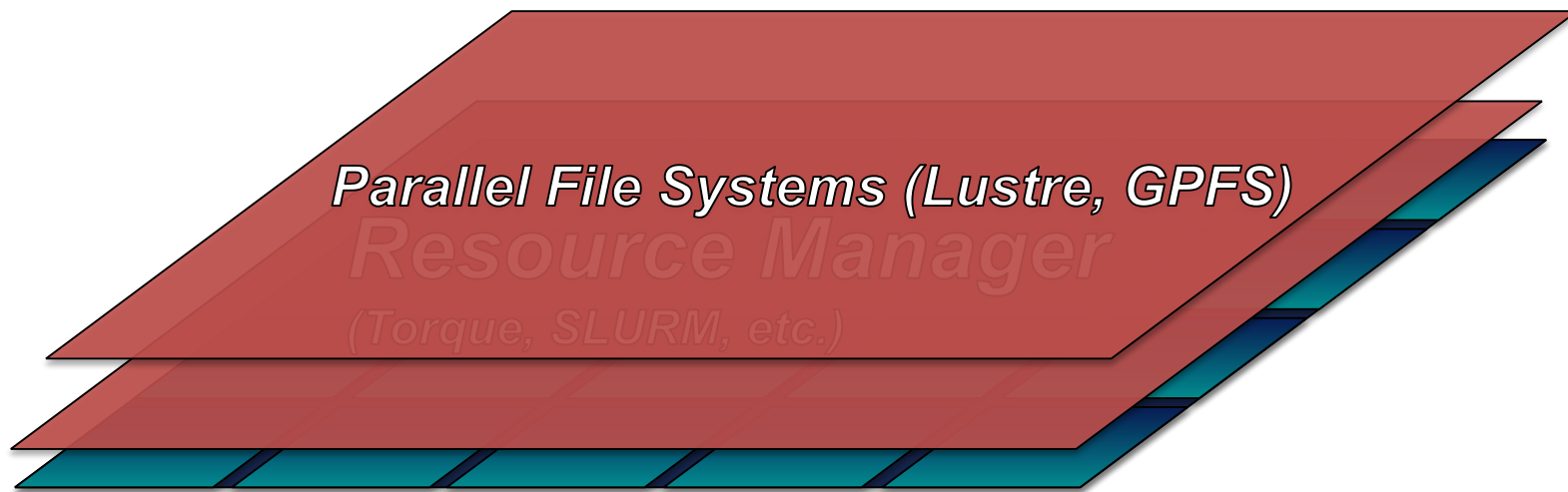
*Physical
Compute*

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?



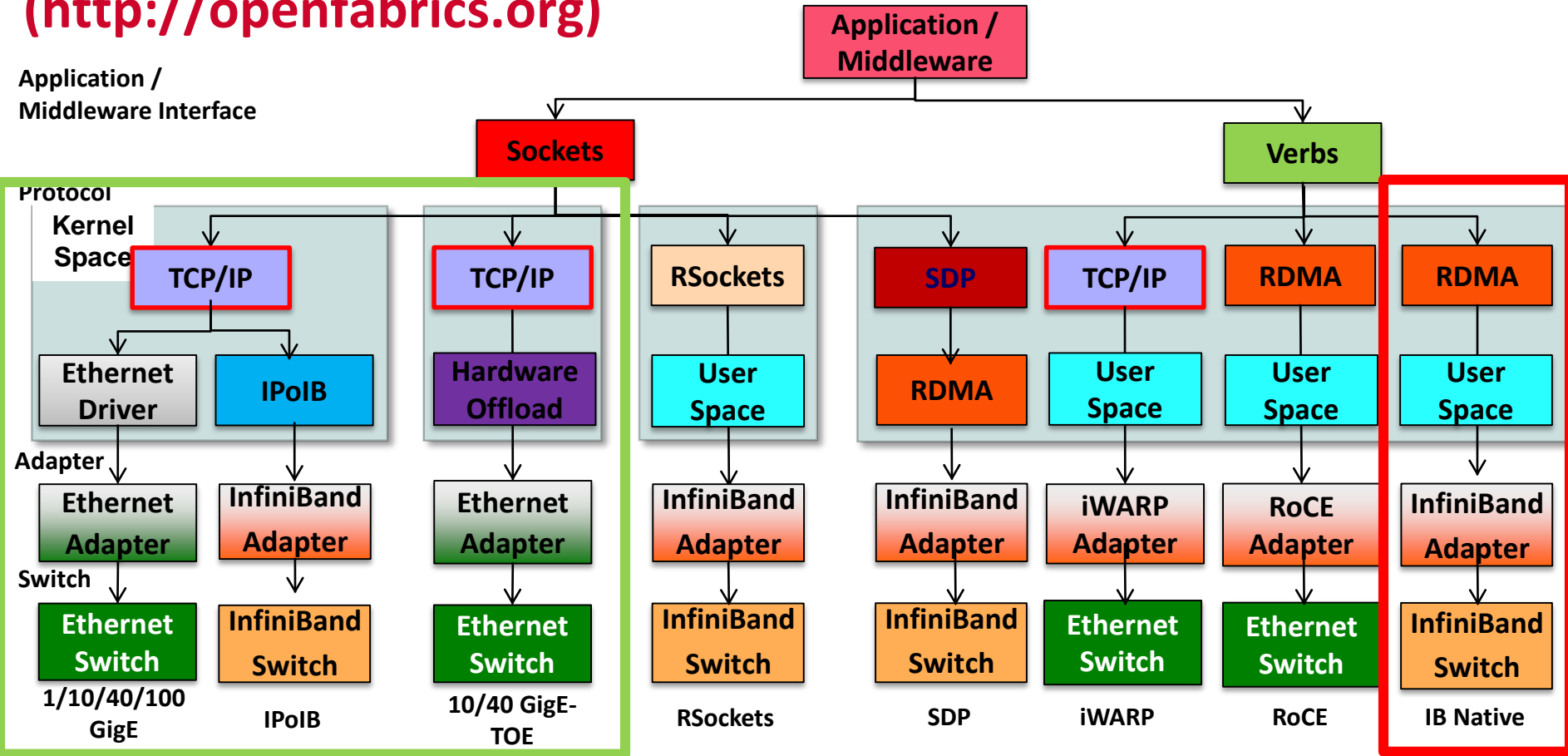
Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



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?

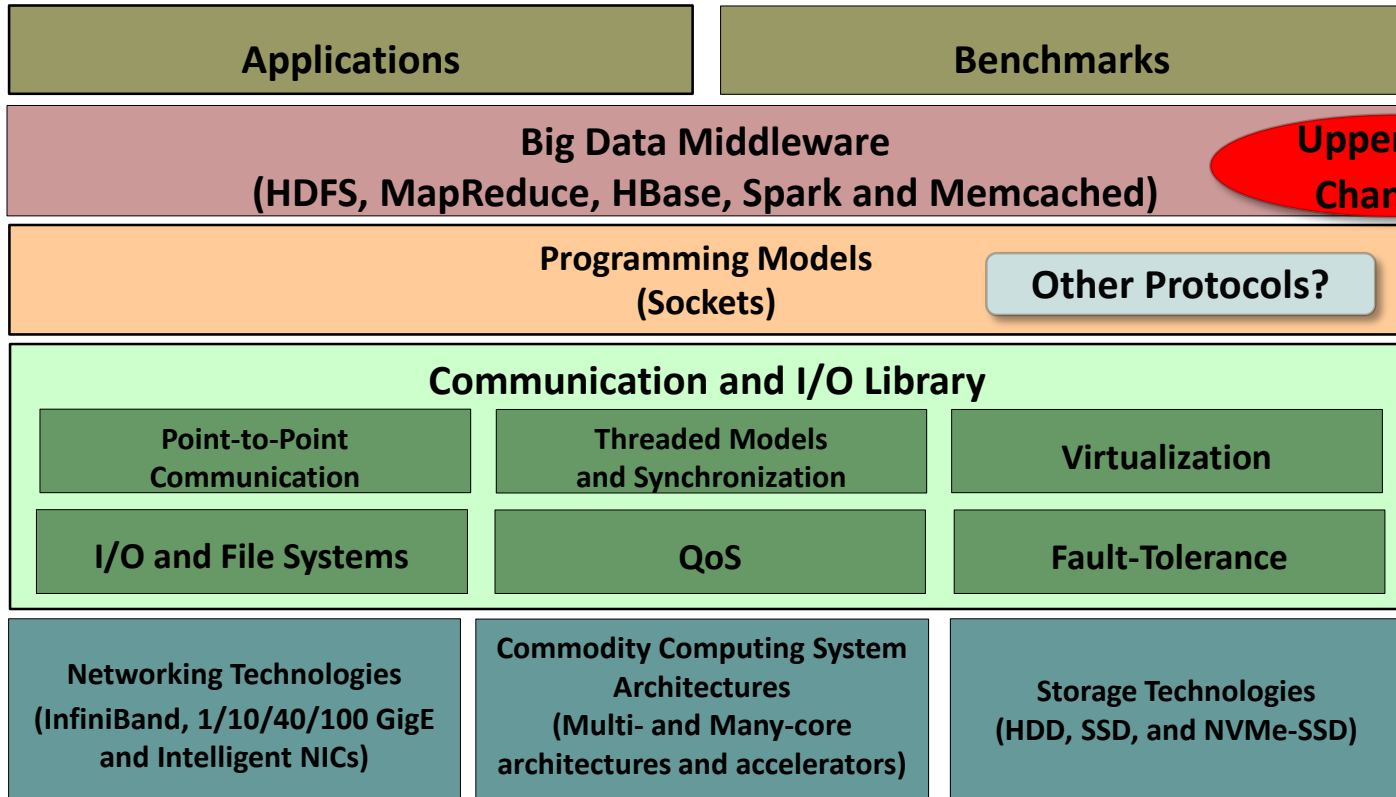
What are the major **bottlenecks** in current Big Data processing middleware (e.g. Hadoop, Spark, and Memcached)?

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



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: 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

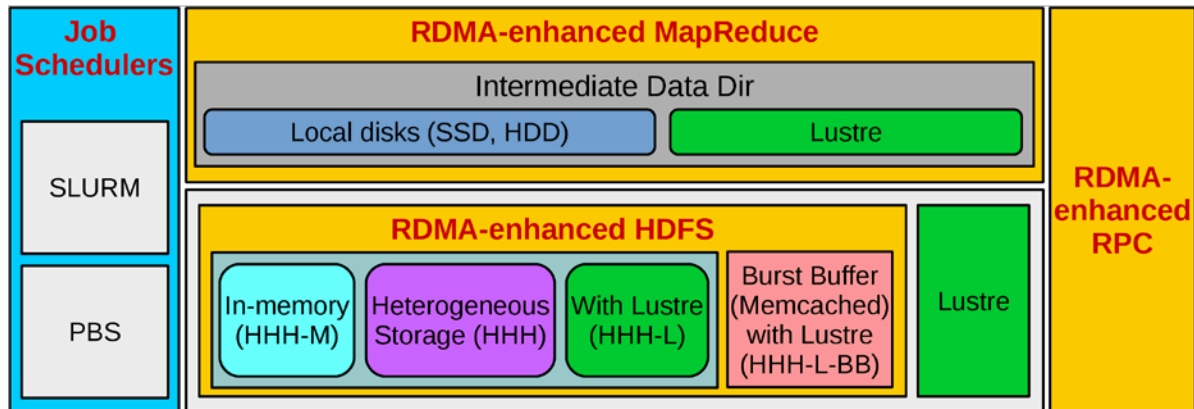


RDMA for Apache Hadoop 2.x Distribution

- 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: **1.2.0**
 - Based on Apache Hadoop **2.8.0**
 - Compliant with Apache Hadoop 2.8.0, 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 Luster 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 Luster, with/without local disks:** Besides, HDFS based solutions, this package also provides support to run MapReduce jobs on top of Luster 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 Luster).

RDMA for Apache Spark Distribution

- High-Performance Design of Spark over RDMA-enabled Interconnects
 - RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for Spark
 - RDMA-based data shuffle and SEDA-based shuffle architecture
 - Non-blocking and chunk-based data transfer
 - RDMA support for Spark SQL
 - Integration with HHH in RDMA for Apache Hadoop
 - 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 and FDR)
 - RoCE support with Mellanox adapters
 - Various multi-core platforms
 - RAM disks, SSDs, and HDD
 - <http://hibd.cse.ohio-state.edu>

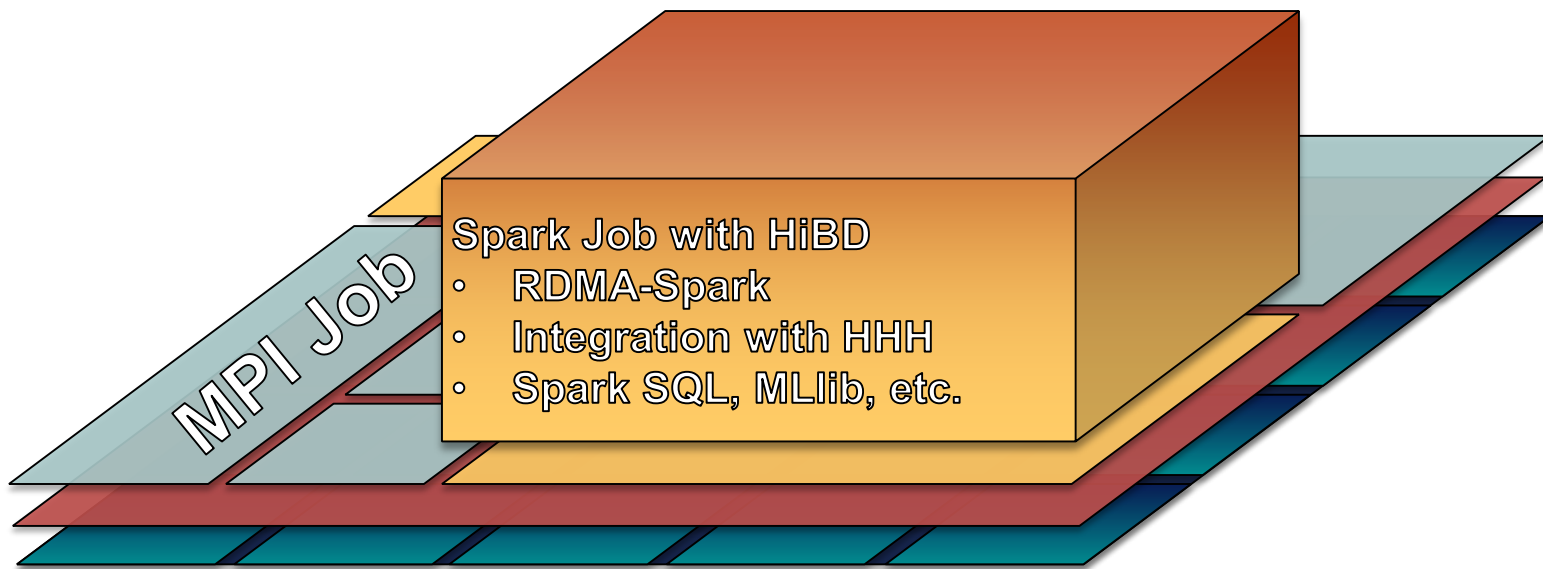
OSU HiBD Micro-Benchmark (OHB) Suite – HDFS, Memcached, HBase, and Spark

- 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
- Micro-benchmarks for Spark
 - GroupBy, SortBy
- Current release: **0.9.2**
- <http://hibd.cse.ohio-state.edu>

Using HiBD Packages for Big Data Processing on Existing HPC Infrastructure



Using HiBD Packages for Big Data Processing on Existing HPC Infrastructure



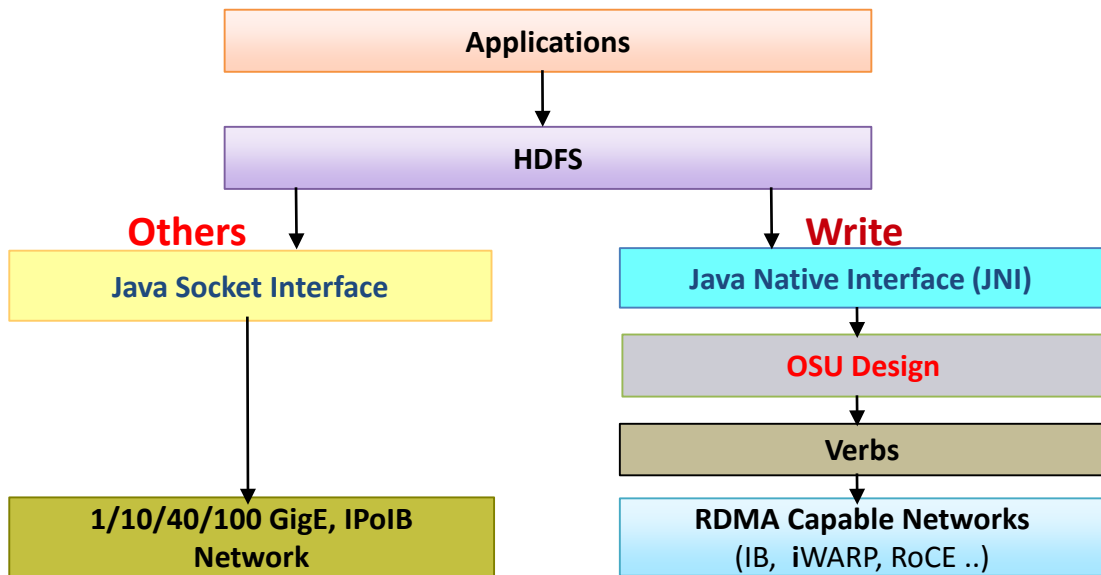
HiBD Packages on SDSC Comet and Chameleon Cloud

- 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/>

Acceleration Case Studies and Performance Evaluation

- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
- BigData + HPC Cloud

Design Overview of HDFS with RDMA



- Design Features

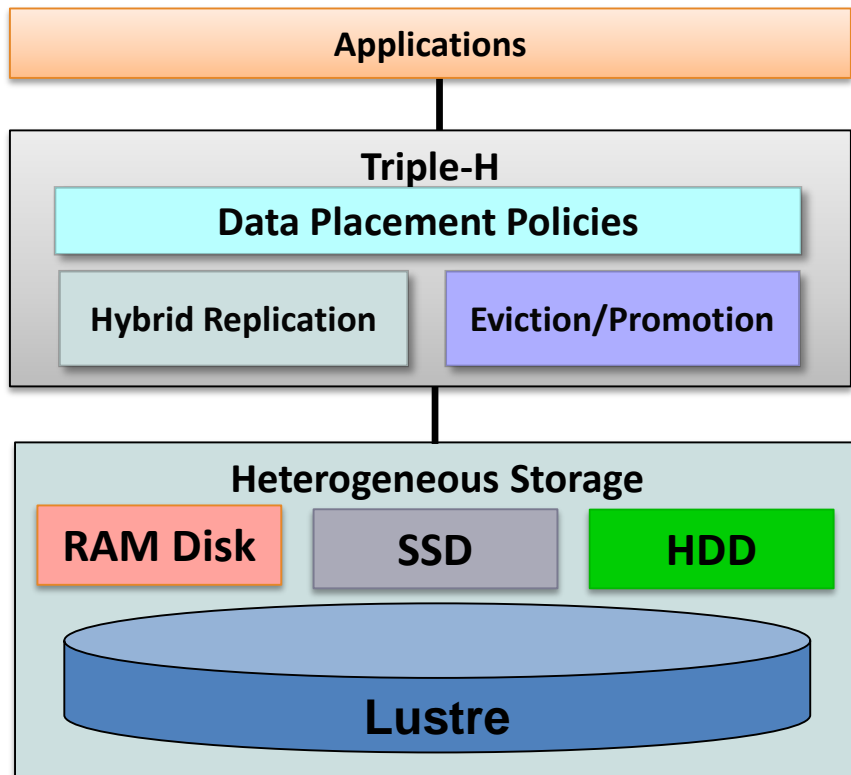
- RDMA-based HDFS write
- RDMA-based HDFS replication
- Parallel replication support
- On-demand connection setup
- InfiniBand/RoCE support

- 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

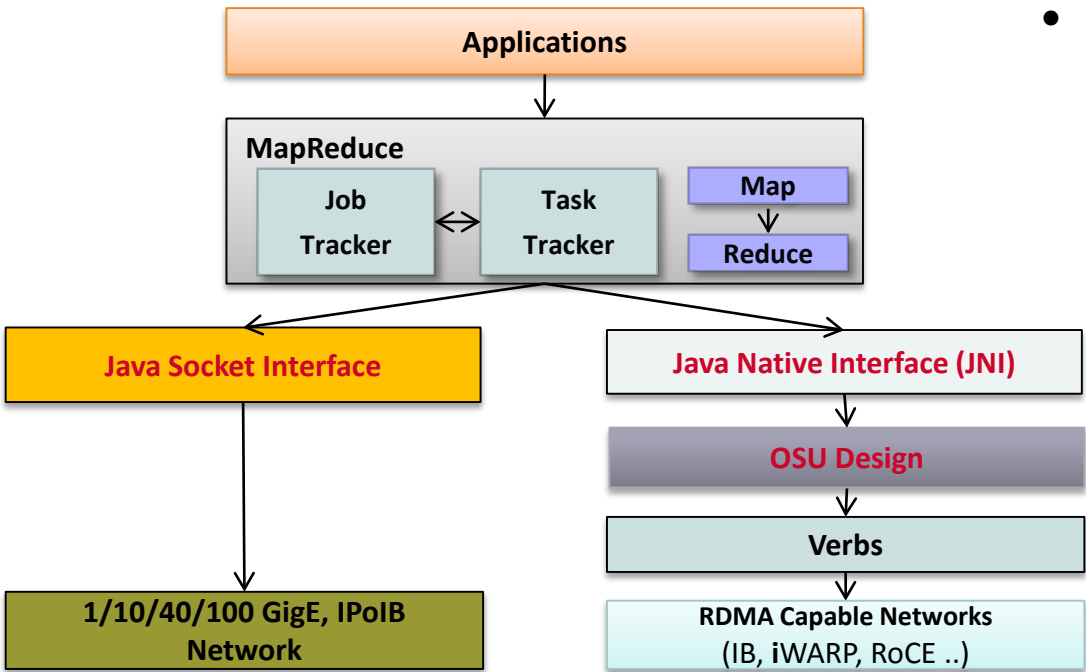
Enhanced HDFS with In-Memory and Heterogeneous Storage



- 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



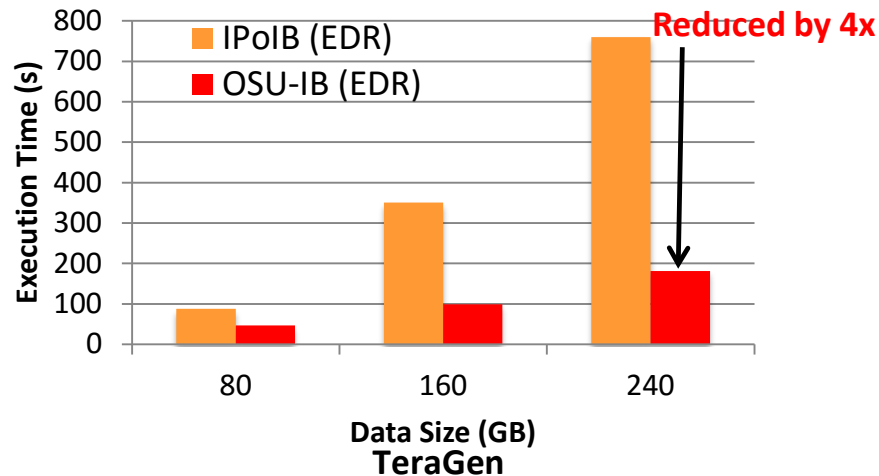
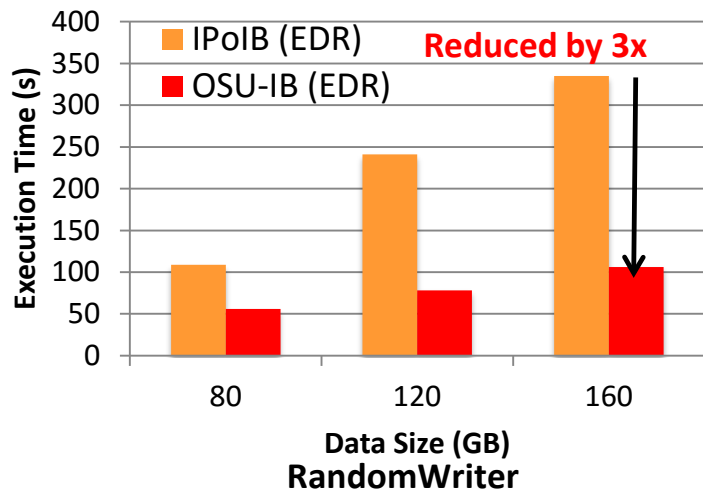
- Design Features

- 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

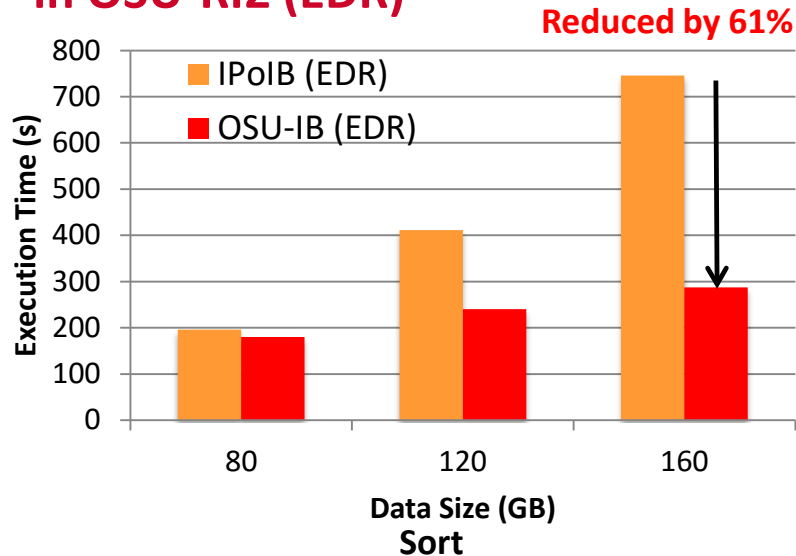
- RandomWriter

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

- TeraGen

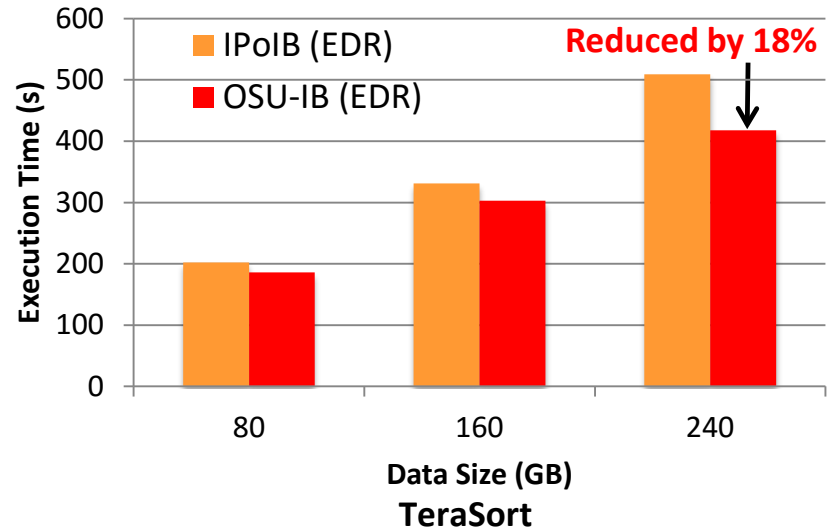
- **4x** improvement over IPoIB for 80-240 GB file size

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



- Sort

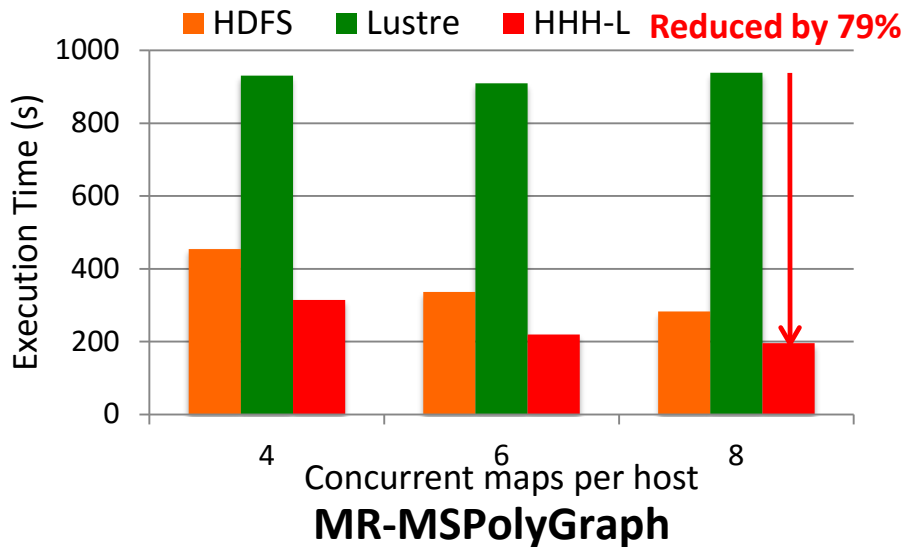
- **61%** improvement over IPoIB for 80-160 GB data



- TeraSort

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

Evaluation of HHH and HHH-L with Applications



HDFS (FDR)	HHH (FDR)
60.24 s	48.3 s

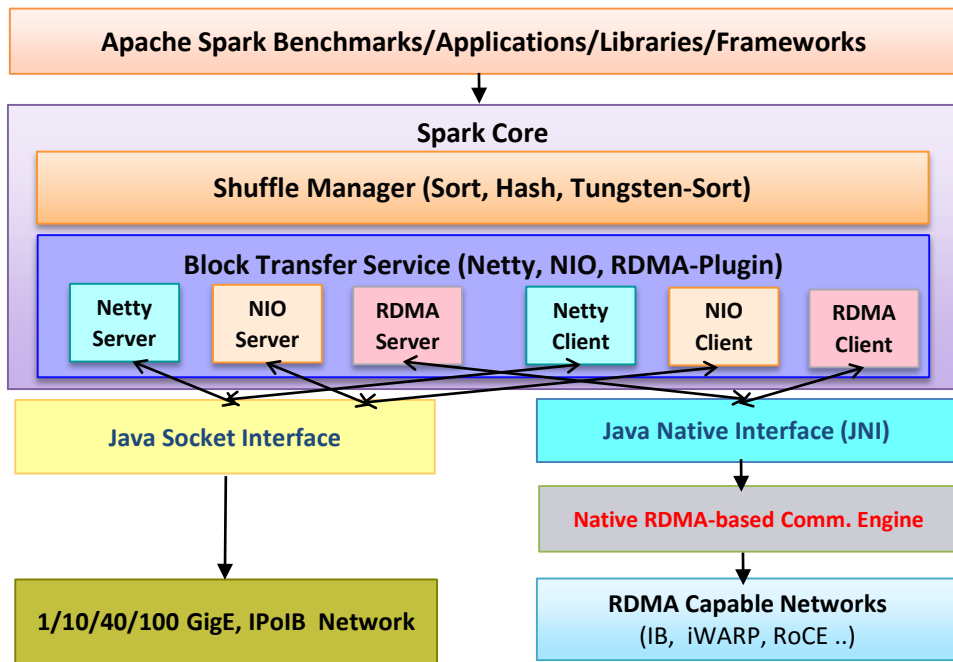
CloudBurst

- MR-MSPolygraph on OSU RI with 1,000 maps
 - HHH-L reduces the execution time by **79%** over Lustre, **30%** over HDFS
- CloudBurst on TACC Stampede
 - With HHH: **19%** improvement over HDFS

Acceleration Case Studies and Performance Evaluation

- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- BigData + HPC Cloud

Design Overview of Spark with RDMA



- Design Features

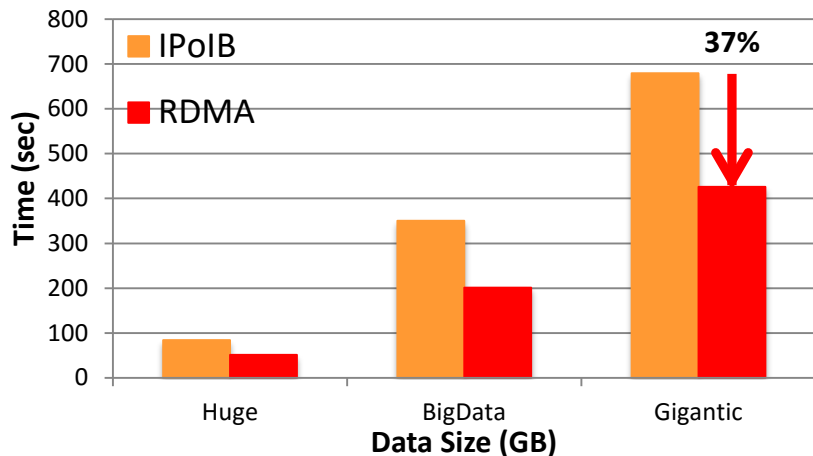
- RDMA based shuffle plugin
- SEDA-based architecture
- Dynamic connection management and sharing
- Non-blocking data transfer
- Off-JVM-heap buffer management
- InfiniBand/RoCE support

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

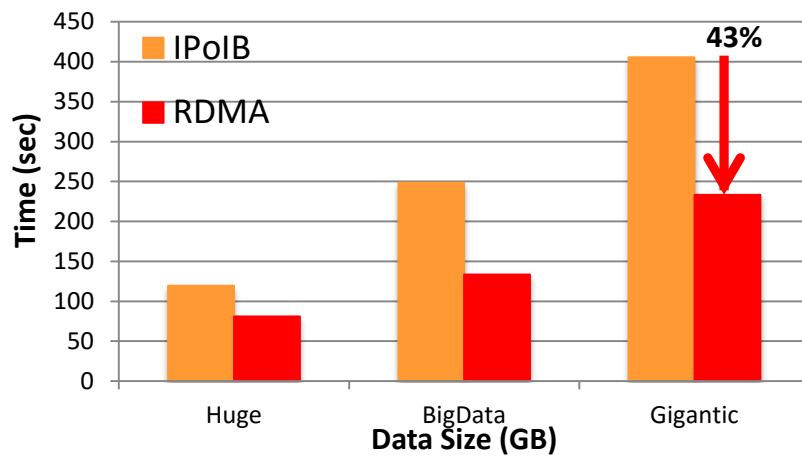
X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, *Accelerating Spark with RDMA for Big Data Processing: Early Experiences*, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

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 – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time



64 Worker Nodes, 1536 cores, PageRank Total Time

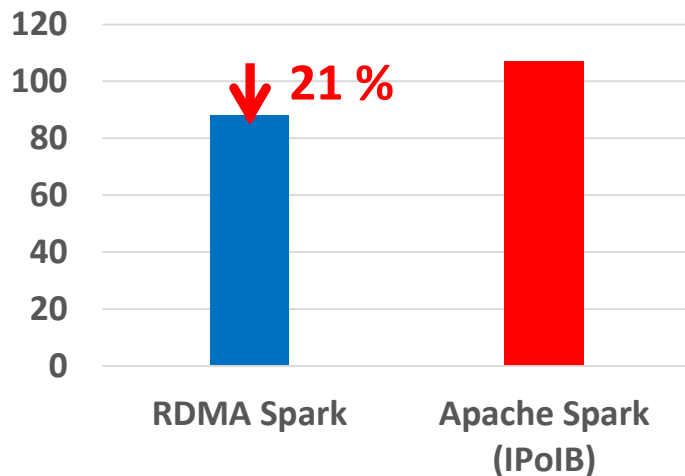
- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA-based design for Spark 1.5.1
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
 - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
 - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

Performance Evaluation on SDSC Comet: Astronomy Application

- **Kira Toolkit¹**: Distributed astronomy image processing toolkit implemented using Apache Spark.
- Source extractor application, using a 65GB dataset from the SDSS DR2 survey that comprises 11,150 image files.
- Compare RDMA Spark performance with the standard apache implementation using IPoIB.

1. Z. Zhang, K. Barbary, F. A. Nothaft, E.R. Sparks, M.J. Franklin, D.A. Patterson, S. Perlmutter. *Scientific Computing meets Big Data Technology: An Astronomy Use Case*. *CoRR*, vol: *abs/1507.03325*, Aug 2015.

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

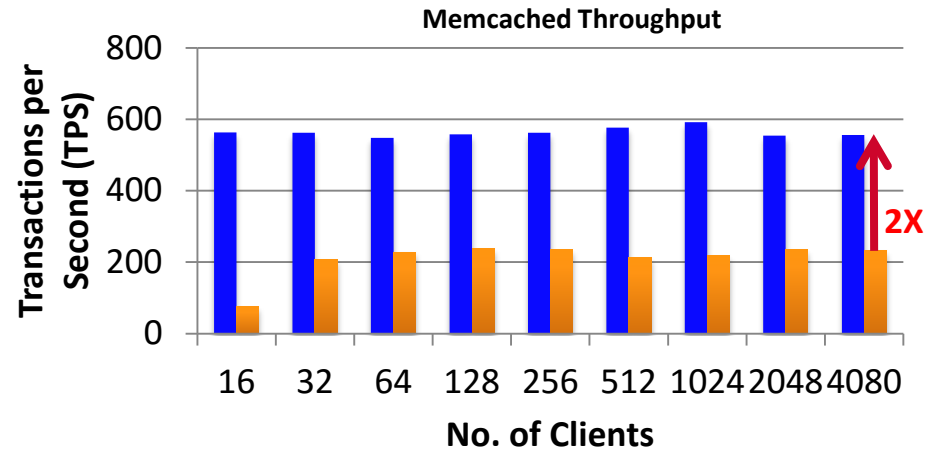
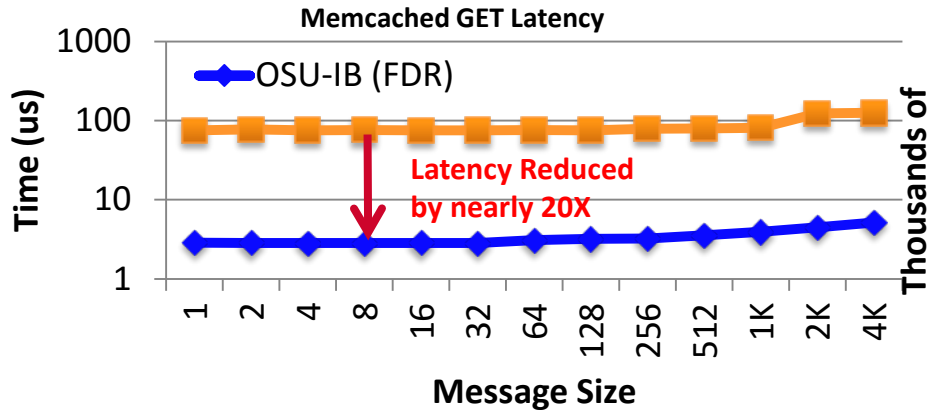


Execution times (sec) for Kira SE benchmark using 65 GB dataset, 48 cores.

Acceleration Case Studies and Performance Evaluation

- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- BigData + HPC Cloud

Memcached Performance (FDR Interconnect)



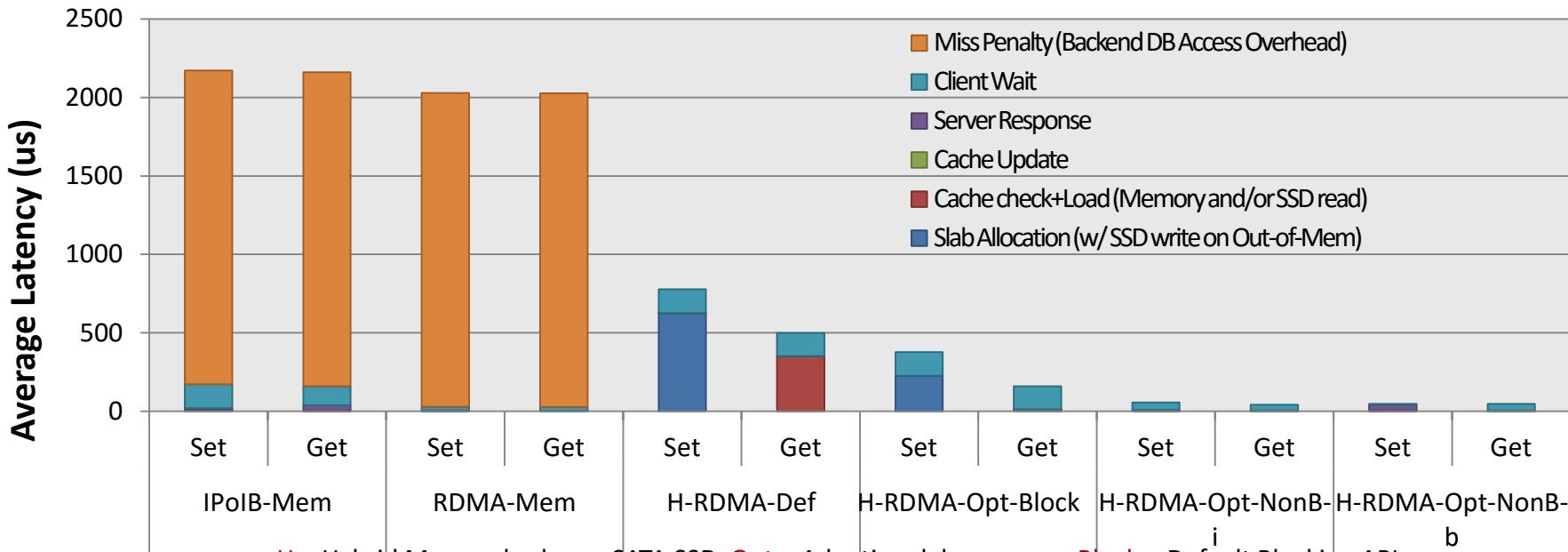
Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- 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

Performance Evaluation with Non-Blocking Memcached API



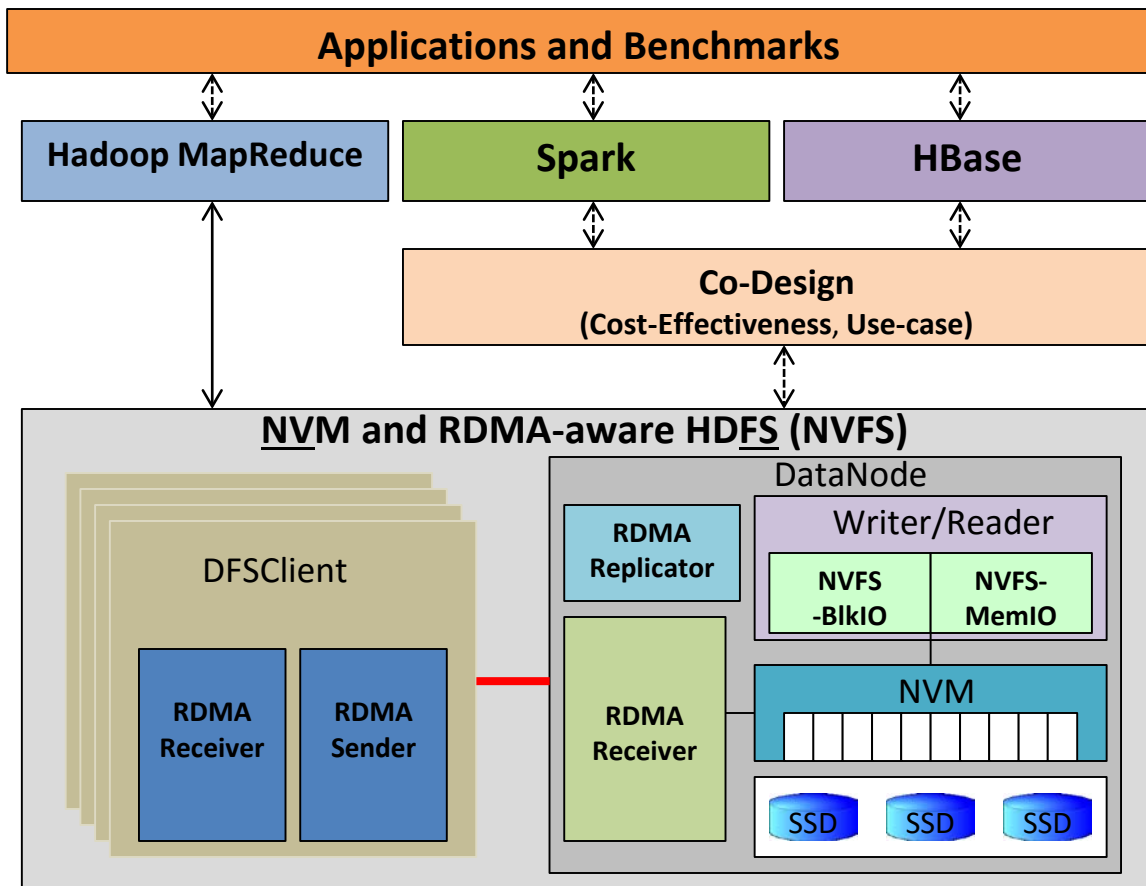
H = Hybrid Memcached over SATA SSD Opt = Adaptive slab manager Block = Default Blocking API
NonB-i = Non-blocking iset/iget API NonB-b = Non-blocking bset/bget API w/ buffer re-use guarantee

- **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
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- BigData + HPC Cloud

Design Overview of NVM and RDMA-aware HDFS (NVFS)

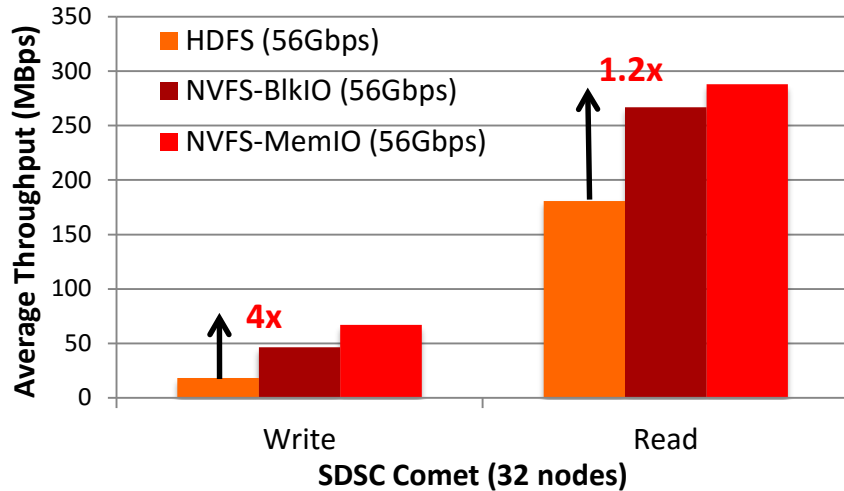


• Design Features

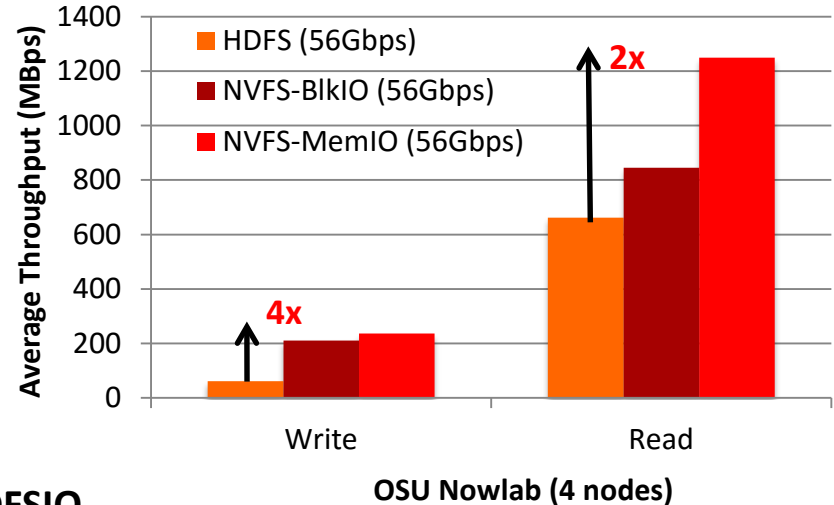
- RDMA over NVM
- HDFS I/O with NVM
 - Block Access
 - Memory Access
- Hybrid design
 - NVM with SSD as a hybrid storage for HDFS I/O
- Co-Design with Spark and HBase
 - Cost-effectiveness
 - Use-case

N. S. Islam, M. W. Rahman, X. Lu, and D. K. Panda, High Performance Design for HDFS with Byte-Addressability of NVM and RDMA, 24th International Conference on Supercomputing (ICS), June 2016

Evaluation with Hadoop MapReduce



TestDFSIO



OSU Nowlab (4 nodes)

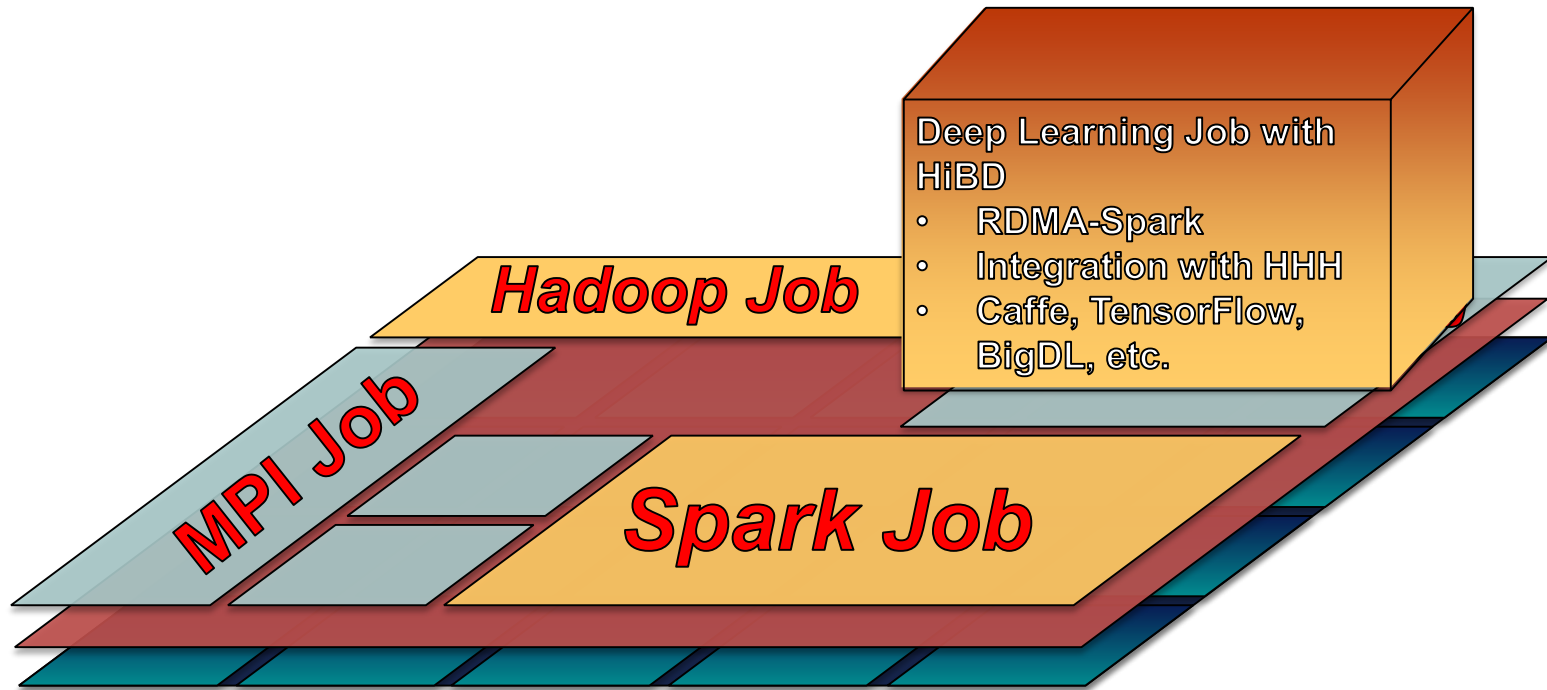
- TestDFSIO on SDSC Comet (32 nodes)
 - Write: NVFS-MemIO gains by **4x** over HDFS
 - Read: NVFS-MemIO gains by **1.2x** over HDFS

- TestDFSIO on OSU Nowlab (4 nodes)
 - Write: NVFS-MemIO gains by **4x** over HDFS
 - Read: NVFS-MemIO gains by **2x** over HDFS

Acceleration Case Studies and Performance Evaluation

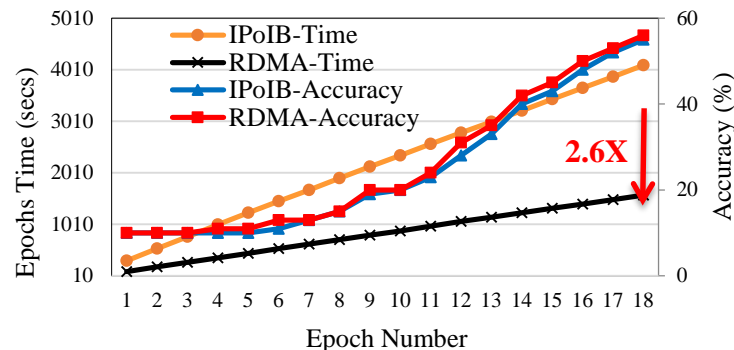
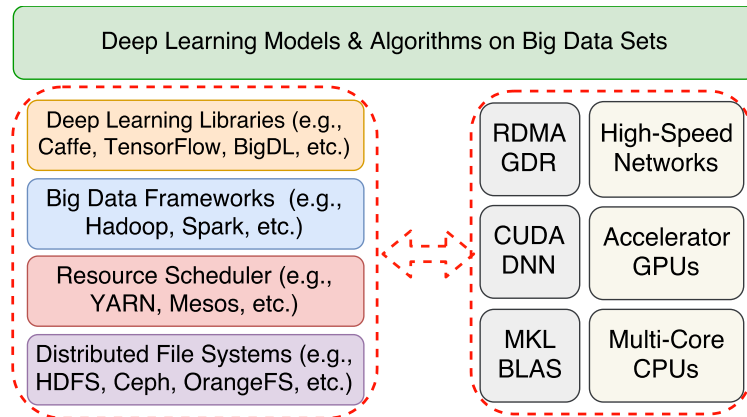
- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- BigData + HPC Cloud

Using HiBD Packages for Deep Learning on Existing HPC Infrastructure



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



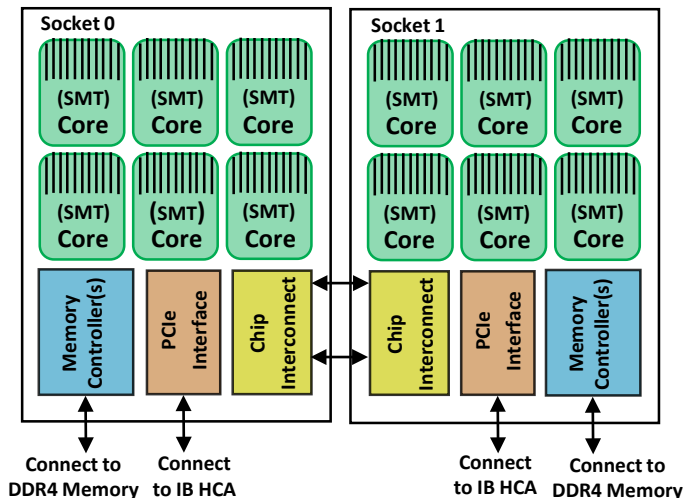
X. Lu, H. Shi, M. H. Javed, R. Biswas, and D. K. Panda, Characterizing Deep Learning over Big Data (DLoBD) Stacks on RDMA-capable Networks, HotI 2017.

Acceleration Case Studies and Performance Evaluation

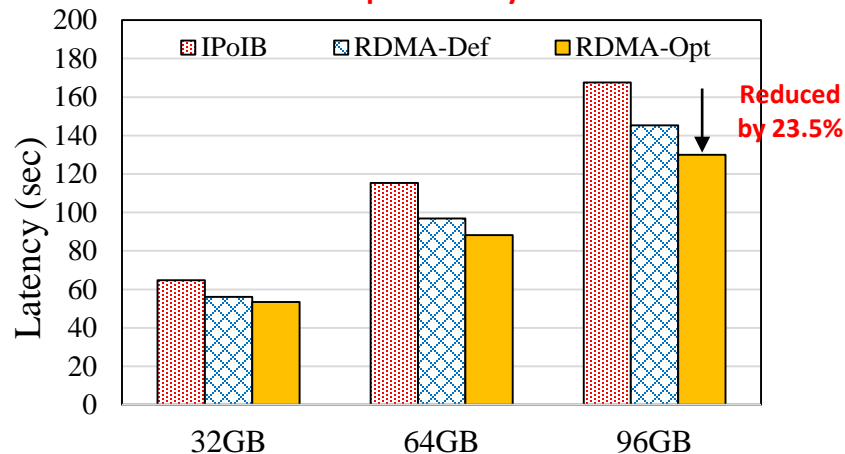
- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- BigData + HPC Cloud

Accelerating Hadoop and Spark with RDMA over OpenPOWER

IBM POWER8 Architecture



Spark SortBy



Challenges

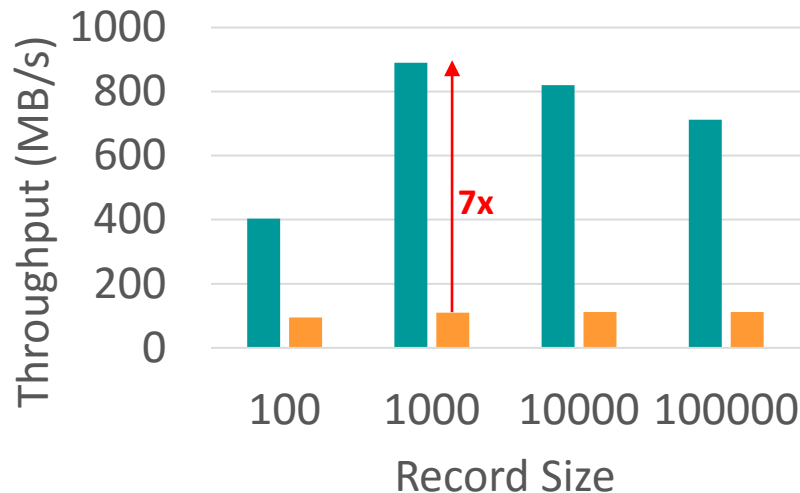
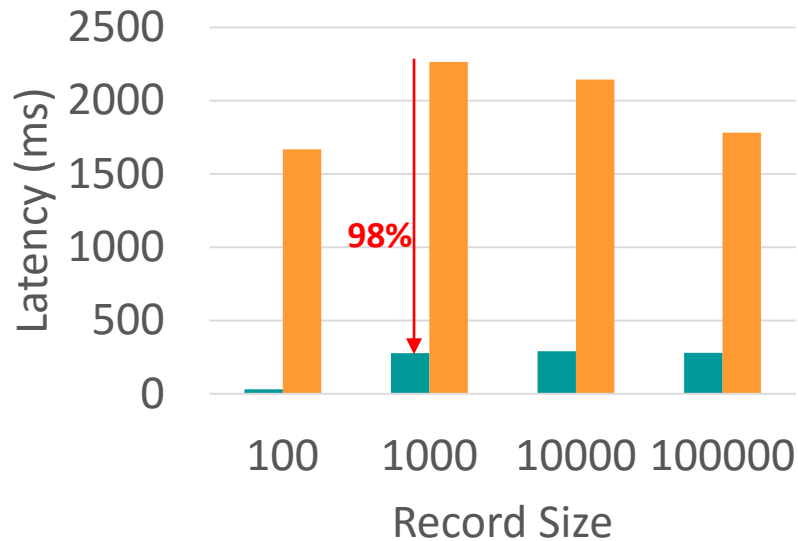
- Performance characteristics of RDMA-based Hadoop and Spark over OpenPOWER systems with IB networks
- Can RDMA-based Big Data middleware demonstrate good performance benefits on OpenPOWER systems?
- Any new accelerations for Big Data middleware on OpenPOWER systems to attain better benefits?

Results

- For the Spark SortBy benchmark, RDMA-Opt outperforms IPoIB and RDMA-Def by 23.5% and 10.5%

X. Lu, H. Shi, D. Shankar and D. K. Panda, Performance Characterization and Acceleration of Big Data Workloads on OpenPOWER System, IEEE BigData 2017.

RDMA-Kafka: High-Performance Message Broker for Streaming Workloads



Kafka Producer Benchmark (1 Broker 4 Producers)

- Experiments run on OSU-RI2 cluster
- 2.4GHz 28 cores, InfiniBand EDR, 512 GB RAM, 400GB SSD
 - Up to 98% improvement in latency compared to IPoIB
 - Up to 7x increase in throughput over IPoIB

Acceleration Case Studies and Performance Evaluation

- Basic Designs
 - HDFS, MapReduce, and RPC
 - Spark
 - Memcached
- Advanced Designs
 - Hadoop with NVRAM
 - Deep Learning Tools (such as Caffe, TensorFlow, BigDL) over RDMA-enabled Hadoop and Spark
 - Big Data Processing over OpenPOWER
 - RDMA Support for Kafka Streaming
- **BigData + HPC Cloud**

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

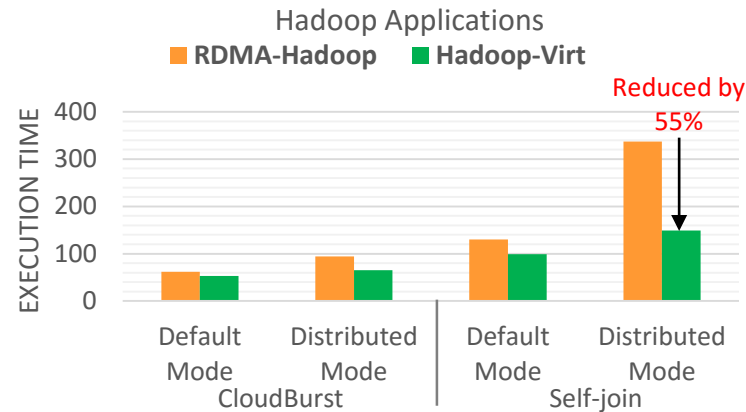
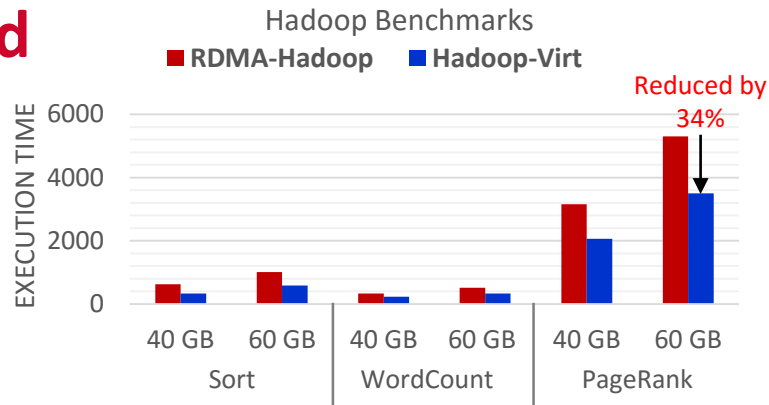
Challenges

- 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

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



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 and Spark
 - OpenPOWER Support
 - Streaming (RDMA-Kafka)
 - MR-Advisor
 - Deep Learning (gRPC and TensorFlow)
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
 - MapReduce, Hadoop RPC, and gRPC
- Advanced designs with upper-level changes and optimizations
 - Boldio (Burst Buffer over Lustre for Big Data I/O Acceleration)
 - 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

One More Presentation

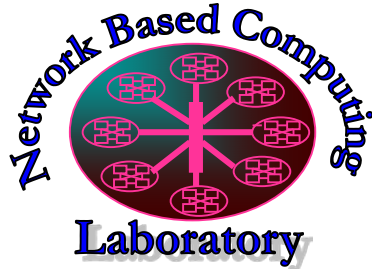
- Thursday (11/16/16) at 10:30am

MVAPICH2-GDR for HPC and Deep Learning

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/>