HiBD
High-Performance Big Data

Sunrise or Sunset: Exploring the Design Space of Big Data Software Stacks

Panel Presentation at HPBDC '17

by

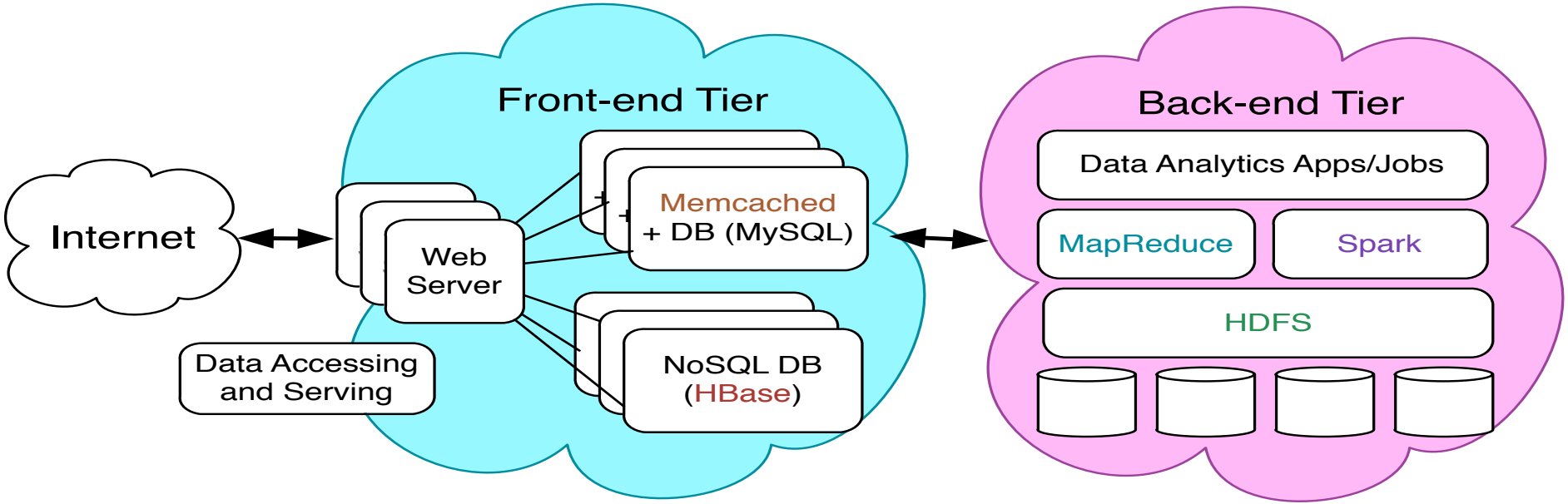
Dhabaleswar K. (DK) Panda
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

Q1: Are Big Data Software Stacks Mature or Not?

- Big Data software stacks like Hadoop, Spark and Memcached have been there for multiple years
  - Hadoop – 11 years (Apache Hadoop 0.1.0 released on April, 2006)
  - Spark – 5 years (Apache Spark 0.5.1 released on June, 2012)
  - Memcached – 14 years (Initial release of Memcached on May 22, 2003)
- Increasingly being used in production environments
- Optimized for commodity clusters with Ethernet and TCP/IP interface
- Not yet able to take full advantage of modern cluster and/or HPC technologies

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

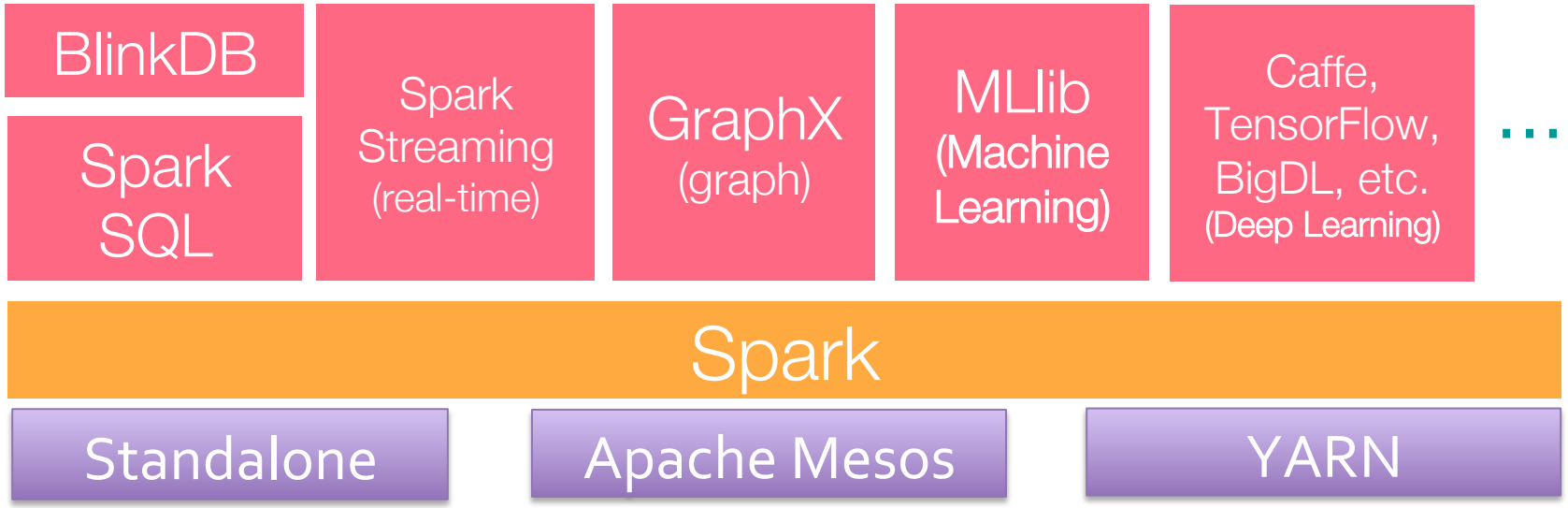


Who Are Using Hadoop?

- Focuses on large data and data analysis
- Hadoop (e.g. HDFS, MapReduce, RPC, HBase) environment is gaining a lot of momentum
- http://wiki.apache.org/hadoop/PoweredBy

Spark Ecosystem

- Generalize MapReduce to support new apps in same engine
- Two Key Observations
  - General task support with DAG
  - Multi-stage and interactive apps require faster **data sharing** across parallel jobs

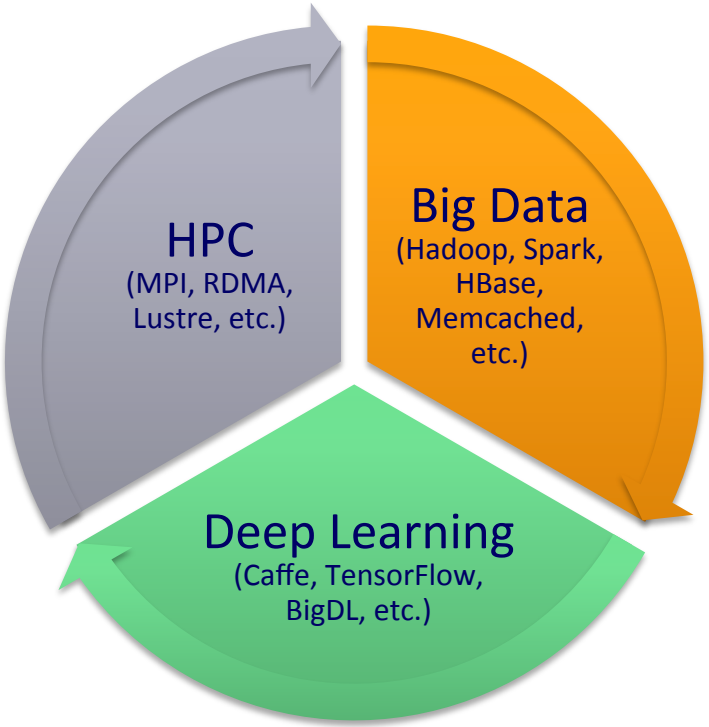


Who Are Using Spark?

- Focuses on large data and data analysis with in-memory techniques
- Apache Spark is gaining a lot of momentum
- http://spark.apache.org/powered-by.html

Q2: What are the Main Driving forces for New-generation Big Data Software Stacks?

Increasing Usage of HPC, Big Data and Deep Learning



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

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

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

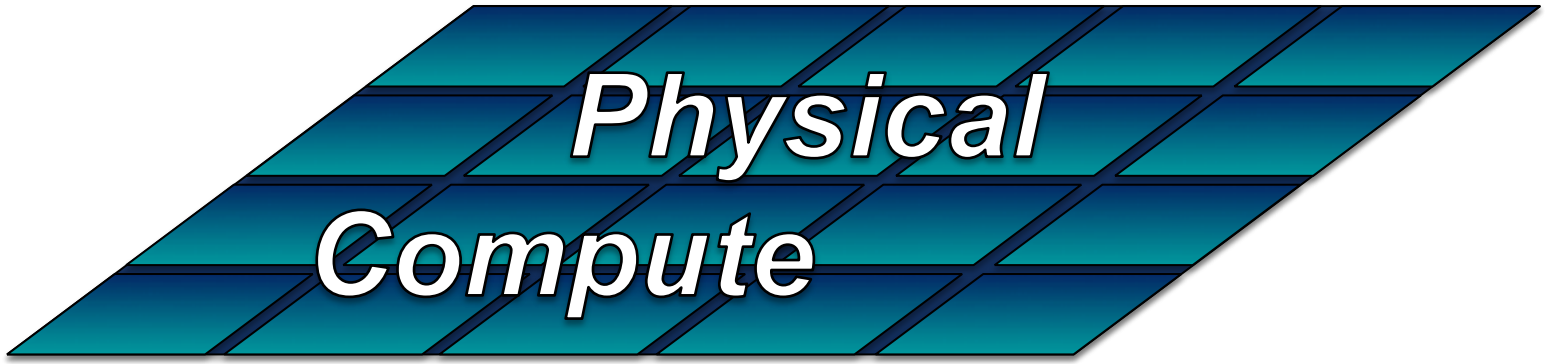
How much performance benefits can be achieved through enhanced designs?

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

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

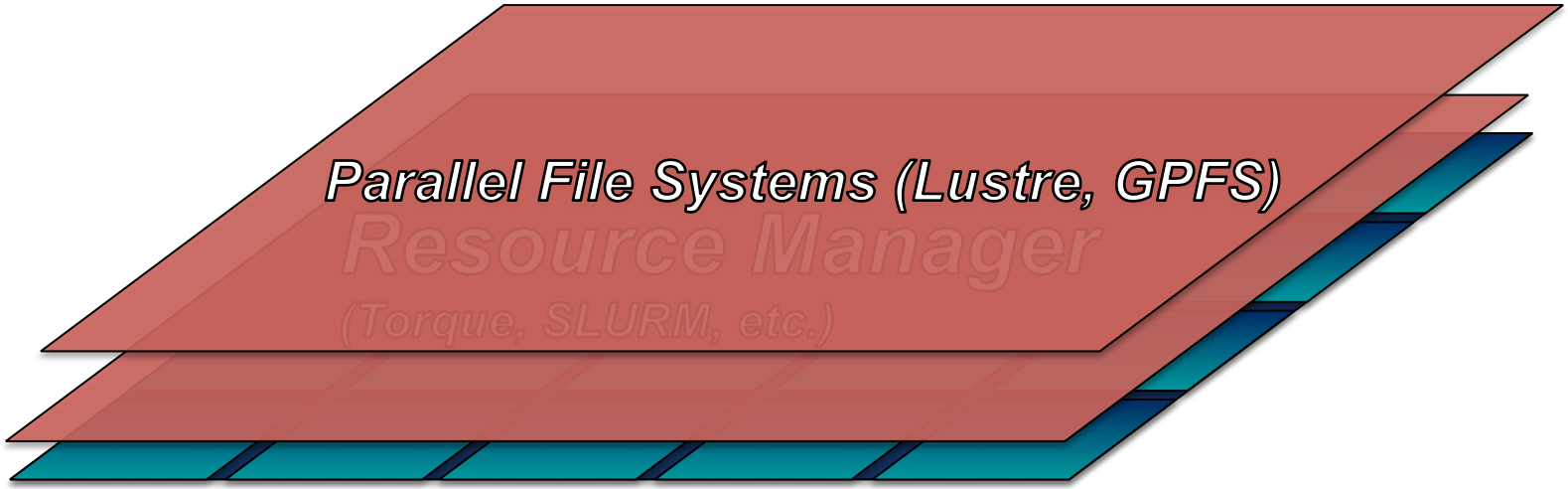
Bring HPC, Big Data processing, and Deep Learning into a "convergent trajectory"!

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?

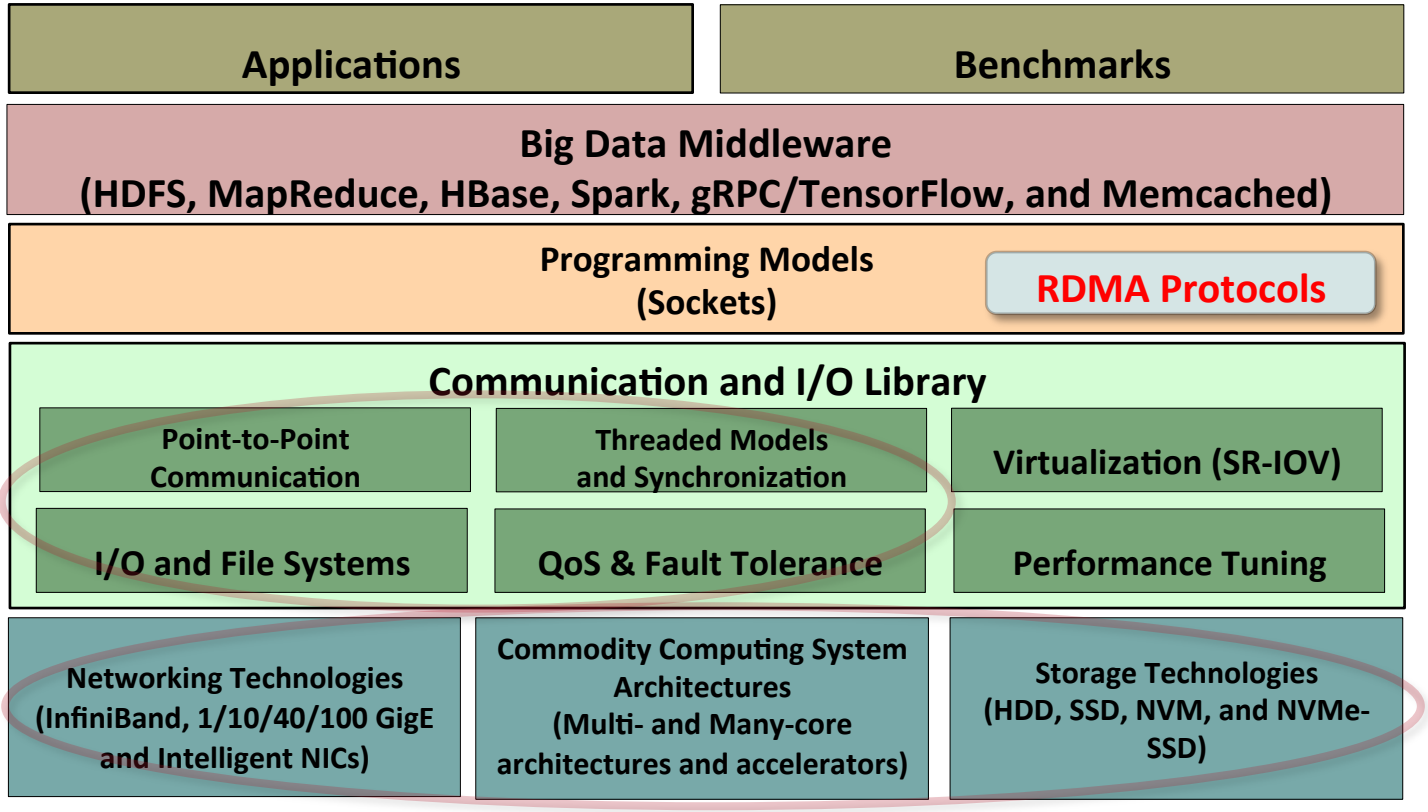
Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



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

Q3: What Chances are Provided for the Academia Communities in Exploring the Design Spaces of Big Data Software Stacks?

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

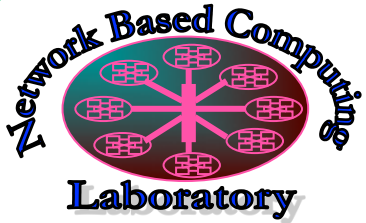


The High-Performance Big Data (HiBD) Project

- 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: 230 organizations from 30 countries
- More than 21,800 downloads from the project site

Available for InfiniBand and RoCE

Also run on Ethernet

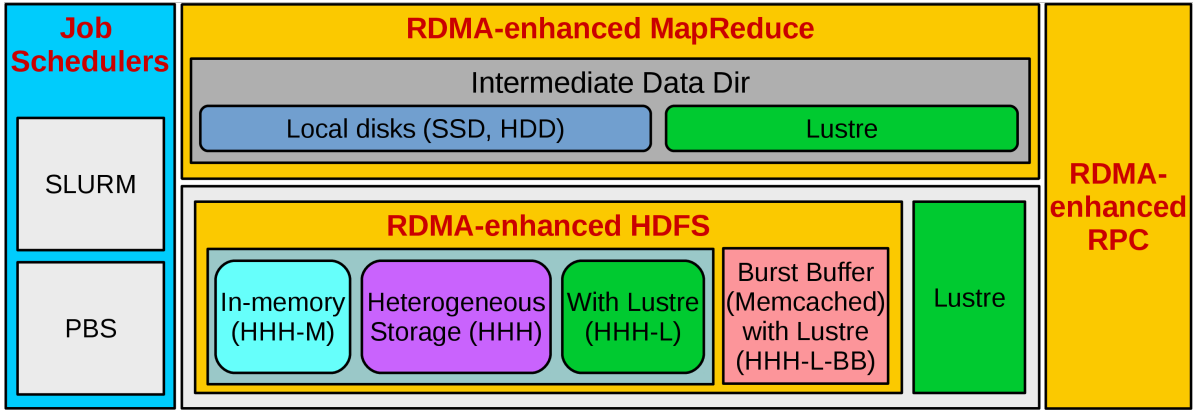


RDMA for Apache Hadoop 2.x Distribution

- 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

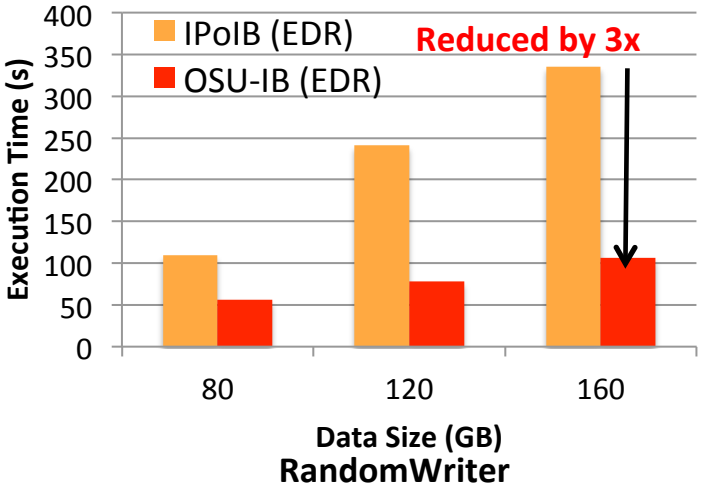
- High-Performance Design of Spark over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for Spark
  - RDMA-based data shuffle and SEDA-based shuffle architecture
  - Support pre-connection, on-demand connection, and connection 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**

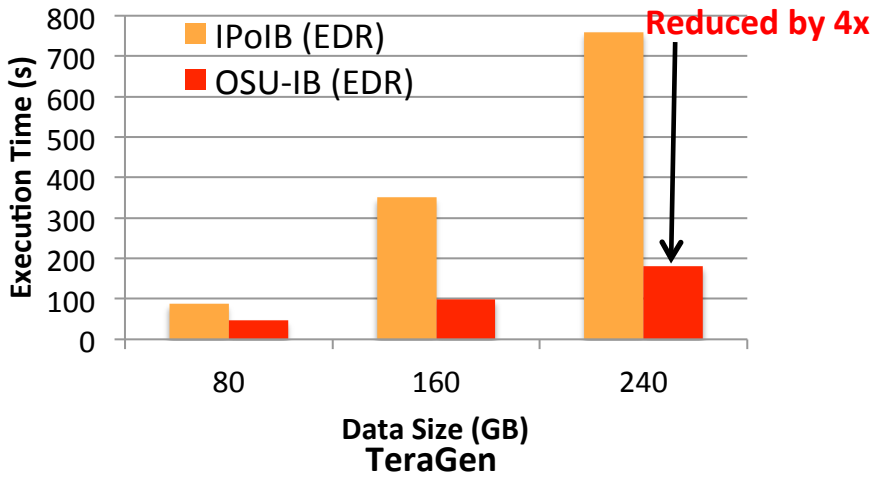
HiBD Packages on SDSC Comet and Chameleon Cloud

- 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 (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

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

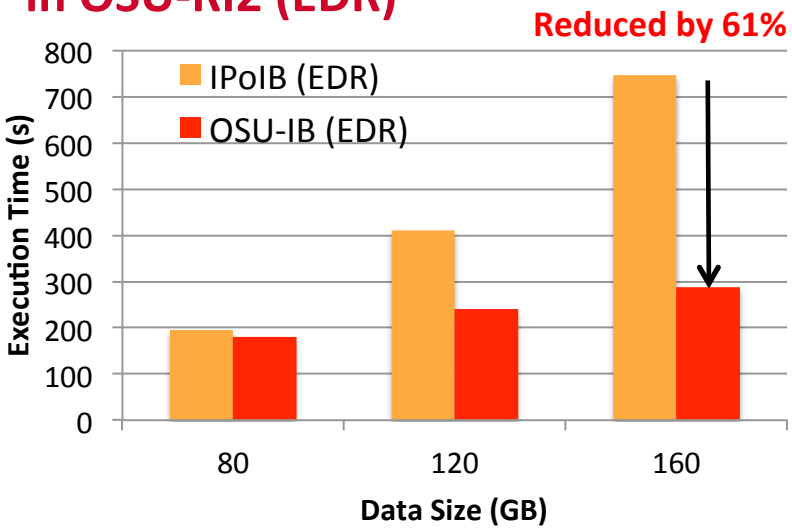




Cluster with 8 Nodes with a total of 64 maps

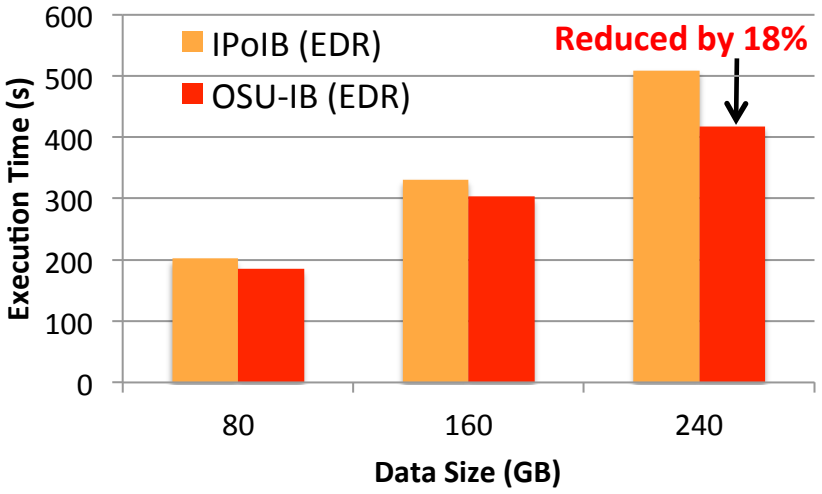
- RandomWriter
  - **3x** improvement over IPoIB for 80-160 GB file size
- TeraGen
  - **4x** improvement over IPoIB for 80-240 GB file size

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



Sort

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

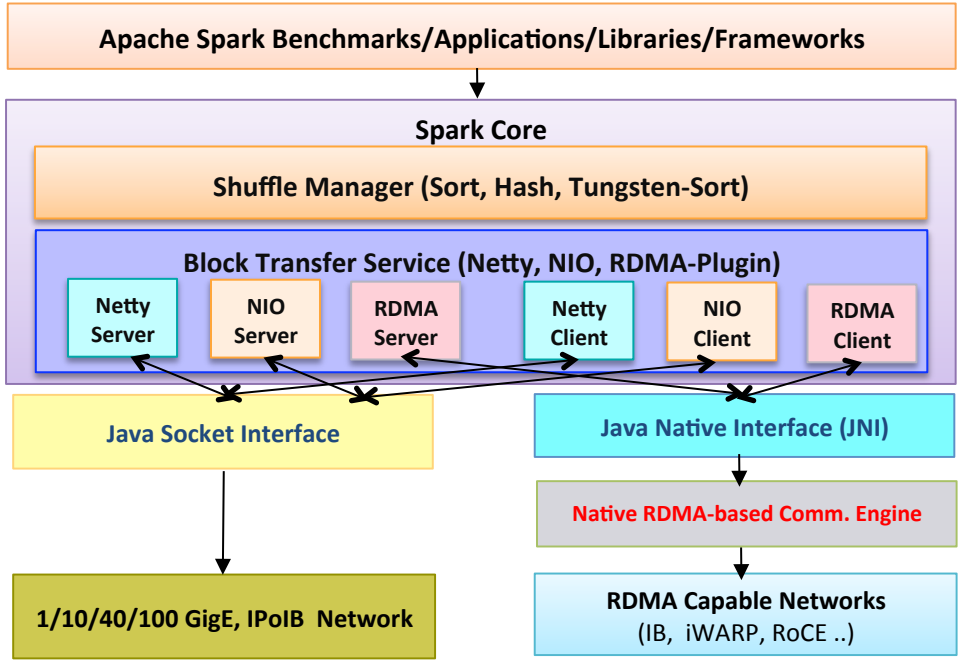


TeraSort

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



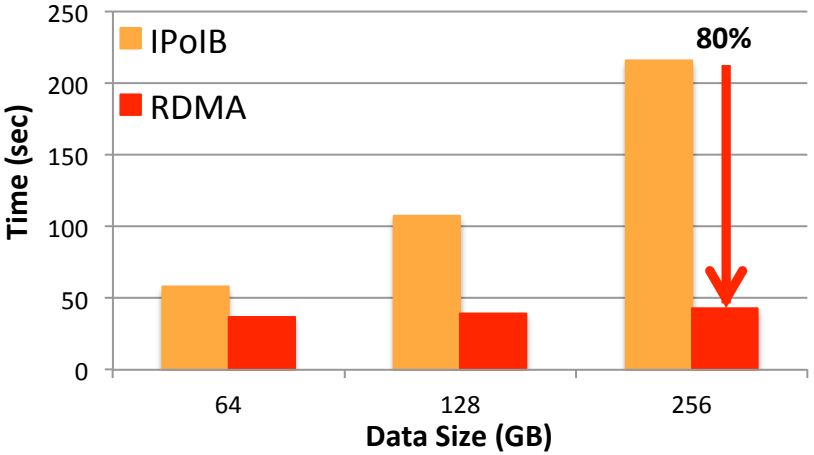
- Design Features
  - RDMA based shuffle plugin
  - SEDA-based architecture
  - Dynamic connection management and sharing
  - Non-blocking data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

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

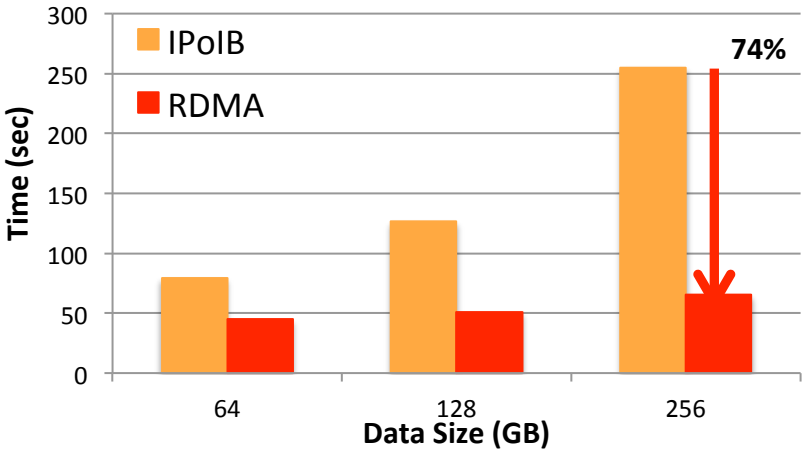
X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

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



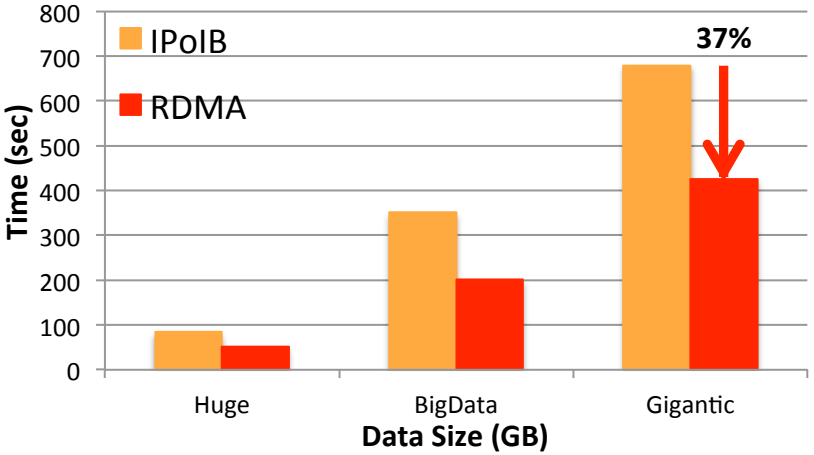
64 Worker Nodes, 1536 cores, SortByTest Total Time



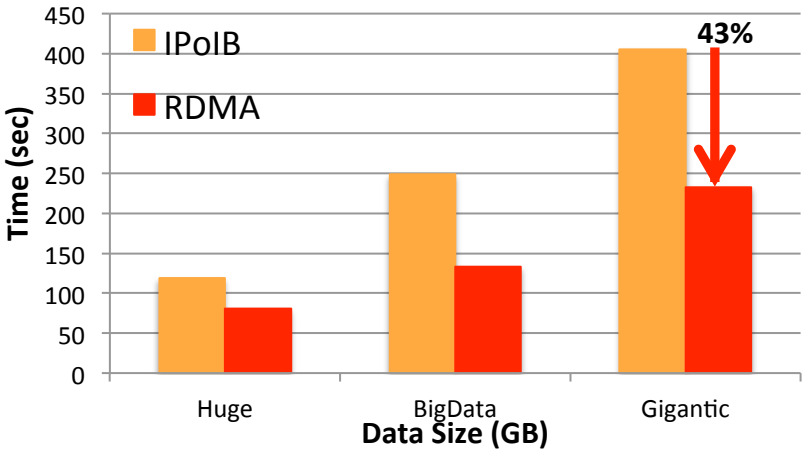
64 Worker Nodes, 1536 cores, GroupByTest Total Time

- 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



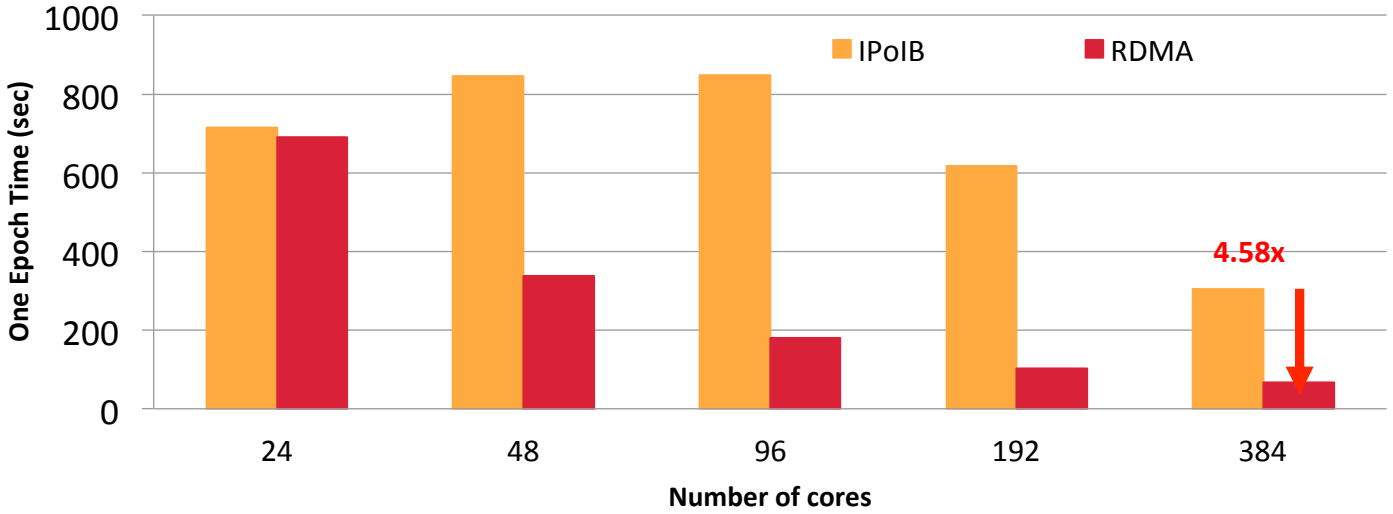
32 Worker Nodes, 768 cores, PageRank Total Time



64 Worker Nodes, 1536 cores, PageRank Total Time

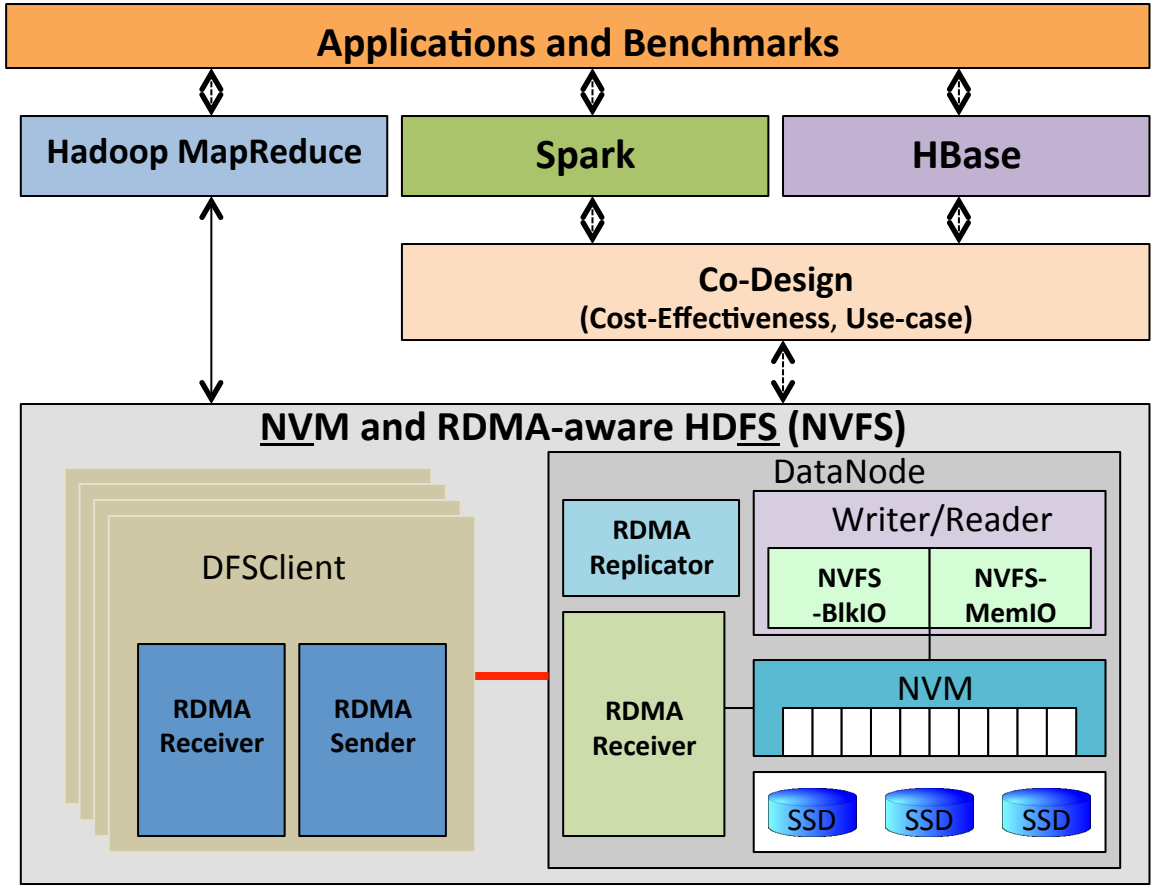
- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

Evaluation with BigDL on RDMA-Spark



- VGG training model on the CIFAR-10 dataset
- Evaluated on SDSC Comet supercomputer
- Initial Results: RDMA-based Spark outperforms default Spark over IPoIB by a factor of 4.58x

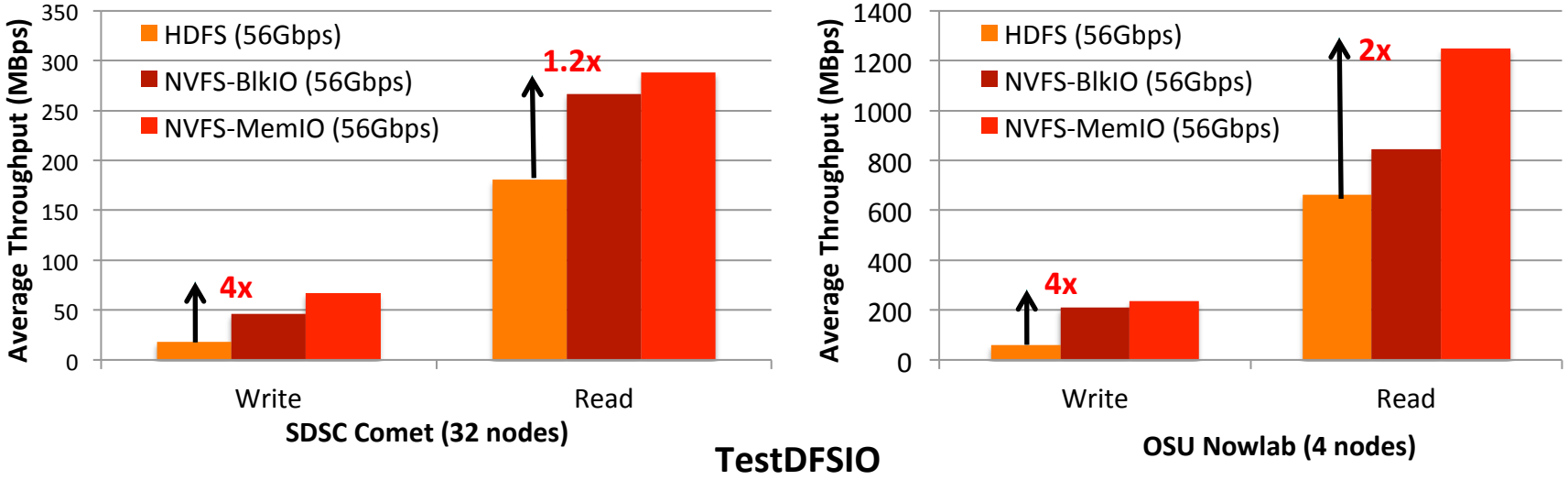
Design Overview of NVM and RDMA-aware HDFS (NVFS)



- **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

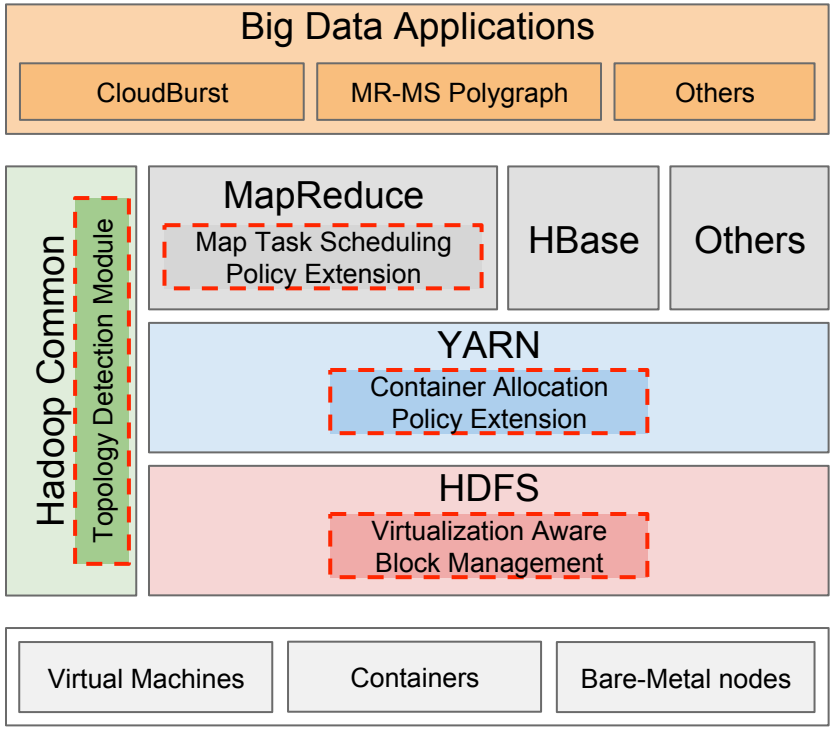
N. S. Islam, M. W. Rahman , X. Lu, and D. K. Panda, High Performance Design for HDFS with Byte-Addressability of NVM and RDMA, 24th International Conference on Supercomputing (ICS), June 2016

Evaluation with Hadoop MapReduce



- TestDFSIO 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

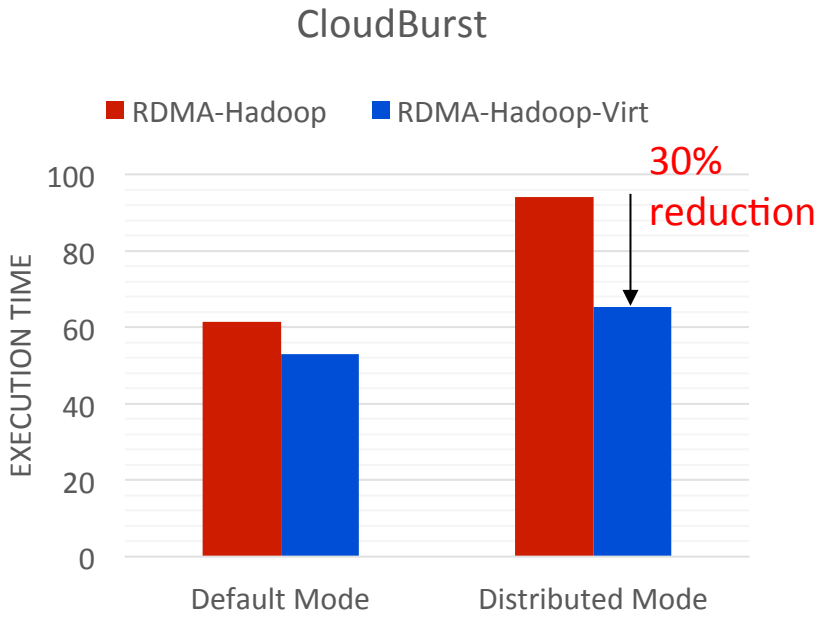
Overview of RDMA-Hadoop-Virt Architecture

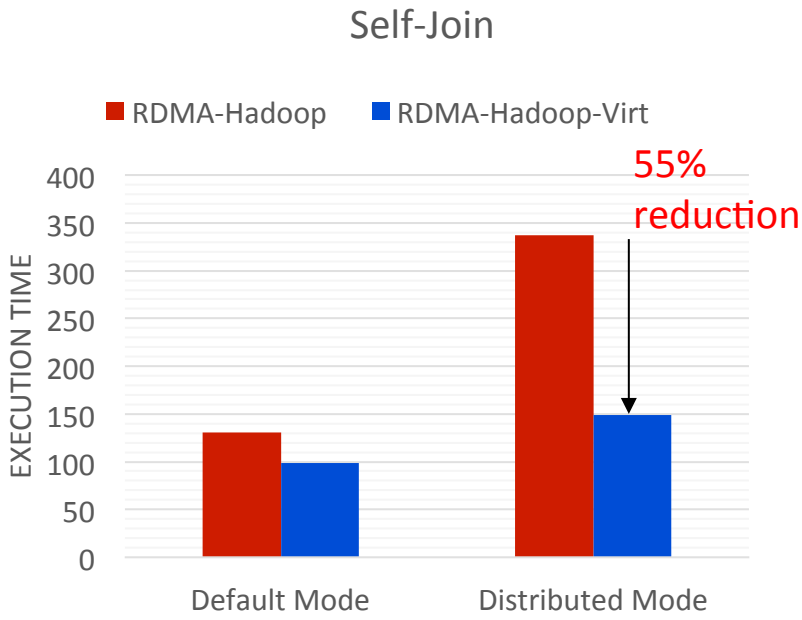


- Virtualization-aware modules in all the four main Hadoop components:
  - **HDFS**: Virtualization-aware Block Management to improve fault-tolerance
  - **YARN**: Extensions to Container Allocation Policy to reduce network traffic
  - **MapReduce**: Extensions to Map Task Scheduling Policy to reduce network traffic
  - **Hadoop Common**: Topology Detection Module for automatic topology detection
- Communications in HDFS, MapReduce, and RPC go through RDMA-based designs over SR-IOV enabled InfiniBand

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

Evaluation with Applications

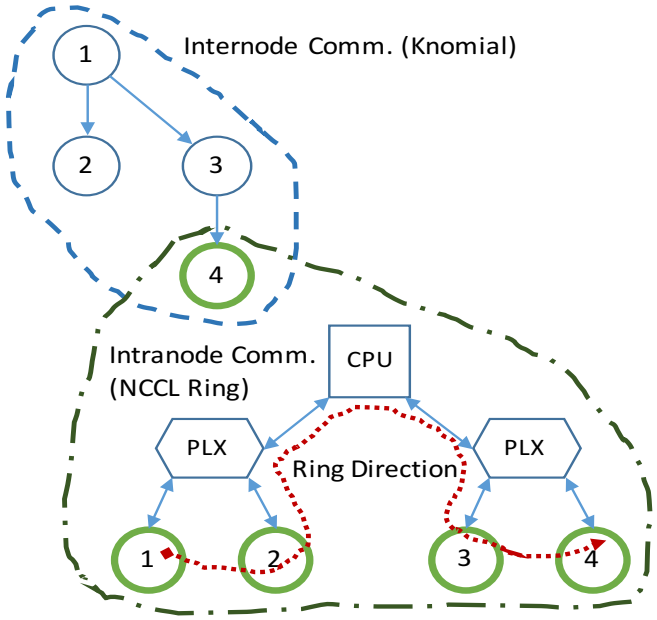




- 14% and 24% improvement with Default Mode for CloudBurst and Self-Join
- 30% and 55% improvement with Distributed Mode for CloudBurst and Self-Join

Deep Learning: New Challenges for MPI Runtimes

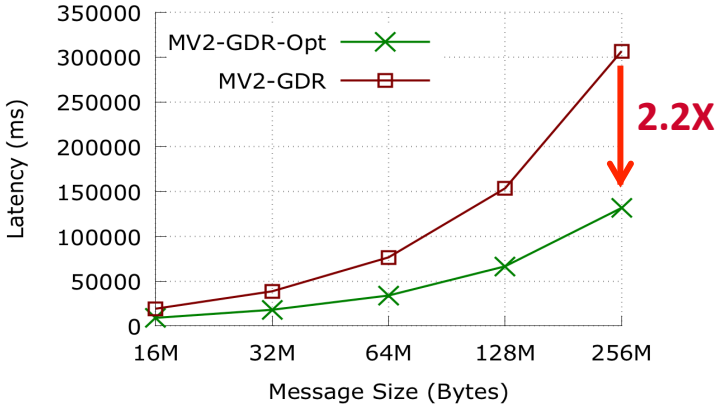
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- How to address these newer requirements?
  - GPU-specific Communication Libraries (NCCL)
    - NVidia's NCCL library provides inter-GPU communication
  - CUDA-Aware MPI (MVAPICH2-GDR)
    - Provides support for GPU-based communication
- Can we exploit CUDA-Aware MPI and NCCL to support Deep Learning applications?



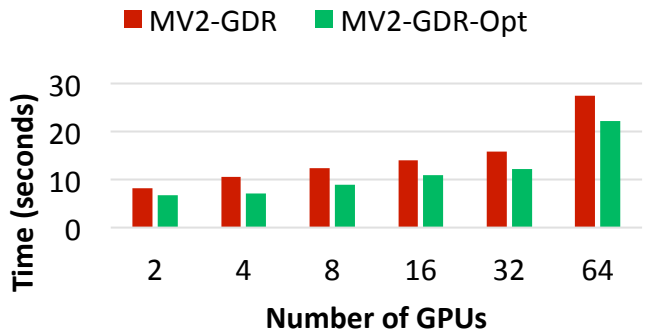
Hierarchical Communication (Knomial + NCCL ring)

Efficient Broadcast: MVAPICH2-GDR and NCCL

- NCCL has some limitations
  - Only works for a single node, thus, no scale-out on multiple nodes
  - Degradation across IOH (socket) for scale-up (within a node)
- We propose optimized MPI_Bcast
  - Communication of very large GPU buffers (order of megabytes)
  - Scale-out on large number of dense multi-GPU nodes
- Hierarchical Communication that efficiently exploits:
  - CUDA-Aware MPI_Bcast in MV2-GDR
  - NCCL Broadcast primitive



Performance Benefits: OSU Micro-benchmarks

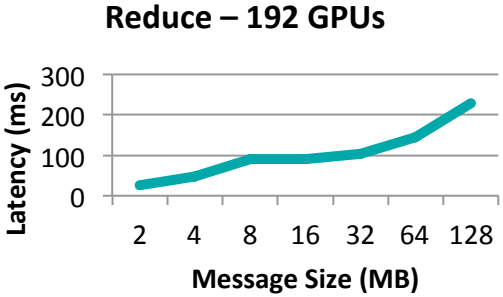


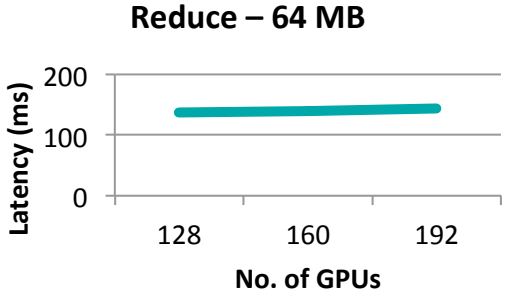
Performance Benefits: Microsoft CNTK DL framework (25% avg. improvement)

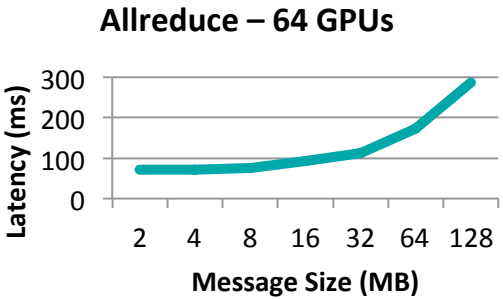
Efficient Large Message Broadcast using NCCL and CUDA-Aware MPI for Deep Learning, A. Awan , K. Hamidouche , A. Venkatesh , and D. K. Panda, The 23rd European MPI Users' Group Meeting (EuroMPI 16), Sep 2016 [Best Paper Runner-Up]

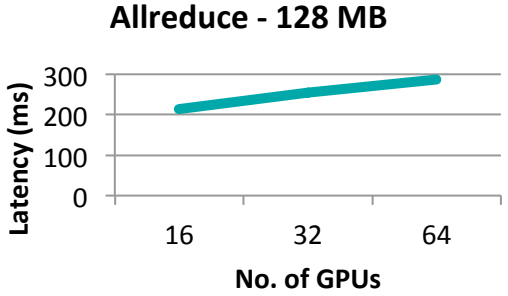
Large Message Optimized Collectives for Deep Learning

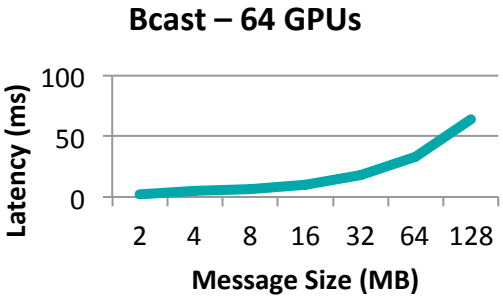
- MV2-GDR provides optimized collectives for **large message sizes**
- Optimized Reduce, Allreduce, and Bcast
- **Good scaling with large number of GPUs**
- **Available with MVAPICH2-GDR 2.2GA**

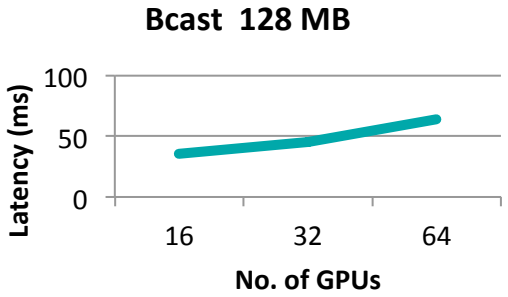










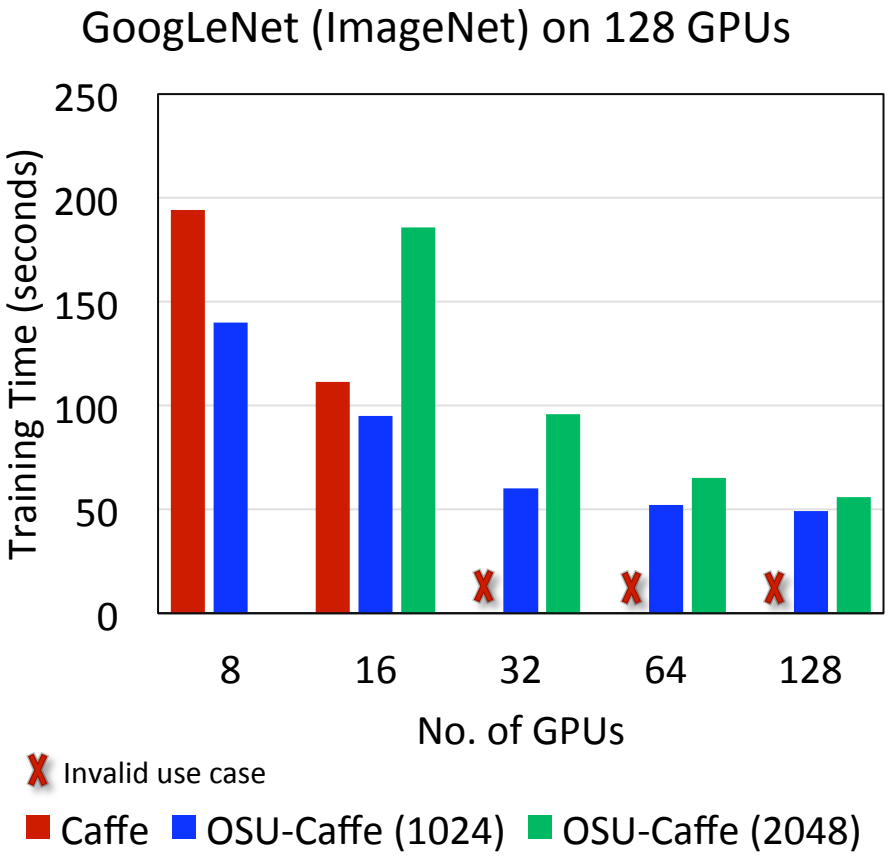


OSU-Caffe: Scalable Deep Learning

- Caffe : A flexible and layered Deep Learning framework.
- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - Limited Scale-out
- OSU-Caffe: MPI-based Parallel Training
  - Enable Scale-up (within a node) and Scale-out (across multi-GPU nodes)
  - network on ImageNet dataset

A. A. Awan, K. Hamidouche, J. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters, PPoPP, Sep 2017

OSU-Caffe is publicly available from:
http://hidl.cse.ohio-state.edu



Open Challenges in Designing Communication and I/O Middleware for High-Performance Big Data Processing

- High-Performance designs for Big Data middleware
  - NVM-aware communication and I/O schemes for Big Data
  - SATA-/PCIe-/NVMe-SSD support
  - High-Bandwidth Memory support
  - Threaded Models and Synchronization
  - Locality-aware designs
- Fault-tolerance/resiliency
  - Migration support with virtual machines
  - Data replication
- Efficient data access and placement policies
- Efficient task scheduling
- Fast deployment and automatic configurations on Clouds
- Optimization for Deep Learning applications

Sunrise or Sunset of Big Data Software?

Assuming 6:00 am as sunrise and

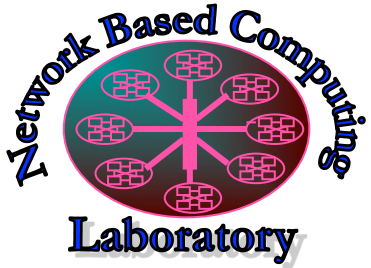
6:00 pm as sunset,

We are at 8:00 am.

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/