HiBD
High-Performance Big Data

Exploiting HPC Technologies to Accelerate Big Data Processing

Talk at Open Fabrics Workshop (April 2016)

by

Dhabaleswar K. (DK) Panda

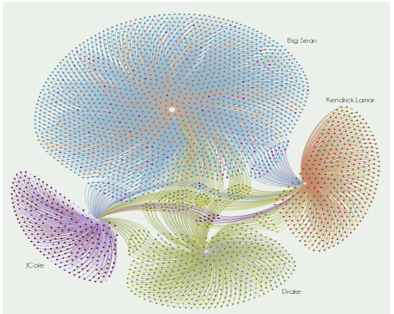
The Ohio State University

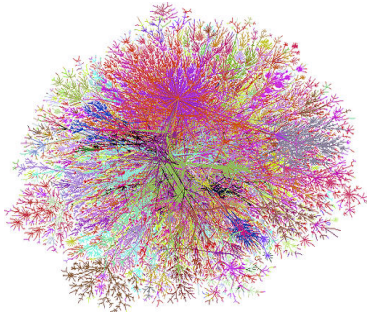
E-mail: panda@cse.ohio-state.edu

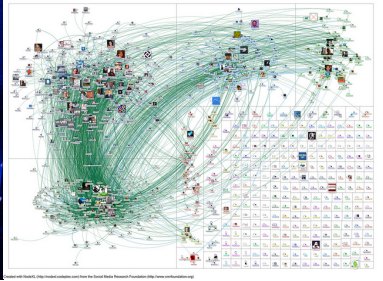
http://www.cse.ohio-state.edu/~panda

Introduction to Big Data 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

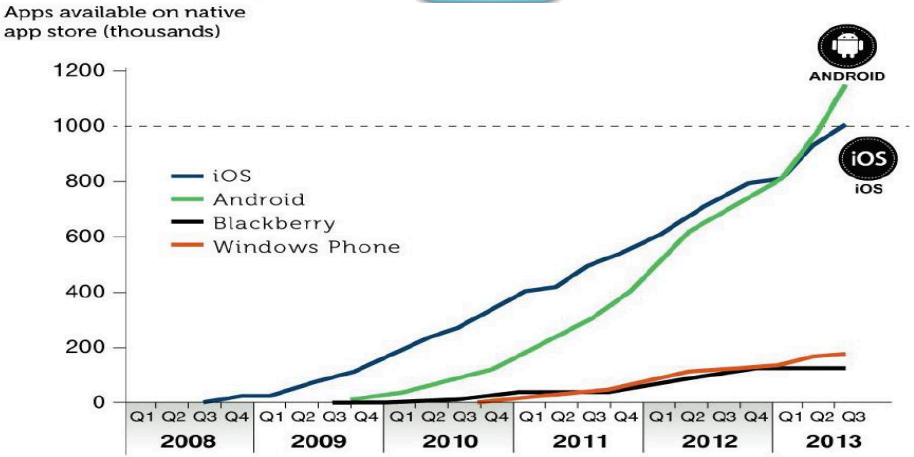






Data Generation in Internet Services and Applications

- Webpages (content, graph)
- Clicks (ad, page, social)
- Users (OpenID, FB Connect, etc.)
- e-mails (Hotmail, Y!Mail, Gmail, etc.)
- Photos, Movies (Flickr, YouTube, Video, etc.)
- Cookies / tracking info (see Ghostery)
- Installed apps (Android market, App Store, etc.)
- Location (Latitude, Loopt, Foursquared, Google Now, etc.)
- User generated content (Wikipedia & co, etc.)
- Ads (display, text, DoubleClick, Yahoo, etc.)
- Comments (Discuss, Facebook, etc.)
- Reviews (Yelp, Y!Local, etc.)
- Social connections (LinkedIn, Facebook, etc.)
- Purchase decisions (Netflix, Amazon, etc.)
- Instant Messages (YIM, Skype, Gtalk, etc.)
- Search terms (Google, Bing, etc.)
- News articles (BBC, NYTimes, Y!News, etc.)
- Blog posts (Tumblr, Wordpress, etc.)
- Microblogs (Twitter, Jaiku, Meme, etc.)
- Link sharing (Facebook, Delicious, Buzz, etc.)



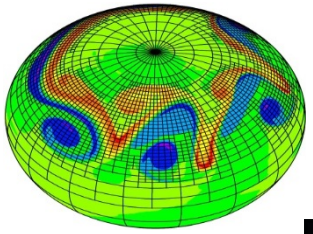
Number of Apps in the Apple App Store, Android Market, Blackberry, and Windows Phone (2013)

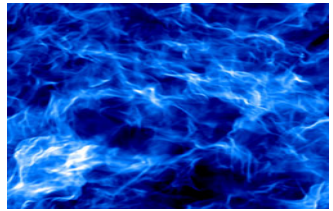
- Android Market: <1200K
- Apple App Store: ~1000K

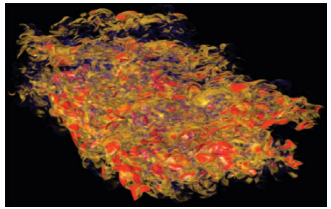
Courtesy: http://dazeinfo.com/2014/07/10/apple-inc-aapl-ios-google-inc-goog-android-growth-mobile-ecosystem-2014/

Not Only in Internet Services - Big Data in Scientific Domains

- Scientific Data Management, Analysis, and Visualization
- Applications examples
  - Climate modeling
  - Combustion
  - Fusion
  - Astrophysics
  - Bioinformatics
- Data Intensive Tasks
  - Runs large-scale simulations on supercomputers
  - Dump data on parallel storage systems
  - Collect experimental / observational data
  - Move experimental / observational data to analysis sites
  - Visual analytics – help understand data visually





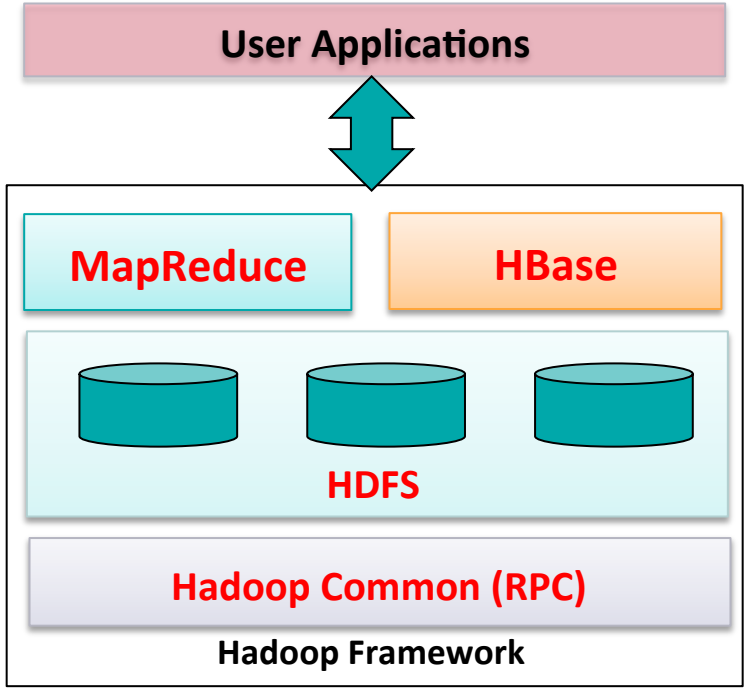


Typical Solutions or Architectures for Big Data Analytics

- Hadoop: http://hadoop.apache.org
  - The most popular framework for Big Data Analytics
  - HDFS, MapReduce, HBase, RPC, Hive, Pig, ZooKeeper, Mahout, etc.
- Spark: http://spark-project.org
  - Provides primitives for in-memory cluster computing; Jobs can load data into memory and query it repeatedly
- Storm: http://storm-project.net
  - A distributed real-time computation system for real-time analytics, online machine learning, continuous computation, etc.
- S4: http://incubator.apache.org/s4
  - A distributed system for processing continuous unbounded streams of data
- GraphLab: http://graphlab.org
  - Consists of a core C++ GraphLab API and a collection of high-performance machine learning and data mining toolkits built on top of the GraphLab API.
- Web 2.0: RDBMS + Memcached (http://memcached.org)
  - Memcached: A high-performance, distributed memory object caching systems

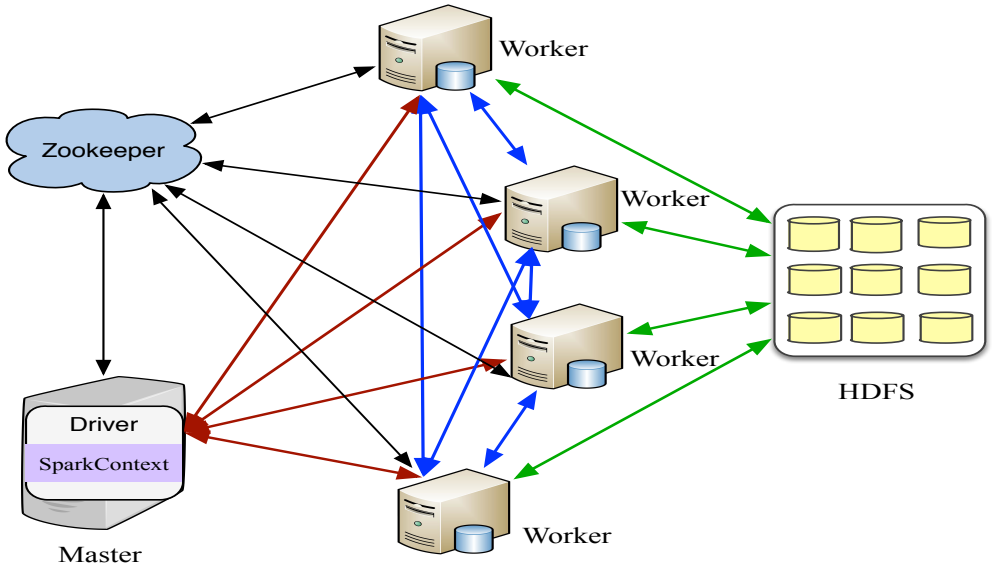
Big Data Processing with Hadoop Components

- Major components included in this tutorial:
  - MapReduce (Batch)
  - HBase (Query)
  - HDFS (Storage)
  - RPC (Inter-process communication)
- Underlying Hadoop Distributed File System (HDFS) used by both MapReduce and HBase
- Model scales but high amount of communication during intermediate phases can be further optimized



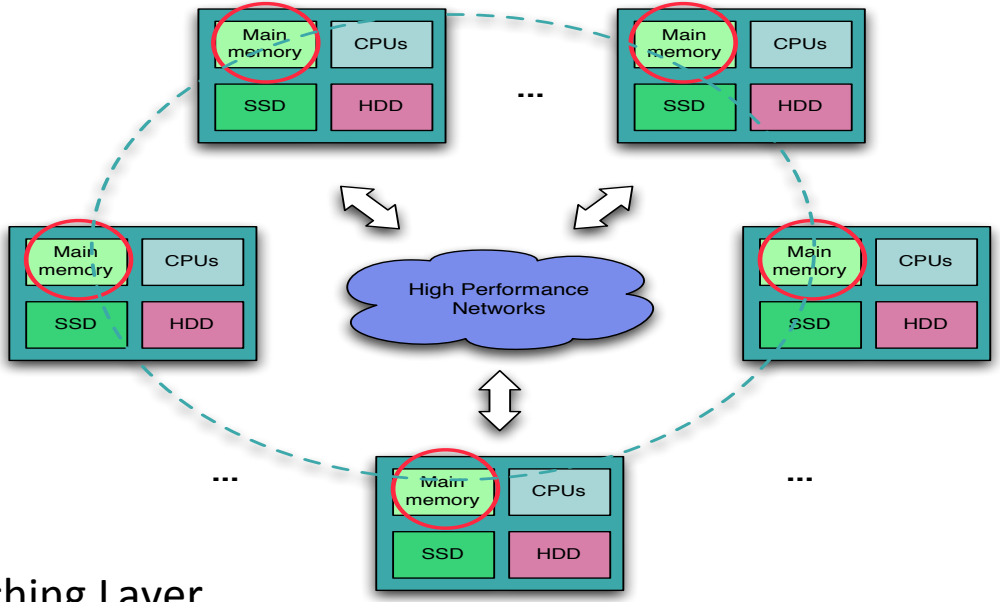
Spark Architecture Overview

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



http://spark.apache.org

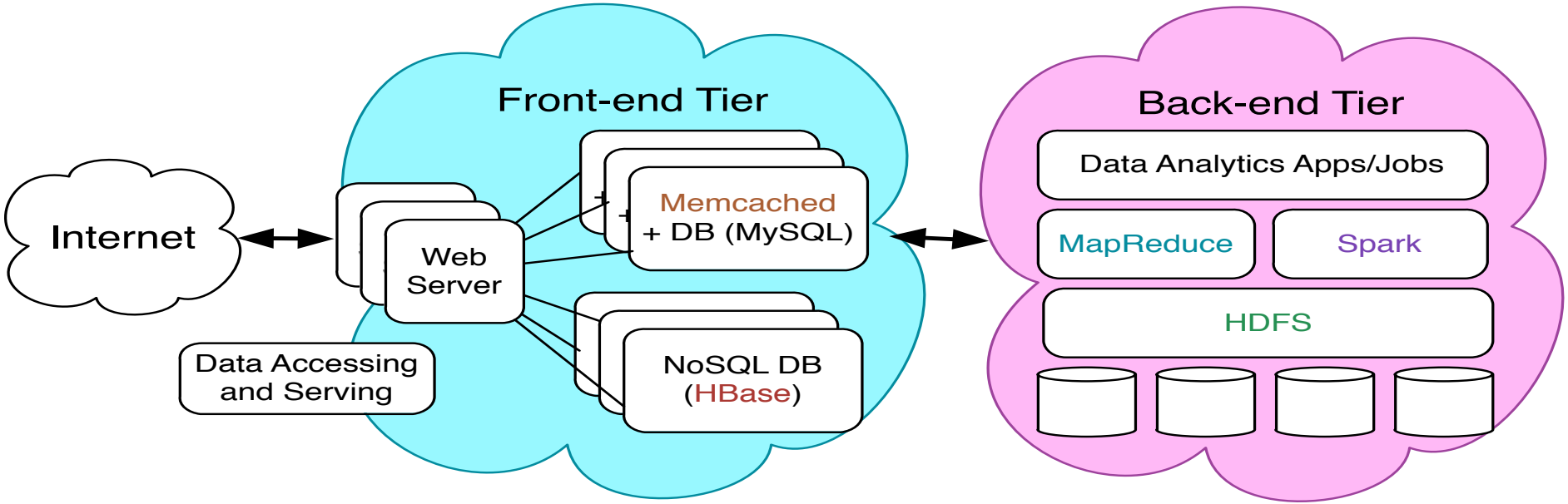
Memcached Architecture



- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

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



Trends in HPC Technologies

- Advanced Interconnects and RDMA protocols
  - InfiniBand
  - 10-40 Gigabit Ethernet/iWARP
  - RDMA over Converged Enhanced Ethernet (RoCE)
- Delivering excellent performance (Latency, Bandwidth and CPU Utilization)
- Has influenced re-designs of enhanced HPC middleware
  - Message Passing Interface (MPI) and PGAS
  - Parallel File Systems (Lustre, GPFS, ..)
- SSDs (SATA and NVMe)
- NVRAM and Burst Buffer

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?

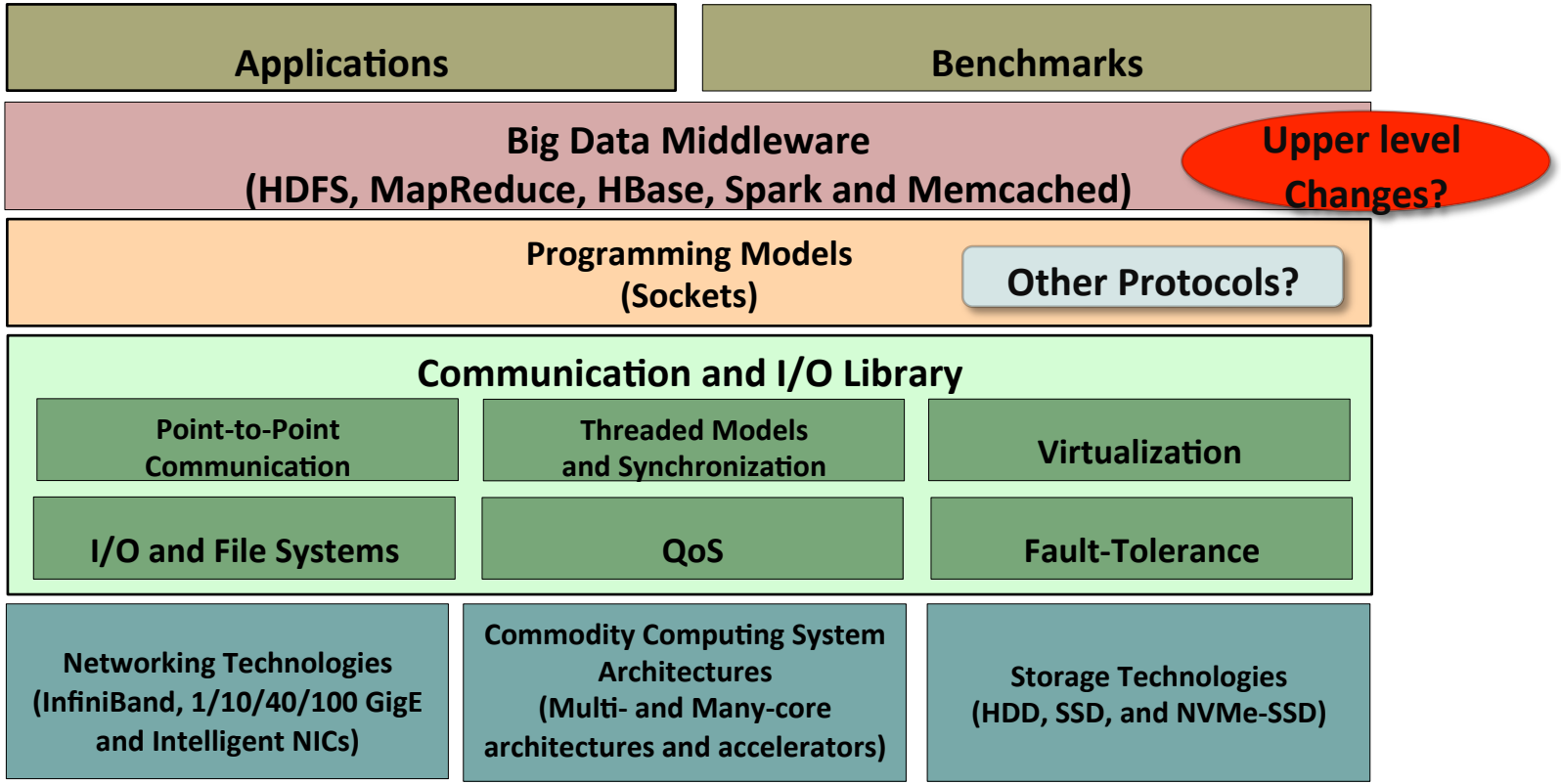
How much performance benefits can be achieved through enhanced designs?

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 Hadoop 1.x (RDMA-Hadoop)
- RDMA for Memcached (RDMA-Memcached)
- OSU HiBD-Benchmarks (OHB)
  - HDFS and Memcached Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 166 organizations from 22 countries
- More than 15,900 downloads from the project site
- RDMA for Apache HBase and Impala (upcoming)

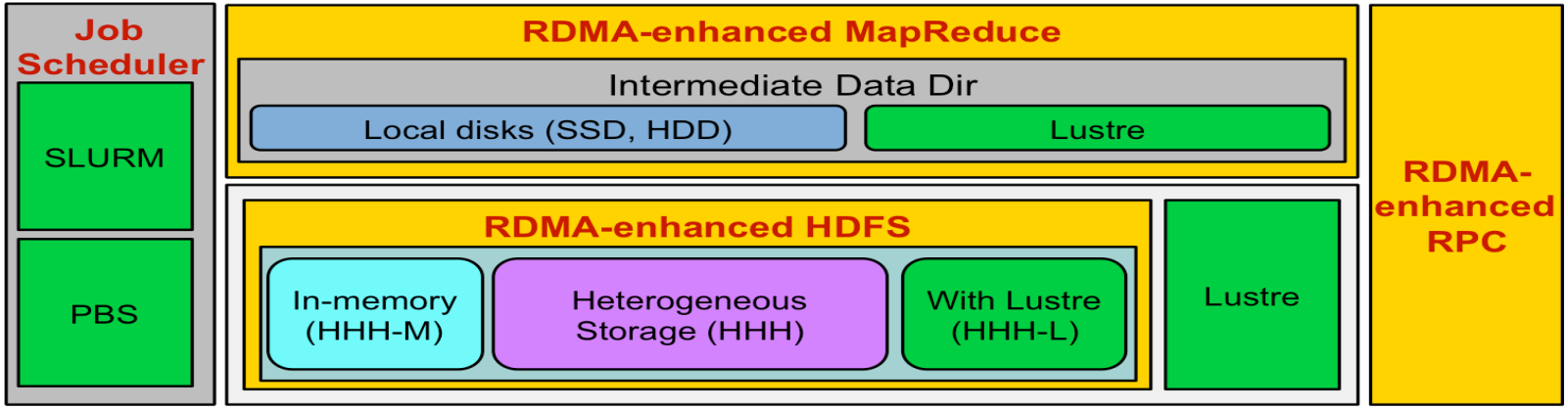
Available for InfiniBand and RoCE

HiBD
High-Performance Big Data

Network Based Computing Laboratory

The Ohio State University

Different Modes of RDMA for Apache Hadoop 2.x

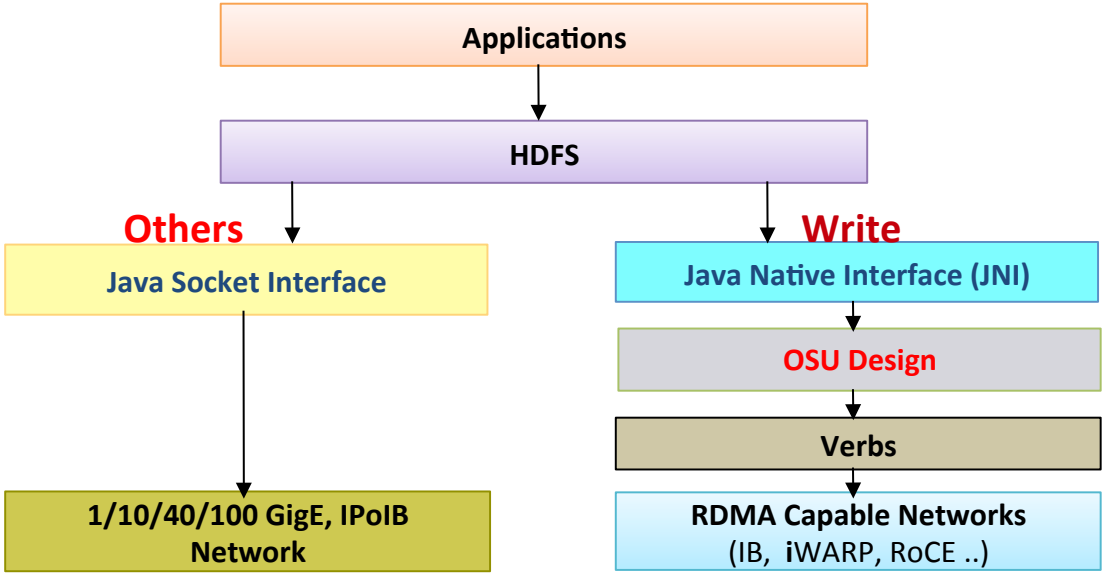


- **HHH**: Heterogeneous storage devices with hybrid replication schemes are supported in this mode of operation to have better fault-tolerance as well as performance. This mode is enabled by **default** in the package.
- **HHH-M**: A high-performance in-memory based setup has been introduced in this 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.
- **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).

Acceleration Case Studies and Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - RPC
  - HBase
  - Spark
  - Memcached (Basic and Hybrid)
  - HDFS + Memcached-based Burst Buffer

Design Overview of HDFS with RDMA



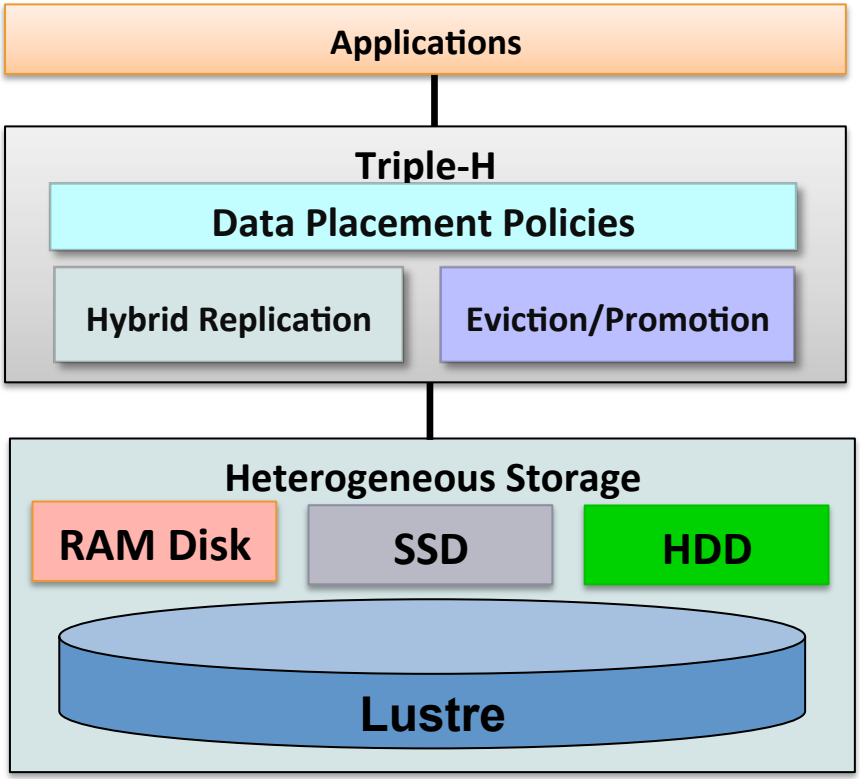
- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - Parallel replication support
  - On-demand connection setup
  - InfiniBand/RoCE support

- 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

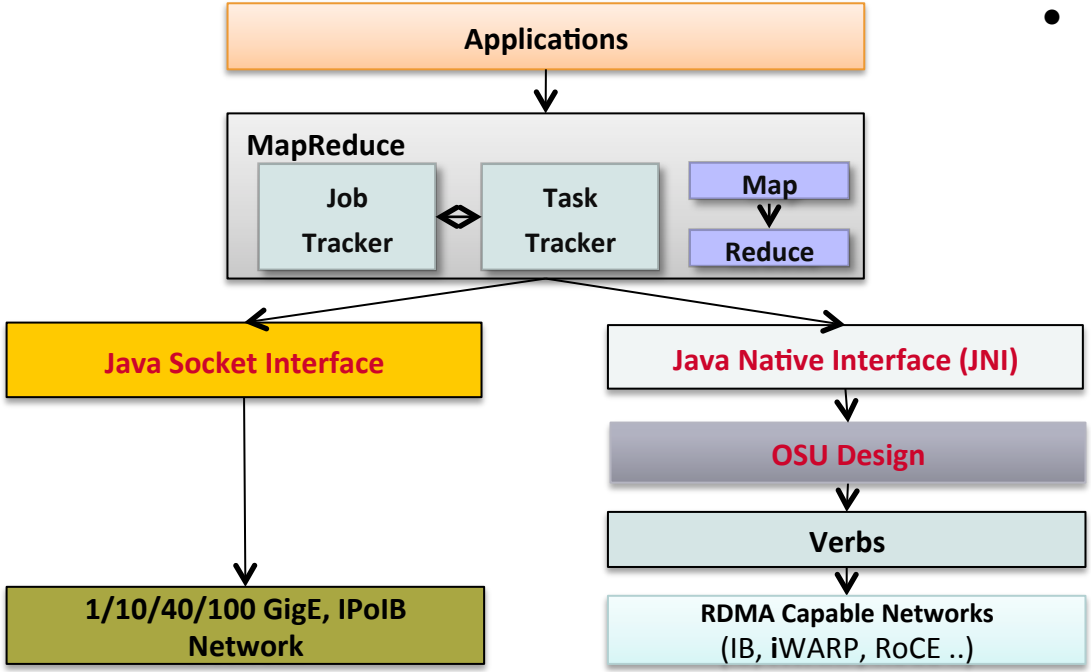
Enhanced HDFS with In-Memory and Heterogeneous Storage



- 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

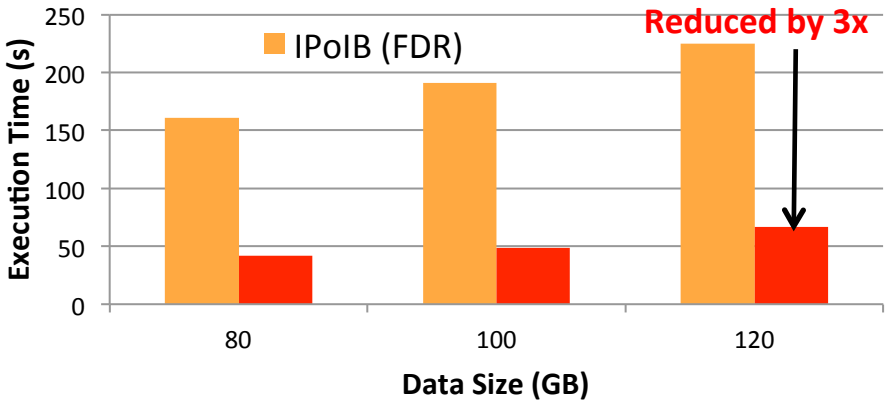


- Design Features
  - RDMA-based shuffle
  - Prefetching and caching map output
  - Efficient Shuffle Algorithms
  - In-memory merge
  - On-demand Shuffle Adjustment
  - Advanced overlapping
    - map, shuffle, and merge
    - shuffle, merge, and reduce
  - On-demand connection setup
  - InfiniBand/RoCE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

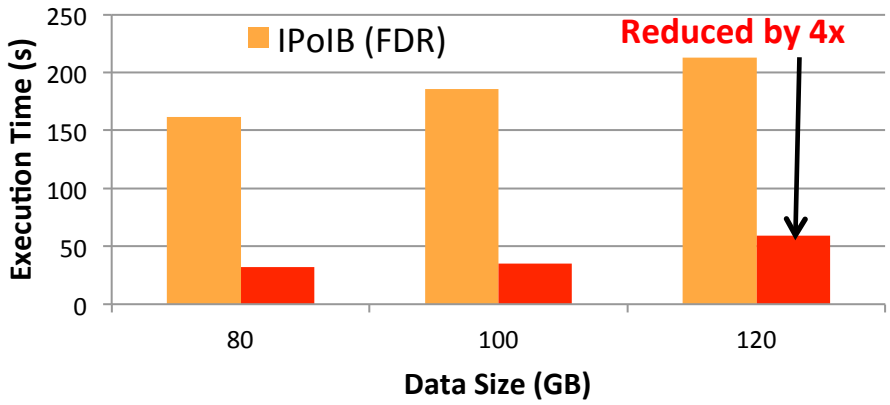
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Java based MapReduce with communication library written in native code

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 Benefits – RandomWriter & TeraGen in TACC-Stampede



RandomWriter

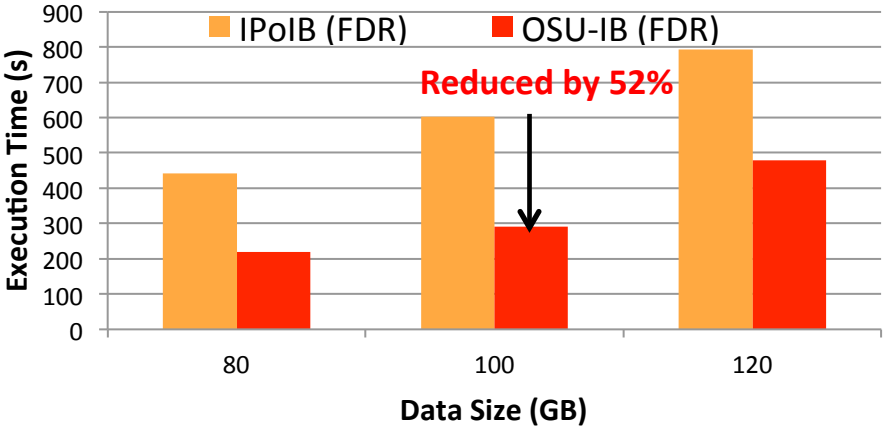


TeraGen

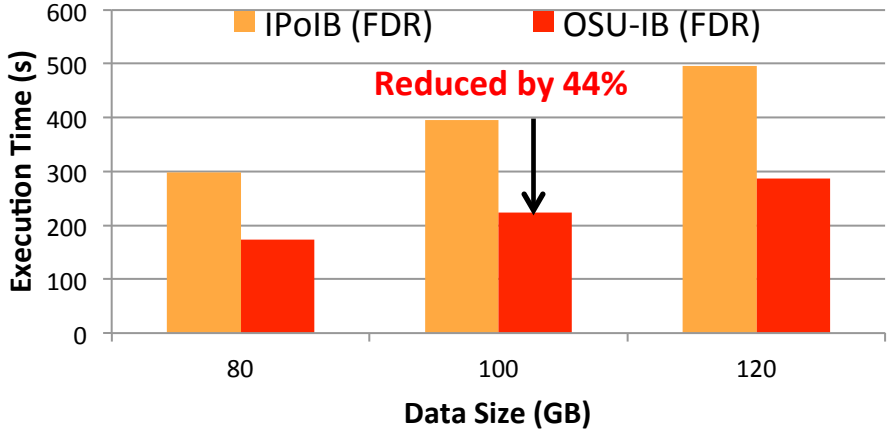
Cluster with 32 Nodes with a total of 128 maps

- RandomWriter
  - **3-4x** improvement over IPoIB for 80-120 GB file size
- TeraGen
  - **4-5x** improvement over IPoIB for 80-120 GB file size

Performance Benefits – Sort & TeraSort in TACC-Stampede



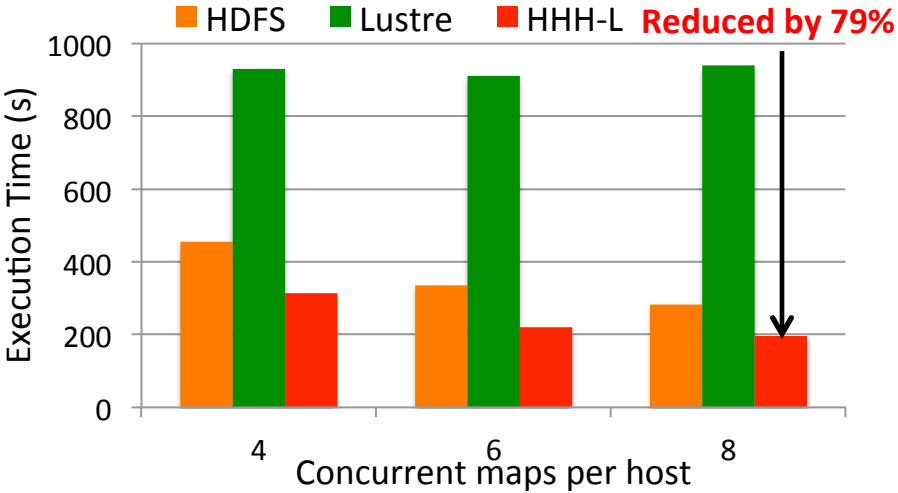
Cluster with 32 Nodes with a total of 128 maps and 57 reduces



Cluster with 32 Nodes with a total of 128 maps and 64 reduces

- Sort with single HDD per node
  - **40-52%** improvement over IPoIB for 80-120 GB data
- TeraSort with single HDD per node
  - **42-44%** improvement over IPoIB for 80-120 GB data

Evaluation of HHH and HHH-L with Applications



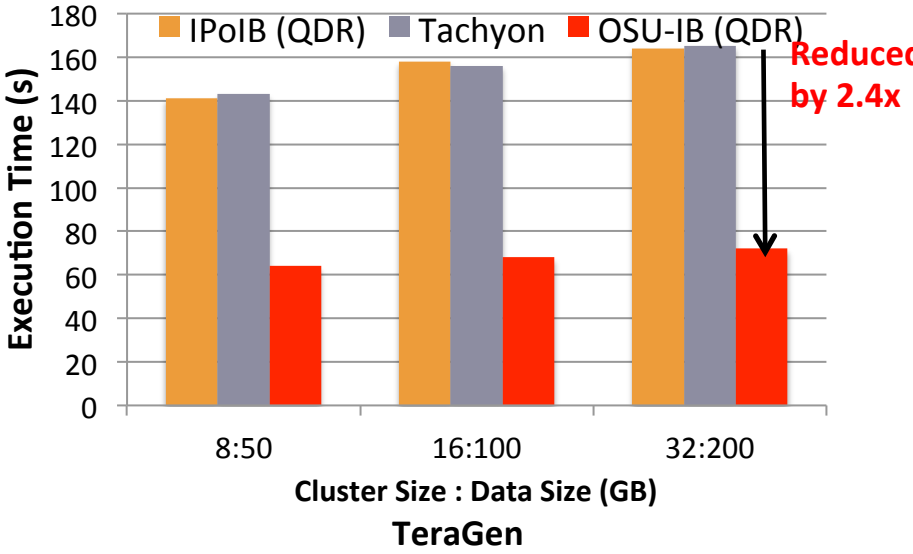
MR-MSPolyGraph

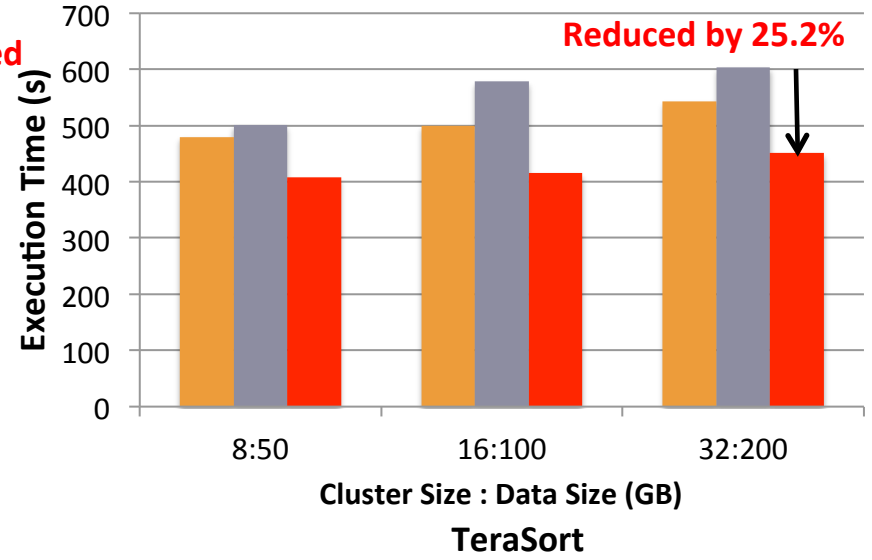
HDFS (FDR)	HHH (FDR)
60.24 s	48.3 s

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

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

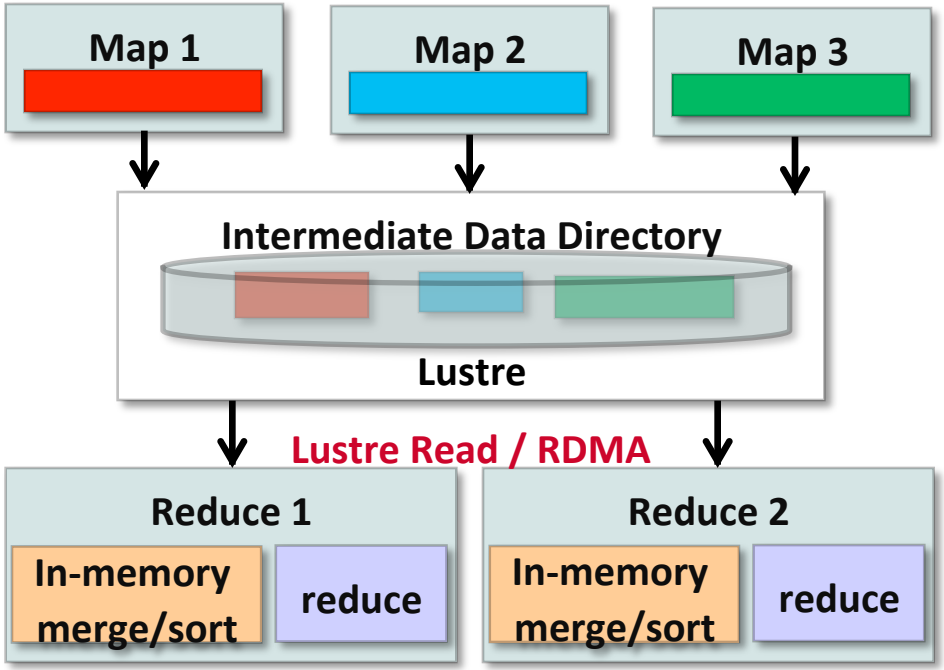




- For 200GB TeraGen on 32 nodes
  - Spark-TeraGen: HHH has 2.4x improvement over Tachyon; 2.3x over HDFS-IPoIB (QDR)
  - Spark-TeraSort: HHH has 25.2% improvement over Tachyon; 17% over HDFS-IPoIB (QDR)

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

Design Overview of Shuffle Strategies for MapReduce over Lustre

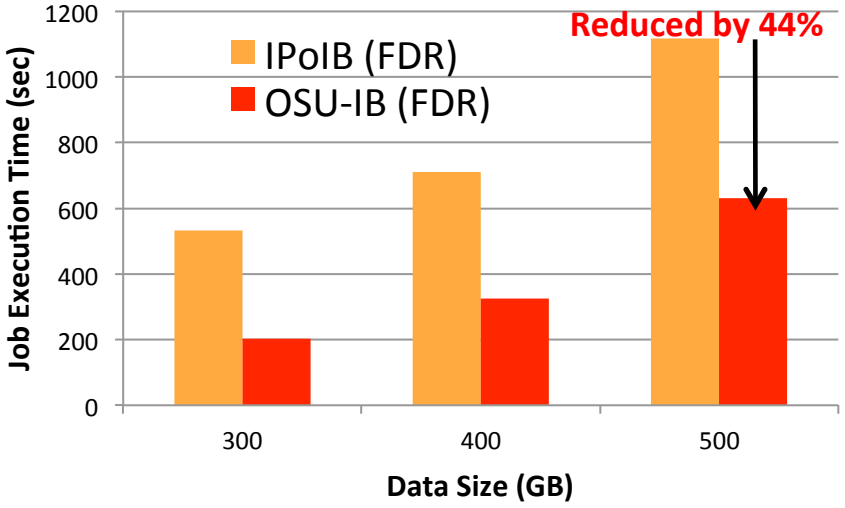


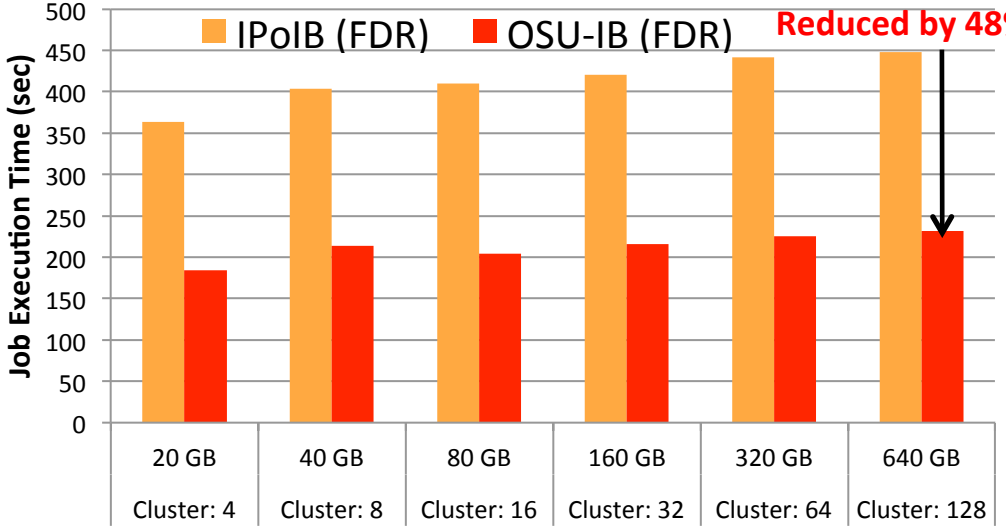
- Design Features
  - Two shuffle approaches
    - Lustre read based shuffle
    - RDMA based shuffle
  - Hybrid shuffle algorithm to take benefit from both shuffle approaches
  - Dynamically adapts to the better shuffle approach for each shuffle request based on profiling values for each Lustre read operation
  - In-memory merge and overlapping of different phases are kept similar to RDMA-enhanced MapReduce design

M. W. Rahman, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, High Performance Design of YARN MapReduce on Modern HPC Clusters with Lustre and RDMA, IPDPS, May 2015

Performance Improvement of MapReduce over Lustre on TACC-Stampede

- Local disk is used as the intermediate data directory



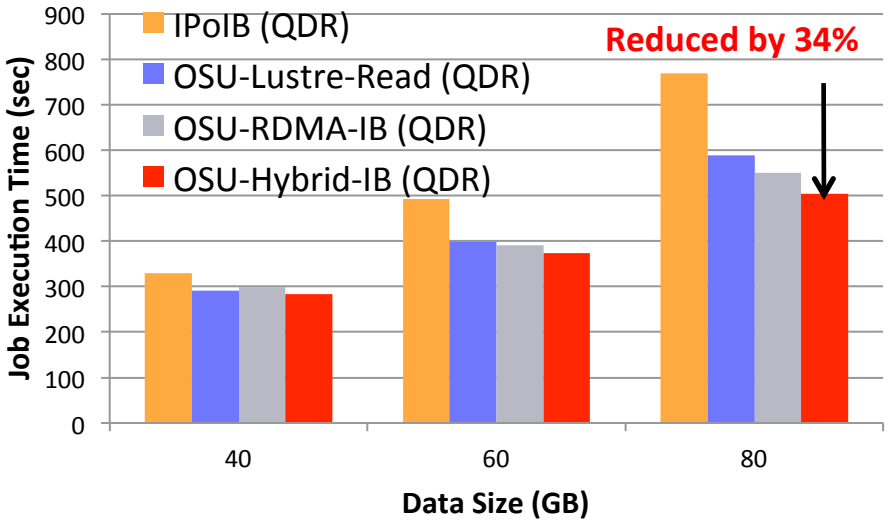


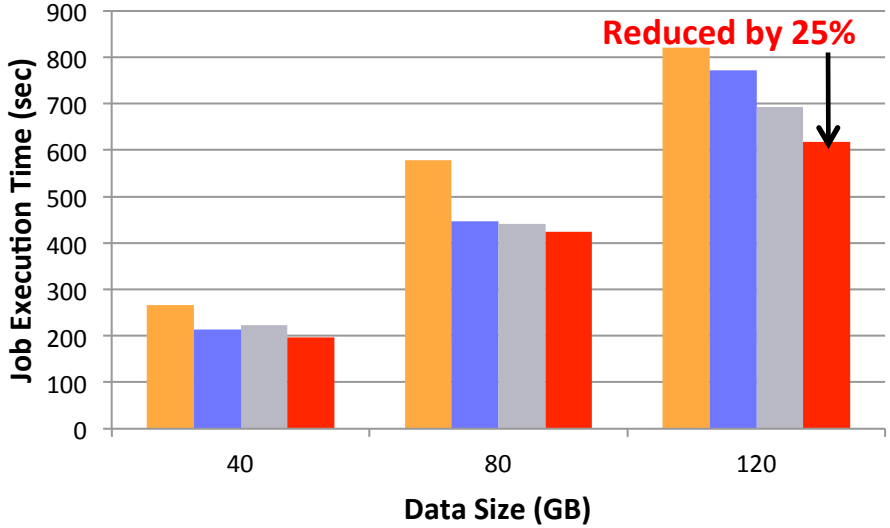
- For 500GB Sort in 64 nodes
  - 44% improvement over IPoIB (FDR)
- For 640GB Sort in 128 nodes
  - 48% improvement over IPoIB (FDR)

M. W. Rahman, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, MapReduce over Lustre: Can RDMA-based Approach Benefit?, Euro-Par, August 2014.

Case Study - Performance Improvement of MapReduce over Lustre on SDSC-Gordon

- Lustre is used as the intermediate data directory



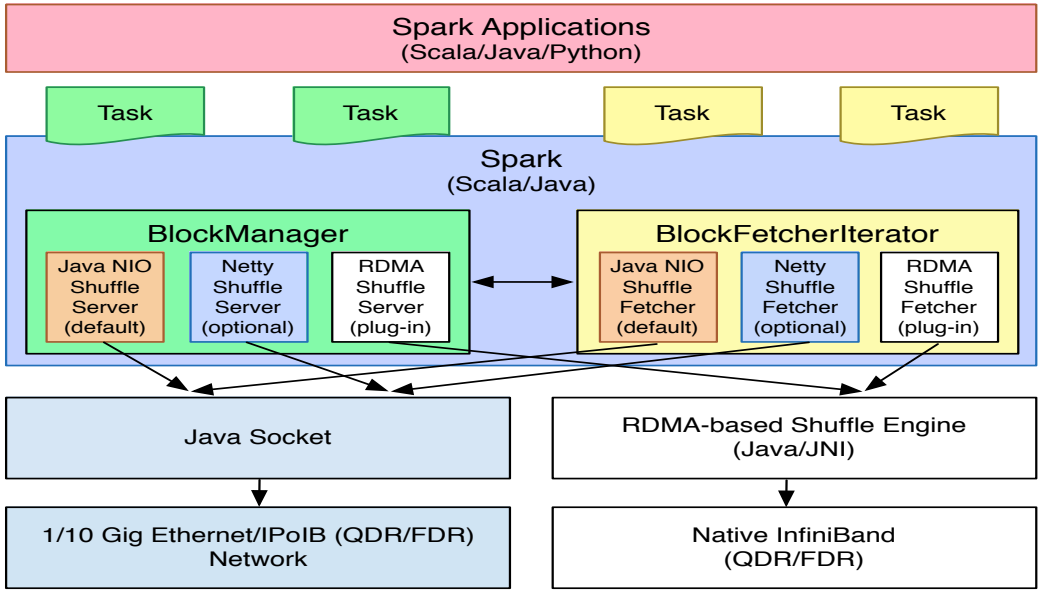


- For 80GB Sort in 8 nodes
  - 34% improvement over IPoIB (QDR)
- For 120GB TeraSort in 16 nodes
  - 25% improvement over IPoIB (QDR)

Acceleration Case Studies and Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - RPC
  - HBase
  - Spark
  - Memcached (Basic and Hybrid)
  - HDFS + Memcached-based Burst Buffer

Design Overview of Spark with RDMA

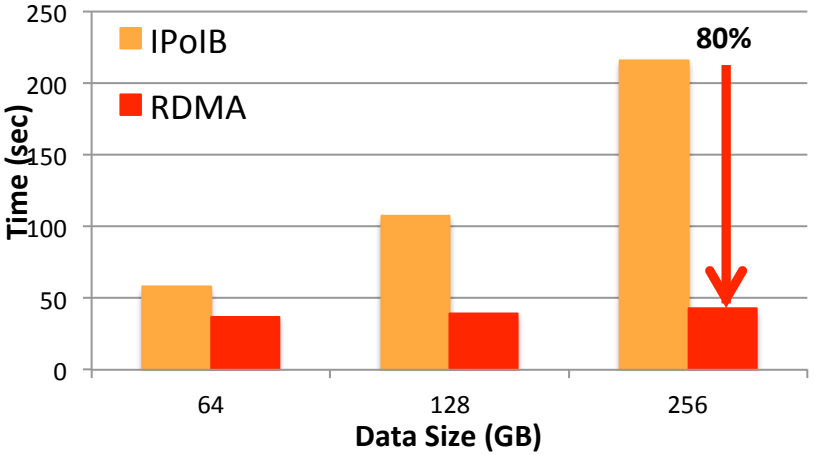


- Design Features
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

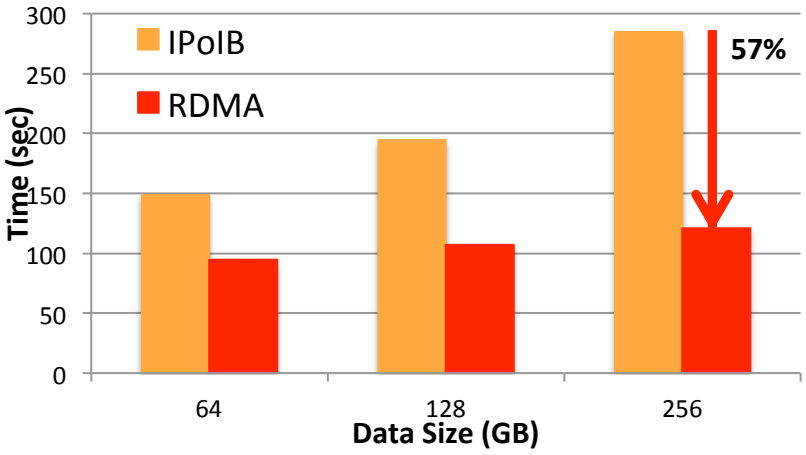
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

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

Performance Evaluation on SDSC Comet – SortBy/GroupBy



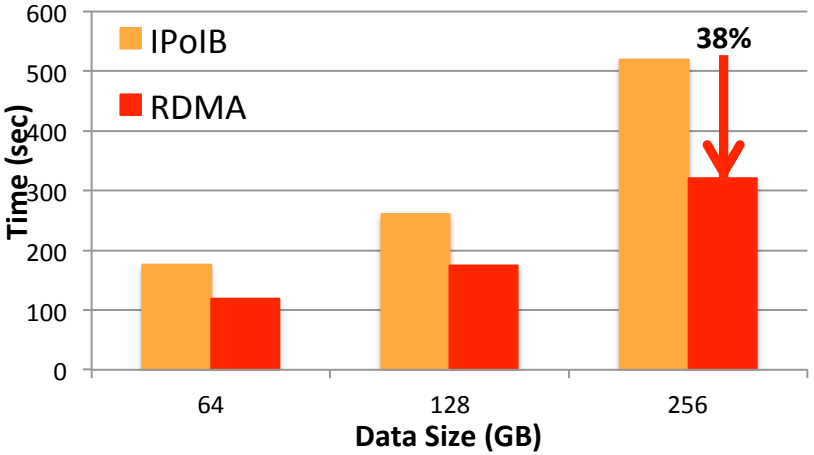
64 Worker Nodes, 1536 cores, SortByTest Total Time



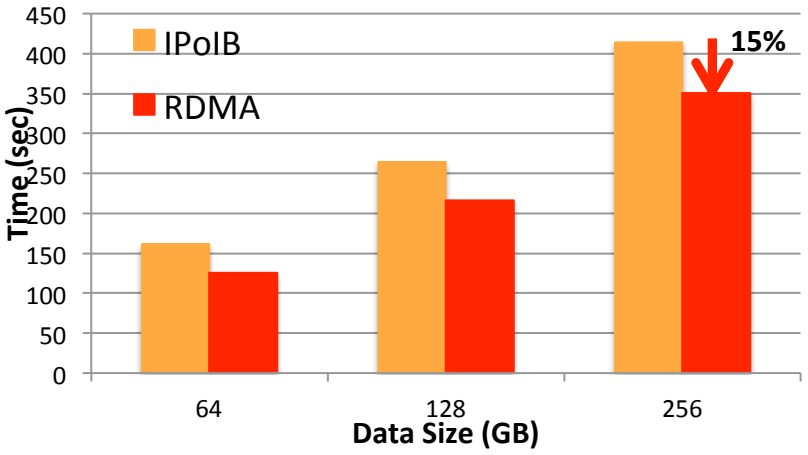
64 Worker Nodes, 1536 cores, GroupByTest Total Time

- InfiniBand FDR, SSD, 64 Worker Nodes, 1536 Cores, (1536M 1536R)
- RDMA-based design for Spark 1.5.1
- 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 57% over IPoIB (56Gbps)

Performance Evaluation on SDSC Comet – HiBench Sort/TeraSort



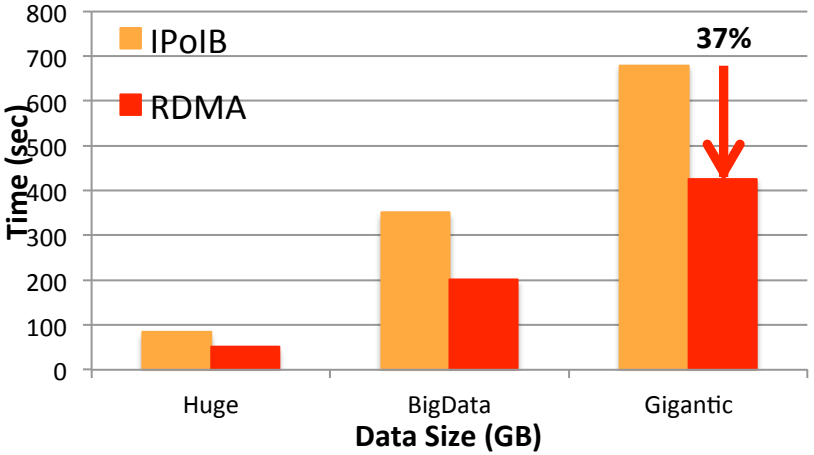
64 Worker Nodes, 1536 cores, Sort Total Time



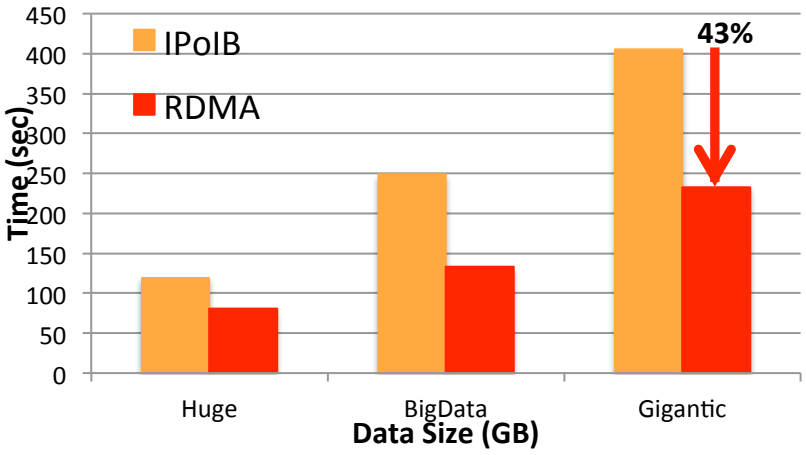
64 Worker Nodes, 1536 cores, TeraSort Total Time

- InfiniBand FDR, SSD, 64 Worker Nodes, 1536 Cores, (1536M 1536R)
- RDMA-based design for Spark 1.5.1
- RDMA vs. IPoIB with 1536 concurrent tasks, single SSD per node.
  - Sort: Total time reduced by 38% over IPoIB (56Gbps)
  - TeraSort: Total time reduced by 15% over IPoIB (56Gbps)

Performance Evaluation on SDSC Comet – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time



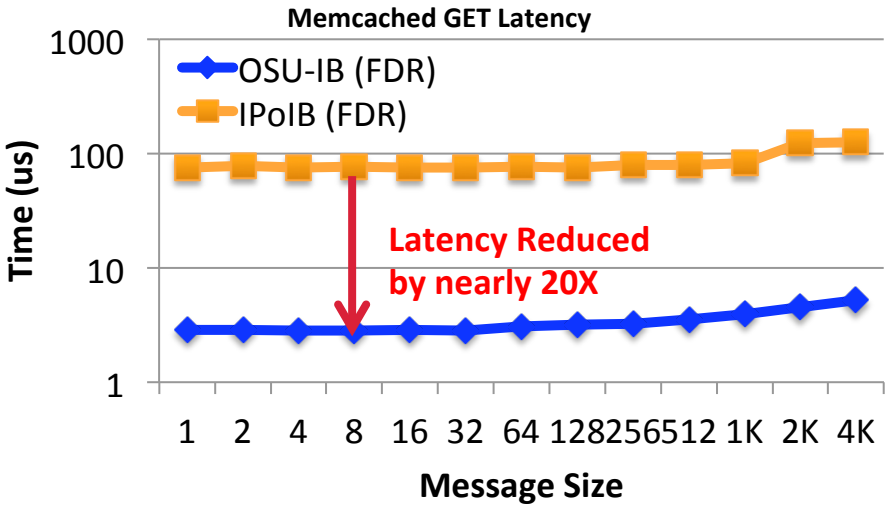
64 Worker Nodes, 1536 cores, PageRank Total Time

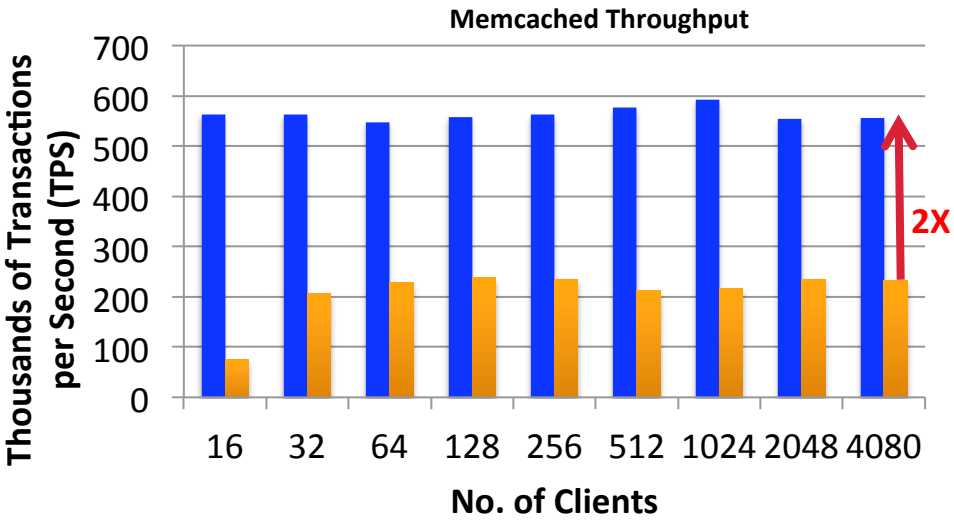
- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA-based design for Spark 1.5.1
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced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

Acceleration Case Studies and Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - RPC
  - HBase
  - Spark
  - Memcached (Basic and Hybrid)
  - HDFS + Memcached-based Burst Buffer

RDMA-Memcached Performance (FDR Interconnect)

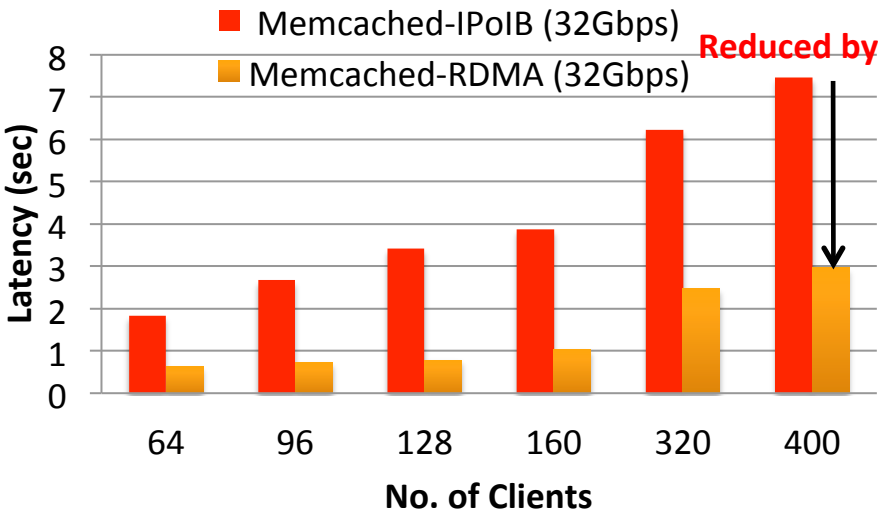


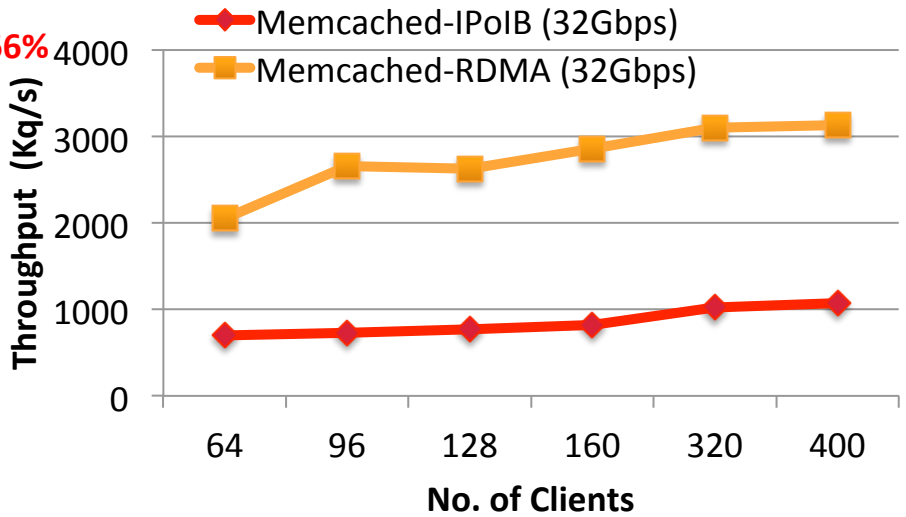


Experiments on TACC Stampede (Intel SandyBridge Cluster, IB: FDR)

- 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

Micro-benchmark Evaluation for OLDP workloads

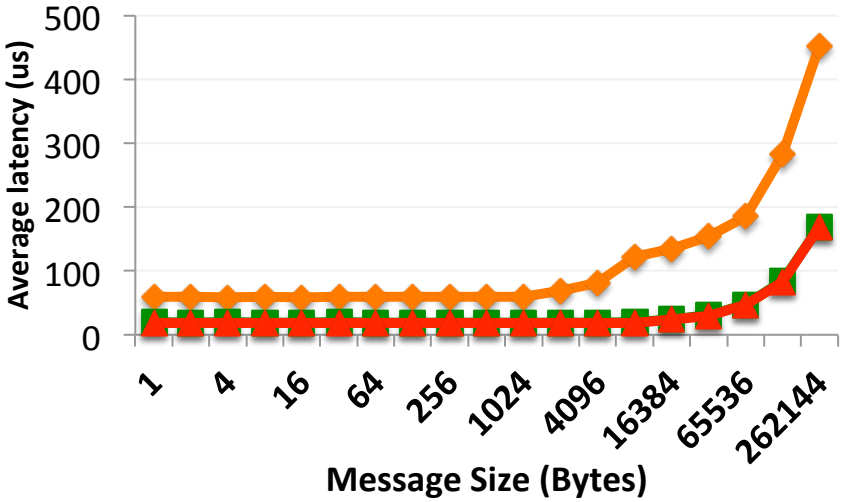


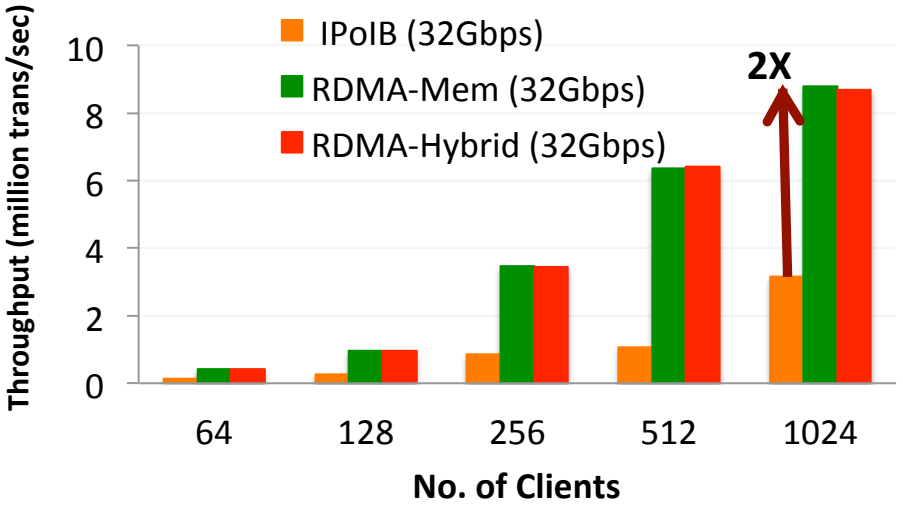


- Illustration with Read-Cache-Read access pattern using modified mysqlslap load testing tool
- Memcached-RDMA can
  - improve query latency by up to 66% over IPoIB (32Gbps)
  - throughput by up to 69% over IPoIB (32Gbps)

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

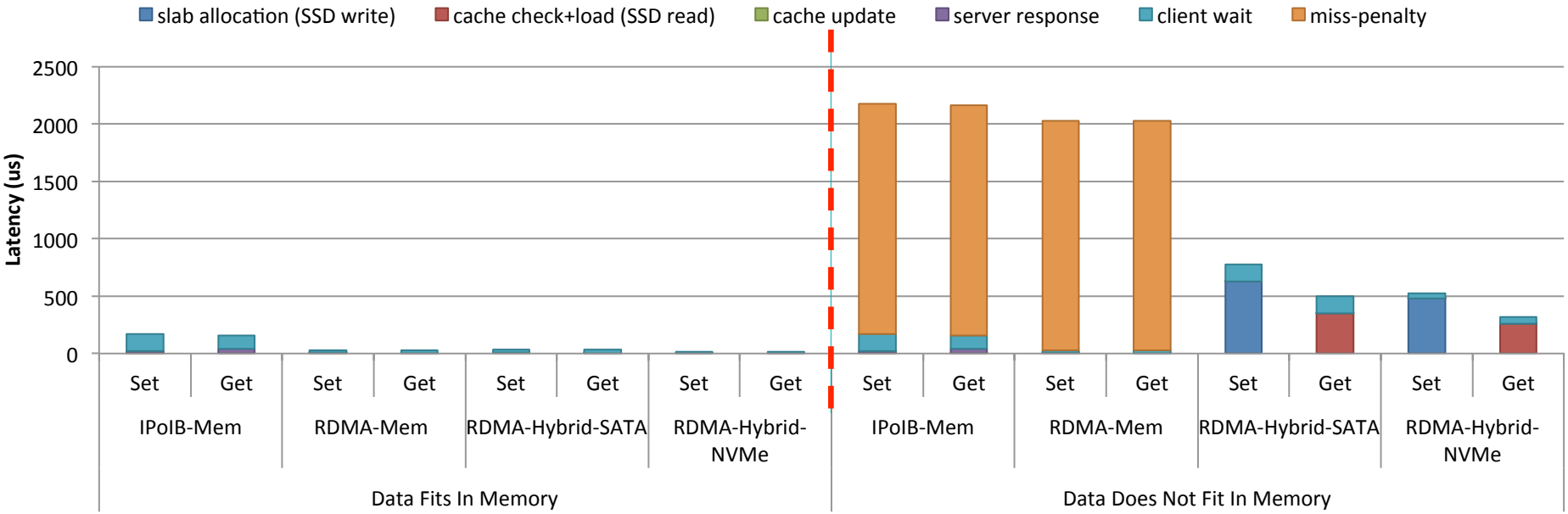
Performance Benefits of Hybrid Memcached (Memory + SSD) on SDSC-Gordon





- ohb_memlat & ohb_memthr latency & throughput micro-benchmarks
- Memcached-RDMA can
  - improve query latency by up to 70% over IPoIB (32Gbps)
  - improve throughput by up to 2X over IPoIB (32Gbps)
  - No overhead in using hybrid mode when all data can fit in memory

Performance Evaluation on IB FDR + SATA/NVMe SSDs



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

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

- Upcoming Releases of RDMA-enhanced Packages will support
  - HBase
  - Impala
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  - MapReduce
  - RPC
- Advanced designs with upper-level changes and optimizations
  - Memcached with Non-blocking API
  - HDFS + Memcached-based Burst Buffer

Concluding Remarks

- Discussed challenges in accelerating Hadoop, Spark and Memcached
- Presented initial designs to take advantage of HPC technologies to accelerate HDFS, MapReduce, RPC, Spark, and Memcached
- 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

Second International Workshop on High-Performance Big Data Computing (HPBDC)

HPBDC 2016 will be held with IEEE International Parallel and Distributed Processing Symposium (IPDPS 2016), Chicago, Illinois USA, May 27th, 2016

Keynote Talk: Dr. Chaitanya Baru,
Senior Advisor for Data Science, National Science Foundation (NSF);
Distinguished Scientist, San Diego Supercomputer Center (SDSC)

Panel Moderator: Jianfeng Zhan (ICT/CAS)
Panel Topic: Merge or Split: Mutual Influence between Big Data and HPC Techniques

Six Regular Research Papers and Two Short Research Papers

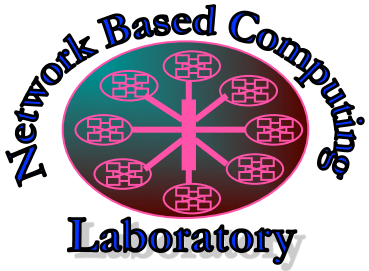
http://web.cse.ohio-state.edu/~luxi/hpbdc2016

HPBDC 2015 was held in conjunction with ICDCS'15

http://web.cse.ohio-state.edu/~luxi/hpbdc2015

Thank You!

panda@cse.ohio-state.edu



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

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