Accelerating and Benchmarking Big Data and Deep Learning Systems on Modern HPC and Cloud Architectures

Invited Talk at Workshop on Benchmarking in the Datacenter,

with HPC Asia (Jan '19)

by

Xiaoyi Lu

The Ohio State University

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

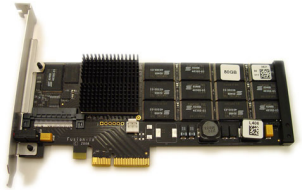
http://www.cse.ohio-state.edu/~luxi

Drivers of Modern HPC Cluster and Cloud Computing Architecture

Multi-/Many-core Processors

High Performance Interconnects – InfiniBand (with SR-IOV)
<1usec latency, 200Gbps Bandwidth>

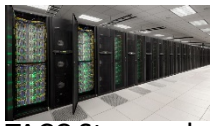
Accelerators / Coprocessors high compute density, high performance/watt
>1 TFlop DP on a chip



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

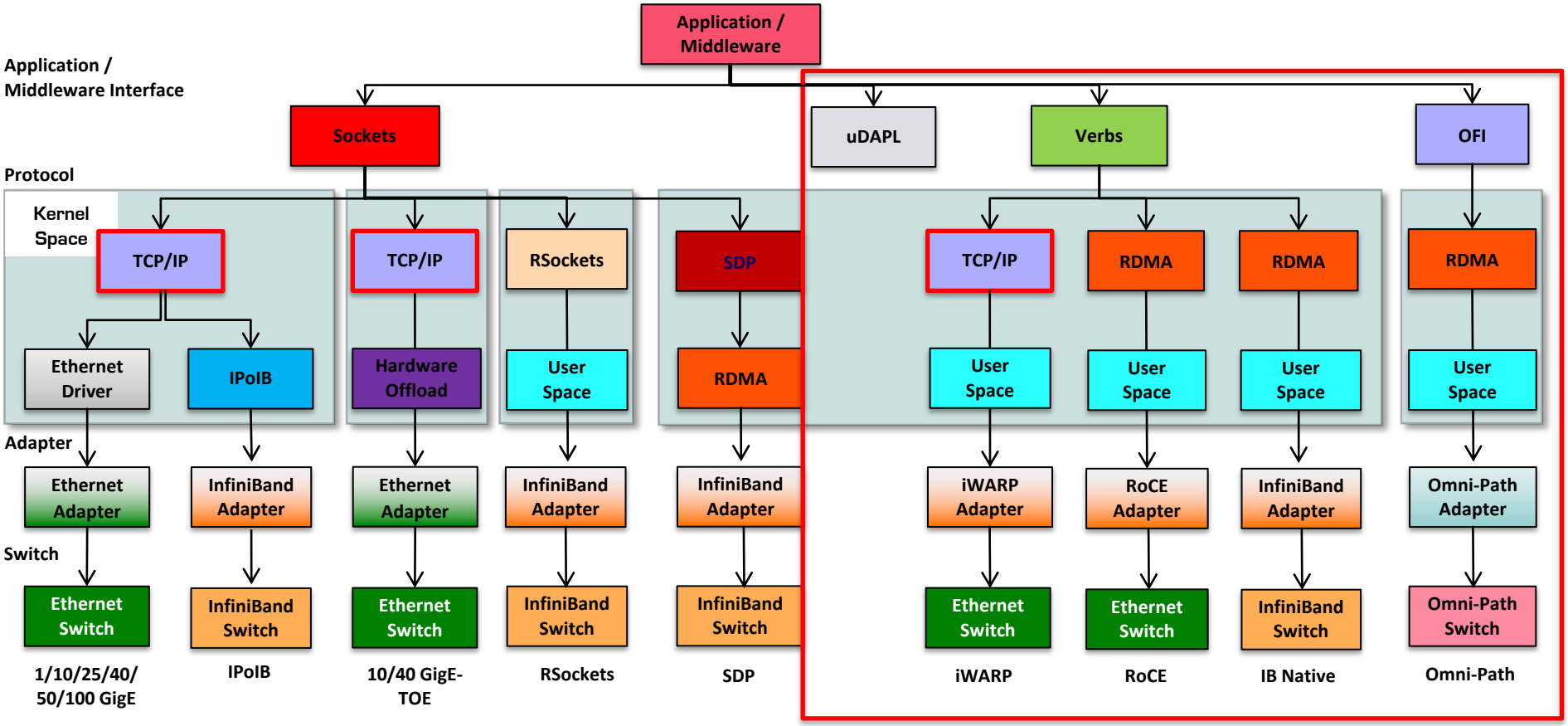
Microsoft Azure

amazon web services EC2

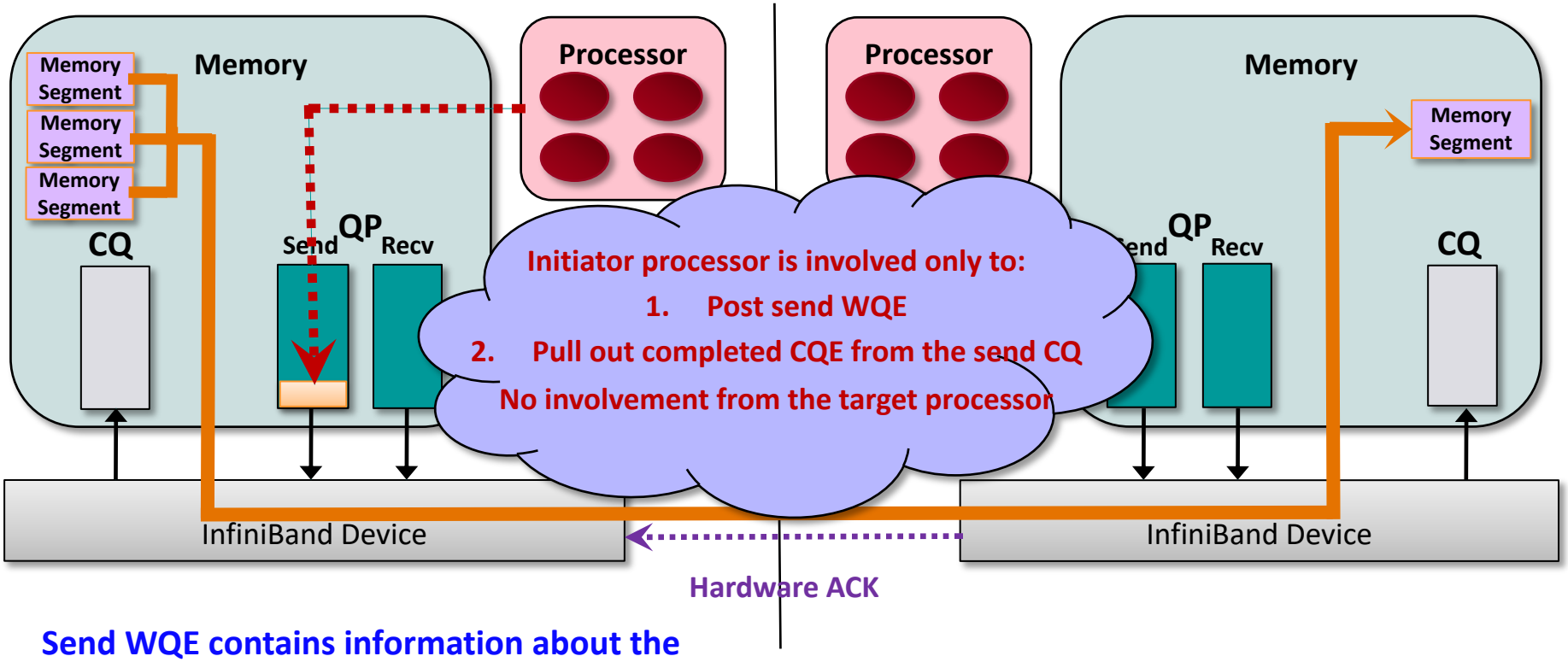
ORACLE Cloud

Chameleon Cloud

Interconnects and Protocols for Datacenters



Communication in the Memory Semantics (RDMA Model)



Send WQE contains information about the send buffer (multiple segments) and the receive buffer (single segment)

Presentation Outline

- Challenges for Accelerating Big Data Processing
- Accelerating Big Data and Deep Learning Systems on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for Hadoop, Spark, and TensorFlow
- Accelerating Big Data Processing on High-Performance Storage
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- Challenges in Designing Benchmarks for Big Data Processing and Deep Learning
  - OSU HiBD Benchmarks
- Conclusion and Q&A

How Can 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"!

The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark
- RDMA for Apache Hadoop 3.x (RDMA-Hadoop-3.x)
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache Kafka
- 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: 300 organizations from 35 countries
- More than 28,900 downloads from the project site

Available for InfiniBand and RoCE

Also run on Ethernet

Available for x86 and OpenPOWER

Support for Singularity and Docker

HiBD
High-Performance Big Data

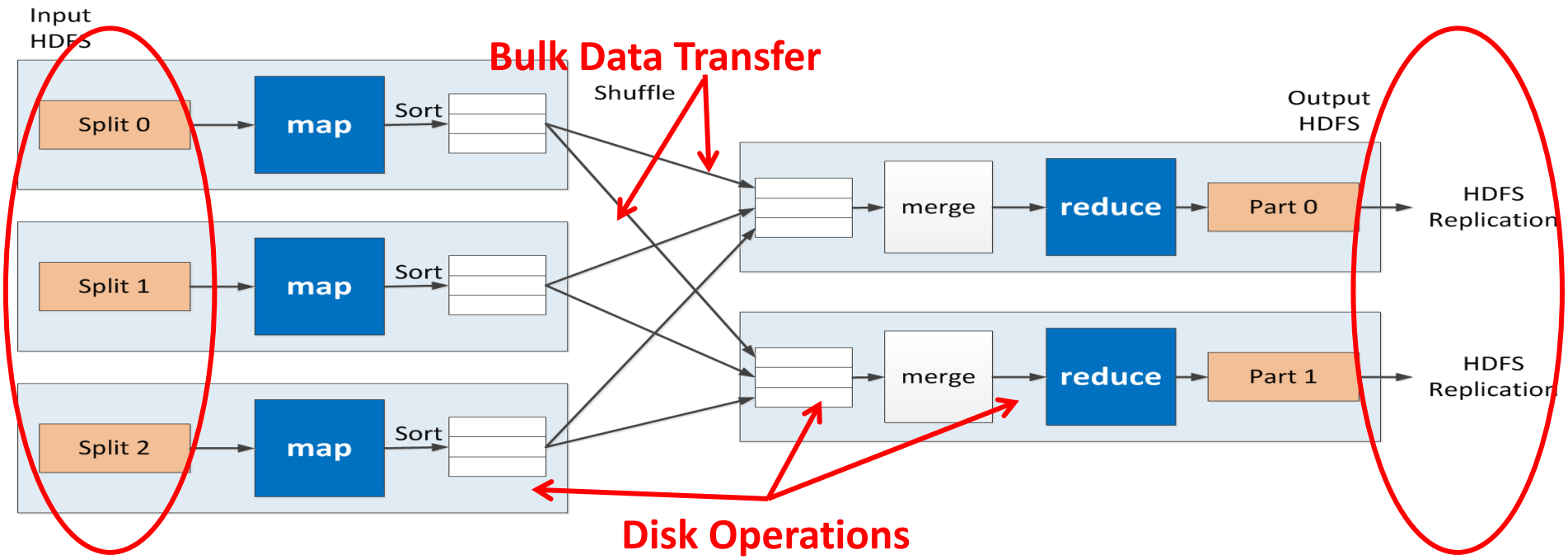
Network Based Computing Laboratory

The Ohio State University

Presentation Outline

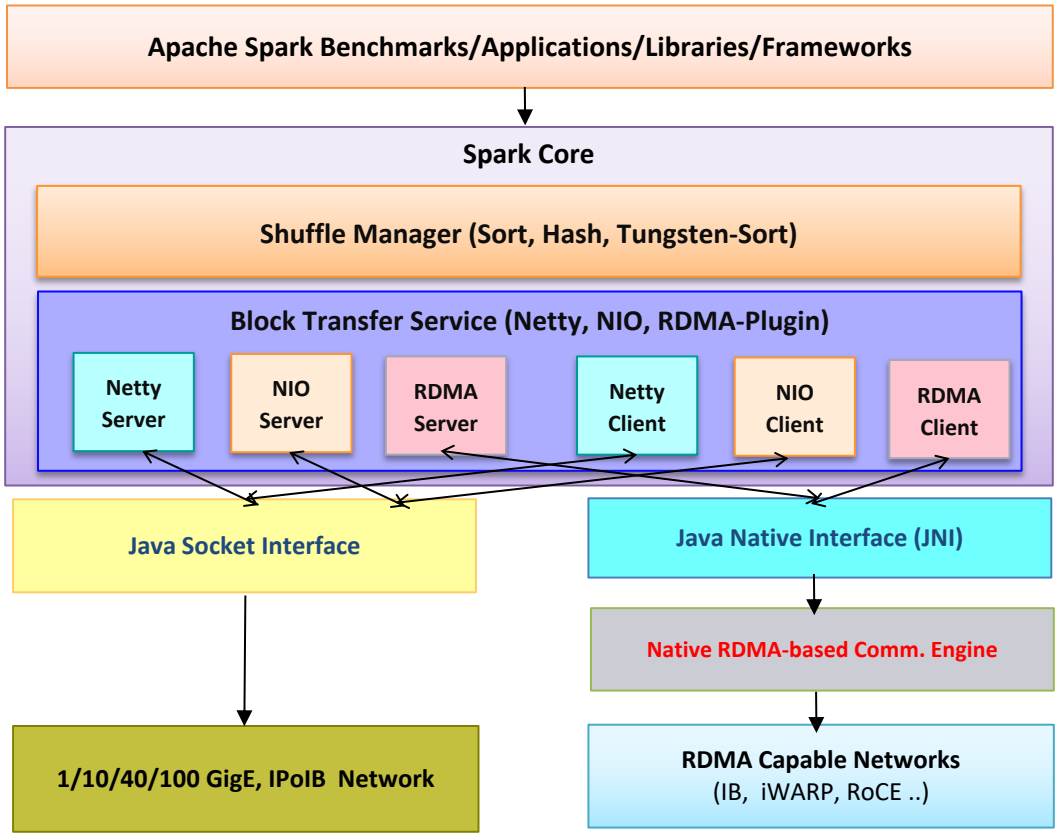
- Challenges for Accelerating Big Data Processing
- Accelerating Big Data and Deep Learning Systems on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for Hadoop, Spark, and TensorFlow
- Accelerating Big Data Processing on High-Performance Storage
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- Challenges in Designing Benchmarks for Big Data Processing and Deep Learning
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Bottlenecks in Data Computing Frameworks (e.g., Hadoop, Spark)



- Map and Reduce Tasks carry out the total job execution
  - Map tasks read from HDFS, operate on it, and write the intermediate data to local disk
  - Reduce tasks get these data by shuffle from TaskTrackers, operate on it and write to HDFS
- Communication in the pipeline
  - Shuffle phase uses HTTP over Java Sockets
  - Replication phase uses Java Sockets

Design Overview of Spark with RDMA



- Design Features
  - Plugin-based design
  - SEDA-based architecture
  - Non-blocking RDMA-based in-memory merge pipeline
  - Dynamic connection management and sharing
  - InfiniBand/RoCE support
  - OpenPOWER support

X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, HotI, 2014

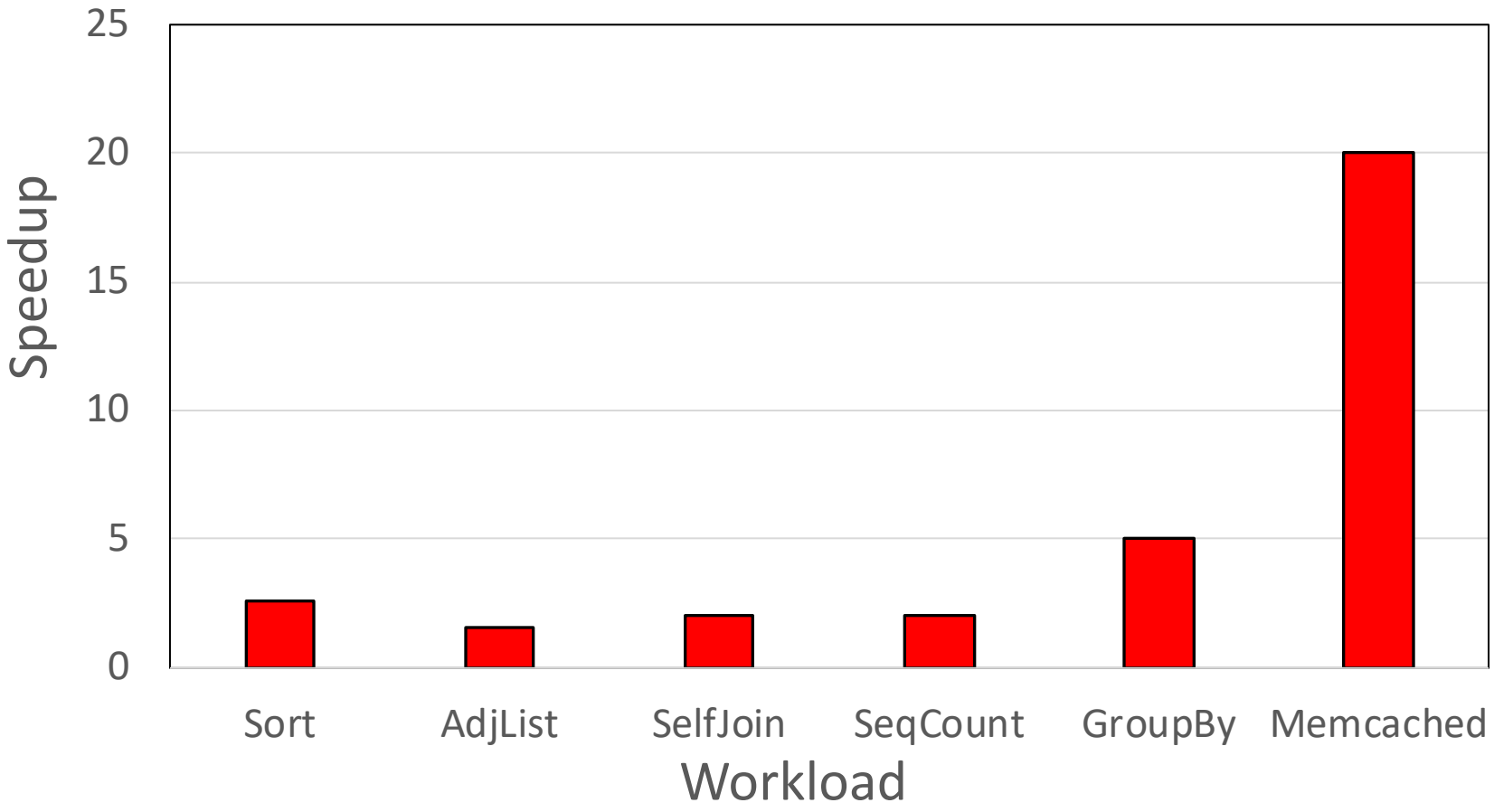
X. Lu, D. Shankar, S. Gugnani, and D. K. Panda, High-Performance Design of Apache Spark with RDMA and Its Benefits on Various Workloads, IEEE BigData, 2016

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

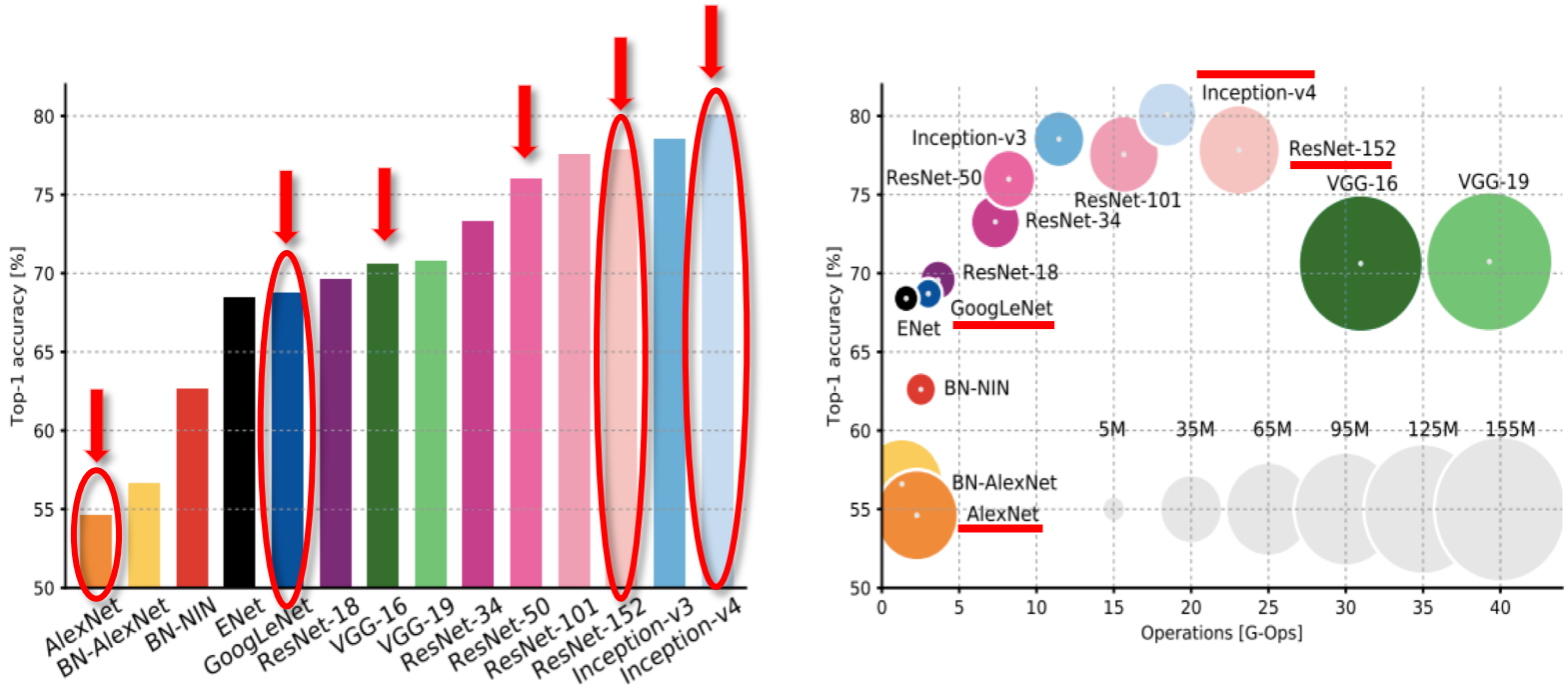
X. Lu, H. Shi, D. Shankar, and D. K. Panda, Performance Characterization and Acceleration of Big Data Workloads on OpenPOWER System, IEEE BigData, 2017

X. Lu, H. Shi, R. Biswas, H. Javed, and D. K. Panda, DLoBD: A Comprehensive Study of Deep Learning over Big Data Stacks on HPC Clusters, IEEE Trans. on MSCS, 2018

Speedup of RDMA Compared to IPoIB

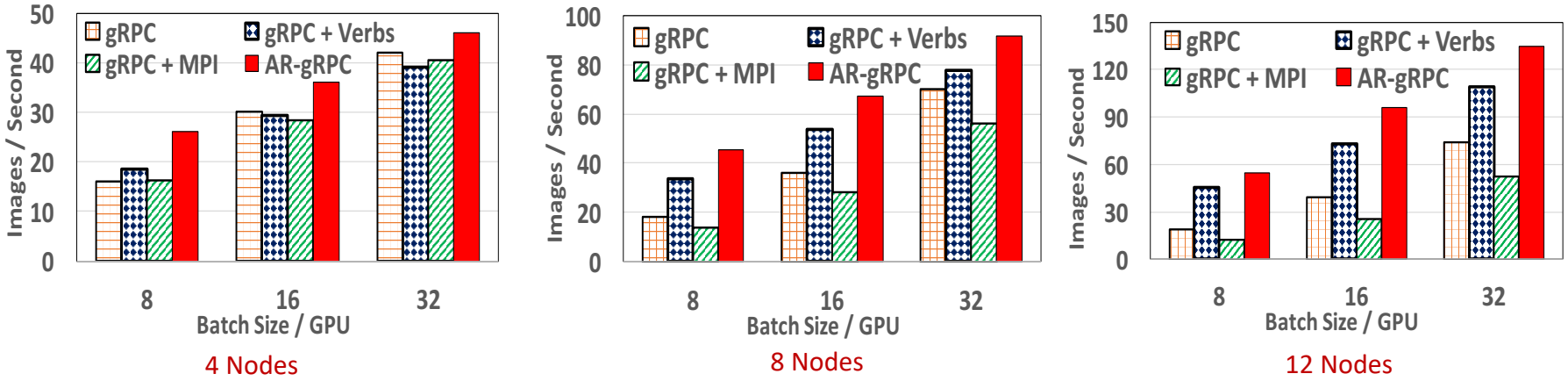


Comparison of Different CNNs



Canziani, Alfredo and Paszke, Adam and Culurciello, Eugenio, "An analysis of deep neural network models for practical applications" in arXiv preprint arXiv:1605.07678, 2016

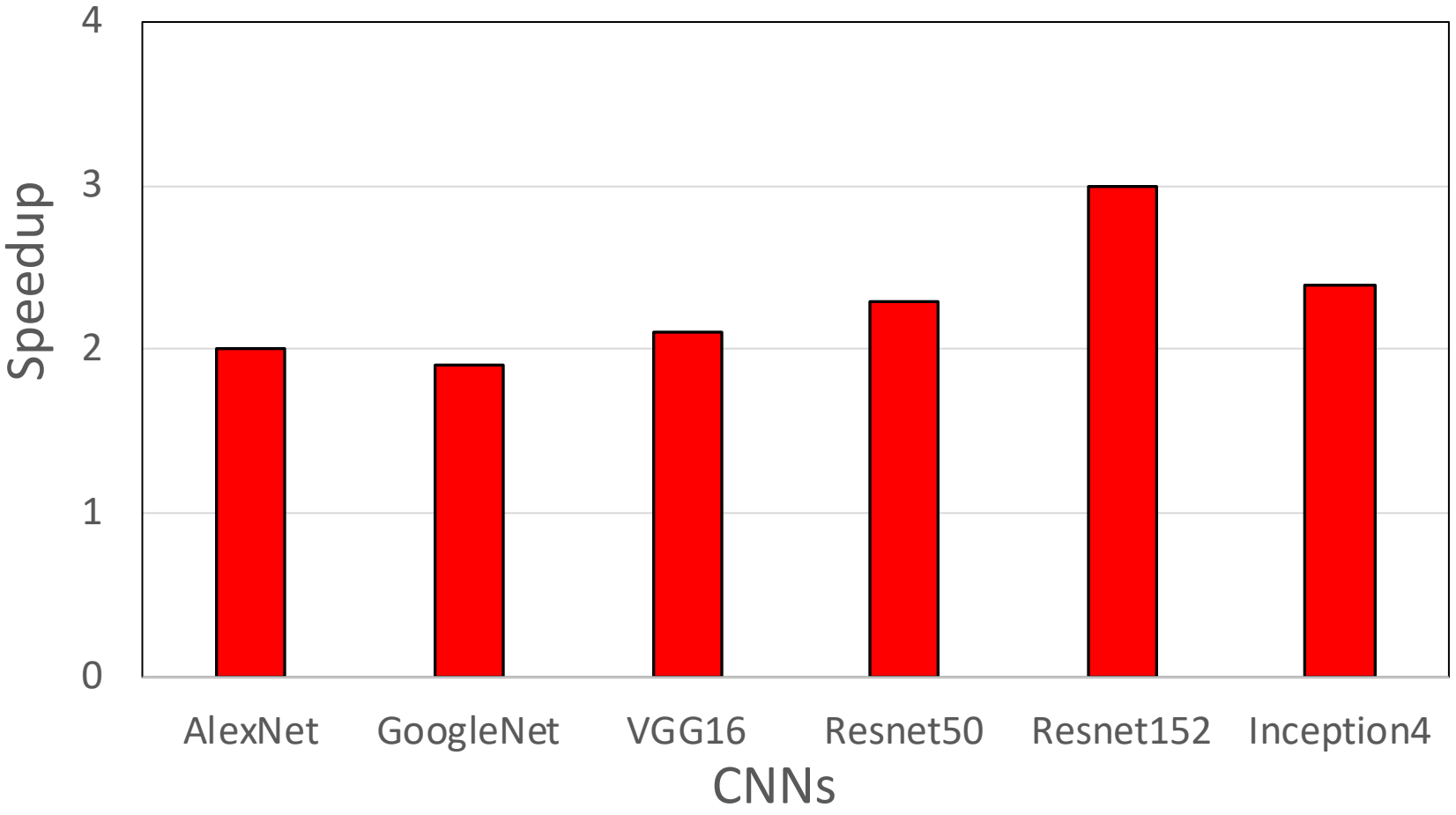
Evaluation of OSU-TensorFlow-RDMA: Resnet152



Resnet152 Evaluation on Cluster A (Higher Better); *TotalBatchSize* = (*BatchSize/GPU*)×*NUMofGPUs*

- AR-gRPC accelerates TensorFlow by 62% (batch size 8/GPU) more compared to default gRPC on 4 nodes
- AR-gRPC improves Resnet152 performance by 32% (batch size 32/GPU) to 147% on 8 nodes
- AR-gRPC incurs a maximum speedup of 3x (55 vs 18 images) compared to default gRPC 12 nodes
  - Even for higher batch size of 32/GPU (total 352) AR-gRPC improves TensorFlow performance by 82% 12 nodes
- AR-gRPC processes a maximum of 40%, 35%, and 30% more images, on 4, 8, and 12 nodes, respectively, than Verbs
- AR-gRPC achieves a maximum speedup of 1.61x, 3.3x and 4.5x compared to MPI channel on 4, 8, and 12 nodes, respectively

Speedup of RDMA Compared to IPoIB for CNNs



R. Biswas, X. Lu, and D. K. Panda, Accelerating TensorFlow with Adaptive RDMA-based gRPC, HiPC, 2018.

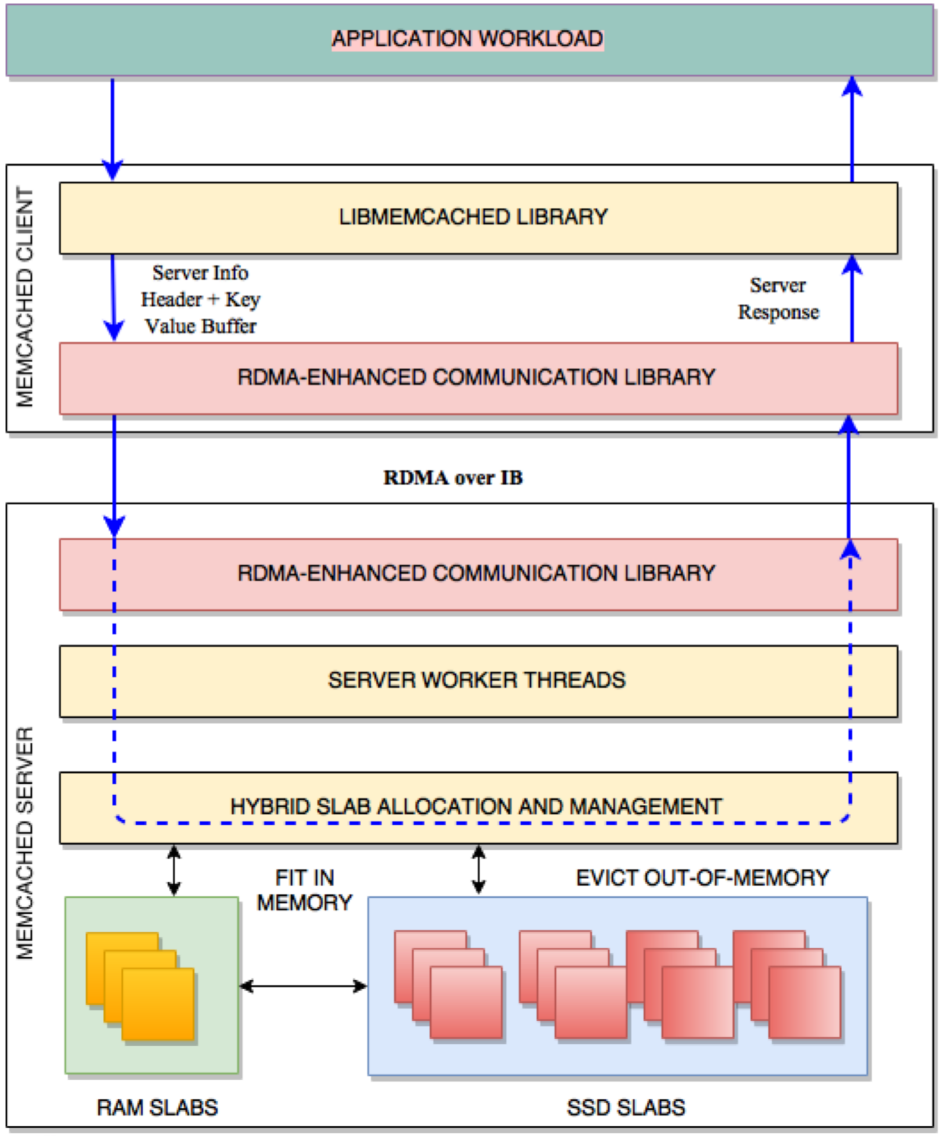
Co-Design Data Computing Frameworks with RDMA

- **HDFS Accelerations with RDMA-capable Networks and High-Speed Storage**
  - 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
  - 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, 2015
  - N. Islam, M. W. Rahman, X. Lu, and D. K. Panda, Efficient Data Access Strategies for Hadoop and Spark on HPC Cluster with Heterogeneous Storage, IEEE BigData, 2016
- **Hadoop MapReduce Accelerations with RDMA-capable Networks and Lustre**
  - 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
  - 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
  - M. W. Rahman, N. S. Islam, X. Lu, and D. K. Panda, A Comprehensive Study of MapReduce over Lustre for Intermediate Data Placement and Shuffle Strategies on HPC Clusters, TPDS, 2017
- **Memcached-Accelerations with RDMA-capable Networks**
  - D. Shankar, X. Lu, D. K. Panda, High-Performance Hybrid Key-Value Store on Modern Clusters with RDMA Interconnects and SSDs: Non-blocking Extensions, Designs, and Benefits, IPDPS, 2016
  - D. Shankar, X. Lu, D. K. Panda, High-Performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads, ICDCS, 2017
  - X. Lu, D. Shankar, D. K. Panda, Scalable and Distributed Key-Value Store-based Data Management Using RDMA-Memcached, TCDE, 2017
- **More: Spark, Kafka, HBase, gRPC/TensorFlow Accelerations with RDMA-capable Networks**

Presentation Outline

- Challenges for Accelerating Big Data Processing
- Accelerating Big Data and Deep Learning Systems on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for Hadoop, Spark, and TensorFlow
- Accelerating Big Data Processing on High-Performance Storage
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- Challenges in Designing Benchmarks for Big Data Processing and Deep Learning
  - OSU HiBD Benchmarks
- Conclusion and Q&A

Overview of SSD-Assisted Hybrid RDMA-Memcached Design



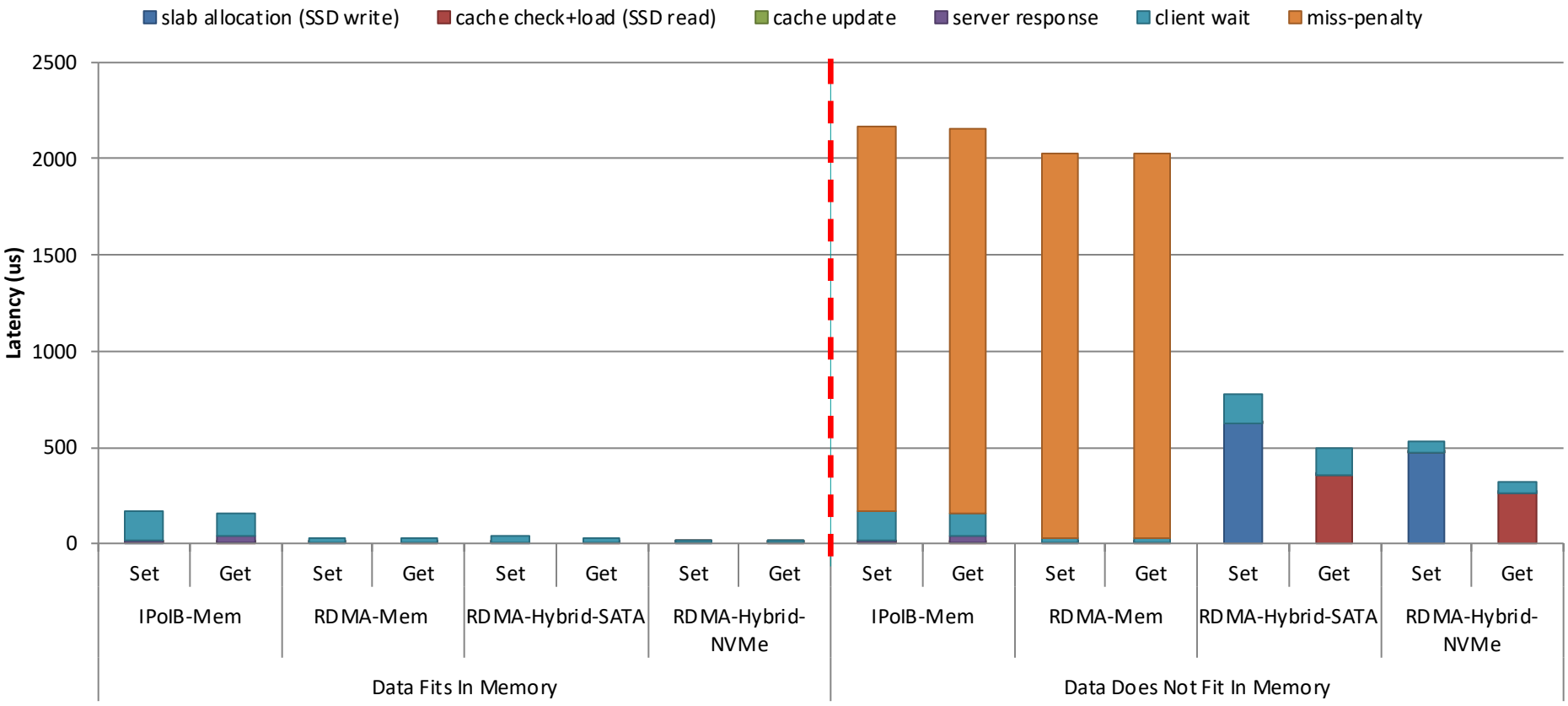
- Design Features
  - Hybrid slab allocation and management for higher data retention
  - Log-structured sequence of blocks flushed to SSD
  - SSD fast random read to achieve low latency object access
  - Uses LRU to evict data to SSD

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

D. Shankar, X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, Benchmarking Key-Value Stores on High-Performance Storage and Interconnects for Web-Scale Workloads, IEEE BigData'15

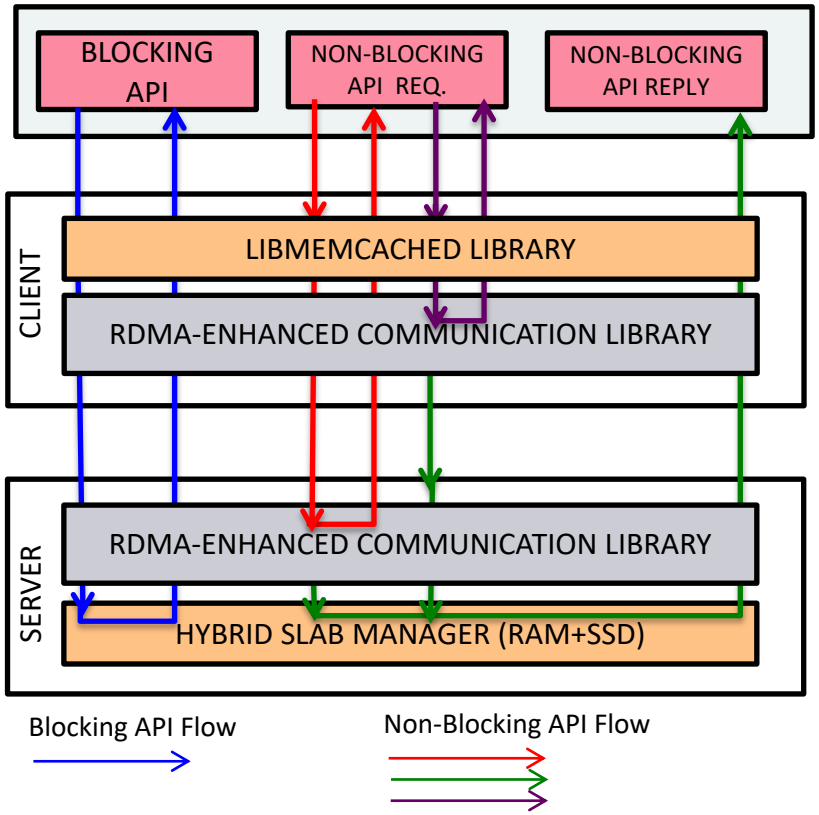
X. Lu, D. Shankar, and D. K. Panda, Scalable and Distributed Key-Value Store-based Data Management Using RDMA-Memcached, TCDE'17 (Invited Paper)

Performance Evaluation on SDSC Comet (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 fix in memory: RDMA-Hybrid gives 2x-2.5x over IPoIB/RDMA-Mem

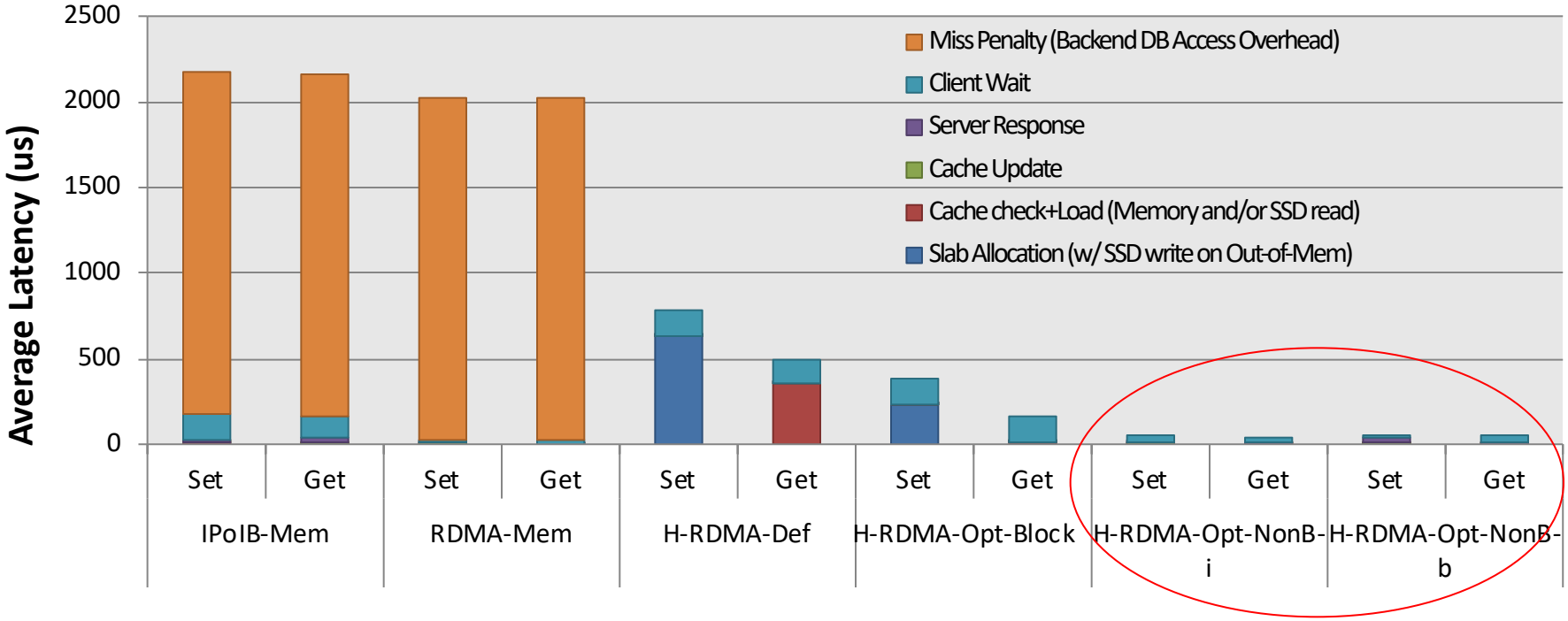
Accelerating Hybrid Memcached with RDMA, Non-blocking Extensions and SSDs



- RDMA-Accelerated Communication for Memcached Get/Set
- Hybrid 'RAM+SSD' slab management for higher data retention
- **Non-blocking API extensions**
  - **memcached_(iset/iget/bset/bget/test/wait)**
  - Achieve near in-memory speeds while hiding bottlenecks of network and SSD I/O
  - Ability to exploit communication/computation overlap
  - Optional buffer re-use guarantees
- Adaptive slab manager with different I/O schemes for higher throughput.

D. Shankar, X. Lu, N. S. Islam, M. W. Rahman, and D. K. Panda, High-Performance Hybrid Key-Value Store on Modern Clusters with RDMA Interconnects and SSDs: Non-blocking Extensions, Designs, and Benefits, IPDPS, May 2016

Performance Evaluation with Non-Blocking Memcached API



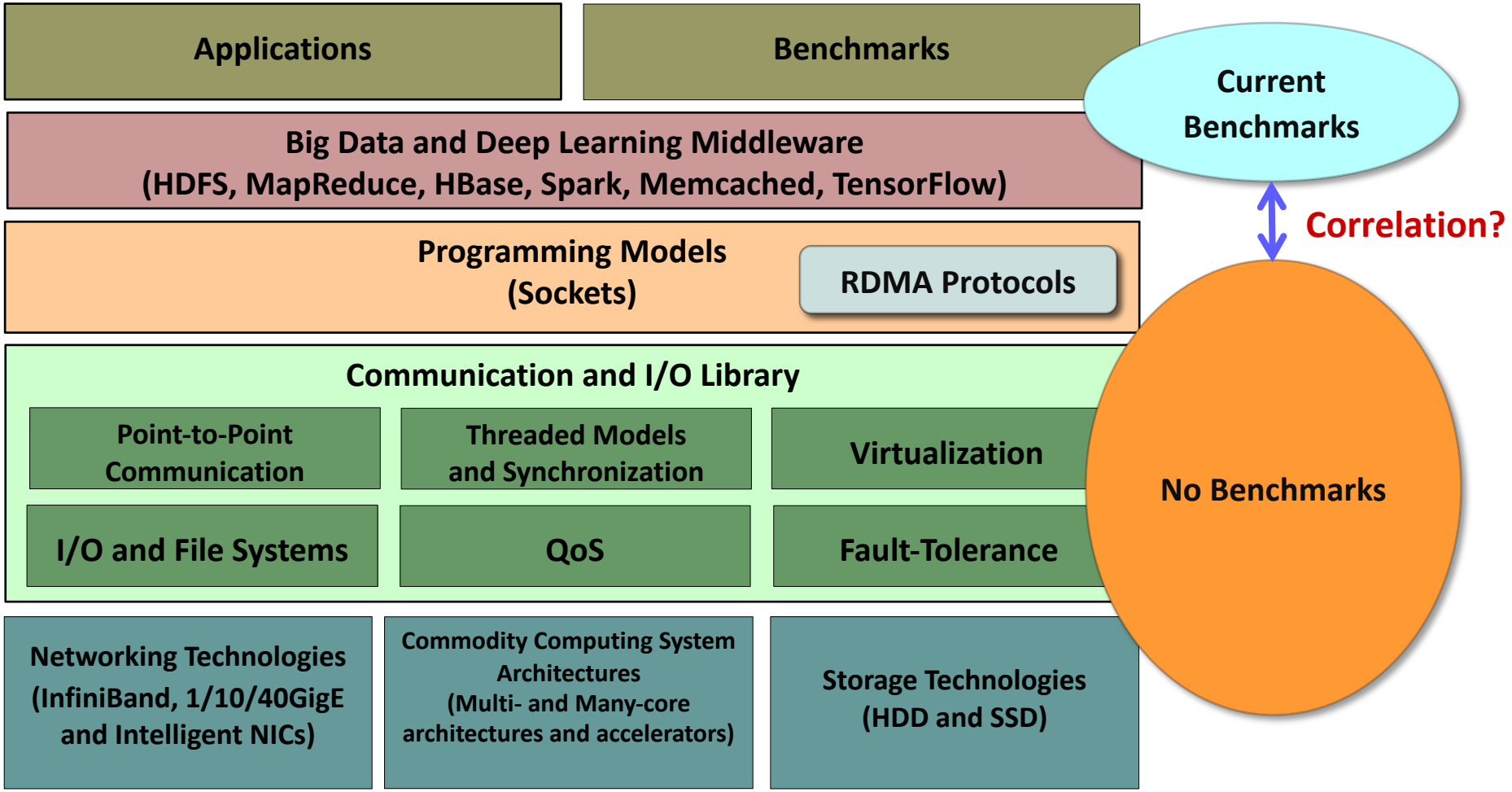
H = Hybrid Memcached over SATA SSD Opt = Adaptive slab manager Block = Default Blocking API
NonB-i = Non-blocking iset/iget API NonB-b = Non-blocking bset/bget API w/ buffer re-use guarantee

- 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

Presentation Outline

- Challenges for Accelerating Big Data Processing
- Accelerating Big Data and Deep Learning Systems on RDMA-enabled High-Performance Interconnects
  - RDMA-enhanced Designs for Hadoop, Spark, and TensorFlow
- Accelerating Big Data Processing on High-Performance Storage
  - SSD-assisted Hybrid Memory for RDMA-based Memcached
- Challenges in Designing Benchmarks for Big Data Processing and Deep Learning
  - OSU HiBD Benchmarks
- Conclusion and Q&A

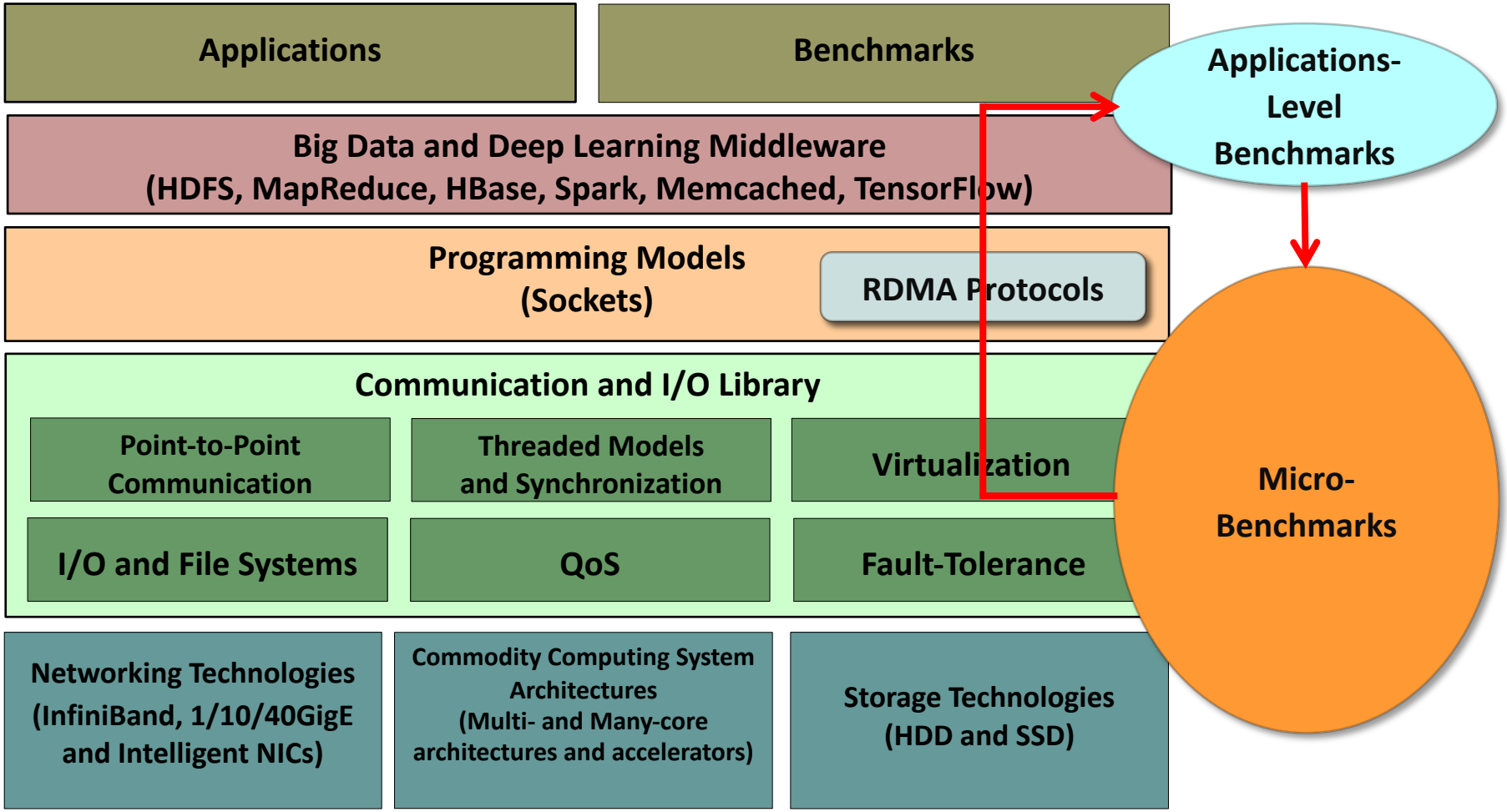
Challenges in Benchmarking of RDMA-based Designs



Are the Current Benchmarks Sufficient for Big Data Management and Processing?

- The current benchmarks provide some performance behavior
- However, do not provide any information to the designer/developer on:
  - What is happening at the lower-layer?
  - Where the benefits are coming from?
  - Which design is leading to benefits or bottlenecks?
  - Which component in the design needs to be changed and what will be its impact?
  - Can performance gain/loss at the lower-layer be correlated to the performance gain/loss observed at the upper layer?

Iterative Process – Requires Deeper Investigation and Design for Benchmarking Next Generation Big Data Systems and Applications



OSU HiBD Micro-Benchmark (OHB) Suite - HDFS

- Evaluate the performance of standalone HDFS
- Five different benchmarks
  - Sequential Write Latency (**SWL**)
  - Sequential or Random Read Latency (**SRL** or **RRL**)
  - Sequential Write Throughput (**SWT**)
  - Sequential Read Throughput (**SRT**)
  - Sequential Read-Write Throughput (**SRWT**)

N. S. Islam, X. Lu, M. W. Rahman, J. Jose, and D. K. Panda, A Micro-benchmark Suite for Evaluating HDFS Operations on Modern Clusters, Int'l Workshop on Big Data Benchmarking (WBDB '12), December 2012.

Benchmark	File Name	File Size	HDFS Parameter	Readers	Writers	Random/ Sequential Read	Seek Interval
SWL	√	√	√				
SRL/RRL	√	√	√			√	√ (RRL)
SWT		√	√		√		
SRT		√	√	√			
SRWT		√	√	√	√		

OSU HiBD Micro-Benchmark (OHB) Suite - MapReduce

- Evaluate the performance of stand-alone MapReduce
- Does not require or involve HDFS or any other distributed file system
- Models shuffle data patterns in real-workload Hadoop application workloads
- Considers various factors that influence the data shuffling phase
  - underlying network configuration, number of map and reduce tasks, intermediate shuffle data pattern, shuffle data size etc.
- Two different micro-benchmarks based on generic intermediate shuffle patterns
  - **MR-AVG:** intermediate data is evenly distributed (or approx. equal) among reduce tasks
    - **MR-RR** i.e., round-robin distribution and **MR-RAND** i.e., pseudo-random distribution
  - **MR-SKEW:** intermediate data is unevenly distributed among reduce tasks
    - Total number of shuffle key/value pairs, max% per reducer, min% per reducer to configure skew

D. Shankar, X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, A Micro-Benchmark Suite for Evaluating Hadoop MapReduce on High-Performance Networks, BPOE-5 (2014)

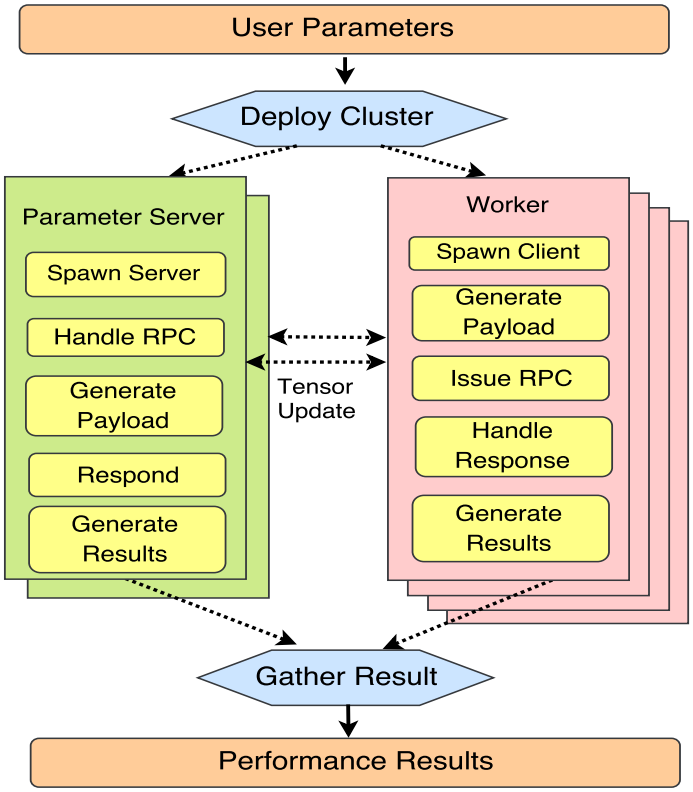
D. Shankar, X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, Characterizing and benchmarking stand-alone Hadoop MapReduce on modern HPC clusters, The Journal of Supercomputing (2016)

OSU HiBD Micro-Benchmark (OHB) Suite - Memcached

- Evaluates the performance of stand-alone Memcached in different modes
- Default API Latency benchmarks for Memcached in-memory mode
  - **SET Micro-benchmark**: Micro-benchmark for memcached set operations
  - **GET Micro-benchmark**: Micro-benchmark for memcached get operations
  - **MIX Micro-benchmark**: Micro-benchmark for a mix of memcached set/get operations (Read:Write ratio is 90:10)
- Latency benchmarks for Memcached hybrid-memory mode
- Non-Blocking API Latency Benchmark for Memcached (both in-memory and hybrid-memory mode)
- YCSB extension for RDMA-Memcached
- Calculates average latency of Memcached operations in different modes

D. Shankar, X. Lu, M. W. Rahman, N. Islam, and D. K. Panda, Benchmarking Key-Value Stores on High-Performance Storage and Interconnects for Web-Scale Workloads, IEEE International Conference on Big Data (IEEE BigData '15), Oct 2015

Design of TF-gRPC-Bench Micro-benchmark Suite



TF-gRPC-Bench Deployment

- Deploys in Parameter Server architecture to exactly model the distributed TensorFlow communication pattern
- Three different benchmarks to measure –
  - Point-to-Point latency
  - Point-to-Point Bandwidth
  - Parameter Server Throughput
- Supports both serialized and non-serialized mode of payload transfer
- Written using gRPC's C++ language binding API's
- Uses gRPC's core C APIs directly to avoid any serialization overhead
- Payload generation Schemes:
  - Uniform
  - Random
  - Skew

R. Biswas, X. Lu, and D. K. Panda, Designing a Micro-Benchmark Suite to Evaluate gRPC for TensorFlow: Early Experiences, BPOE-9, Mar 2018.

Concluding Remarks

- Discussed communication and I/O challenges in accelerating Big Data and Deep Learning systems
- Presented initial designs to take advantage of InfiniBand/RDMA and high-performance storage architectures for Hadoop, Spark, Memcached, TensorFlow, and many others
- Presented challenges in designing benchmarks
- Results are promising
- Many other open issues need to be solved
- Will enable Big Data and Deep Learning community to take advantage of modern HPC technologies to carry out their analytics in a fast and scalable manner

The 5th International Workshop on High-Performance Big Data and Cloud Computing (HPBDC)

In conjunction with IPDPS'19, Rio de Janeiro, Brazil, Monday, May 20th, 2019

Deadline	Important Date
Abstract (Optional)	January 15th, 2019
Paper Submission	February 1st, 2019
Acceptance notification	March 1st, 2019
Camera-Ready deadline	March 15th, 2019

HPBDC 2018 was held in conjunction with IPDPS'18

http://web.cse.ohio-state.edu/~luxi/hpbdc2018

HPBDC 2017 was held in conjunction with IPDPS'17

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

HPBDC 2016 was held in conjunction with IPDPS'16

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

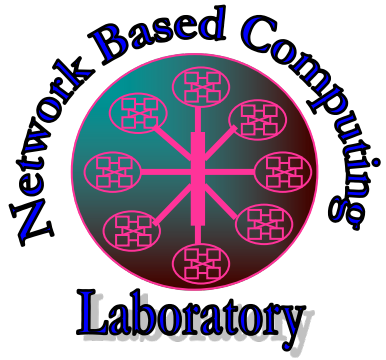
HPBDC 2015 was held in conjunction with ICDCS'15

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

Thank You!

luxi@cse.ohio-state.edu

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



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

High-Performance
Big Data

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