High Performance RDMA based Design for Big Data and Web 2.0 Memcached

Talk at OFA Developer Workshop (2013)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Big Data

- Provides groundbreaking opportunities for enterprise information management and decision making
- The rate of information growth appears to be exceeding Moore's Law
- The amount of data is exploding; companies are capturing and digitizing more information that ever
- 35 zettabytes of data will be generated and consumed by the end of this decade

Courtesy: John Gantz and David Reinsel, "The Digital Universe Decade – Are You Ready?" May 2010. http://www.emc.com/collateral/analyst-reports/idc-digital-universe-are-you-ready.pdf

Use of High-Performance Networks for Scientific Computing

- Message Passing Interface (MPI)
- Parallel File Systems
- Storage Systems
- Almost 12.5 years of Research and Development since InfiniBand was introduced in October 2000
- Other Programming Models are emerging to take advantage of High-Performance Networks
  - UPC
  - OpenSHMEM
  - Hybrid MPI+PGAS (OpenSHMEM and UPC)

Big Data/Enterprise/Commercial Computing

- Focuses on large data and data analysis
- Hadoop (HDFS, HBase, MapReduce) environment is gaining a lot of momentum
- Memcached is also used for Web 2.0

Can High-Performance Interconnects Benefit Enterprise Computing?

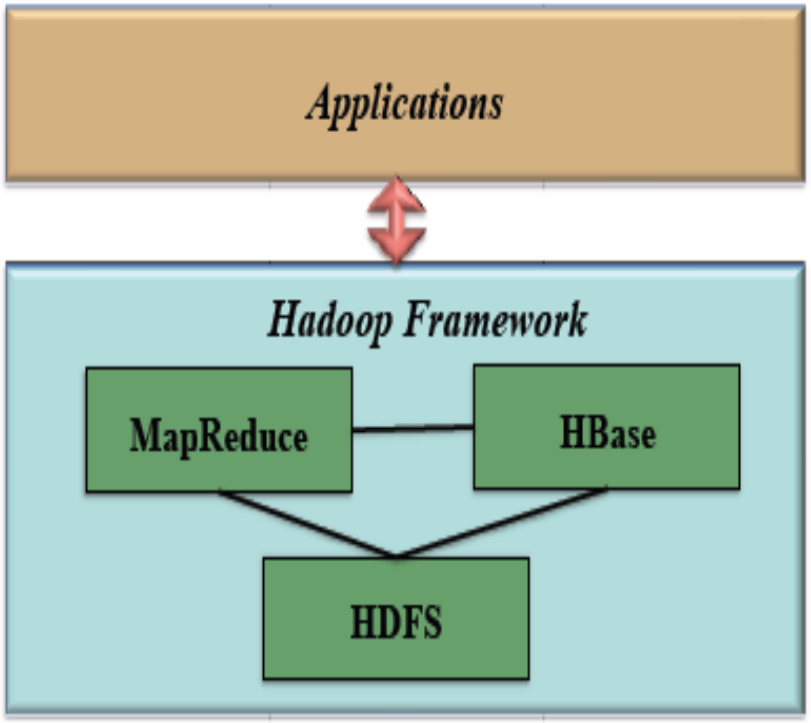
- Most of the current enterprise systems use 1GE
- Concerns for performance and scalability
- Usage of High-Performance Networks is beginning to draw interest
  - Oracle, IBM, Google are working along these directions
- What are the challenges?
- Where do the bottlenecks lie?
- Can these bottlenecks be alleviated with new designs (similar to the designs adopted for MPI)?

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

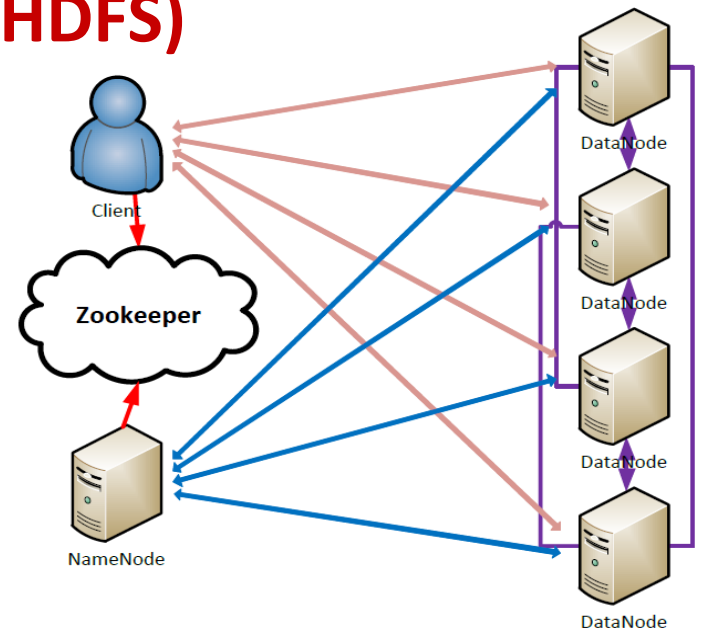
Big Data Processing with Hadoop

- The open-source implementation of MapReduce programming model for Big Data Analytics
- http://hadoop.apache.org/
- Framework includes: MapReduce, HDFS, and HBase
- Underlying Hadoop Distributed File System (HDFS) used by MapReduce and HBase
- Model scales but high amount of communication during intermediate phases

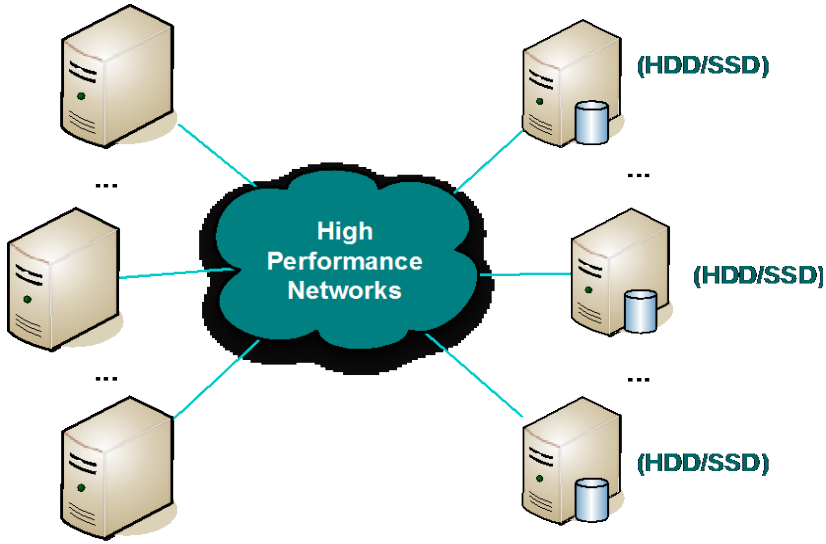


Hadoop Distributed File System (HDFS)

- Primary storage of Hadoop; highly reliable and fault-tolerant
- Adopted by many reputed organizations
  - eg:- Facebook, Yahoo!
- NameNode: stores the file system namespace
- DataNode: stores data blocks
- Developed in Java for platform-independence and portability
- Uses sockets for communication!

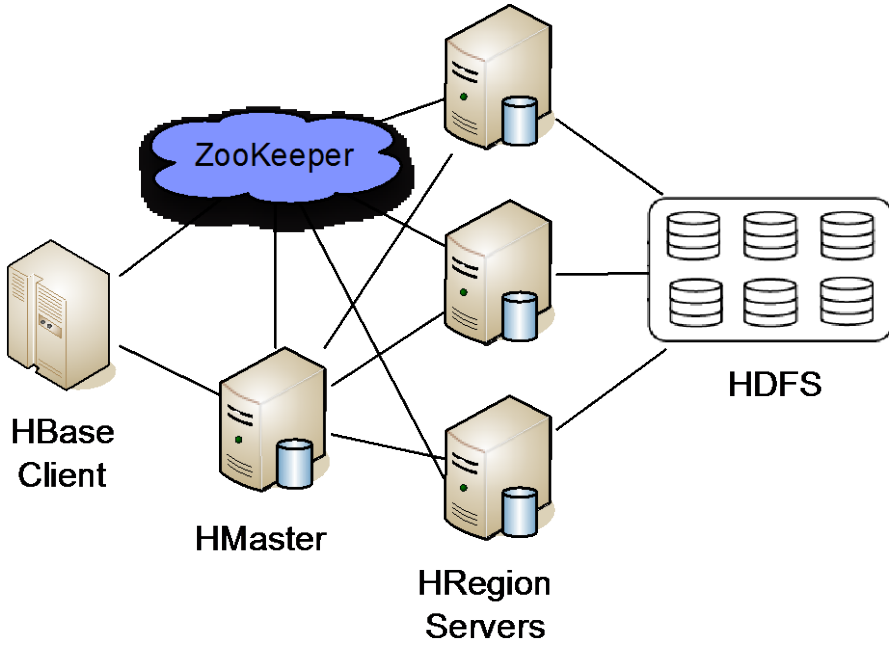


(HDFS Architecture)

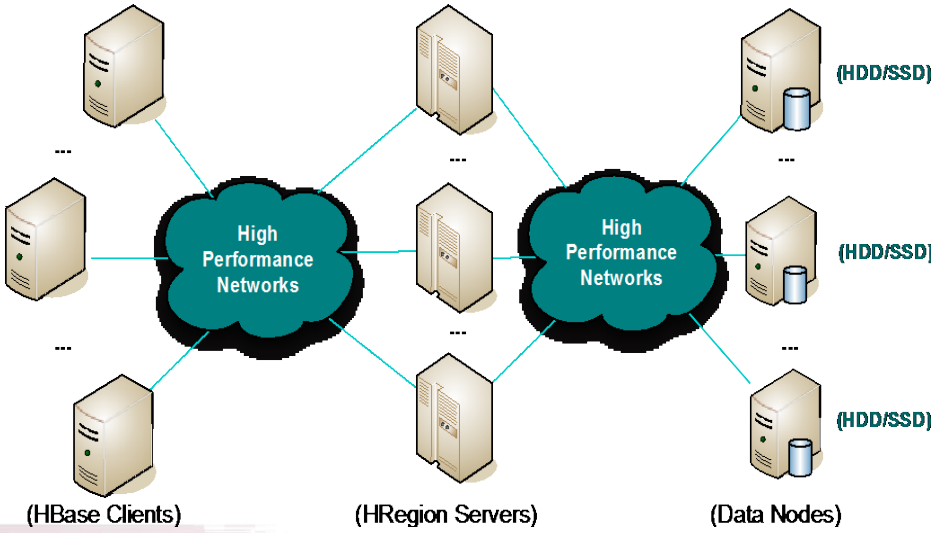


HBase

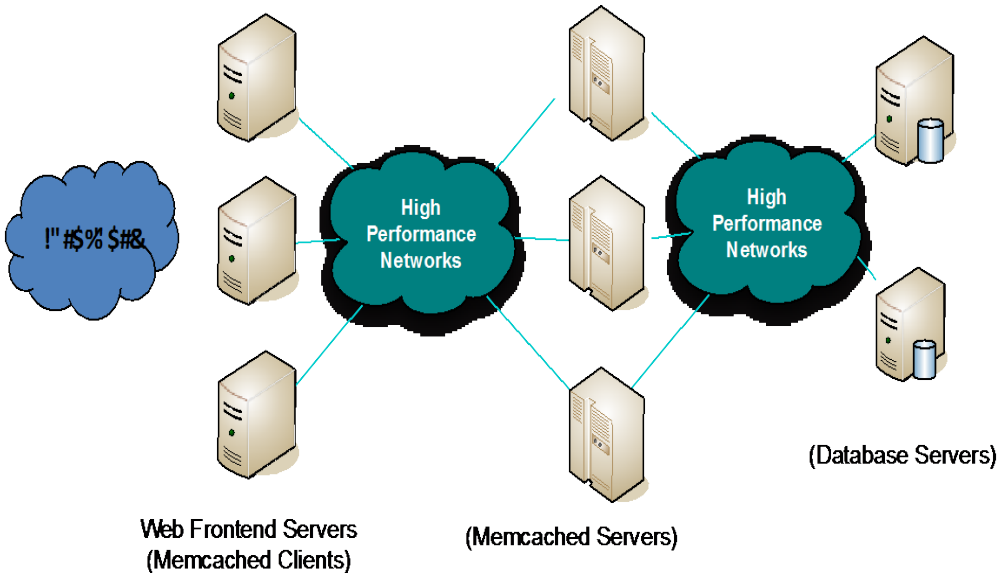
- Apache Hadoop Database (http://hbase.apache.org/)
- Semi-structured database, which is highly scalable
- Integral part of many datacenter applications
  - eg:- Facebook Social Inbox
- Developed in Java for platform-independence and portability
- Uses sockets for communication!

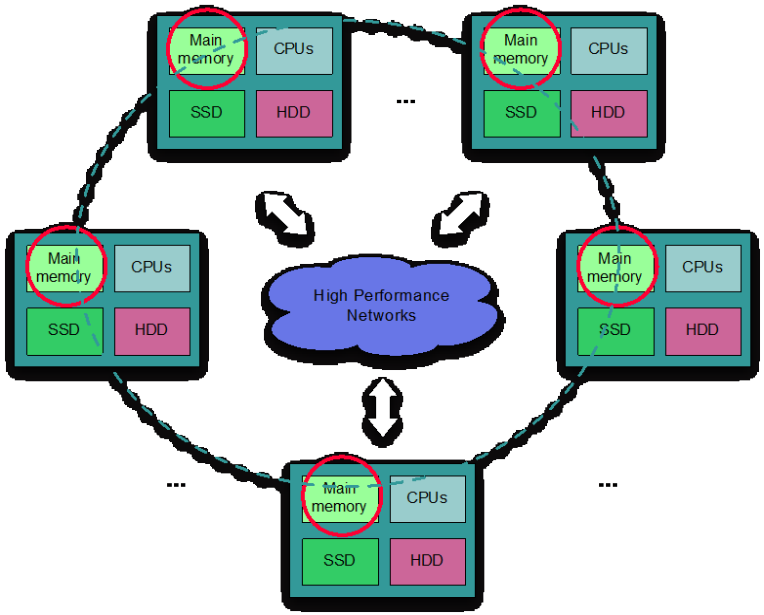


(HBase Architecture)



Memcached Architecture



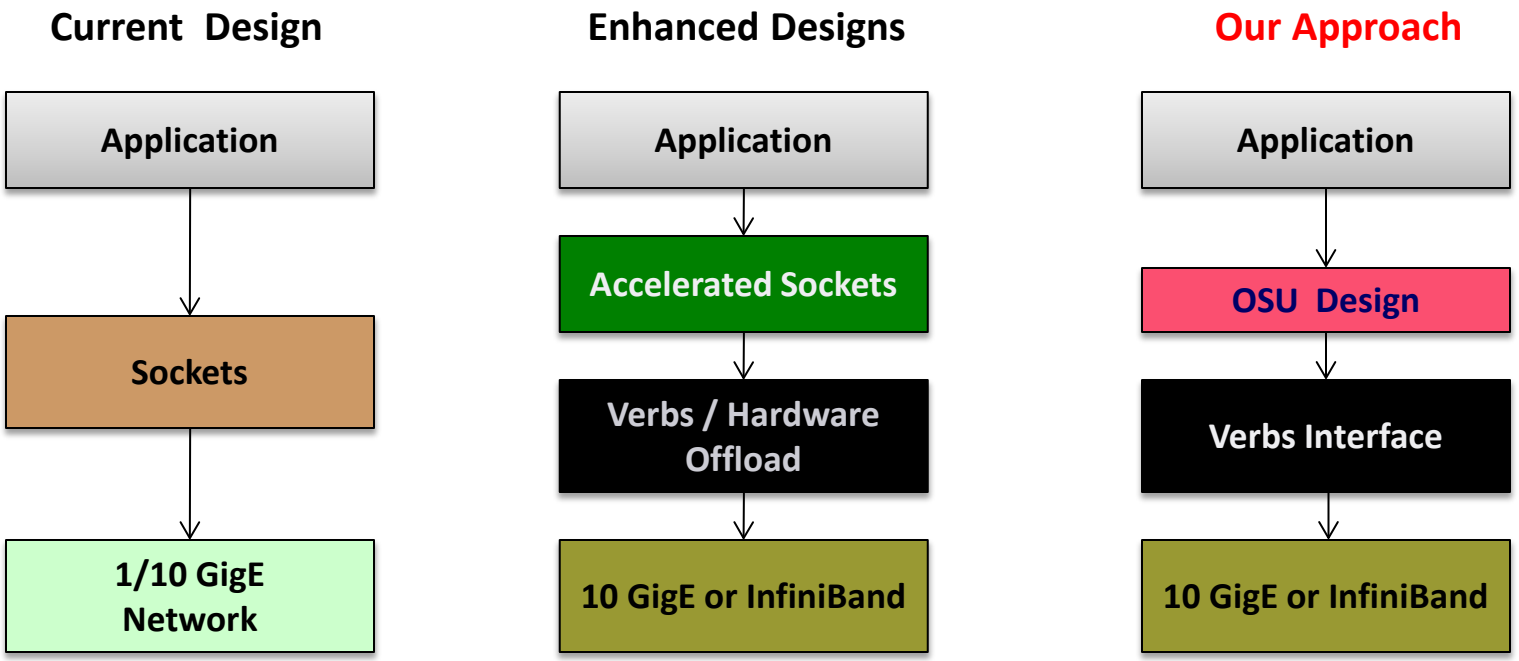


- Integral part of Web 2.0 architecture
- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS and HBase)
  - Memcached
- Conclusion and Q&A

Can Big Data Processing Systems be Designed with High-Performance Networks and Protocols?



- Sockets not designed for high-performance
  - Stream semantics often mismatch for upper layers (Memcached, HBase, Hadoop)
  - Zero-copy not available for non-blocking sockets

Interplay between Storage and Interconnect/Protocols

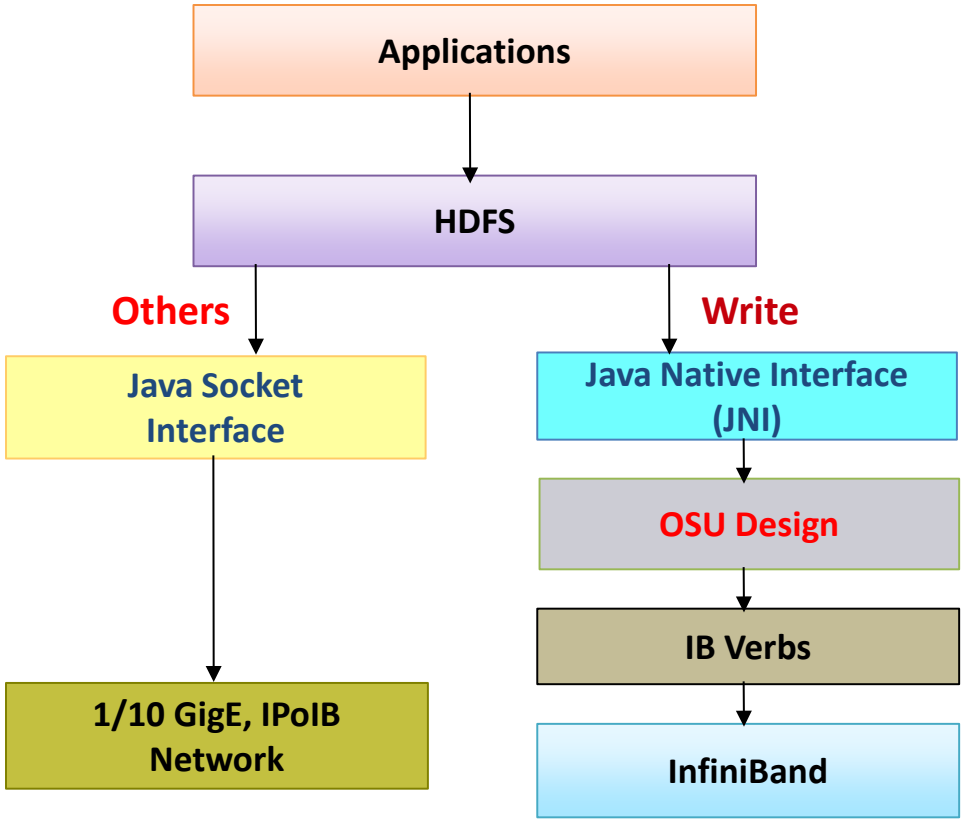
- Most of the current generation enterprise systems use the traditional hard disks
- Since hard disks are slower, high performance communication protocols may not have impact
- SSDs and other storage technologies are emerging
- Does it change the landscape?

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

HDFS-RDMA Design Overview

Enables high performance RDMA communication, while supporting traditional socket interface

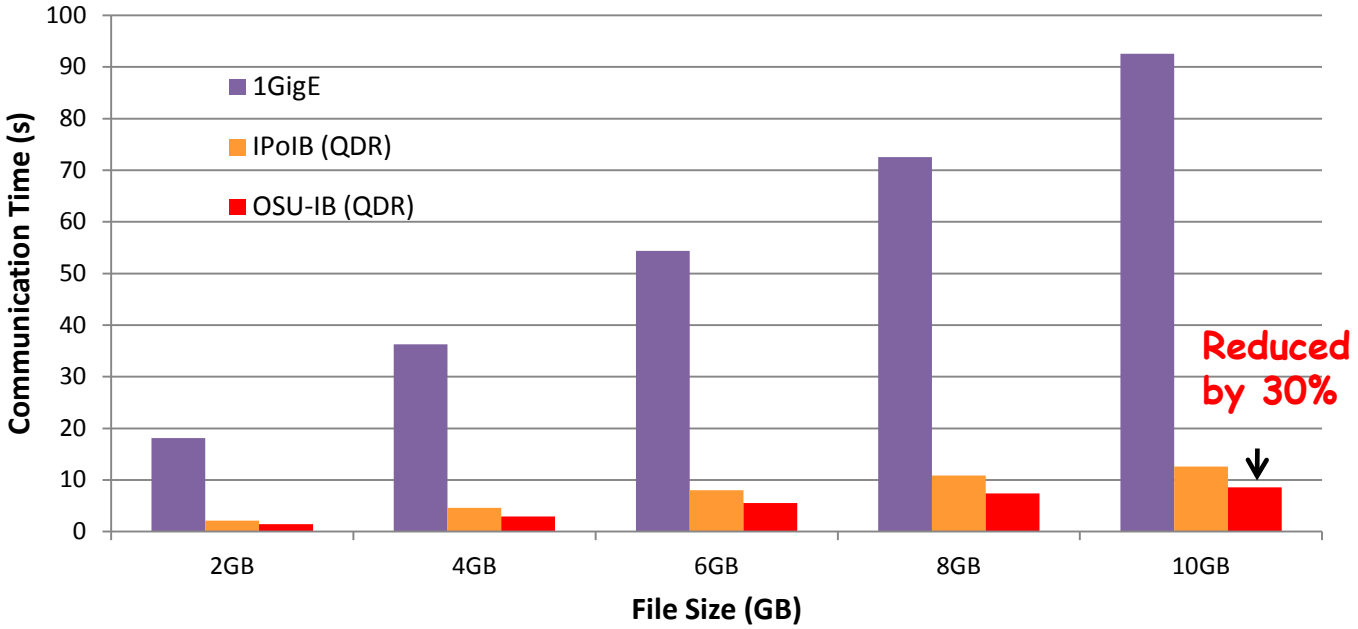


HDFS Write involves replication; more network intensive

HDFS Read is mostly node-local

- JNI Layer bridges Java based HDFS with communication library written in native code
- **Only the communication part of HDFS Write is modified; No change in HDFS architecture**

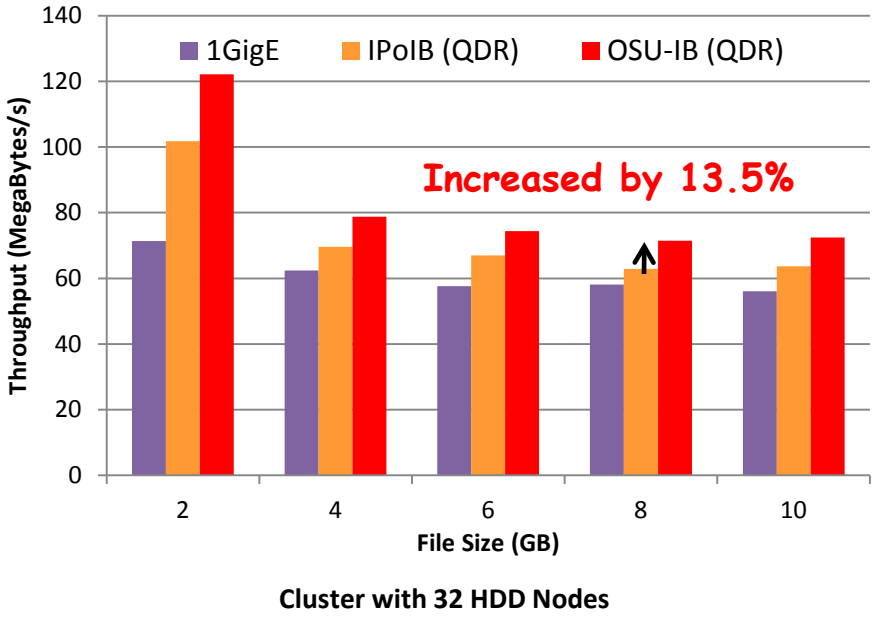
Communication Times in HDFS



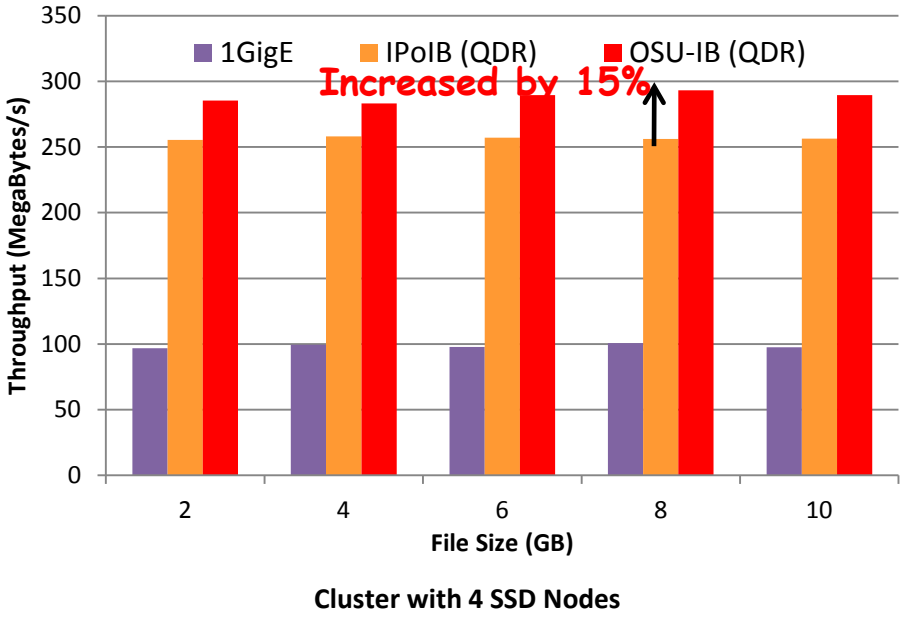
- Cluster B with 32 HDD DataNodes
  - **30%** improvement in communication time over IPoIB (32Gbps)
  - **87%** improvement in communication time over 1GigE
- Similar improvements are obtained for SSD DataNodes

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

Evaluations using TestDFSIO



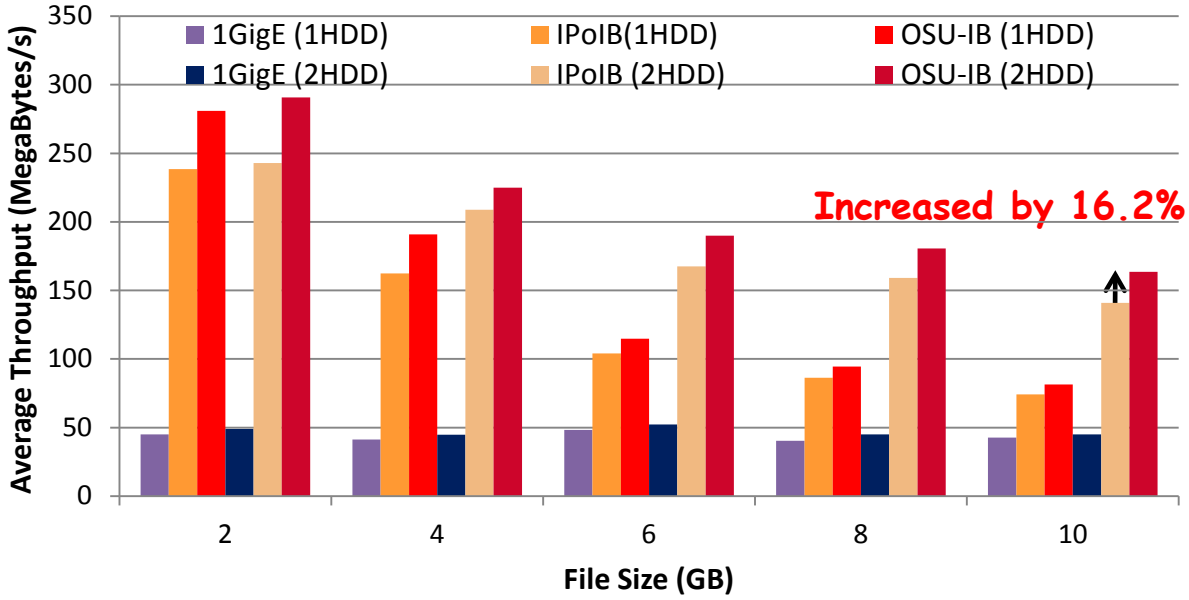
Cluster with 32 HDD Nodes



Cluster with 4 SSD Nodes

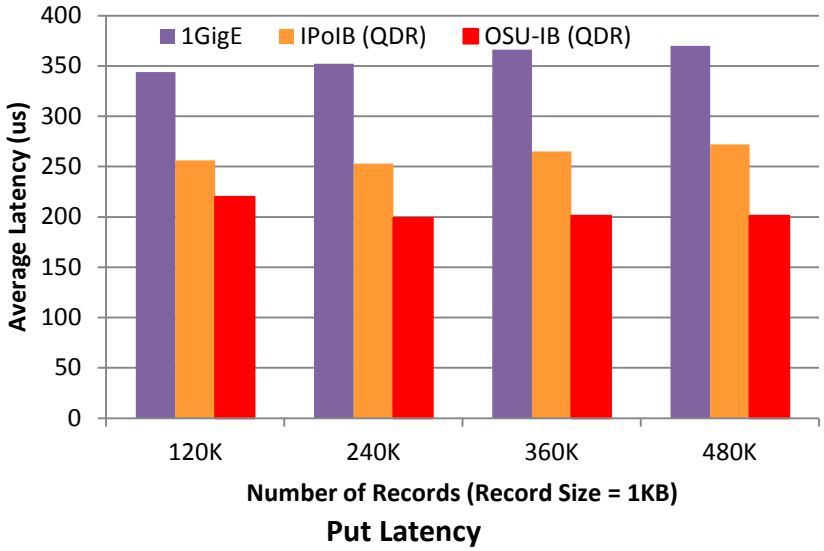
- Cluster with 32 HDD DataNodes
  - **13.5%** improvement over IPoIB (32Gbps) for 8GB file size
- Cluster with 4 SSD DataNodes
  - **15%** improvement over IPoIB (32Gbps) for 8GB file size

Evaluations using TestDFSIO

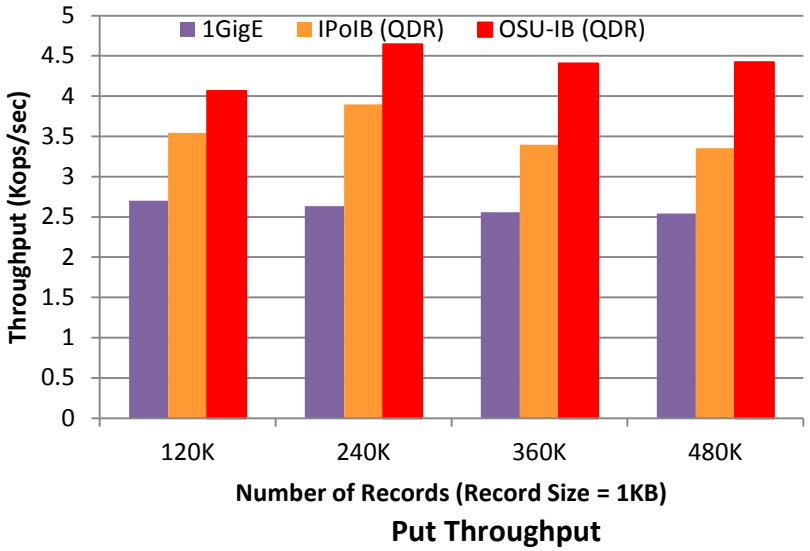


- Cluster with 4 DataNodes, **1 HDD** per node
  - **10%** improvement over IPoIB (32Gbps) for 10GB file size
- Cluster with 4 DataNodes, **2 HDD** per node
  - **16.2%** improvement over IPoIB (32Gbps) for 10GB file size
- 2 HDD vs 1 HDD
  - **2.01x** improvement for OSU-IB (32Gbps)
  - **1.8x** improvement for IPoIB (32Gbps)

Evaluations using YCSB (32 Region Servers: 100% Update)



Put Latency



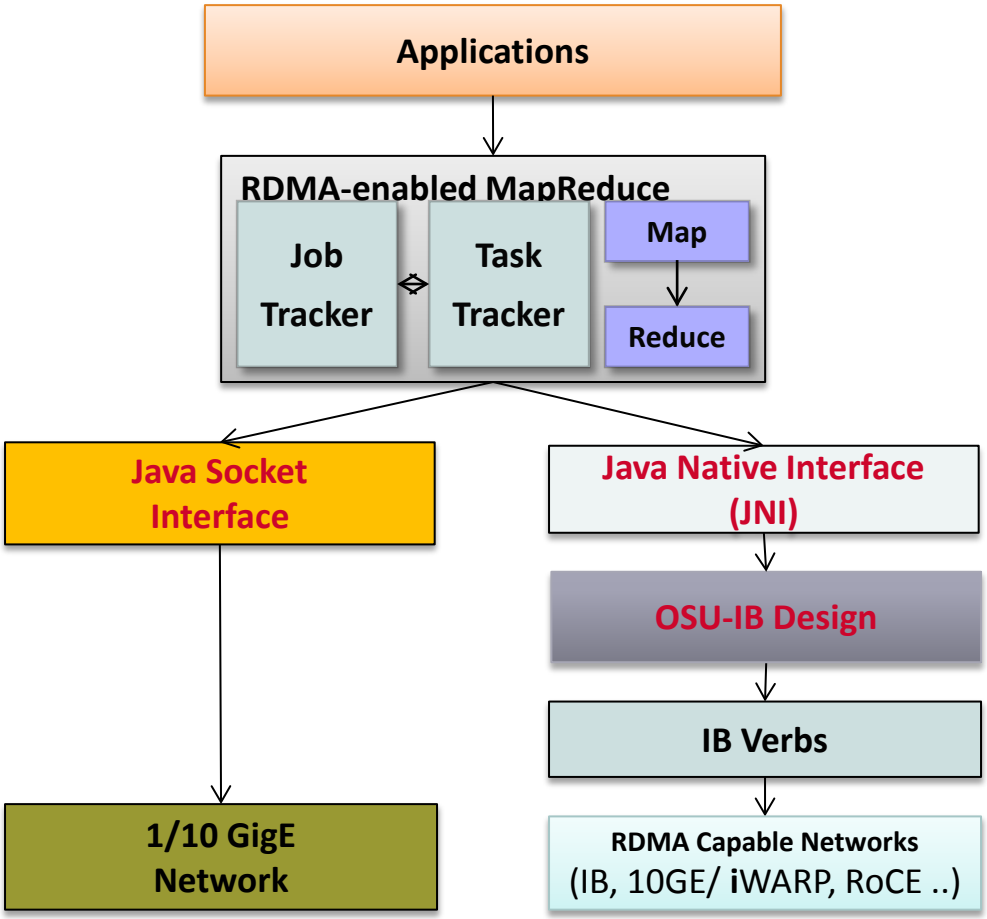
Put Throughput

- HBase using TCP/IP, running over HDFS-IB
- HBase ***Put*** latency for 480K records
  - **201** us **for OSU Design; 272 us for IPoIB (32Gbps)**
- HBase ***Put*** throughput for 480K records
  - **4.42** Kops/sec for OSU Design; **3.63** Kops/sec for IPoIB (32Gbps)
- **26%** improvement in average latency; **24%** improvement in throughput

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

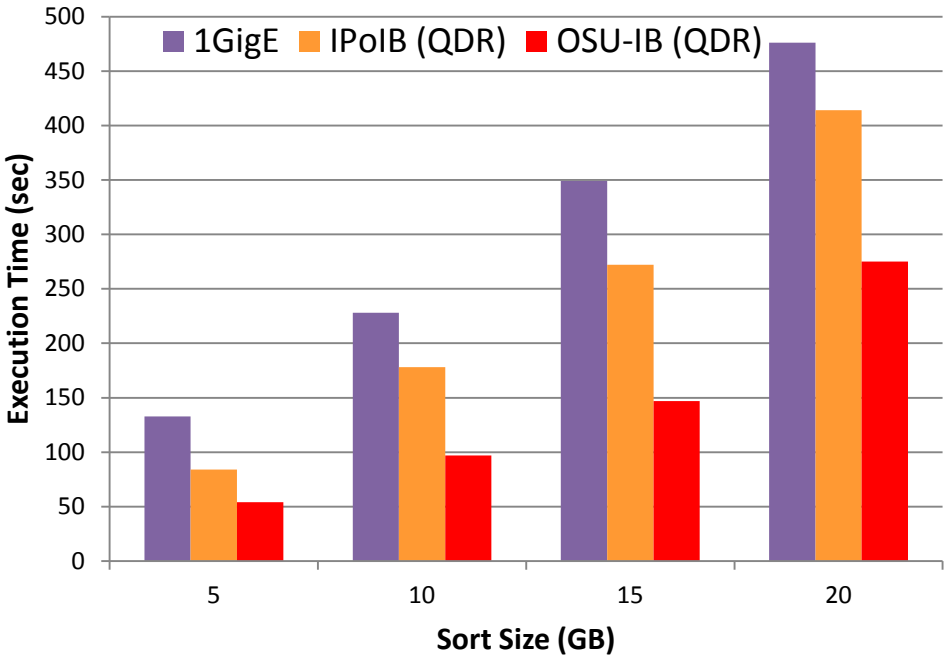
MapReduce-RDMA Design Overview

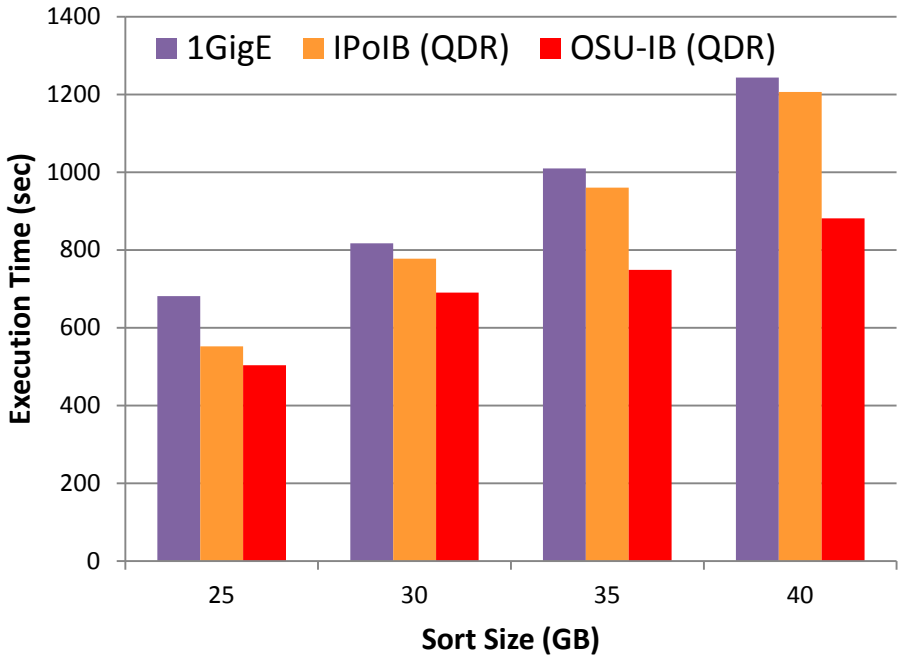


- Enables high performance RDMA communication, while supporting traditional socket interface

M.-W Rahman, N. S. Islam, X. Lu, J. Jose, H. Subramon, H. Wang and D. K. Panda, High-Performance RDMA-based Design of Hadoop MapReduce over InfiniBand, Int'l Workshop on High Performance Data Intensive Computing (HPDIC), held in conjunction with Int'l Parallel and Distributed Processing Symposium (IPDPS '13), May 2013.

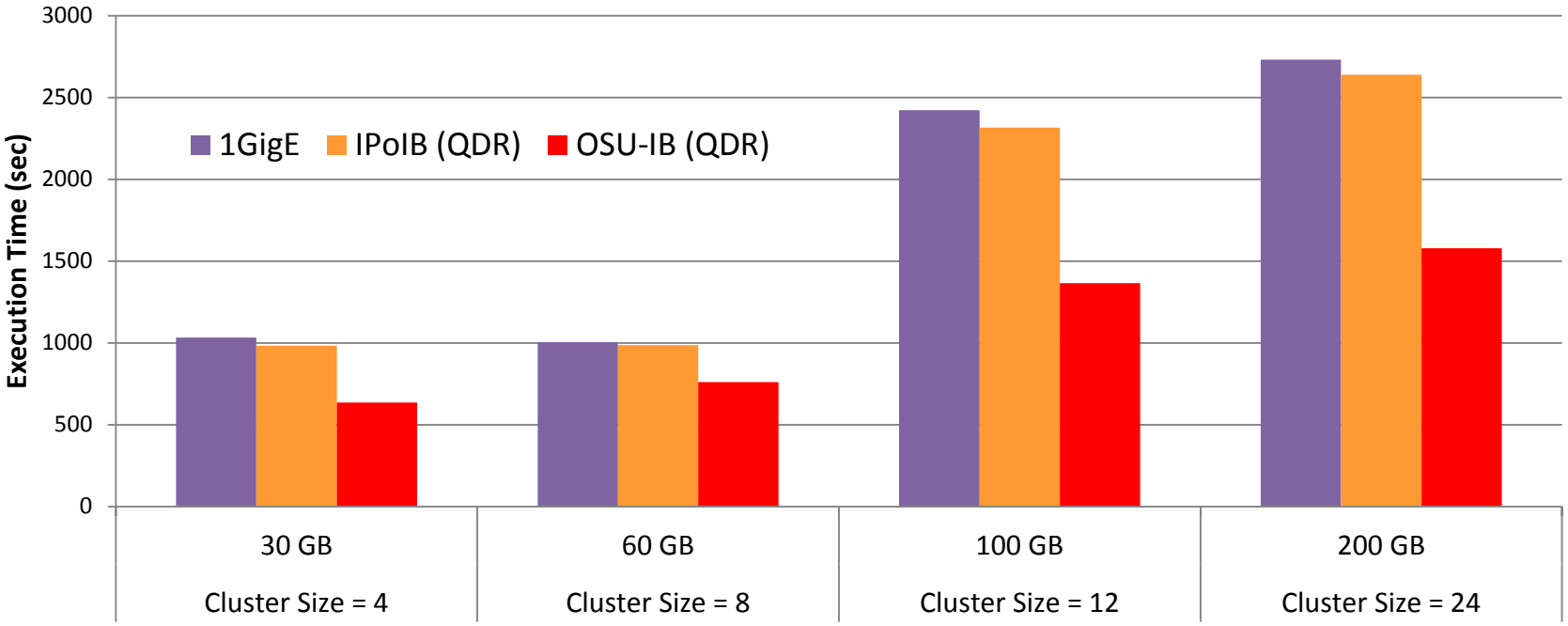
Evaluations using Sort





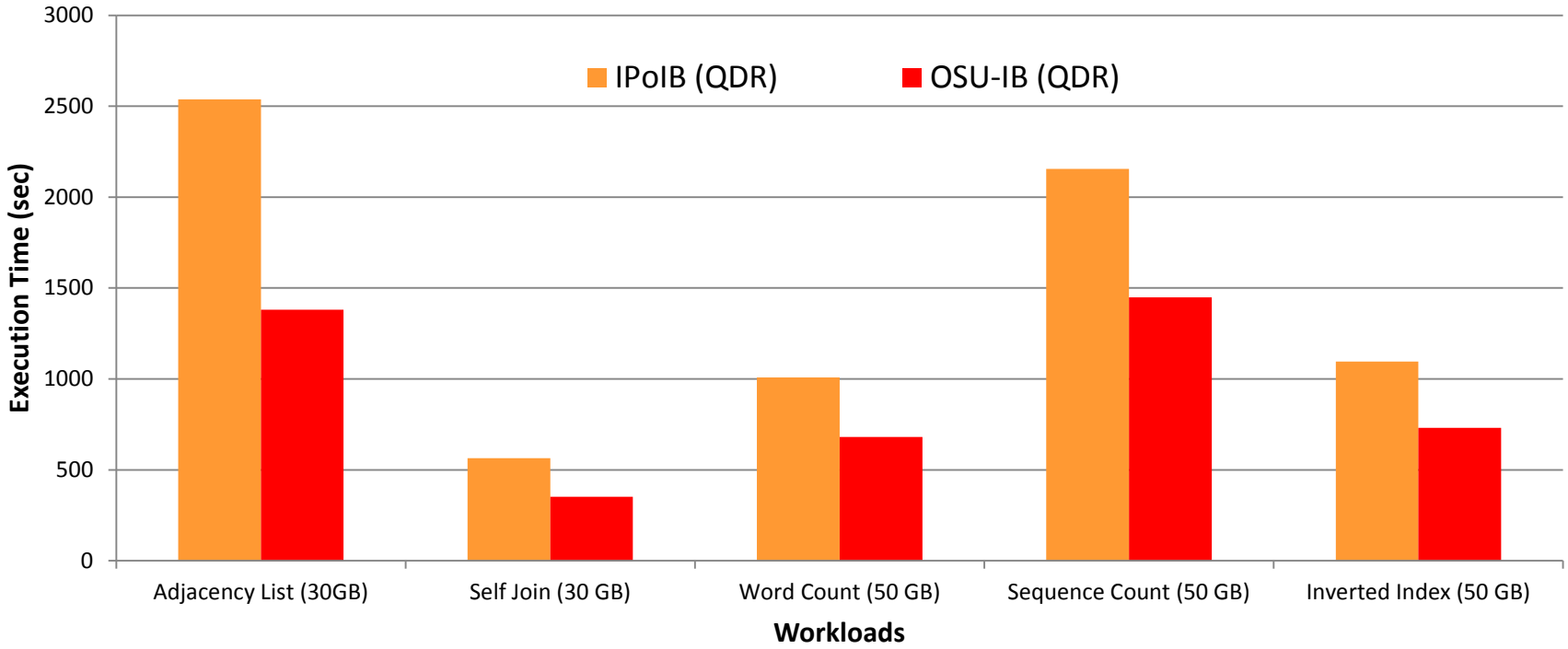
- For 5-20 GB Sort, 4-node cluster with a single SSD per node
- For 25-40 GB Sort, 8-node cluster with a single HDD per node
- 46% improvement over IPoIB (32 Gbps) for 15 GB Sort
- 27% improvement over IPoIB (32 Gbps) for 40 GB Sort

Evaluations using TeraSort



- Cluster Size 4 and 8 have 24 GB RAM in each node, Cluster Size 12 and 24 have 12 GB RAM in each node, all the nodes have single HDD
- 41% improvement over IPoIB (32Gbps) for 100 GB Terasort

Evaluations using PUMA Workload

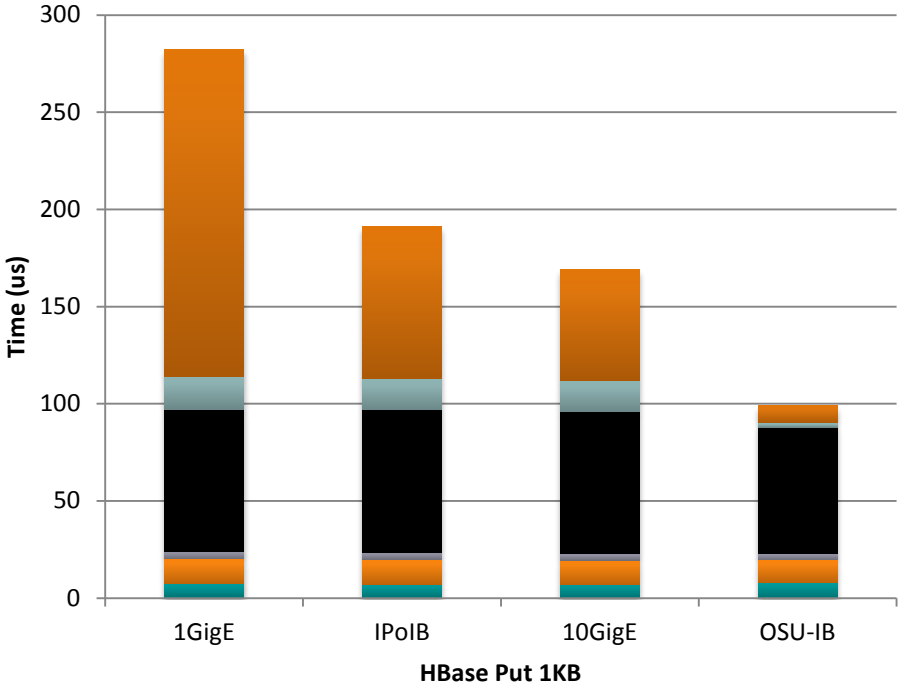


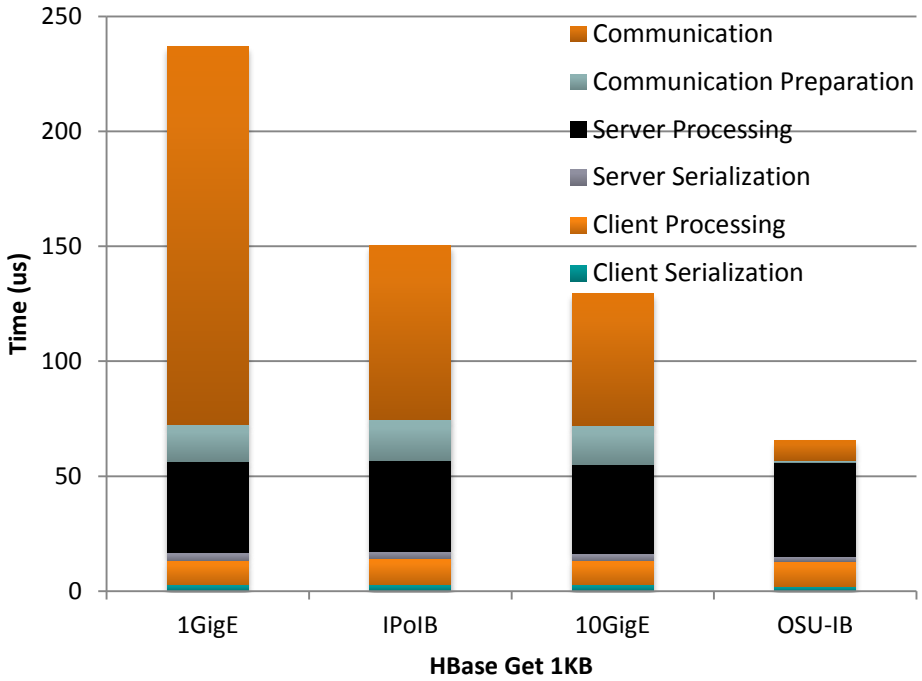
- The DataSet for these workloads are taken from PUMA (Purdue MapReduce Benchmark Suite)
- 46% improvement in Adjacency List over IPoIB (32Gbps) for 30 GB data size
- 33% improvement in Sequence Count over IPoIB (32 Gbps) for 50 GB data size

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

HBase Put/Get – Detailed Analysis

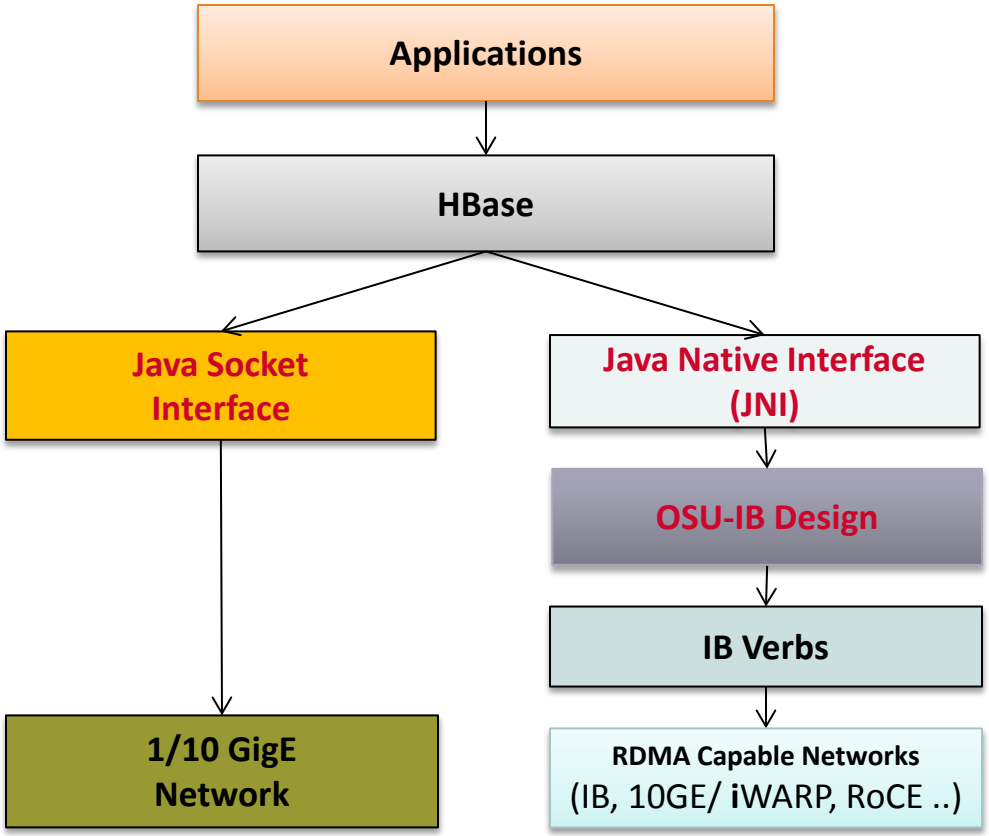




- HBase 1KB Put
  - Communication Time – 8.9 us
  - A factor of 6X improvement over 10GE for communication time
- HBase 1KB Get
  - Communication Time – 8.9 us
  - A factor of 6X improvement over 10GE for communication time

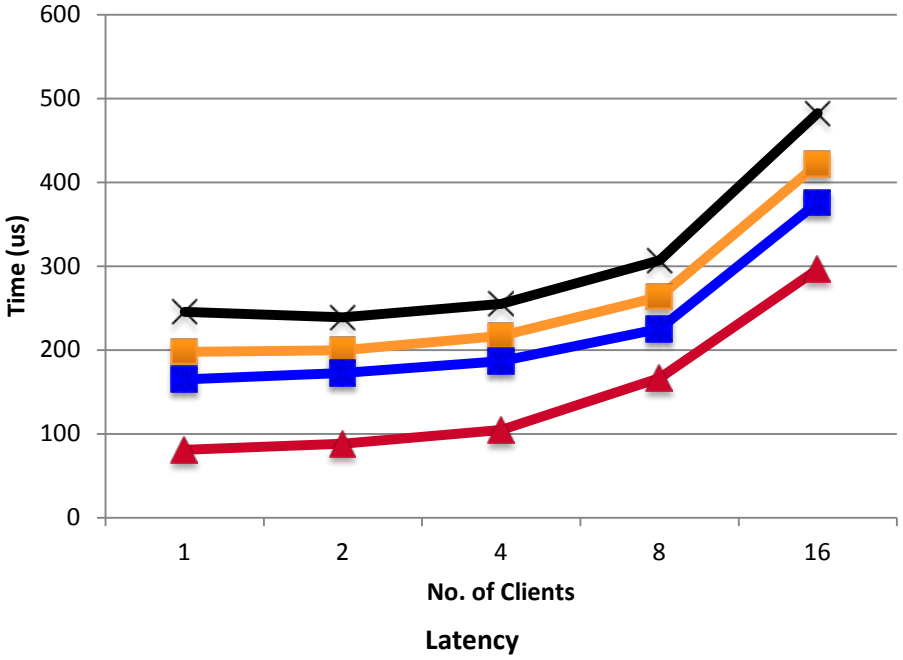
W. Rahman, J. Huang, J. Jose, X. Ouyang, H. Wang, N. Islam, H. Subramoni, Chet Murthy and D. K. Panda, Understanding the Communication Characteristics in HBase: What are the Fundamental Bottlenecks?, ISPASS'12

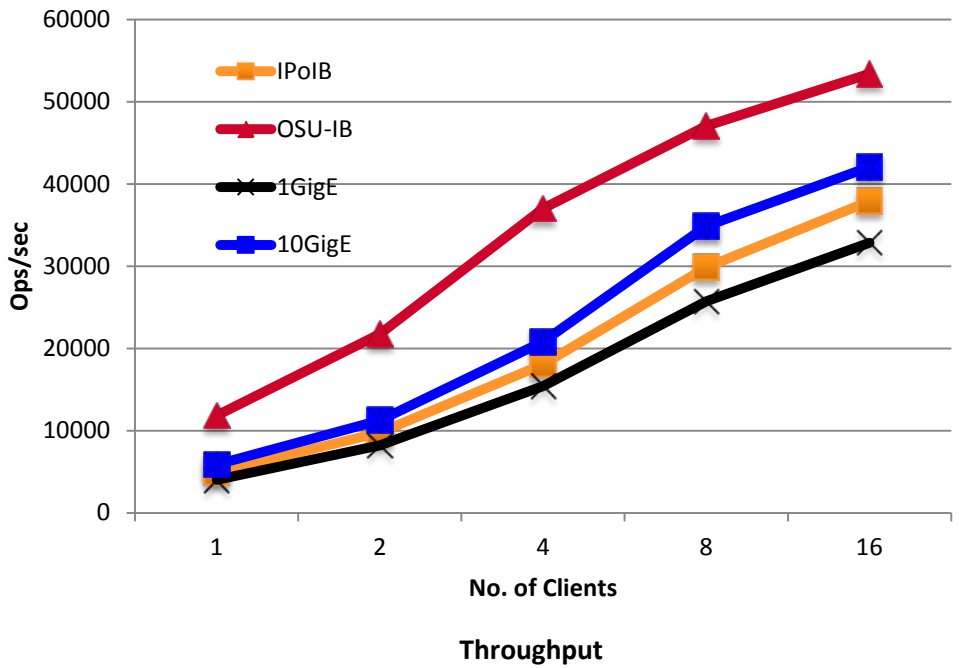
HBase-RDMA Design Overview



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

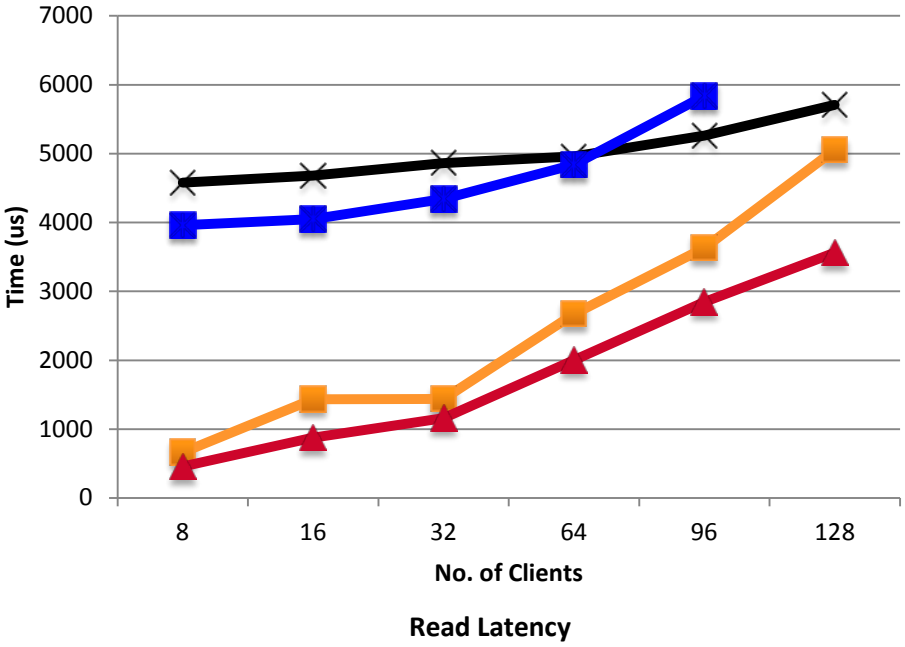
HBase Single Server-Multi-Client Results



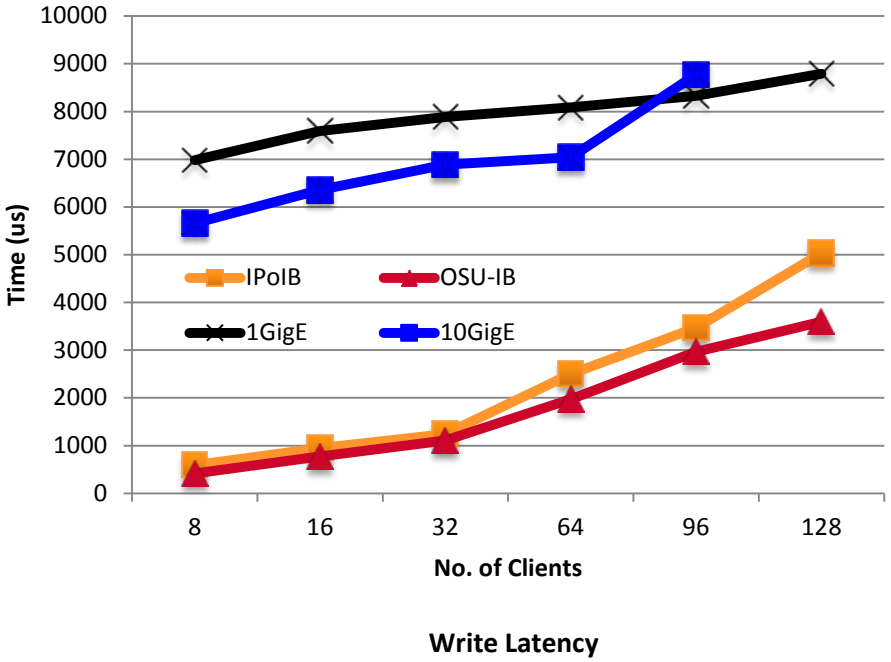


- HBase Get latency
  - 4 clients: 104.5 us; 16 clients: 296.1 us
- HBase Get throughput
  - 4 clients: 37.01 Kops/sec; 16 clients: 53.4 Kops/sec
- 27% improvement in throughput for 16 clients over 10GE

HBase – YCSB Read-Write Workload



Read Latency



Write Latency

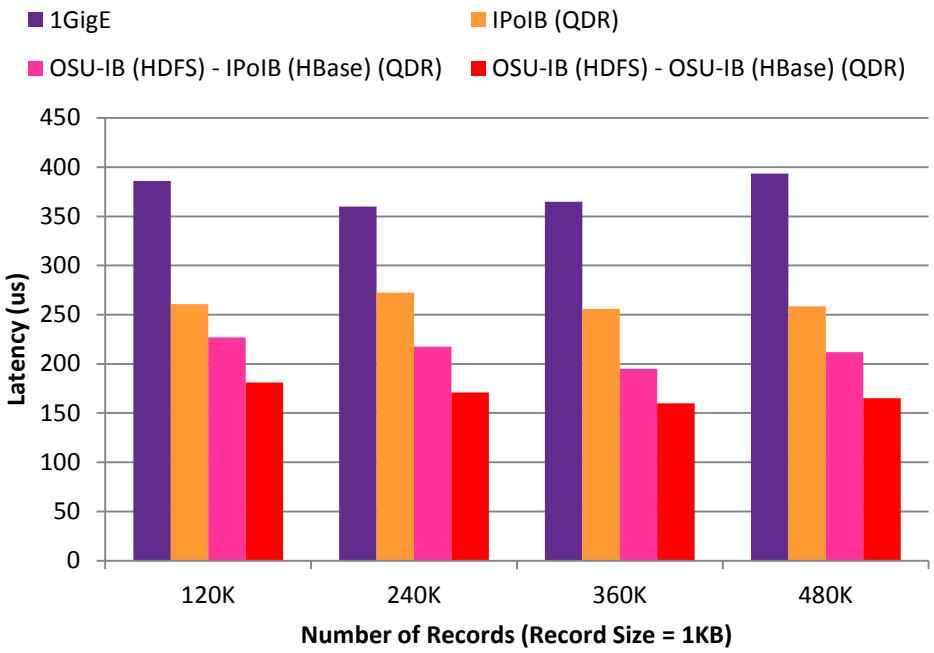
- HBase Get latency (Yahoo! Cloud Service Benchmark)
  - 64 clients: 2.0 ms; 128 Clients: 3.5 ms
  - 42% improvement over IPoIB for 128 clients
- HBase Get latency
  - 64 clients: 1.9 ms; 128 Clients: 3.5 ms
  - 40% improvement over IPoIB for 128 clients

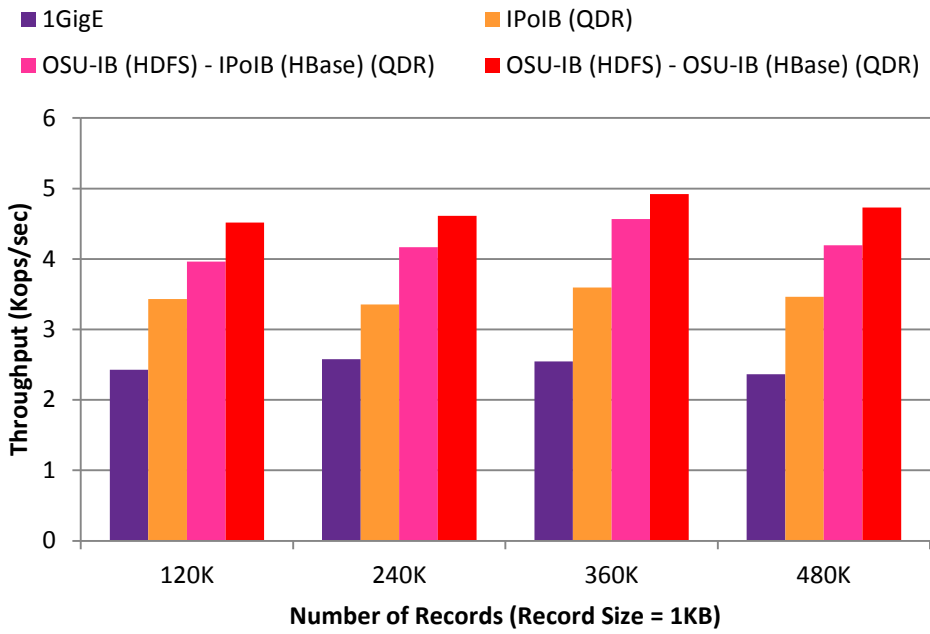
J. Huang, X. Ouyang, J. Jose, W. Rahman, H. Wang, M. Luo, H. Subramoni, Chet Murthy and D. K. Panda, High-Performance Design of HBase with RDMA over InfiniBand, IPDPS'12

Presentation Outline

- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

HDFS and HBase Integration over IB (OSU-IB)



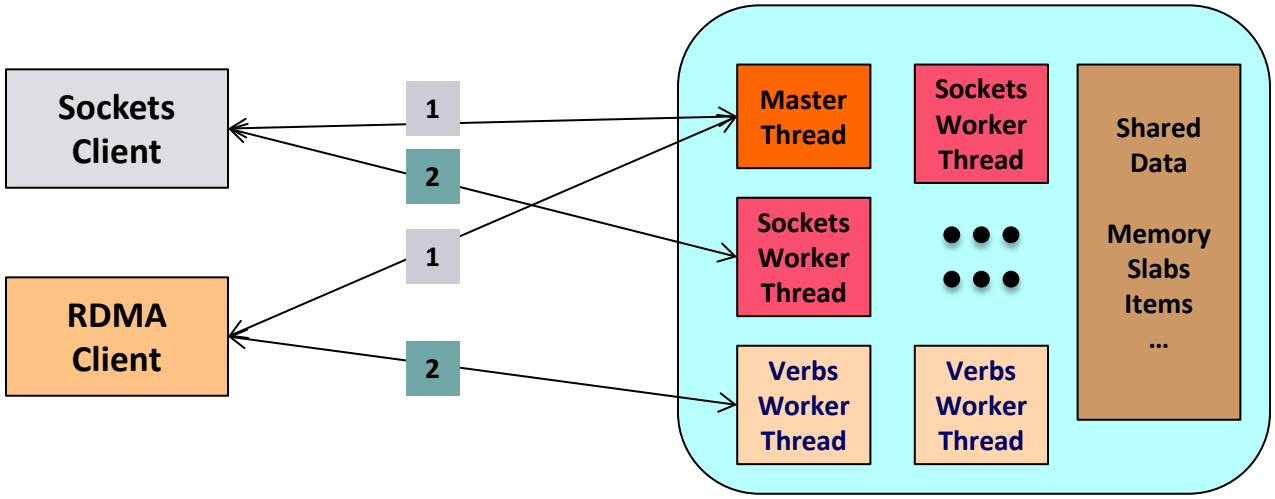


- YCSB Evaluation with 4 RegionServers (100% update)
- HBase Put Latency and Throughput for 360K records
  - 37% improvement over IPoIB (32Gbps)
  - 18% improvement over OSU-IB HDFS only

Presentation Outline

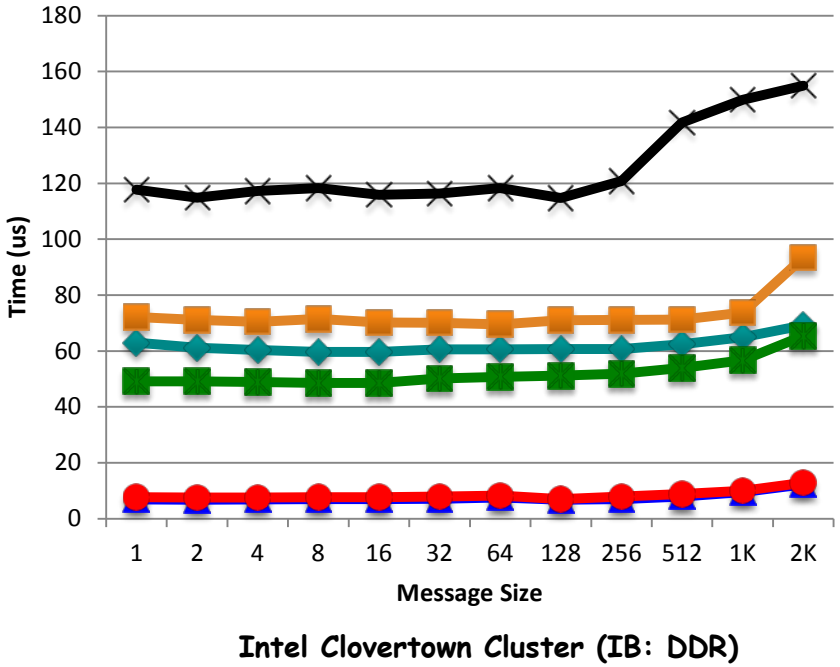
- Overview of Hadoop (HDFS, MapReduce and HBase) and Memcached
- Challenges in Accelerating Enterprise Middleware
- Designs and Case Studies
  - Hadoop
    - HDFS
    - MapReduce
    - HBase
    - Combination (HDFS + HBase)
  - Memcached
- Conclusion and Q&A

Memcached-RDMA Design

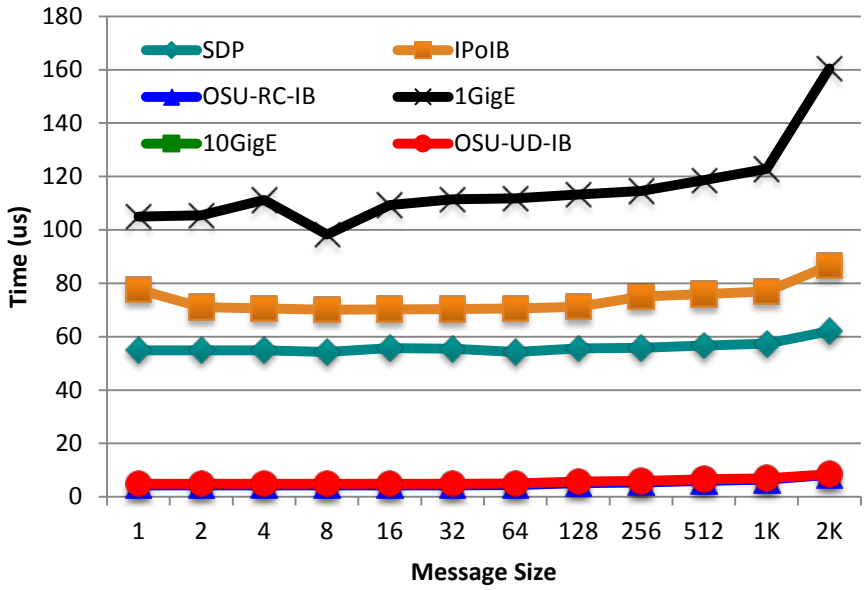


- Memcached applications need not be modified; uses verbs interface if available
- Memcached Server can serve both socket and verbs clients simultaneously
- Native IB-verbs-level Design and evaluation with
  - Memcached Server: 1.4.9
  - Memcached Client: (libmemcached) 0.52

Memcached Get Latency (Small Message)



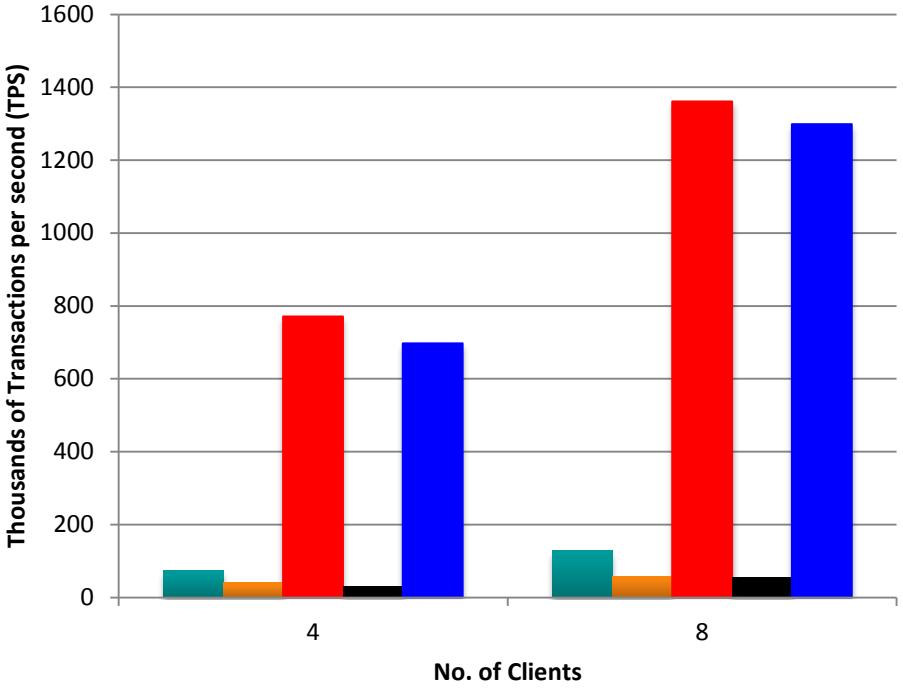
Intel Clovertown Cluster (IB: DDR)

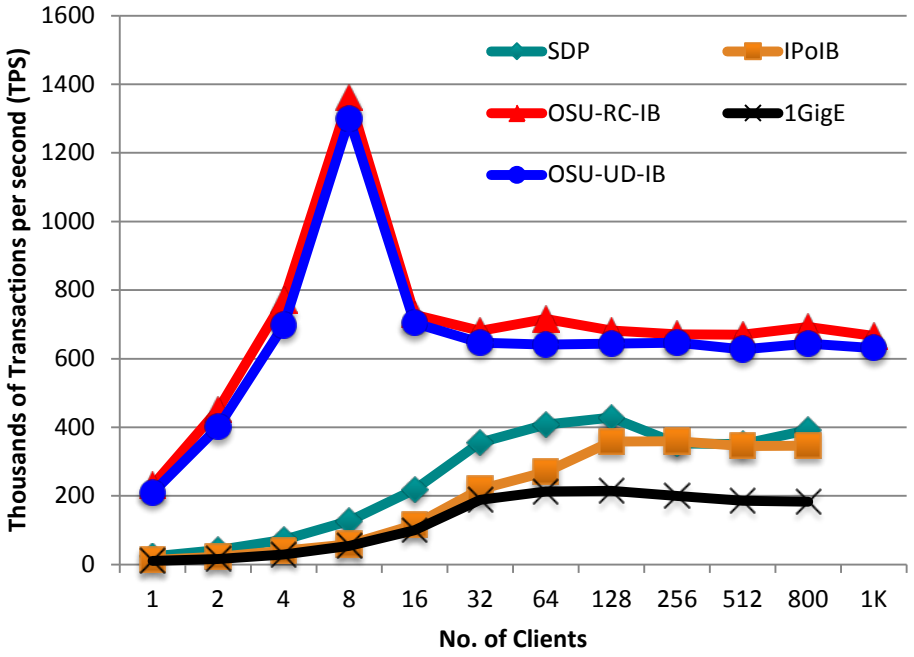


Intel Westmere Cluster (IB: QDR)

- Memcached Get latency
  - 4 bytes RC/UD – DDR: 6.82/7.55 us; QDR: 4.28/4.86 us
  - 2K bytes RC/UD – DDR: 12.31/12.78 us; QDR: 8.19/8.46 us
- Almost factor of *four* improvement over 10GE (TOE) for 2K bytes on the DDR cluster

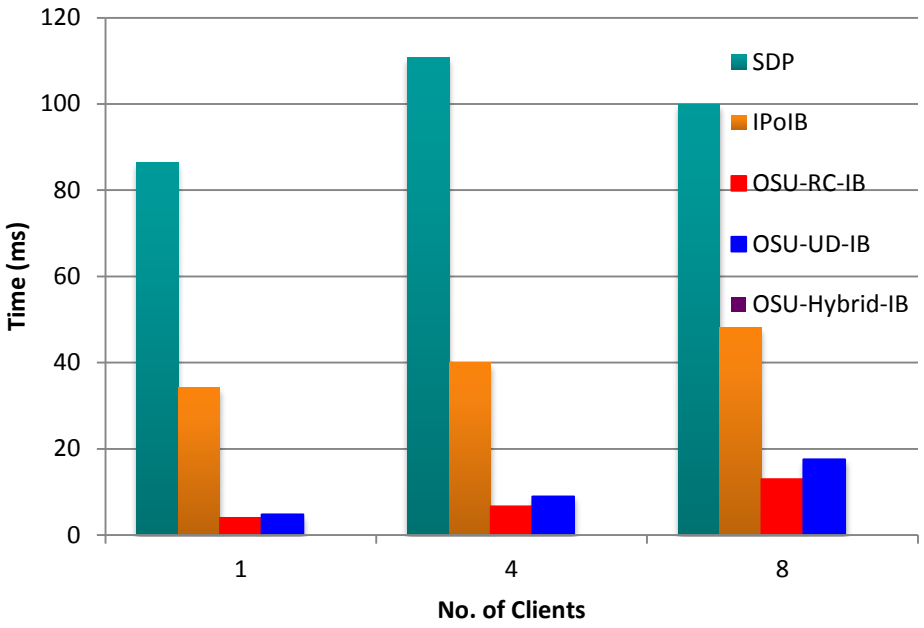
Memcached Get TPS (4byte)

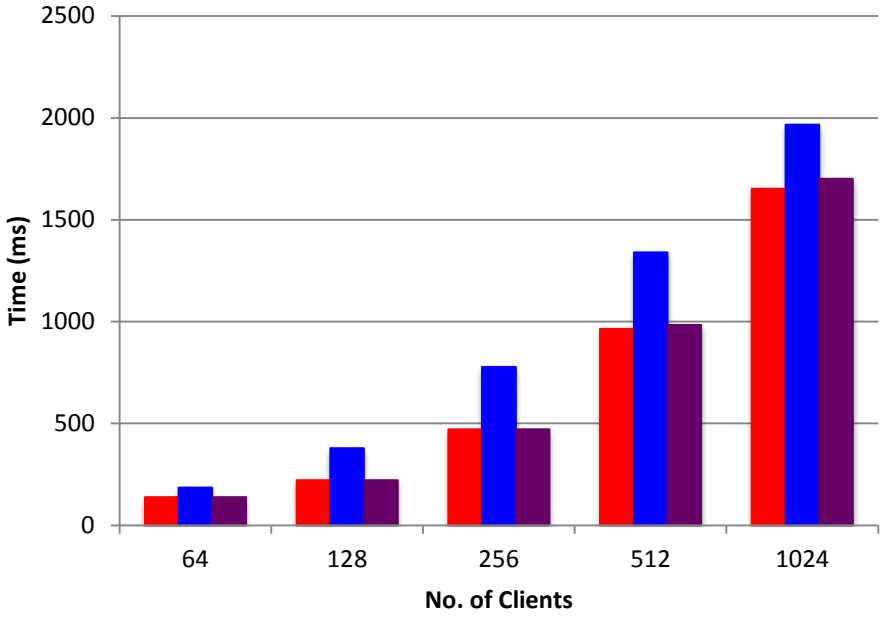




- Memcached Get transactions per second for 4 bytes
  - On IB QDR 1.4M/s (RC), 1.3 M/s (UD) for 8 clients
- Significant improvement with native IB QDR compared to SDP and IPoIB

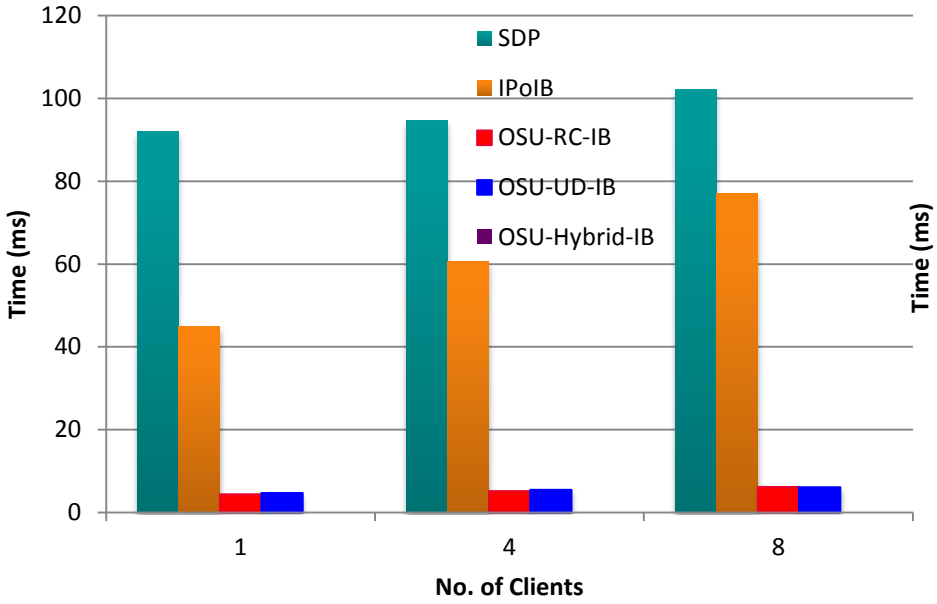
Application Level Evaluation – Olio Benchmark

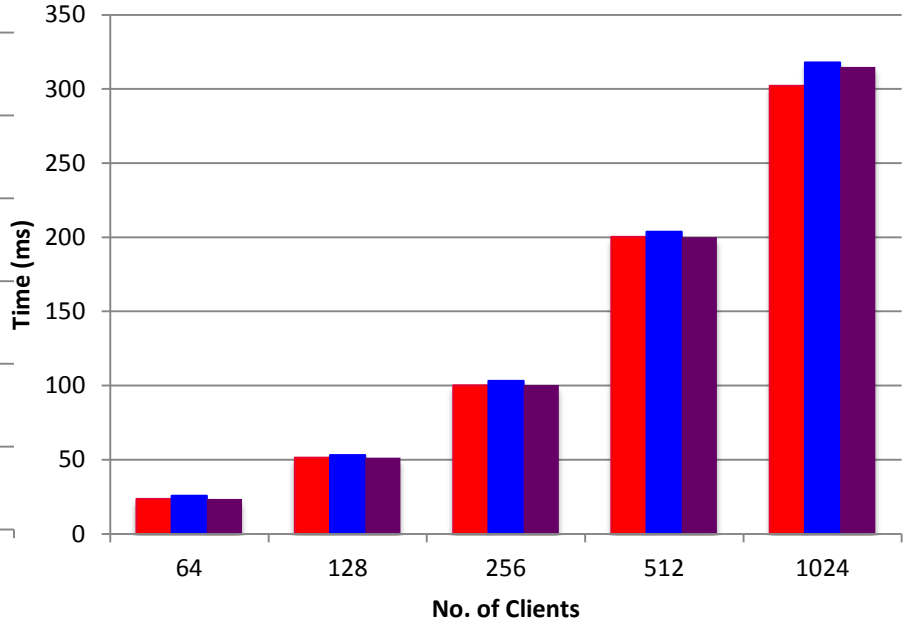




- Olio Benchmark
  - RC – 1.6 sec, UD – 1.9 sec, Hybrid – 1.7 sec for 1024 clients
- 4X times better than IPoIB for 8 clients
- Hybrid design achieves comparable performance to that of pure RC design

Application Level Evaluation – Real Application Workloads





- Real Application Workload
  - RC – 302 ms, UD – 318 ms, Hybrid – 314 ms for 1024 clients
- 12X times better than IPoIB for 8 clients
- Hybrid design achieves comparable performance to that of pure RC design

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

Concluding Remarks

- Presented initial designs to take advantage of InfiniBand/RDMA for HDFS, MapReduce, HBase and Memcached
- Results are promising
- Working on Integrated designs of all components
- Many other open issues need to be solved including design changes at the upper layers
- Will enable BigData community to take advantage of modern clusters and Hadoop middleware to carry out their analytics in a fast and scalable manner

Web Pointers

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

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

MVAPICH Web Page

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

panda@cse.ohio-state.edu