Accelerating Data Management and Processing on Modern Clusters with RDMA-Enabled Interconnects

Keynote Talk at ADMS 2014

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 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
4V Characteristics of Big Data

- Commonly accepted 3V’s of Big Data
  - **Volume, Velocity, Variety**
    

- 4/5V’s of Big Data – 3V + *Veracity, *Value

http://api.ning.com/files/tRHkwQN7s-Xz5cxylXG004GLGJdjoPd6bVfVBwvgu*F5MwDDUCiHHdmBW-JTEz0cfJjGurJucBTkIUNdL3jcZT8IPfNWfN9/dv1.jpg
Velocity of Big Data – How Much Data Is Generated Every Minute on the Internet?

The global Internet population grew 6.59% from 2010 to 2011 and now represents 2.1 Billion People.

http://www.domo.com/blog/2012/06/how-much-data-is-created-every-minute
Data Management and Processing in Modern Datacenters

- Substantial impact on designing and utilizing modern 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
Overview of Web 2.0 Architecture and Memcached

• Three-layer architecture of Web 2.0
  – Web Servers, Memcached Servers, Database Servers

• Memcached is a core component of Web 2.0 architecture
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
HBase Overview

- 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!
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!
Disk Operations

- 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 shuffle phase uses HTTP over Java Sockets
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
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
  - RDMA-based Memcached
  - Case study with OLTP
  - SSD-assisted hybrid Memcached
  - RDMA-based HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
High-End Computing (HEC): PetaFlop to ExaFlop

Projected Performance Development

Expected to have an ExaFlop system in 2020-2022!

100 PFlops in 2015

1 EFlops in 2018?

Expected to have an ExaFlop system in 2020-2022!
Trends for Commodity Computing Clusters in the Top 500 List (http://www.top500.org)
High End Computing (HEC)

- High End Computing (HEC) grows dramatically
  - High Performance Computing
  - Big Data Computing

- Technology Advancement
  - Multi-core/many-core technologies and accelerators
  - Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
  - Solid State Drives (SSDs) and Non-Volatile Random-Access Memory (NVRAM)
  - Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)

Tianhe – 2 (1)  Titan (2)  Stampede (6)  Tianhe – 1A (10)
Overview of High Performance Interconnects

- High-Performance Computing (HPC) has adopted advanced interconnects and protocols
  - InfiniBand
  - 10 Gigabit Ethernet/iWARP
  - RDMA over Converged Enhanced Ethernet (RoCE)
- Very Good Performance
  - Low latency (few micro seconds)
  - High Bandwidth (100 Gb/s with dual FDR InfiniBand)
  - Low CPU overhead (5-10%)
- OpenFabrics software stack (www.openfabrics.org) with IB, iWARP and RoCE interfaces are driving HPC systems
- Many such systems in Top500 list
All interconnects and protocols in OpenFabrics Stack

Application / Middleware Interface

- Sockets
- Verbs

Kernel Space

- TCP/IP
- IPoIB
- Hardware Offload
- User Space

Application / Middleware

- Ethernet Adapter
- InfiniBand Adapter
- Ethernet Switch
- User Space

Protocol

- Ethernet
- InfiniBand
- TCP/IP
- IPoIB
- iWARP
- RoCE
- IB Native

Adapter

- Ethernet Adapter
- InfiniBand Adapter
- Ethernet Switch
- IPoIB
- 1/10/40 GigE

Switch

- Ethernet Switch
- InfiniBand Switch
- 10/40 GigE-TOE
Trends of Networking Technologies in TOP500 Systems

Percentage share of InfiniBand is steadily increasing

Interconnect Family – Systems Share
Large-scale InfiniBand Installations

- 223 IB Clusters (44.3%) in the June 2014 Top500 list (http://www.top500.org)
- Installations in the Top 50 (25 systems):

<table>
<thead>
<tr>
<th>Installations</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>519,640 cores (Stampede) at TACC (7th)</td>
<td>120,640 cores (Nebulae) at China/NCS (28th)</td>
</tr>
<tr>
<td>62,640 cores (HPC2) in Italy (11th)</td>
<td>72,288 cores (Yellowstone) at NCAR (29th)</td>
</tr>
<tr>
<td>147,456 cores (Super MUC) in Germany (12th)</td>
<td>70,560 cores (Helios) at Japan/IFERC (30th)</td>
</tr>
<tr>
<td>76,032 cores (Tsubame 2.5) at Japan/GSIC (13th)</td>
<td>138,368 cores (Tera-100) at France/CEA (35th)</td>
</tr>
<tr>
<td>194,616 cores (Cascade) at PNNL (15th)</td>
<td>222,072 cores (QUARTETTO) in Japan (37th)</td>
</tr>
<tr>
<td>110,400 cores (Pangea) at France/Total (16th)</td>
<td>53,504 cores (PRIMERGY) in Australia (38th)</td>
</tr>
<tr>
<td>96,192 cores (Pleiades) at NASA/Ames (21st)</td>
<td>77,520 cores (Conte) at Purdue University (39th)</td>
</tr>
<tr>
<td>73,584 cores (Spirit) at USA/Air Force (24th)</td>
<td>44,520 cores (Spruce A) at AWE in UK (40th)</td>
</tr>
<tr>
<td>77,184 cores (Curie thin nodes) at France/CEA (26th)</td>
<td>48,896 cores (MareNostrum) at Spain/BSC (41st)</td>
</tr>
<tr>
<td>65,320-cores, iDataPlex DX360M4 at Germany/Max-Planck (27th)</td>
<td>and many more!</td>
</tr>
</tbody>
</table>
Open Standard InfiniBand Networking Technology

• Introduced in Oct 2000
• High Performance Data Transfer
  – Interprocessor communication and I/O
  – Low latency (<1.0 microsec), High bandwidth (up to 12.5 GigaBytes/sec -> 100Gbps), and low CPU utilization (5-10%)
• Flexibility for LAN and WAN communication
• Multiple Transport Services
  – Reliable Connection (RC), Unreliable Connection (UC), Reliable Datagram (RD), Unreliable Datagram (UD), and Raw Datagram
  – Provides flexibility to develop upper layers
• Multiple Operations
  – Send/Recv
  – RDMA Read/Write
  – Atomic Operations (very unique)
    • high performance and scalable implementations of distributed locks, semaphores, collective communication operations
• Leading to big changes in designing HPC clusters, file systems, cloud computing systems, grid computing systems, ....
Communication in the Channel Semantics (Send/Receive Model)

Send WQE contains information about the send buffer (multiple non-contiguous segments)

Receive WQE contains information on the receive buffer (multiple non-contiguous segments); Incoming messages have to be matched to a receive WQE to know where to place

Processor is involved only to:
1. Post receive WQE
2. Post send WQE
3. Pull out completed CQEs from the CQ
Send WQE contains information about the send buffer (multiple segments) and the receive buffer (single segment)

Initiator processor is involved only to:
1. Post send WQE
2. Pull out completed CQE from the send CQ
No involvement from the target processor
Presented Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Wide Adaptation of RDMA Technology

- Message Passing Interface (MPI) for HPC
- Parallel File Systems
  - Lustre
  - GPFS
- Delivering excellent performance (latency, bandwidth, and CPU Utilization)
- Delivering excellent scalability
MVAPICH2/MVAPICH2-X Software

• High Performance open-source MPI Library for InfiniBand, 10Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  – MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Available since 2002
  – MVAPICH2-X (MPI + PGAS), Available since 2012
  – Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC)
  – Used by more than 2,200 organizations in 73 countries
  – More than 221,000 downloads from OSU site directly
  – Empowering many TOP500 clusters
    • 7th ranked 519,640-core cluster (Stampede) at TACC
    • 13th ranked 74,358-core cluster (Tsubame 2.5) at Tokyo Institute of Technology
    • 23th ranked 96,192-core cluster (Pleiades) at NASA
    • many others ...
  – Available with software stacks of many IB, HSE, and server vendors including
    Linux Distros (RedHat and SuSE)
    – http://mvapich.cse.ohio-state.edu
  – Partner in the U.S. NSF-TACC Stampede System
One-way Latency: MPI over IB with MVAPICH2

**Small Message Latency**

- **Latency (us)**
  - 1.82
  - 1.66
  - 1.64
  - 1.56
  - 1.09
  - 0.99
  - 1.12

**Message Size (bytes)**

```plaintext
0 4 8 16 32 64 128 256 512 1K
```

**Large Message Latency**

- **Latency (us)**
  - 26

**Message Size (bytes)**

```plaintext
2K 4K 8K 16K 32K 64K 128K 256K
```

**Platforms:**
- DDR, QDR - 2.4 GHz Quad-core (Westmere) Intel PCI Gen2 with IB switch
- FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch
- ConnectIB-Dual FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch
- ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
Bandwidth: MPI over IB with MVAPICH2

Unidirectional Bandwidth

Bidirectional Bandwidth

DDR, QDR - 2.4 GHz Quad-core (Westmere) Intel PCI Gen2 with IB switch
FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch
ConnectIB-Dual FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch
ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
Can High-Performance Interconnects Benefit Data Management and Processing?

• Most of the current Big Data systems use Ethernet Infrastructure with Sockets

• Concerns for performance and scalability

• Usage of High-Performance Networks is beginning to draw interest
  – Oracle, IBM, Google, Intel 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)?

• Can HPC Clusters with High-Performance networks be used for Big Data applications using Hadoop and Memcached?
Designing Communication and I/O Libraries for Big Data Systems: Challenges

- Big Data Middleware
  - (HDFS, MapReduce, HBase, Spark and Memcached)
- Networking Technologies
  - (InfiniBand, 1/10/40GigE and Intelligent NICs)
- Storage Technologies
  - (HDD and SSD)
- Programming Models
  - (Sockets)
- Other Protocols?
- Applications
- Benchmarks
- Communication and I/O Library
  - Point-to-Point Communication
  - Threaded Models and Synchronization
  - Virtualization
  - I/O and File Systems
  - QoS
  - Fault-Tolerance
- Commodity Computing System Architectures
  - (Multi- and Many-core architectures and accelerators)
- Benchmarks
Can Big Data Processing Systems be Designed with High-Performance Networks and Protocols?

- **Current Design**
  - Application
  - Sockets
  - 1/10 GigE Network

- **Enhanced Designs**
  - Application
  - Accelerated Sockets
  - Verbs / Hardware Offload
  - 10 GigE or InfiniBand

- **Our Approach**
  - Application
  - OSU Design
  - Verbs Interface
  - 10 GigE or InfiniBand

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

• Overview of Modern Clusters, Interconnects and Protocols
• Challenges for Accelerating Data Management and Processing
• The High-Performance Big Data (HiBD) Project
• RDMA-based design for Memcached and HBase
• RDMA-based designs for Apache Hadoop and Spark
  – Case studies with HDFS, MapReduce, and Spark
  – RDMA-based MapReduce on HPC Clusters with Lustre
• Ongoing and Future Activities
• Conclusion and Q&A
The High-Performance Big Data (HiBD) Project

- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
- [http://hibd.cse.ohio-state.edu](http://hibd.cse.ohio-state.edu)
- RDMA for Apache HBase and Spark
RDMA for Memcached Distribution

• High-Performance Design of Memcached over RDMA-enabled Interconnects
  – High performance design with native InfiniBand and RoCE support at the verbs-level for Memcached and libMemcached components
  – Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)

• Current release: 0.9.1
  – Based on Memcached 1.4.20 and libMemcached 1.0.18.
  – Compliant with Memcached APIs and applications
  – Tested with
    • Mellanox InfiniBand adapters (DDR, QDR and FDR)
    • RoCE support with Mellanox adapters
    • Various multi-core platforms
  – [http://hibd.cse.ohio-state.edu](http://hibd.cse.ohio-state.edu)
RDMA for Apache Hadoop 1.x/2.x Distributions

• High-Performance Design of Hadoop over RDMA-enabled Interconnects
  – High performance design with native InfiniBand and RoCE support at the verbs-level for HDFS, MapReduce, and RPC components
  – Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)

• Current release: 0.9.9/0.9.1
  – Based on Apache Hadoop 1.2.1/2.4.1
  – Compliant with Apache Hadoop 1.2.1/2.4.1 APIs and applications
  – Tested with
    • Mellanox InfiniBand adapters (DDR, QDR and FDR)
    • RoCE support with Mellanox adapters
    • Various multi-core platforms
    • Different file systems with disks and SSDs
  – http://hibd.cse.ohio-state.edu
OSU HiBD Micro-Benchmark (OHB) Suite - Memcached

- Released in OHB 0.7.1 (ohb_memlat)
- Evaluates the performance of stand-alone Memcached
- Three different micro-benchmarks
  - 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)
- Calculates average latency of Memcached operations
- Can measure throughput in Transactions Per Second
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
  - RDMA-based Memcached
  - Case study with OLTP
  - SSD-assisted hybrid Memcached
  - RDMA-based HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Design Challenges of RDMA-based Memcached

- Can Memcached be re-designed from the ground up to utilize RDMA capable networks?
- How efficiently can we utilize RDMA for Memcached operations?
- Can we leverage the best features of RC and UD to deliver both high performance and scalability to Memcached?
- Memcached applications need not be modified; the middleware uses verbs interface if available
Memcached-RDMA Design

- Server and client perform a negotiation protocol
  - Master thread assigns clients to appropriate worker thread
- Once a client is assigned a verbs worker thread, it can communicate directly and is “bound” to that thread
- All other Memcached data structures are shared among RDMA and Sockets worker threads

- Native IB-verbs-level Design and evaluation with
  - Server: Memcached (http://memcached.org)
  - Client: libmemcached (http://libmemcached.org)
  - Different networks and protocols: 10GigE, IPoIB, native IB (RC, UD)
Memcached Get Latency (Small Message)

- 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 (4 bytes)

- 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 IPoIB
Memcached Performance (FDR Interconnect)

- 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

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


Memcached-RDMA – Case Studies with OLTP Workloads

- Design of scalable architectures using Memcached and MySQL

- Case study to evaluate benefits of using RDMA-Memcached for traditional OLTP workloads
Illustration with Read-Cache-Read access pattern using modified mysqlslap load testing tool

Up to 40 nodes, 10 concurrent threads per node, 20 queries per client

Memcached-RDMA can
- improve query latency by up to 66% over IPoIB (32Gbps)
- throughput by up to 69% over IPoIB (32Gbps)
Traditional Memcached Deployments

Existing Approach (Basic)

Proxy Server (Memcached Client) → Memcached Server
(1) Internet
(2)(frequent) Database Server
(3)(frequent)

Existing memcached deployment

• Low hit ratio at Memcached server due to Limited main memory size
• Frequent access to backend DB

Virtual Memory Swap (VMS)

Proxy Server (Memcached Client) → Database Server
Internet
(1) host memory
(2)(seldom) Virtual Memory Swap System
(3)(seldom) SSD

Mmap() SSD into virtual memory

• Higher hit ratio due to SSD Mapped virtual memory
• Significant overhead at virtual memory management
SSD-Assisted Hybrid Memory for Memcached

### Get Latency (us)

<table>
<thead>
<tr>
<th></th>
<th>IB Verbs</th>
<th>IPoIB</th>
<th>10GigE</th>
<th>1GigE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySQL</strong></td>
<td>N/A</td>
<td>10763</td>
<td>10724</td>
<td>11220</td>
</tr>
<tr>
<td><strong>Memcached (In RAM)</strong></td>
<td>10</td>
<td>60</td>
<td>40</td>
<td>150</td>
</tr>
<tr>
<td><strong>Memcached (SSD virtual memory)</strong></td>
<td>347</td>
<td>387</td>
<td>362</td>
<td>455</td>
</tr>
</tbody>
</table>

### SSD Basic Performance (us) (Fusionio ioDrive)

<table>
<thead>
<tr>
<th></th>
<th>Random Read</th>
<th>Random Write</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SSD Latency</strong></td>
<td>68</td>
<td>70</td>
</tr>
</tbody>
</table>

*SSD-Assisted Hybrid Memory to expand Memory size
*A user-level library that bypasses the kernel-based virtual memory management overhead*
Evaluation on SSD-Assisted Hybrid Memory for Memcached - Latency

- Memcached with InfiniBand DDR transport (16Gbps)
- 30GB data in SSD, 256 MB read/write buffer
- Get / Put a random object

Evaluation on SSD-Assisted Hybrid Memory for Memcached - Throughput

- Memcached with InfiniBand DDR transport (16Gbps)
- 30GB data in SSD, object size = 1KB, 256 MB read/write buffer
- 1, 2, 4 and 8 client process to perform random get()
Motivation – Detailed Analysis on HBase Put/Get

- 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

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-RDMA Design - Communication Flow

- RDMA design components
  - Network Selector, IB Reader, Helper threads, JNI Adaptive Interface
HBase Micro-benchmark (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

- **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 Put latency**
  - 64 clients: **1.9 ms**; 128 Clients: **3.5 ms**
  - 40% improvement over IPoIB for 128 clients
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark
Design Overview of HDFS with RDMA

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

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

• Cluster with HDD DataNodes
  – 30% improvement in communication time over IPoIB (QDR)
  – 56% improvement in communication time over 10GigE

• Similar improvements are obtained for SSD DataNodes

Evaluations using OHB HDFS Micro-benchmark

- Cluster with 4 HDD DataNodes, single disk per node
  - 25% improvement in latency over IPoIB (QDR) for 10GB file size
  - 50% improvement in throughput over IPoIB (QDR) for 10GB file size

Evaluations using TestDFSIO

- Cluster with 8 HDD DataNodes, single disk per node
  - 24% improvement over IPoIB (QDR) for 20GB file size
- Cluster with 4 SSD DataNodes, single SSD per node
  - 61% improvement over IPoIB (QDR) for 20GB file size
Evaluations using Enhanced DFSIO of Intel HiBench on TACC-Stampede

- Cluster with 64 DataNodes, single HDD per node
  - 64% improvement in throughput over IPoIB (FDR) for 256GB file size
  - 37% improvement in latency over IPoIB (FDR) for 256GB file size
Evaluations using YCSB
(32 Region Servers: 100% Update)

- 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
**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
Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark
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
Advanced Overlapping among different phases

- A hybrid approach to achieve maximum possible overlapping in MapReduce across all phases compared to other approaches
  - Efficient Shuffle Algorithms
  - Dynamic and Efficient Switching
  - On-demand Shuffle Adjustment

• Stand-alone MapReduce micro-benchmark (MR-AVG)
• 1 KB key/value pair size
• For 8 slave nodes, RDMA has up to 30% over IPoIB (56Gbps)
• For 16 slave nodes, RDMA has up to 28% over IPoIB (56Gbps)

Evaluations using Sort (HDD, SSD)

- With 8 HDD DataNodes for 40GB sort
  - 43% improvement over IPoIB (QDR)
  - 44% improvement over UDA-IB (QDR)

- With 8 SSD DataNodes for 40GB sort
  - 52% improvement over IPoIB (QDR)
  - 45% improvement over UDA-IB (QDR)
Evaluations with TeraSort

- 100 GB TeraSort with 8 DataNodes with 2 HDD per node
  - 49% benefit compared to UDA-IB (QDR)
  - 54% benefit compared to IPoIB (QDR)
  - 56% benefit compared to 10GigE
Performance Evaluation on Larger Clusters

Sort in OSU Cluster

- For 240GB Sort in 64 nodes
  - 40% improvement over IPoIB (QDR) with HDD used for HDFS

TeraSort in TACC Stampede

- For 320GB TeraSort in 64 nodes
  - 38% improvement over IPoIB (FDR) with HDD used for HDFS
Evaluations using PUMA Workload

- **50%** improvement in Self Join over IPoIB (QDR) for 80 GB data size
- **49%** improvement in Sequence Count over IPoIB (QDR) for 30 GB data size
• 50 small MapReduce jobs executed in a cluster size of 4
• Maximum performance benefit 24% over IPoIB (QDR)
• Average performance benefit 13% over IPoIB (QDR)
Acceleration Case Studies and In-Depth Performance Evaluation

- RDMA-based Designs and Performance Evaluation
  - HDFS
  - MapReduce
  - Spark
Design Overview of Spark with RDMA

- **Design Features**
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking and out-of-order 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

Preliminary Results of Spark-RDMA Design - GroupBy

Cluster with 4 HDD Nodes, GroupBy with 32 cores
- 18% improvement over IPoIB (QDR) for 10GB data size

Cluster with 8 HDD Nodes, GroupBy with 64 cores
- 20% improvement over IPoIB (QDR) for 20GB data size
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Optimize Apache Hadoop over Parallel File Systems

- **HPC Cluster Deployment**
  - Hybrid topological solution of Beowulf architecture with separate I/O nodes
  - Lean compute nodes with light OS; more memory space; small local storage
  - Sub-cluster of dedicated I/O nodes with parallel file systems, such as Lustre

- **MapReduce over Lustre**
  - Local disk is used as the intermediate data directory
  - Lustre is used as the intermediate data directory
Case Study - 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)

Case Study - Performance Improvement of MapReduce over Lustre on TACC-Stampede

- Lustre is used as the intermediate data directory

- For 160GB Sort in 16 nodes
  - 35% improvement over IPoIB (FDR)

- For 320GB Sort in 32 nodes
  - 33% improvement over IPoIB (FDR)

- Can more optimizations be achieved by leveraging more features of Lustre?
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Ongoing and Future Activities for Hadoop Accelerations

- Some other existing Hadoop solutions
  - Hadoop-A (UDA)
    - RDMA-based implementation of Hadoop MapReduce shuffle engine
    - Uses plug-in based solution: UDA (Unstructured Data Accelerator)
  - Cloudera Distributions of Hadoop
    - CDH: Open Source distribution
    - Cloudera Standard: CDH with automated cluster management
    - Cloudera Enterprise: Cloudera Standard + enhanced management capabilities and support
    - Ethernet/IPoIB
  - Hortonworks Data Platform
    - HDP: Open Source distribution
    - Developed as projects through the Apache Software Foundation (ASF), NO proprietary extensions or add-ons
    - Functional areas: Data Management, Data Access, Data Governance and Integration, Security, and Operations.
    - Ethernet/IPoIB
    - [http://hortonworks.com/hdp](http://hortonworks.com/hdp)
Designing Communication and I/O Libraries for Big Data Systems: Solved a Few Initial Challenges

Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)

Programming Models
(Sockets)

RDMA Protocol

Applications

Benchmarks

Communication and I/O Library

Point-to-Point Communication

Threaded Models and Synchronization

Virtualization

I/O and File Systems

QoS

Fault-Tolerance

Networking Technologies
(InfiniBand, 1/10/40GigE and Intelligent NICs)

Commodity Computing System Architectures
(Multi- and Many-core architectures and accelerators)

Storage Technologies
(HDD and SSD)

Upper level Changes?
More Challenges

• Multi-threading and Synchronization
  – Multi-threaded model exploration
  – Fine-grained synchronization and lock-free design
  – Unified helper threads for different components
  – Multi-endpoint design to support multi-threading communications

• QoS and Virtualization
  – Network virtualization and locality-aware communication for Big Data middleware
  – Hardware-level virtualization support for End-to-End QoS
  – I/O scheduling and storage virtualization
  – Live migration
More Challenges (Cont’d)

• Support of Accelerators
  – Efficient designs for Big Data middleware to take advantage of NVIDIA GPGPUs and Intel MICs
  – Offload computation-intensive workload to accelerators
  – Explore maximum overlapping between communication and offloaded computation

• Fault Tolerance Enhancements
  – Exploration of light-weight fault tolerance mechanisms for Big Data

• Support of Parallel File Systems
  – Optimize Big Data middleware over parallel file systems (e.g. Lustre) on modern HPC clusters

• Big Data Benchmarking
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?
Challenges in Benchmarking of RDMA-based Designs

- **Applications**
- **Benchmarks**
- **Big Data Middleware** (HDFS, MapReduce, HBase, Spark and Memcached)
- **Programming Models** (Sockets)
- **RDMA Protocols**
- **Communication and I/O Library**
  - Point-to-Point Communication
  - Threaded Models and Synchronization
  - Virtualization
  - QoS
  - Fault-Tolerance
- **Networking Technologies** (InfiniBand, 1/10/40GigE and Intelligent NICs)
- **Commodity Computing System Architectures** (Multi- and Many-core architectures and accelerators)
- **Storage Technologies** (HDD and SSD)

**Current Benchmarks**

**Correlation?**

**No Benchmarks**

**ADMS ’14**
OSU MPI Micro-Benchmarks (OMB) Suite

• A comprehensive suite of benchmarks to
  – Compare performance of different MPI libraries on various networks and systems
  – Validate low-level functionalities
  – Provide insights to the underlying MPI-level designs

• Started with basic send-recv (MPI-1) micro-benchmarks for latency, bandwidth and bi-directional bandwidth

• Extended later to
  – MPI-2 one-sided
  – Collectives
  – GPU-aware data movement
  – OpenSHMEM (point-to-point and collectives)
  – UPC

• Has become an industry standard

• Extensively used for design/development of MPI libraries, performance comparison of MPI libraries and even in procurement of large-scale systems

• Available from http://mvapich.cse.ohio-state.edu/benchmarks

• Available in an integrated manner with MVAPICH2 stack
Iterative Process – Requires Deeper Investigation and Design for Benchmarking Next Generation Big Data Systems and Applications

Applications

Big Data Middleware
(HDFS, MapReduce, HBase, Spark and Memcached)

Programming Models
(Sockets)

Communication and I/O Library

Point-to-Point Communication

Threaded Models and Synchronization

Virtualization

I/O and File Systems

QoS

Fault-Tolerance

Networking Technologies
(InfiniBand, 1/10/40GigE and Intelligent NICs)

Commodity Computing System Architectures
(Multi- and Many-core architectures and accelerators)

Storage Technologies
(HDD and SSD)

Applications

Benchmarks

RDMA Protocols

Applications-Level Benchmarks

Micro-Benchmarks
Future Plans of OSU High Performance Big Data Project

• Upcoming Releases of RDMA-enhanced Packages will support
  – Hadoop 2.x MapReduce & RPC
  – Spark
  – HBase

• Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  – HDFS
  – MapReduce
  – RPC

• Exploration of other components (Threading models, QoS, Virtualization, Accelerators, etc.)

• Advanced designs with upper-level changes and optimizations
Presentation Outline

- Overview of Modern Clusters, Interconnects and Protocols
- Challenges for Accelerating Data Management and Processing
- The High-Performance Big Data (HiBD) Project
- RDMA-based design for Memcached and HBase
- RDMA-based designs for Apache Hadoop and Spark
  - Case studies with HDFS, MapReduce, and Spark
  - RDMA-based MapReduce on HPC Clusters with Lustre
- Ongoing and Future Activities
- Conclusion and Q&A
Concluding Remarks

- Presented an overview of data management and processing middleware in different tiers
- Provided an overview of modern networking technologies
- Discussed challenges in accelerating Big Data middleware
- Presented initial designs to take advantage of InfiniBand/RDMA for Memcached, HBase, Hadoop and Spark
- Results are promising
- Many other open issues need to be solved
- Will enable Big Data management and processing community to take advantage of modern HPC technologies to carry out their analytics in a fast and scalable manner
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/