High-Performance Big Data Analytics with RDMA over NVM and NVMe-SSD

Talk at OFA Workshop 2018

by

Xiaoyi Lu
The Ohio State University
E-mail: luxi@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~luxi
Big 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
Big Data Processing with Apache Big Data Analytics Stacks

- **Major components included:**
  - **MapReduce** (Batch)
  - **Spark** (Iterative and Interactive)
  - **HBase** (Query)
  - **HDFS** (Storage)
  - **RPC** (Inter-process communication)

- **Underlying** **Hadoop Distributed File System (HDFS)** used by MapReduce, Spark, HBase, and many others

- **Model scales but high amount of communication and I/O can be further optimized!**
Drivers of Modern HPC Cluster and Data Center Architecture

- 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 FPGAs)
The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
  - HDFS, Memcached, HBase, and Spark Micro-benchmarks
- [http://hibd.cse.ohio-state.edu](http://hibd.cse.ohio-state.edu)
- Users Base: 280 organizations from 34 countries
- More than 25,750 downloads from the project site

Available for InfiniBand and RoCE

Available for x86 and OpenPOWER

Significant performance improvement with ‘RDMA+DRAM’ compared to default Sockets-based designs;
How about RDMA+NVRAM?
Non-Volatile Memory (NVM) and NVMe-SSD

3D XPoint from Intel & Micron

Samsung NVMe SSD

Performance of PMC Flashtec NVRAM [*]

- Non-Volatile Memory (NVM) provides byte-addressability with persistence
- The huge explosion of data in diverse fields require fast analysis and storage
- NVMs provide the opportunity to build high-throughput storage systems for data-intensive applications
- Storage technology is moving rapidly towards NVM

NVRAM Emulation based on DRAM

- Popular methods employed by recent works to emulate NVRAM performance model over DRAM
- Two ways:
  - Emulate byte-addressable NVRAM over DRAM
  - Emulate block-based NVM device over DRAM
Presentation Outline

• NRCIO: NVM-aware RDMA-based Communication and I/O Schemes
• NRCIO for Big Data Analytics
• NVMe-SSD based Big Data Analytics
• Conclusion and Q&A
Design Scope (NVM for RDMA)

**D-to-D over RDMA:** Communication buffers for client and server are allocated in DRAM (Common)

- **D-to-N over RDMA:** Communication buffers for client are allocated in DRAM; Server uses NVM
- **N-to-D over RDMA:** Communication buffers for client are allocated in NVM; Server uses DRAM
- **N-to-N over RDMA:** Communication buffers for client and server are allocated in NVM
NVRAM-aware Communication in NRCIO

NRCIO Send/Recv over NVRAM

Client

Server

Application buffer in DRAM/NVRAM

Message

memcpy

Hdr Message

DRAM buffer

Header + Payload

Message

memcp

cflush + delay

Server buffer in NVRAM

ACK

NRCIO RDMA_Read over NVRAM

Client

Server

Hdr

Header (RTS)

Reserve NVRAM Buffer

Message

Application buffer in DRAM/NVRAM

RDMA Read

cflush + delay

Message

Message

FIN

FIN

Server buffer in NVRAM
Comparison of communication latency using NRCIO send/receive semantics over InfiniBand QDR network and PCM memory

- High communication latencies due to slower writes to non-volatile persistent memory
  - NVRAM-to-Remote-NVRAM (NVRAM-NVRAM) => ~10x overhead vs. DRAM-DRAM
  - DRAM-to-Remote-NVRAM (DRAM-NVRAM) => ~8x overhead vs. DRAM-DRAM
  - NVRAM-to-Remote-DRAM (NVRAM-DRAM) => ~4x overhead vs. DRAM-DRAM
• Communication latency with NRCIO RDMA-Read over InfiniBand QDR + PCM memory
• Communication overheads for large messages due to slower writes into NVRAM from remote memory; similar to Send/Receive
• RDMA-Read outperforms Send/Receive for large messages; as observed for DRAM-DRAM
Presentation Outline

• NRCIO: NVM-aware RDMA-based Communication and I/O Schemes
• NRCIO for Big Data Analytics
• NVMe-SSD based Big Data Analytics
• Conclusion and Q&A
Opportunities of Using NVRAM+RDMA in HDFS

- Files are divided into fixed sized blocks
  - Blocks divided into packets
- NameNode: stores the file system namespace
- DataNode: stores data blocks in local storage devices
- Uses block replication for fault tolerance
  - Replication enhances data-locality and read throughput
- Communication and I/O intensive
- Java Sockets based communication
- Data needs to be persistent, typically on SSD/HDD
Design Overview of NVM and RDMA-aware HDFS (NVFS)

- **Design Features**
  - RDMA over NVM
  - HDFS I/O with NVM
    - Block Access
    - Memory Access
  - Hybrid design
    - NVM with SSD as a hybrid storage for HDFS I/O
  - Co-Design with Spark and HBase
    - Cost-effectiveness
    - Use-case

Evaluation with Hadoop MapReduce

- TestDFSIO on SDSC Comet (32 nodes)
  - Write: NVFS-MemIO gains by 4x over HDFS
  - Read: NVFS-MemIO gains by 1.2x over HDFS

- TestDFSIO on OSU Nowlab (4 nodes)
  - Write: NVFS-MemIO gains by 4x over HDFS
  - Read: NVFS-MemIO gains by 2x over HDFS

<table>
<thead>
<tr>
<th></th>
<th>Write</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSC Comet (32 nodes)</td>
<td></td>
<td>4x</td>
</tr>
<tr>
<td>TestDFSIO</td>
<td></td>
<td>1.2x</td>
</tr>
<tr>
<td>OSU Nowlab (4 nodes)</td>
<td></td>
<td>4x</td>
</tr>
<tr>
<td></td>
<td>2x</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation with HBase

- **YCSB 100% Insert on SDSC Comet (32 nodes)**
  - NVFS-BlkIO gains by **21%** by storing only WALs to NVM
- **YCSB 50% Read, 50% Update on SDSC Comet (32 nodes)**
  - NVFS-BlkIO gains by **20%** by storing only WALs to NVM
Opportunities to Use NVRAM+RDMA in MapReduce

- 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 (persistent)
  - Reduce tasks get these data by shuffle from NodeManagers, operate on it and write to HDFS (persistent)
- Communication and I/O intensive; Shuffle phase uses HTTP over Java Sockets; I/O operations take place in SSD/HDD typically
Opportunities to Use NVRAM in MapReduce-RDMA Design

Opportunities exist to improve the performance with NVRAM

All Operations are In-Memory

RDMA
NVRAM-Assisted Map Spilling in MapReduce-RDMA

- Minimizes the disk operations in Spill phase


Comparison with Sort and TeraSort

- RMR-NVM achieves **2.37x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **55%** compared to MR-IPoIB, **28%** compared to RMR

- RMR-NVM achieves **2.48x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **51%** compared to MR-IPoIB, **31%** compared to RMR
Evaluation of Intel HiBench Workloads

- We evaluate different HiBench workloads with Huge data sets on 8 nodes
- Performance benefits for Shuffle-intensive workloads compared to MR-IPoIB:
  - Sort: 42% (25 GB)
  - TeraSort: 39% (32 GB)
  - PageRank: 21% (5 million pages)
- Other workloads:
  - WordCount: 18% (25 GB)
  - KMeans: 11% (100 million samples)
Evaluation of PUMA Workloads

- We evaluate different PUMA workloads on 8 nodes with 30GB data size.
- Performance benefits for Shuffle-intensive workloads compared to MR-IPoIB:
  - AdjList: 39%
  - SelfJoin: 58%
  - RankedInvIndex: 39%
- Other workloads:
  - SeqCount: 32%
  - InvIndex: 18%
Presentation Outline

- NRCIO: NVM-aware RDMA-based Communication and I/O Schemes
- NRCIO for Big Data Analytics
- NVMe-SSD based Big Data Analytics
- Conclusion and Q&A
Overview of NVMe Standard

- NVMe is the standardized interface for PCIe SSDs
- Built on ‘RDMA’ principles
  - Submission and completion I/O queues
  - Similar semantics as RDMA send/recv queues
  - Asynchronous command processing
- Up to 64K I/O queues, with up to 64K commands per queue
- Efficient small random I/O operation
- MSI/MSI-X and interrupt aggregation

Source: NVMExpress.org
Overview of NVMe-over-Fabric

- Remote access to flash with NVMe over the network
- RDMA fabric is of most importance
  - Low latency makes remote access feasible
  - 1 to 1 mapping of NVMe I/O queues to RDMA send/recv queues
Design Challenges with NVMe-SSD

- **QoS**
  - Hardware-assisted QoS
- **Persistence**
  - Flushing buffered data
- **Performance**
  - Consider flash related design aspects
  - Read/Write performance skew
  - Garbage collection
- **Virtualization**
  - SR-IOV hardware support
  - Namespace isolation
- **New software systems**
  - Disaggregated Storage with NVMf
  - Persistent Caches
Evaluation with RocksDB

- 20%, 33%, 61% improvement for Insert, Write Sync, and Read Write
- Overwrite: Compaction and flushing in background
  - Low potential for improvement
- Read: Performance much worse; Additional tuning/optimization required
Evaluation with RocksDB

- 25%, 50%, 160% improvement for Insert, Write Sync, and Read Write
- Overwrite: Compaction and flushing in background
  - Low potential for improvement
- Read: Performance much worse; Additional tuning/optimization required
QoS-aware SPDK Design

- Synthetic application scenarios with different QoS requirements
  - Comparison using SPDK with Weighted Round Robbin NVMe arbitration
- Near desired job bandwidth ratios
- Stable and consistent bandwidth

S. Gugnani, X. Lu, and D. K. Panda, Analyzing, Modeling, and Provisioning QoS for NVMe SSDs, (Under Review)
Conclusion and Future Work

• Exploring NVM-aware RDMA-based Communication and I/O Schemes for Big Data Analytics
• Proposed a new library, NRCIO (work-in-progress)
• Re-design HDFS storage architecture with NVRAM
• Re-design RDMA-MapReduce with NVRAM
• Design Big Data analytics stacks with NVMe and NVMf protocols
• Results are promising
• Further optimizations in NRCIO
• Co-design with more Big Data analytics frameworks
The 4th International Workshop on High-Performance Big Data Computing (HPBDC)

HPBDC 2018 will be held with IEEE International Parallel and Distributed Processing Symposium (IPDPS 2018), Vancouver, British Columbia CANADA, May, 2018

Workshop Date: May 21st, 2018

Keynote Talk: Prof. Geoffrey Fox, Twister2: A High-Performance Big Data Programming Environment

Six Regular Research Papers and Two Short Research Papers

Panel Topic: Which Framework is the Best for High-Performance Deep Learning:
  Big Data Framework or HPC Framework?
  
  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
Thank You!

luxi@cse.ohio-state.edu

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

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

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