Designing High-Performance Non-Volatile Memory-aware RDMA Communication Protocols for Big Data Processing

Talk at Storage Developer Conference | SNIA 2018

by

Xiaoyi Lu

The Ohio State University

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

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

Dhabaleswar K. (DK) Panda

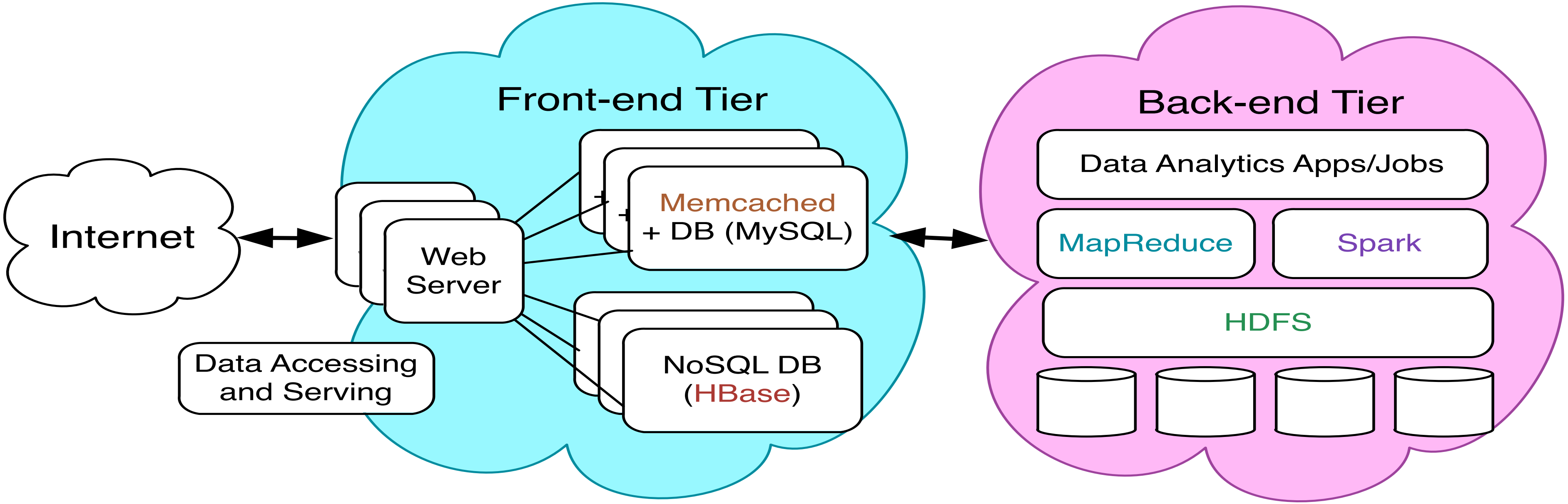
The Ohio State University

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

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

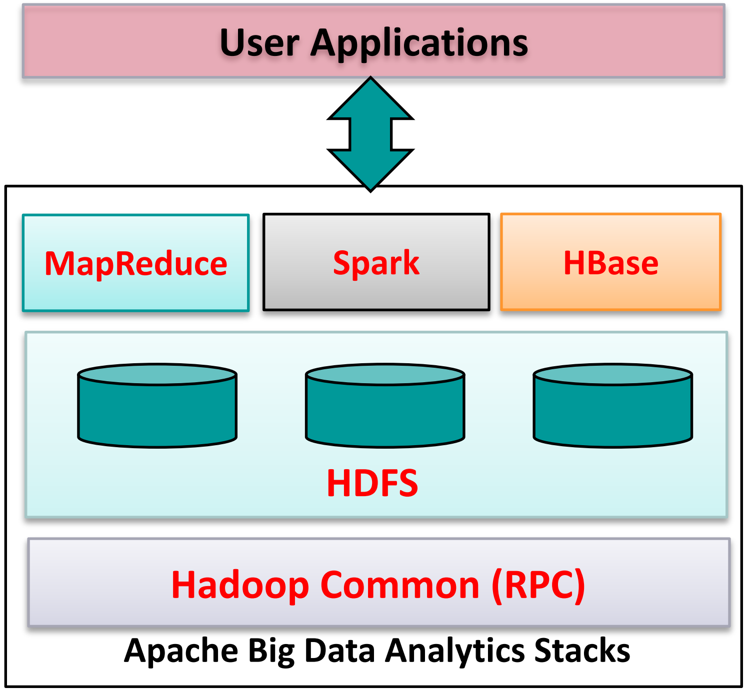
Big Data Management and Processing on Modern Clusters

- 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!

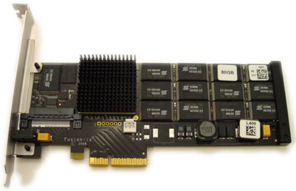


Drivers of Modern HPC Cluster and Data Center Architecture

Multi-/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)
- **NVM and NVMe-SSD**
- Accelerators (NVIDIA GPGPUs and FPGAs)

SDSC Comet

TACC Stampede

Microsoft Azure | amazon web services EC2 | ORACLE Cloud | Chameleon Cloud

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
- Users Base: 290 organizations from 34 countries
- More than 27,800 downloads from the project site

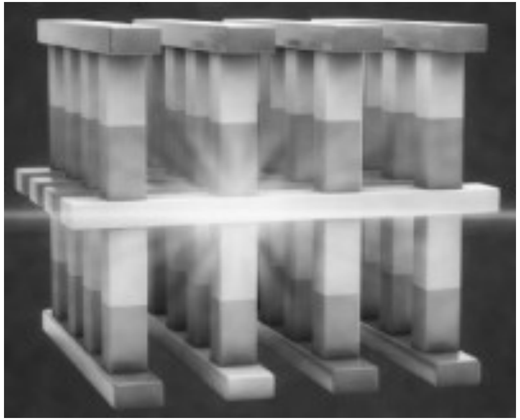
Available for InfiniBand and RoCE

Available for x86 and OpenPOWER

Significant performance improvement with 'RDMA+DRAM' compared to default Sockets-based designs;

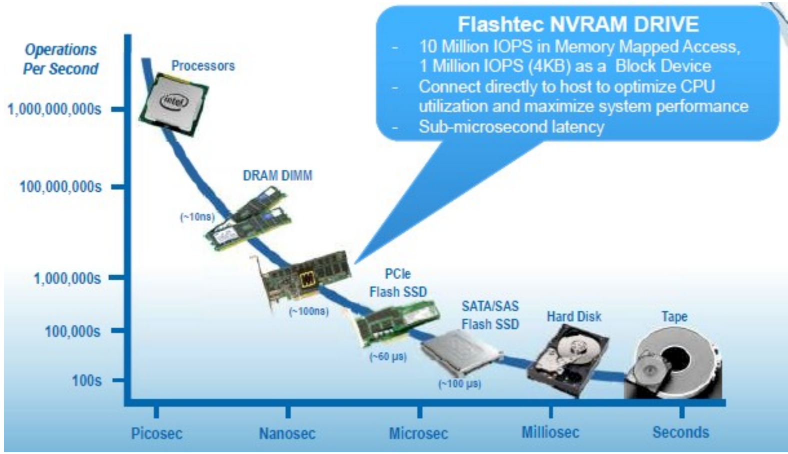
How about RDMA+NVRAM?

Non-Volatile Memory (NVM) and NVMe-SSD



3D XPoint from Intel & Micron

Samsung NVMe SSD



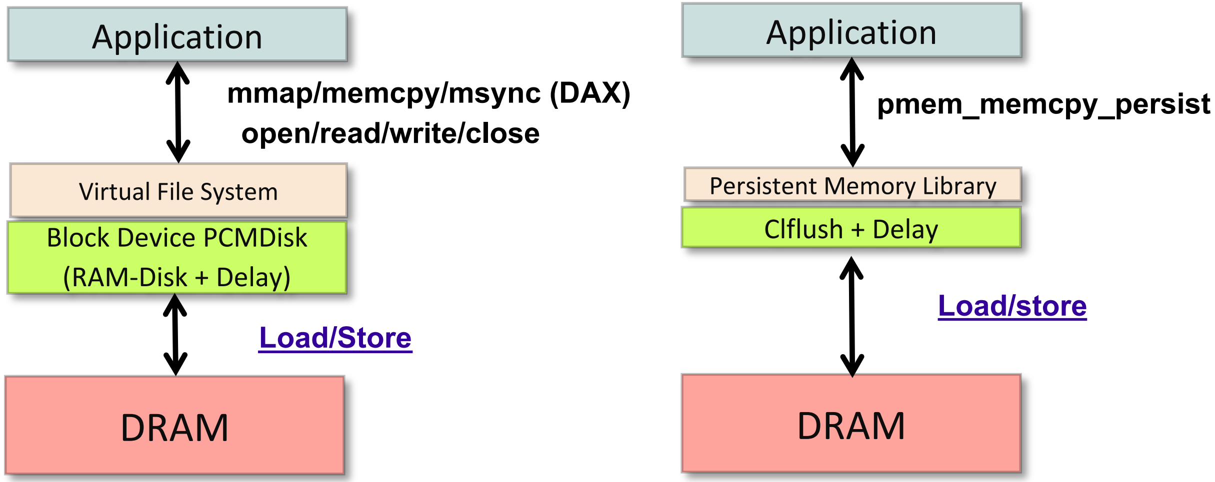
Performance of PMC Flashtec NVRAM [*]

- 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

[*] http://www.enterprisetech.com/2014/08/06/ flashtec-nvram-15-million-iops-sub-microsecond- latency/

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

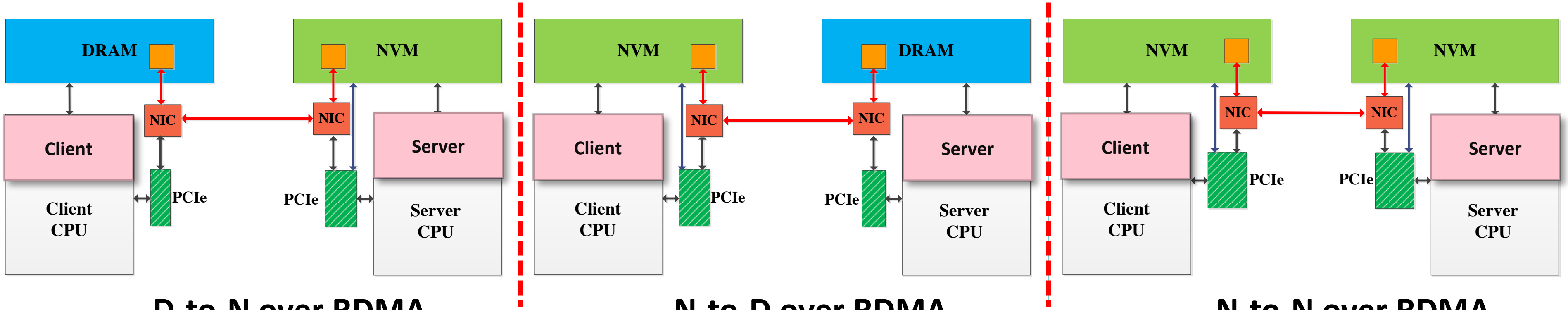


Presentation Outline

- 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 | **N-to-D over RDMA** | **N-to-N over RDMA**

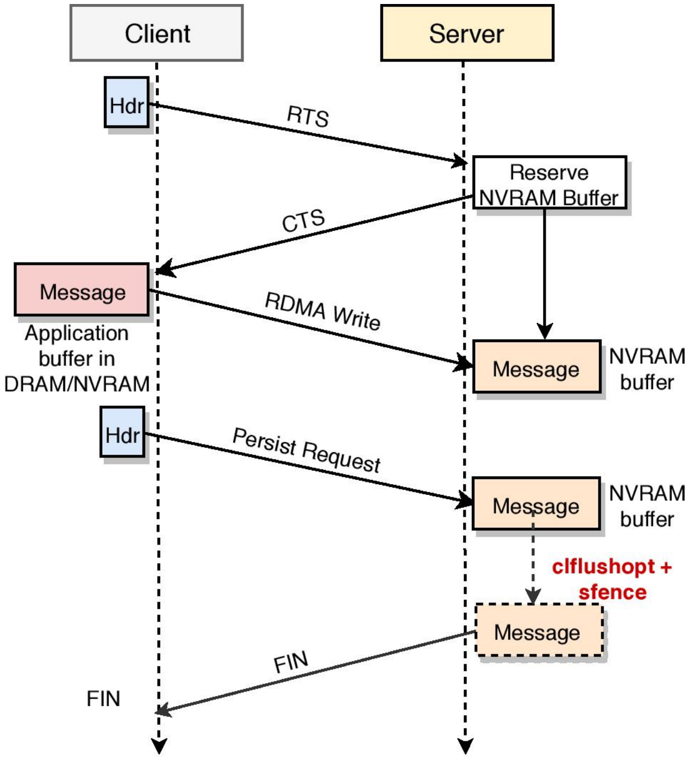
D-to-N over RDMA: Communication buffers for client are allocated in DRAM; Server uses NVM

N-to-D over RDMA: Communication buffers for client are allocated in NVM; Server uses DRAM

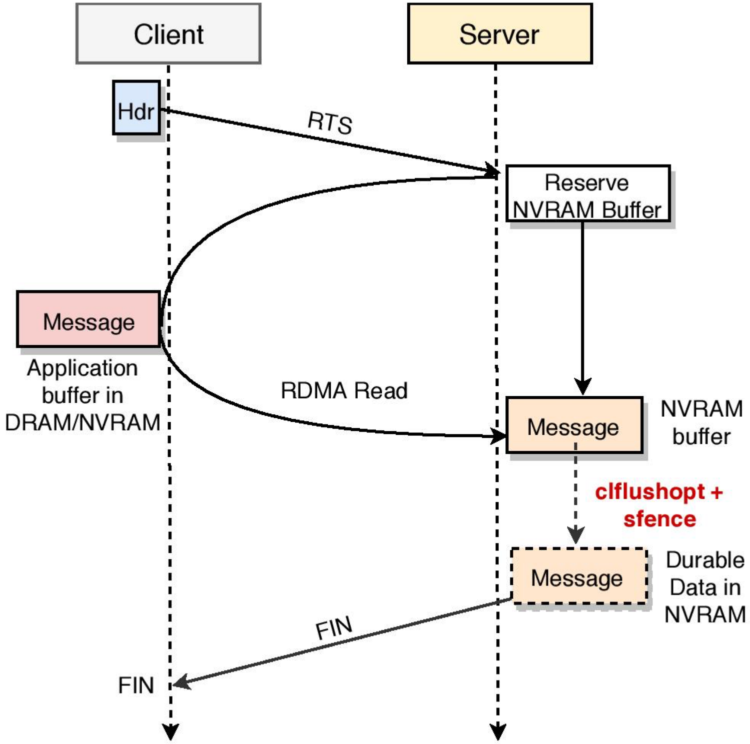
N-to-N over RDMA: Communication buffers for client and server are allocated in NVM

NVRAM-aware RDMA-based Communication in NRCIO

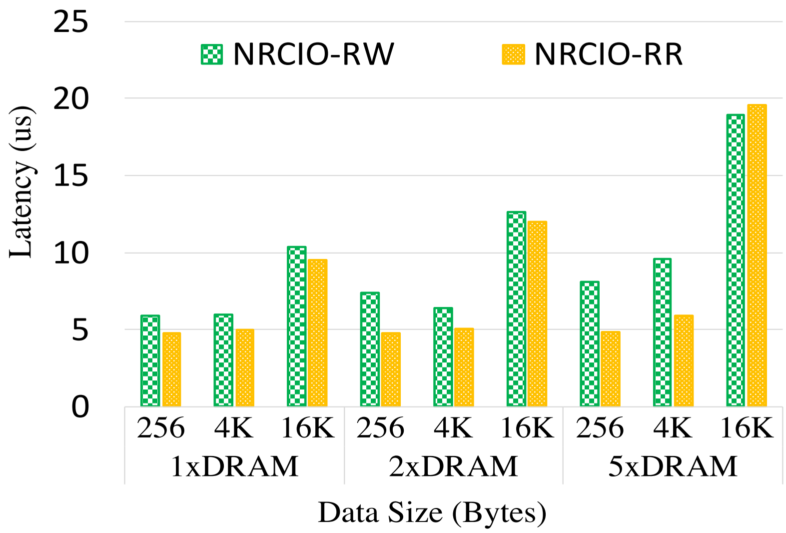
NRCIO RDMA Write over NVRAM

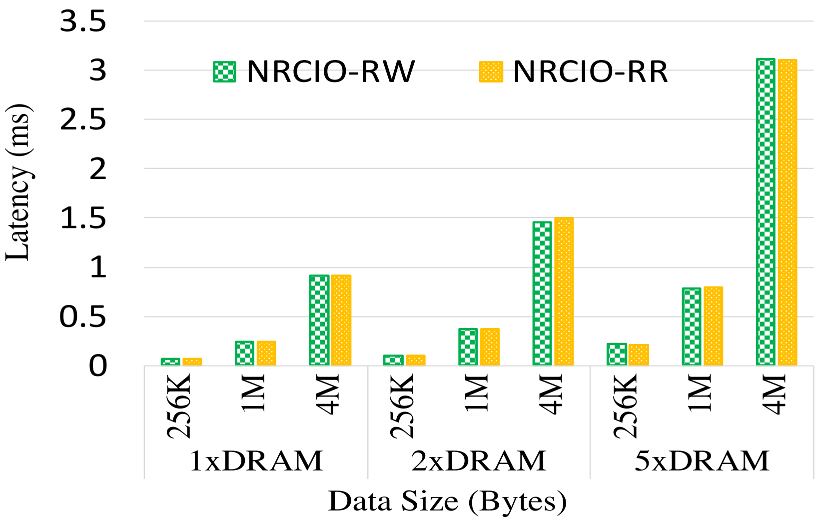


NRCIO RDMA Read over NVRAM



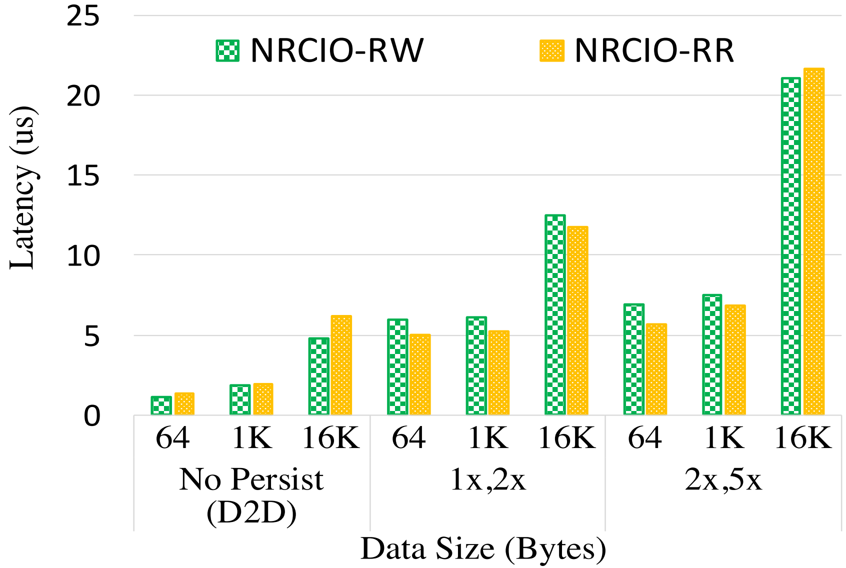
DRAM-TO-NVRAM RDMA-Aware Communication with NRCIO

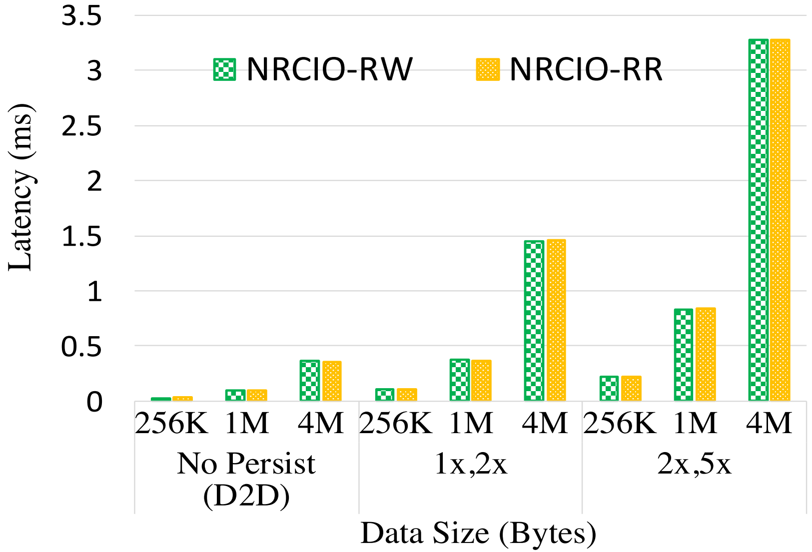




- Comparison of communication latency using NRCIO RDMA read and write communication protocols over InfiniBand EDR HCA with DRAM as source and NVRAM as destination
- {NxDRAM} NVRAM emulation mode = Nx NVRAM write slowdown vs. DRAM with clflushopt (emulated) + sfence
- Smaller impact of time-for-persistence on the end-to-end latencies for small messages vs. large messages => larger number of cache lines to flush

NVRAM-TO-NVRAM RDMA-Aware Communication with NRCIO





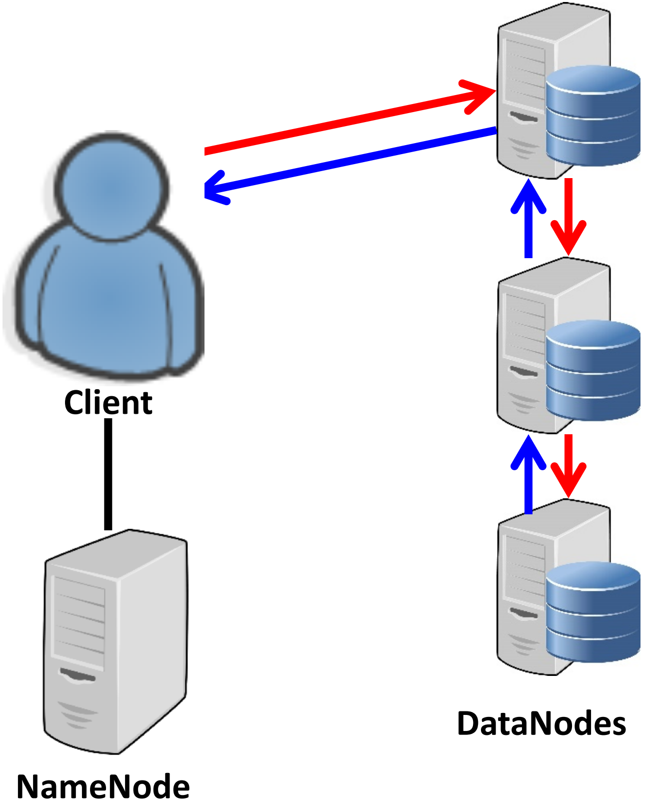
- Comparison of communication latency using NRCIO RDMA read and write communication protocols over InfiniBand EDR HCA vs. DRAM
- {Ax, By} NVRAM emulation mode = Ax NVRAM read slowdown and Bx NVRAM write slowdown vs. NVRAM
- High end-to-end latencies due to slower writes to non-volatile persistent memory
  - E.g., 3.9x for {1x,2x} and 8x for {2x,5x}

Presentation Outline

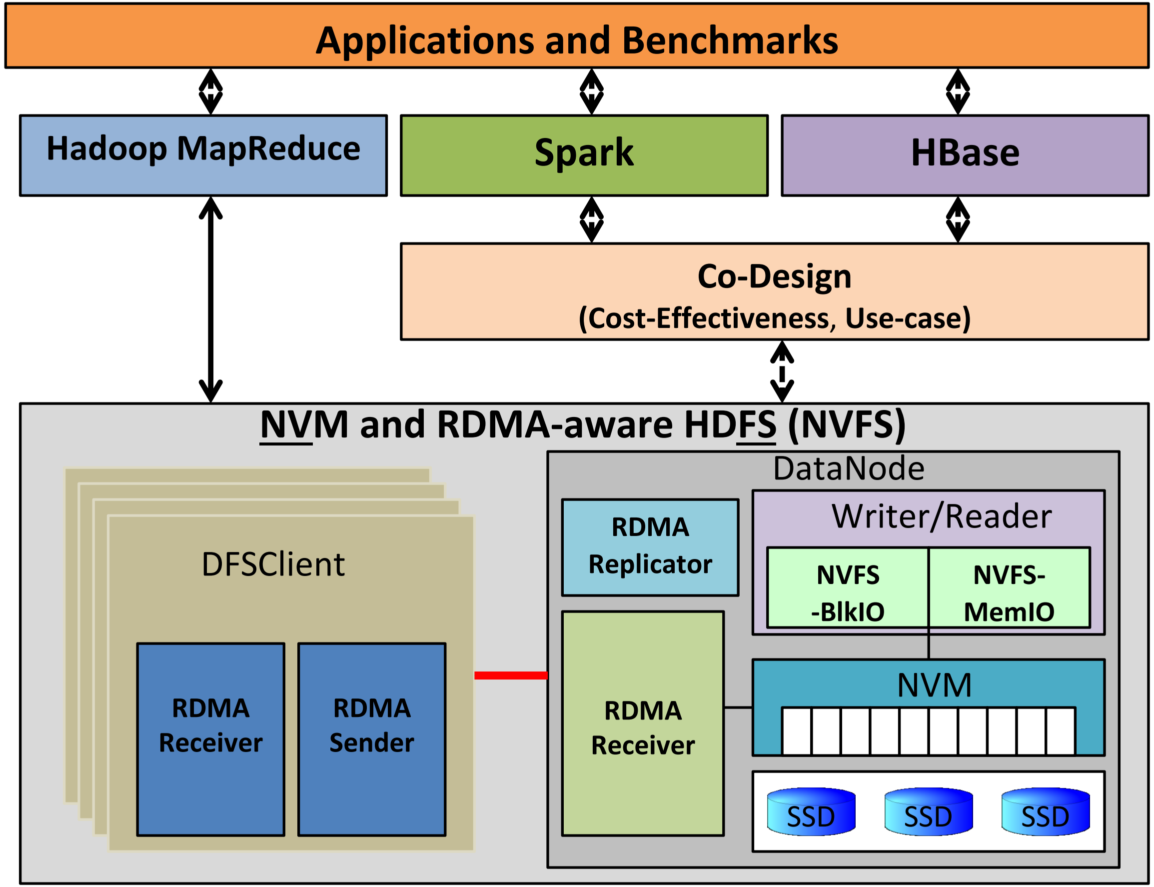
- NRCIO: NVM-aware RDMA-based Communication and I/O Schemes
- NRCIO for Big Data Analytics
- NVMe-SSD based Big Data Analytics
- Conclusion and Q&A

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



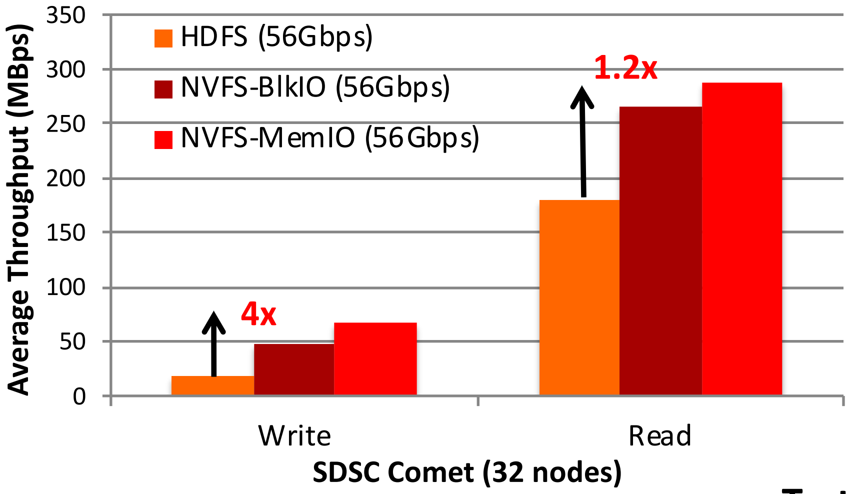
Design Overview of NVM and RDMA-aware HDFS (NVFS)

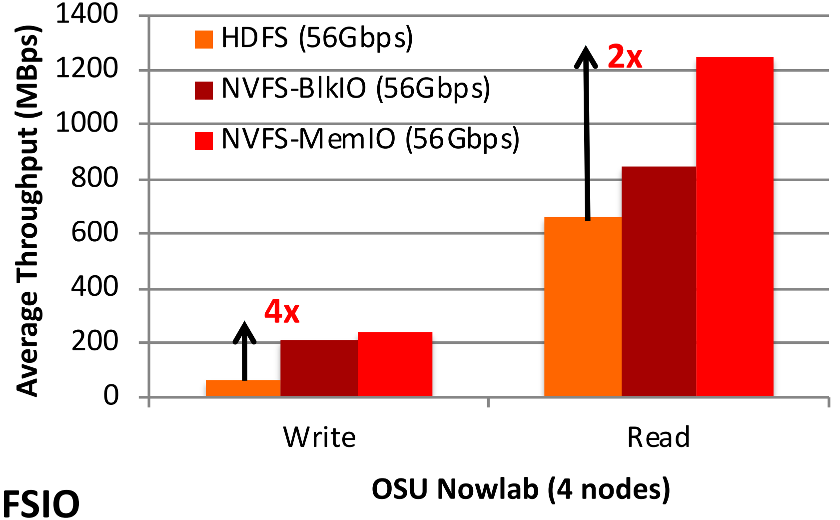


- **Design Features**
  - RDMA over NVM
  - HDFS I/O with NVM
    - Block Access
    - Memory Access
  - Hybrid design
    - NVM with SSD as a hybrid storage for HDFS I/O
  - Co-Design with Spark and HBase
    - Cost-effectiveness
    - Use-case

N. S. Islam, M. W. Rahman , X. Lu, and D. K. Panda, High Performance Design for HDFS with Byte-Addressability of NVM and RDMA, 24th International Conference on Supercomputing (ICS), June 2016

Evaluation with Hadoop MapReduce

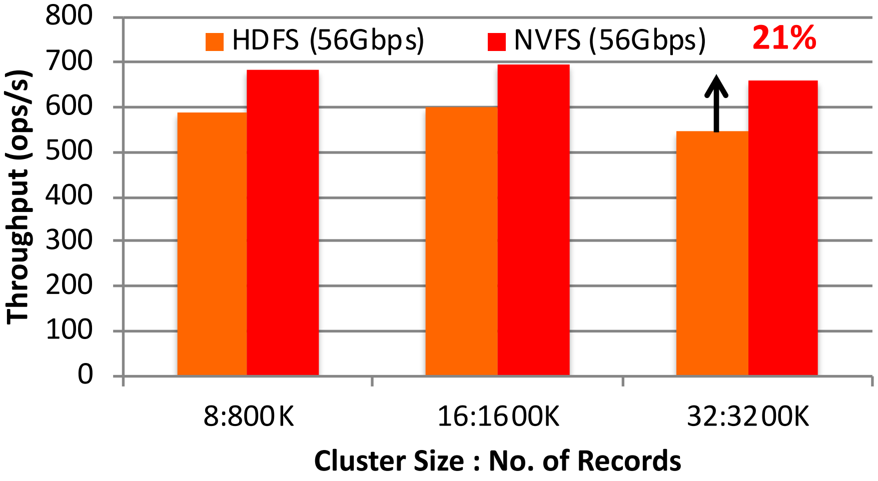




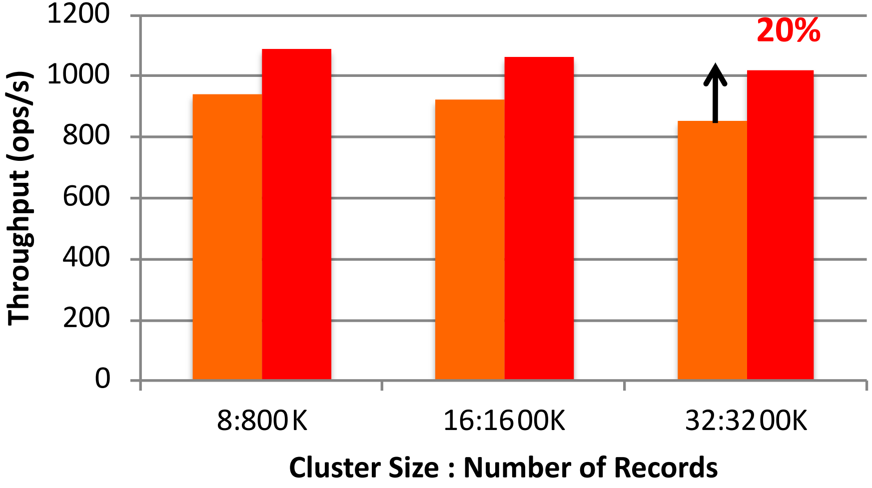
TestDFSIO

- 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

Evaluation with HBase



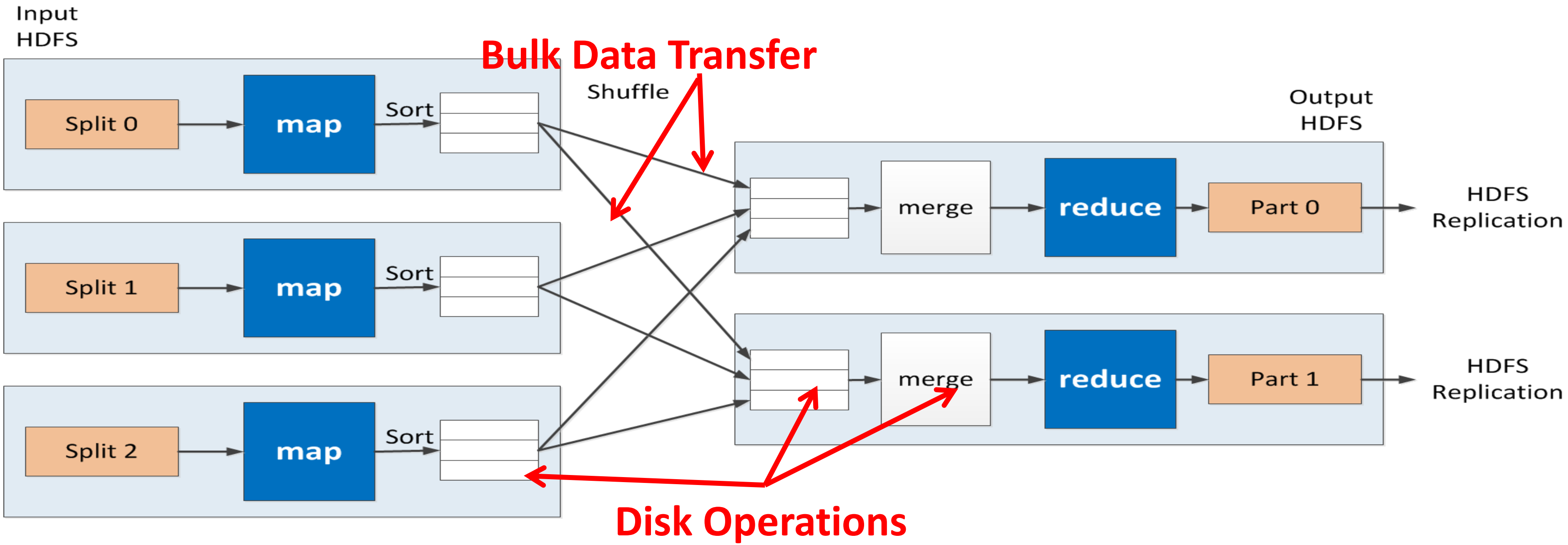
HBase 100% insert



HBase 50% read, 50% update

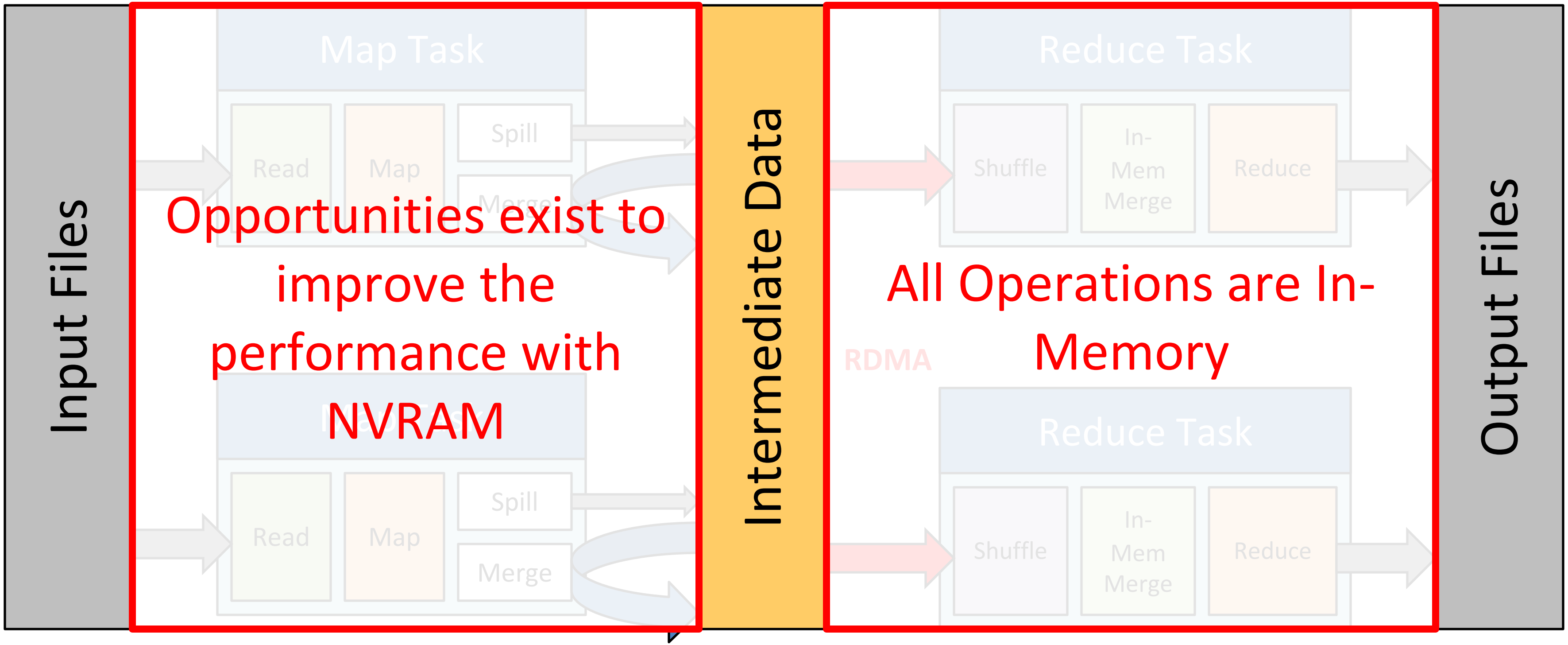
- YCSB 100% Insert on SDSC Comet (32 nodes)
  - NVFS-BlkIO gains by **21%** by storing only WALs to NVM
- YCSB 50% Read, 50% Update on SDSC Comet (32 nodes)
  - NVFS-BlkIO gains by **20%** by storing only WALs to NVM

Opportunities to Use NVRAM+RDMA in MapReduce

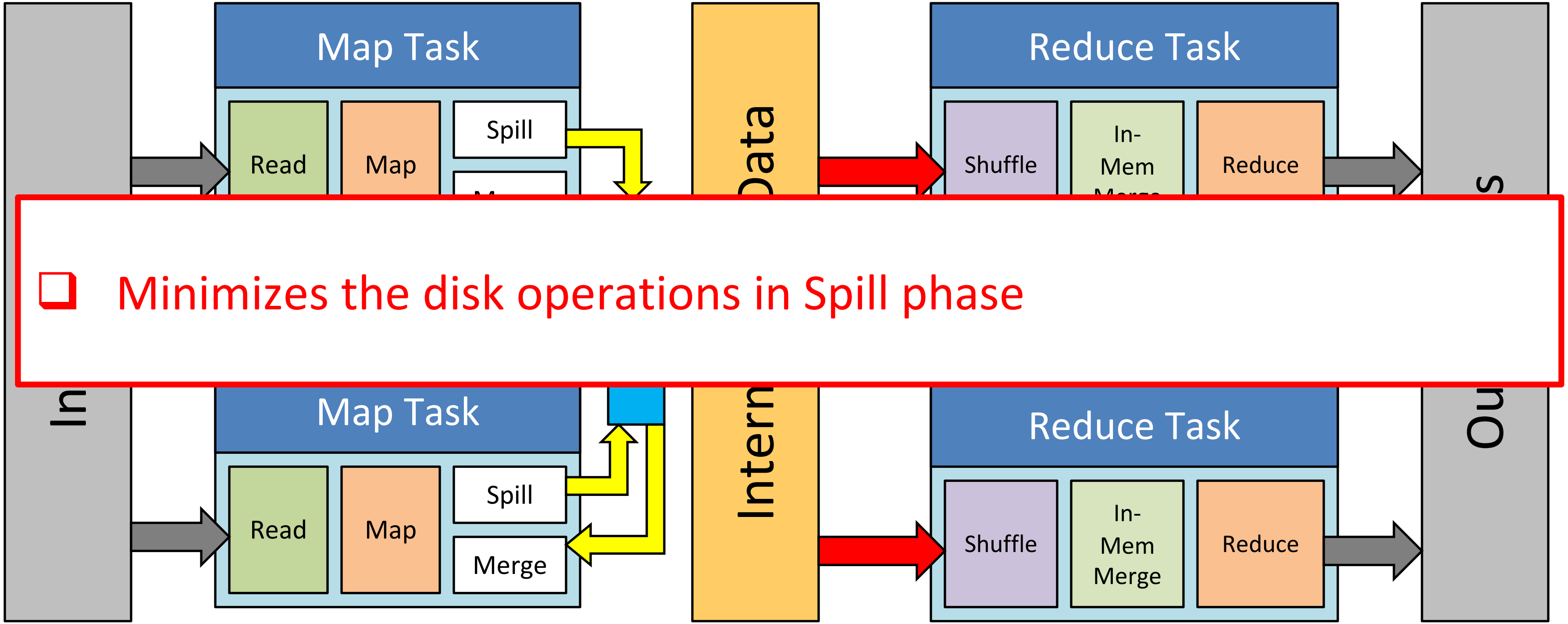


- Map and Reduce Tasks carry out the total job execution
  - Map tasks read from HDFS, operate on it, and write the intermediate data to local disk (persistent)
  - Reduce tasks get these data by shuffle from NodeManagers, operate on it and write to HDFS (persistent)
- Communication and I/O intensive; Shuffle phase uses HTTP over Java Sockets; I/O operations take place in SSD/HDD typically

Opportunities to Use NVRAM in MapReduce-RDMA Design



NVRAM-Assisted Map Spilling in MapReduce-RDMA

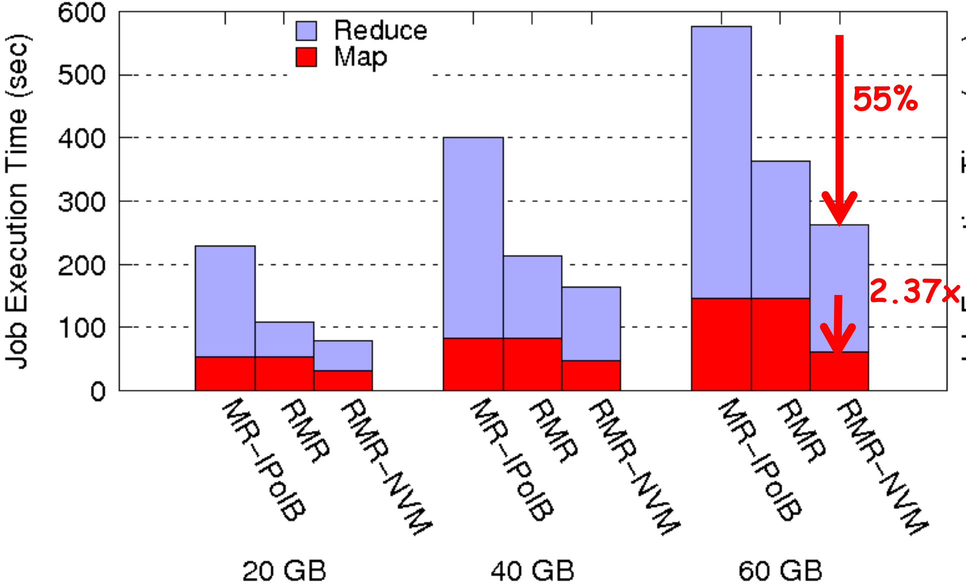


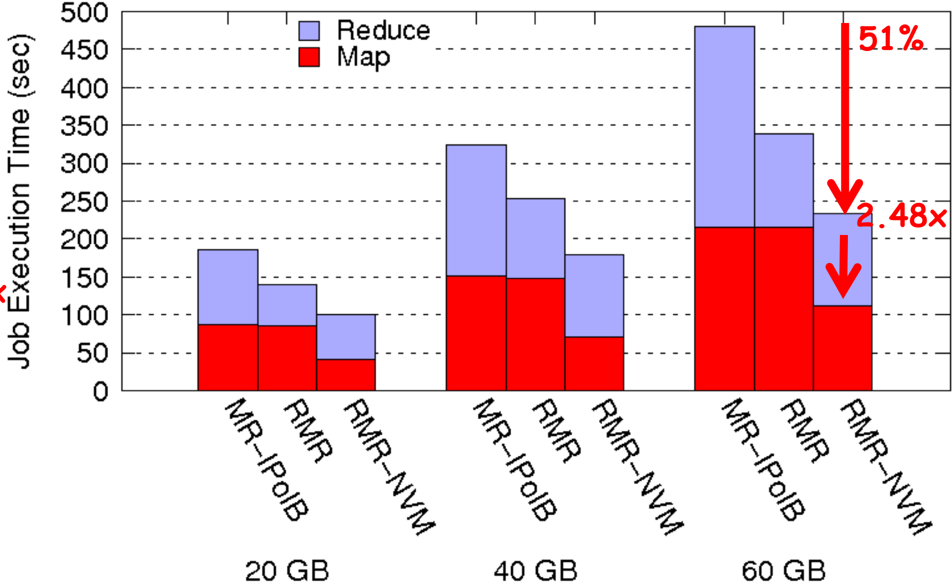
- ❑ Minimizes the disk operations in Spill phase

M. W. Rahman, N. S. Islam, X. Lu, and D. K. Panda, Can Non-Volatile Memory Benefit MapReduce Applications on HPC Clusters? PDSW-DISCS, with SC 2016.

M. W. Rahman, N. S. Islam, X. Lu, and D. K. Panda, NVMD: Non-Volatile Memory Assisted Design for Accelerating MapReduce and DAG Execution Frameworks on HPC Systems? IEEE BigData 2017.

Comparison with Sort and TeraSort

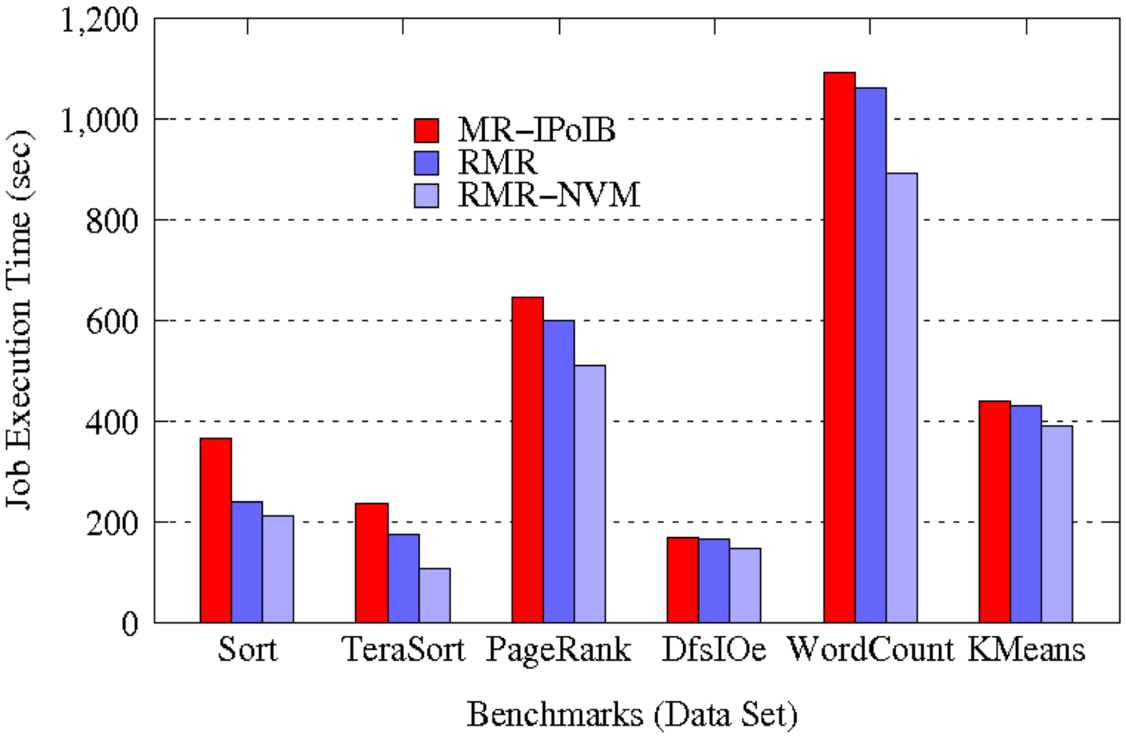




- RMR-NVM achieves **2.37x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **55%** compared to MR-IPoIB, **28%** compared to RMR
- RMR-NVM achieves **2.48x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **51%** compared to MR-IPoIB, **31%** compared to RMR

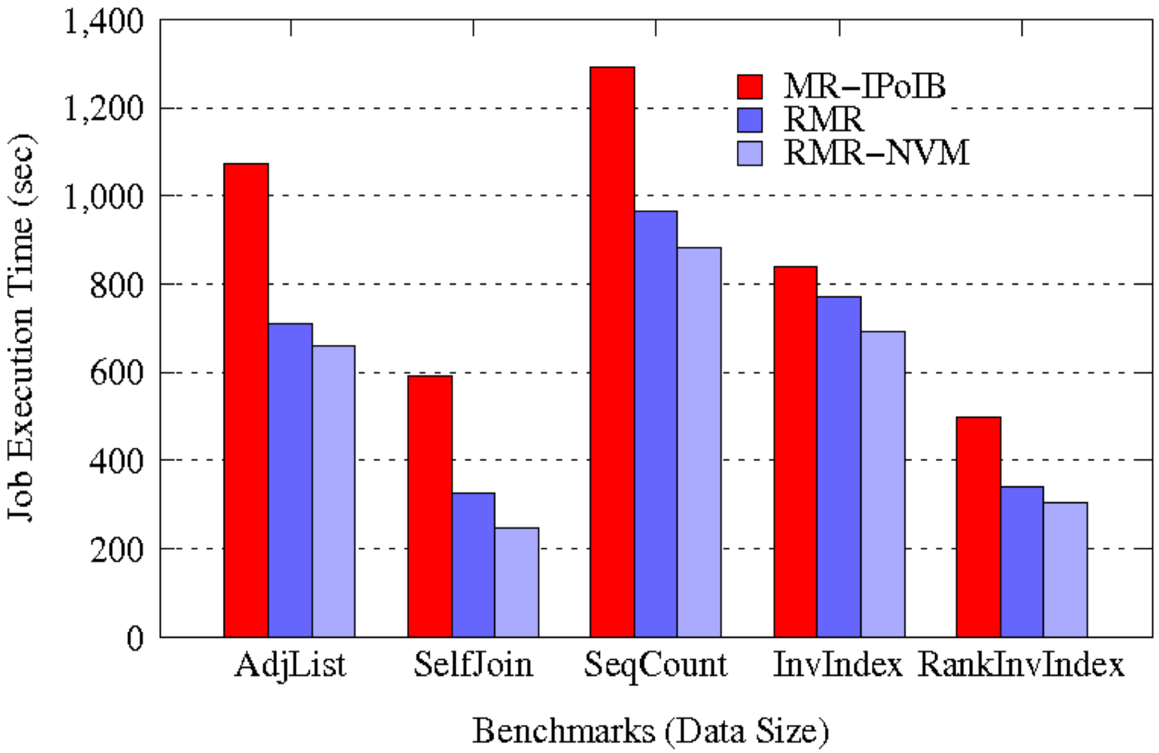
Evaluation of Intel HiBench Workloads

- 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%**

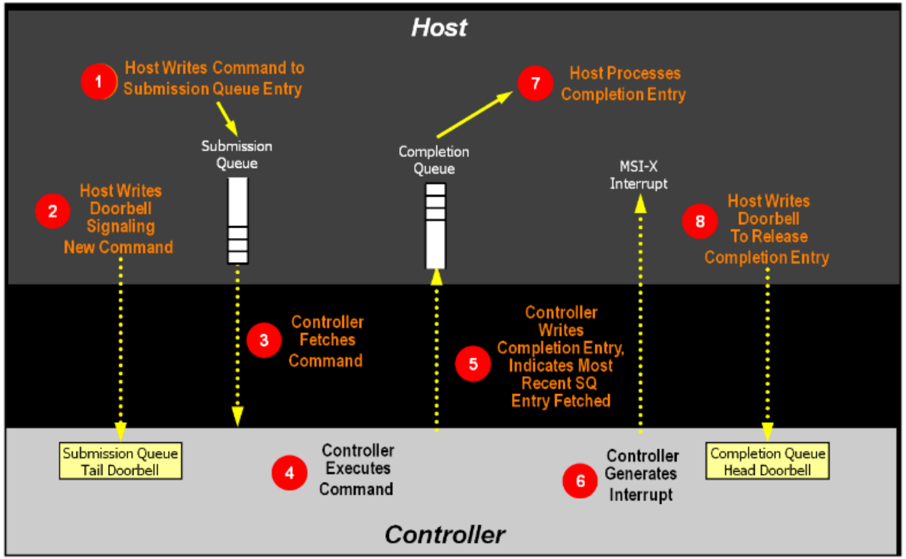


Presentation Outline

- 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

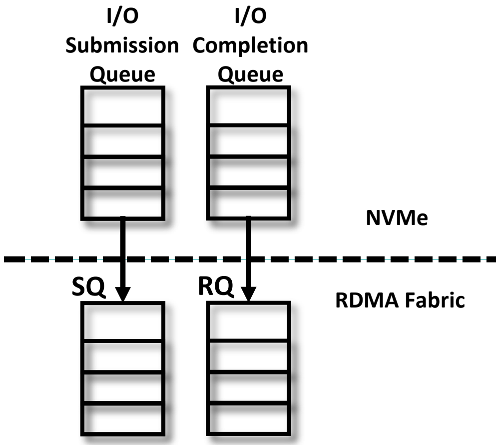


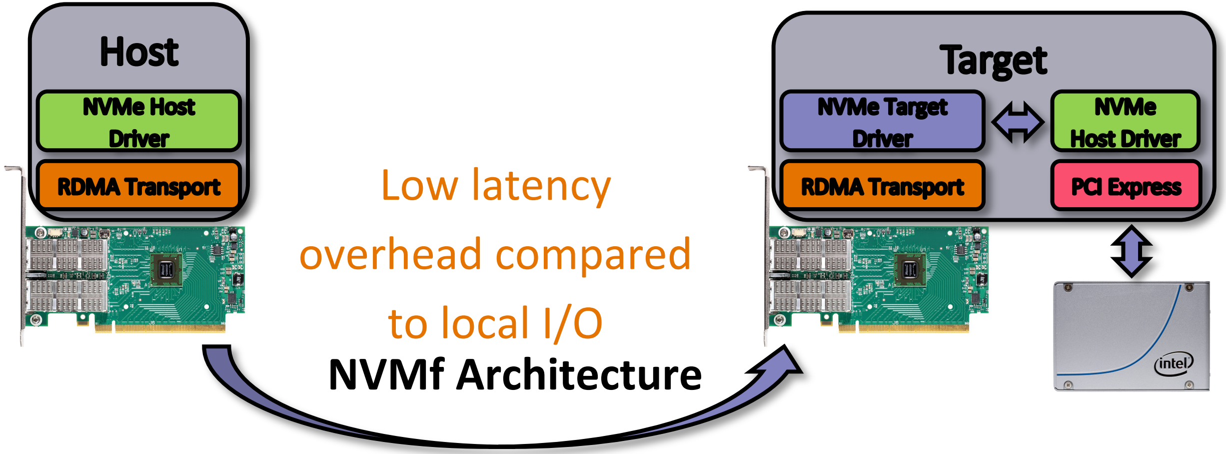
NVMe Command Processing

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

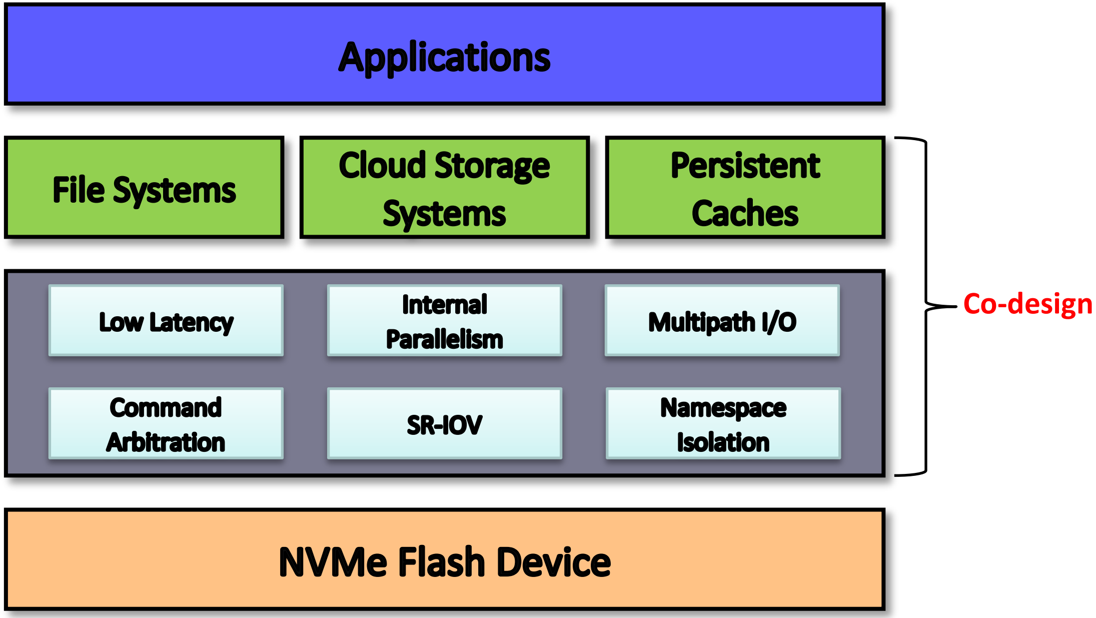




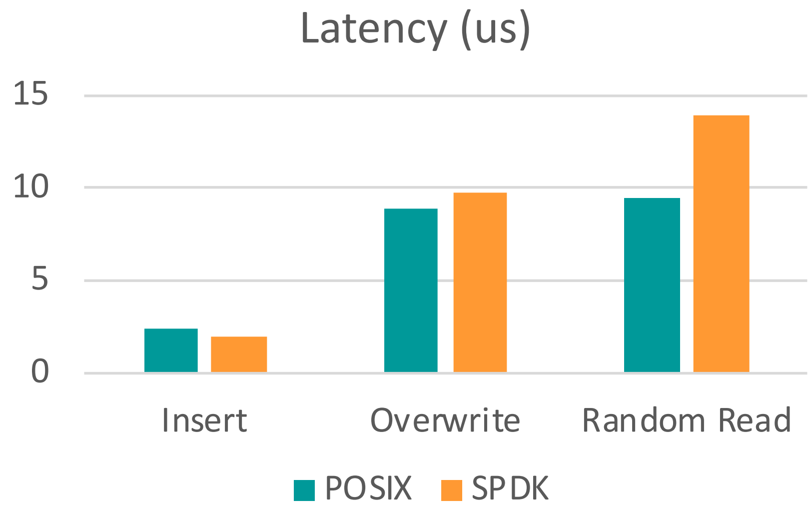
NVMf Architecture

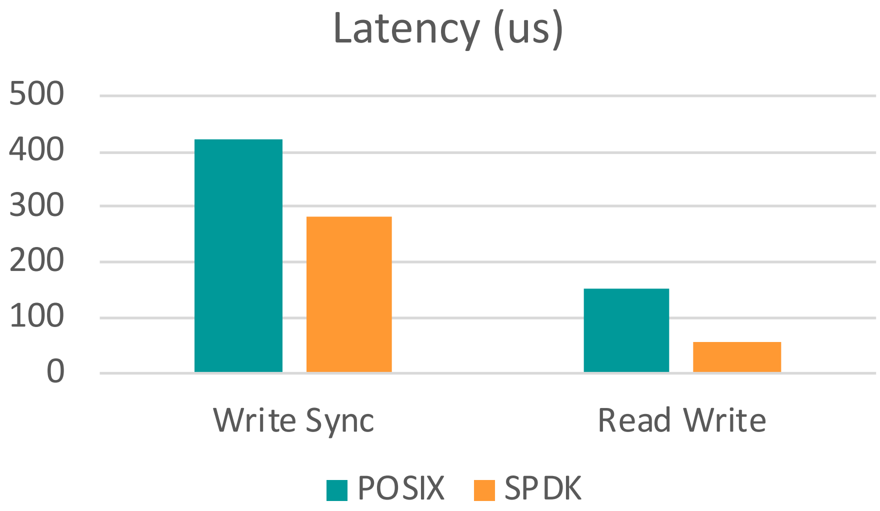
Design Challenges with NVMe-SSD

- 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



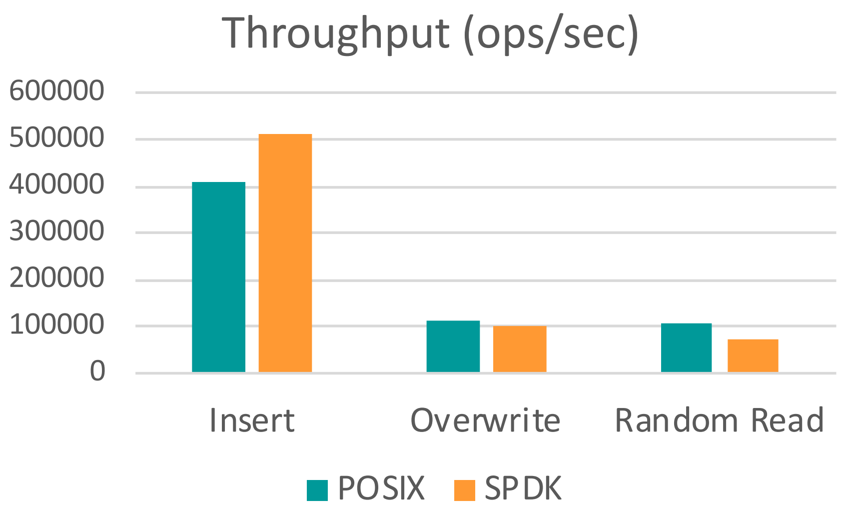
Evaluation with RocksDB

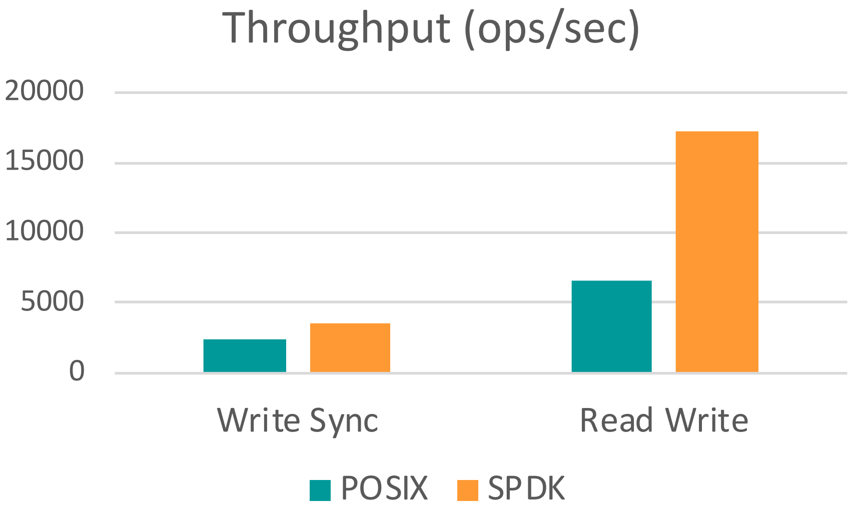




- 20%, 33%, 61% improvement for Insert, Write Sync, and Read Write
- Overwrite: Compaction and flushing in background
  - Low potential for improvement
- Read: Performance much worse; Additional tuning/optimization required

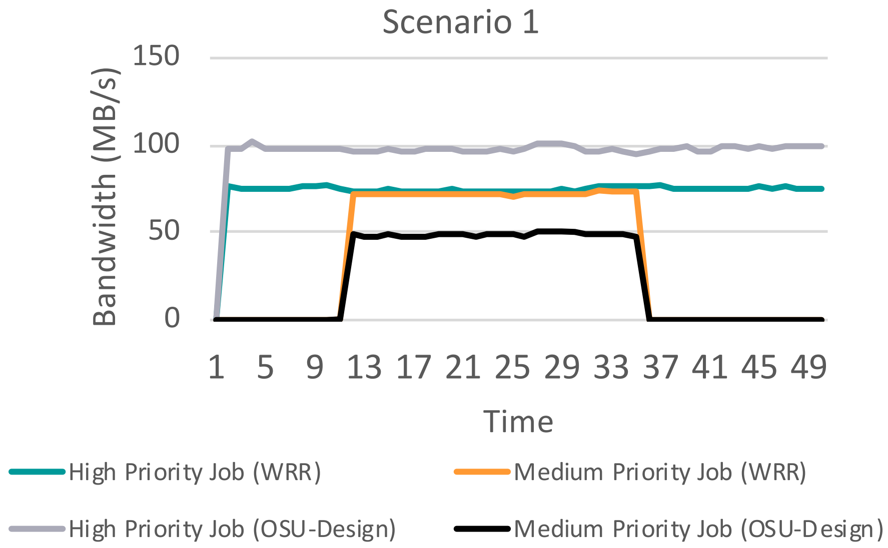
Evaluation with RocksDB

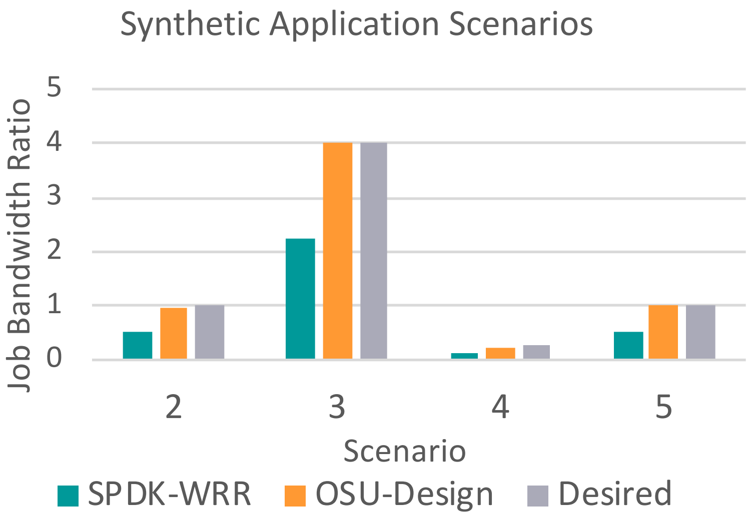




- 25%, 50%, 160% improvement for Insert, Write Sync, and Read Write
- Overwrite: Compaction and flushing in background
  - Low potential for improvement
- Read: Performance much worse; Additional tuning/optimization required

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

- Big Data Analytics needs high-performance NVM-aware RDMA-based Communication and I/O Schemes
- 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
    - TensorFlow, Object Storage, Database, etc.

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/