Can Non-Volatile Memory Benefit MapReduce Applications on HPC Clusters?

Md. Wasi-ur- Rahman, Nusrat Sharmin Islam, Xiaoyi Lu, and Dhabaleswar K. (DK) Panda

Department of Computer Science and Engineering

The Ohio State University

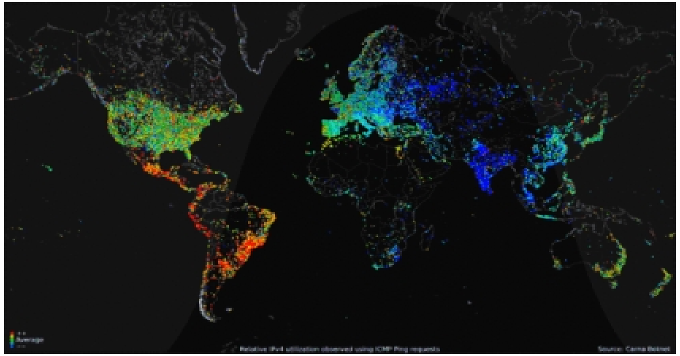
Columbus, OH, USA

Outline

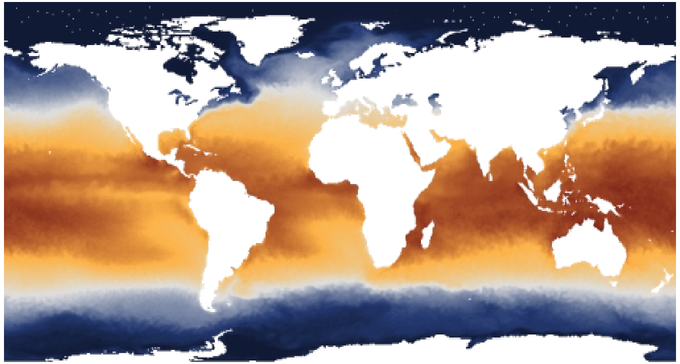
- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
- Performance Evaluation
- Conclusion and Future Work

Introduction

- Big Data has become one of the most important elements in business analytics
- The rate of information growth appears to be exceeding Moore’s Law
- Every day ~2.5 quintillion (2.5×10^{18}) bytes of data are created
- Big Data and High Performance Computing (HPC) are converging to meet large scale data processing challenges
- According to IDC, 67% of HPC centers are running High Performance Data Analysis (HPDA) workloads
- The revenues of these workloads are expected to grow exponentially



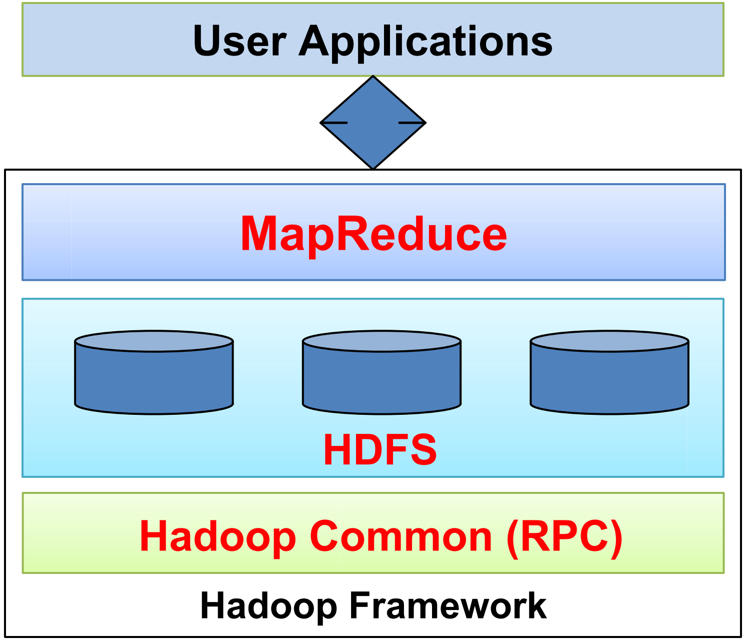
http://www.coolinfographics.com/blog/tag/data?currentPage=3



http://www.climatecentral.org/news/white-house-brings-together-big-data-and-climate-change-17194

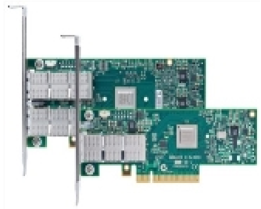
Big Data Processing with Hadoop

- The open-source implementation of MapReduce programming model for Big Data Analytics
- Major components
  - HDFS
  - MapReduce
- Underlying Hadoop Distributed File System (HDFS) can be used by both MapReduce and end applications



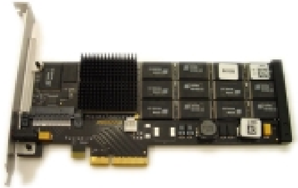
Drivers of Modern HPC Cluster Architectures

Multi-core Processors



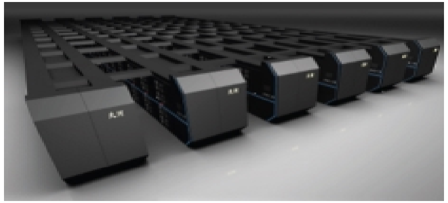
High Performance Interconnects - InfiniBand
<1usec latency, 100Gbps Bandwidth>

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)
- Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), Parallel File Systems
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)



Tianhe – 2

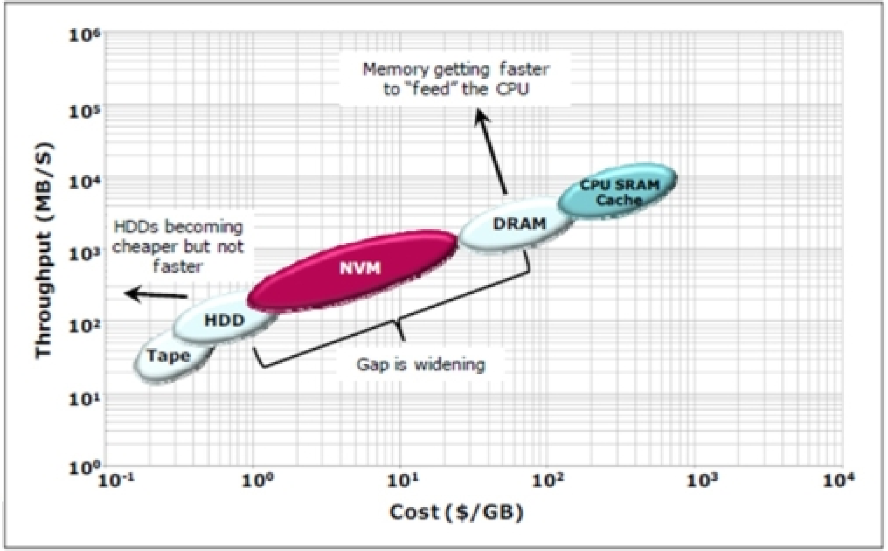
Titan

Stampede

Gordon

Non-Volatile Memory Trends

http://www.slideshare.net/Yole_Developpement/yole-emerging-nonvolatile-memory-2016-report-by-yole-developpement?next_slideshow=2

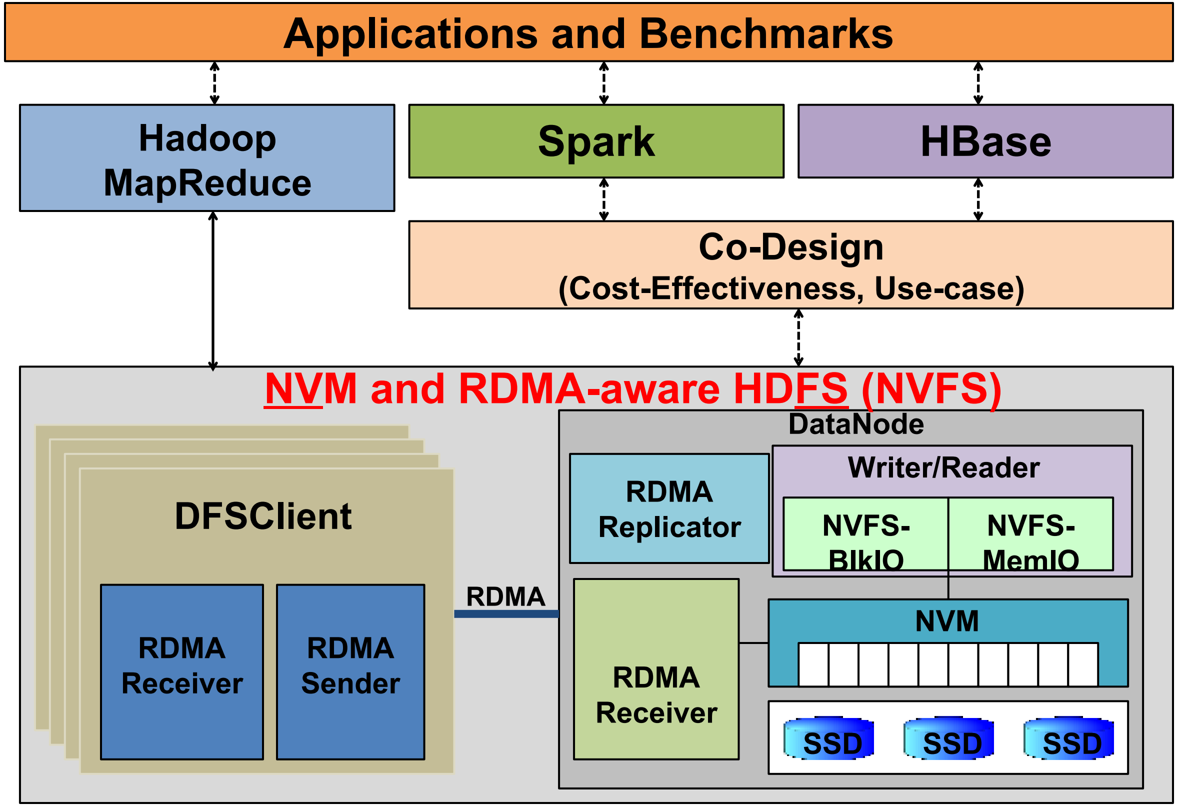


http://www.chipdesignmag.com/bursky/?paged=2

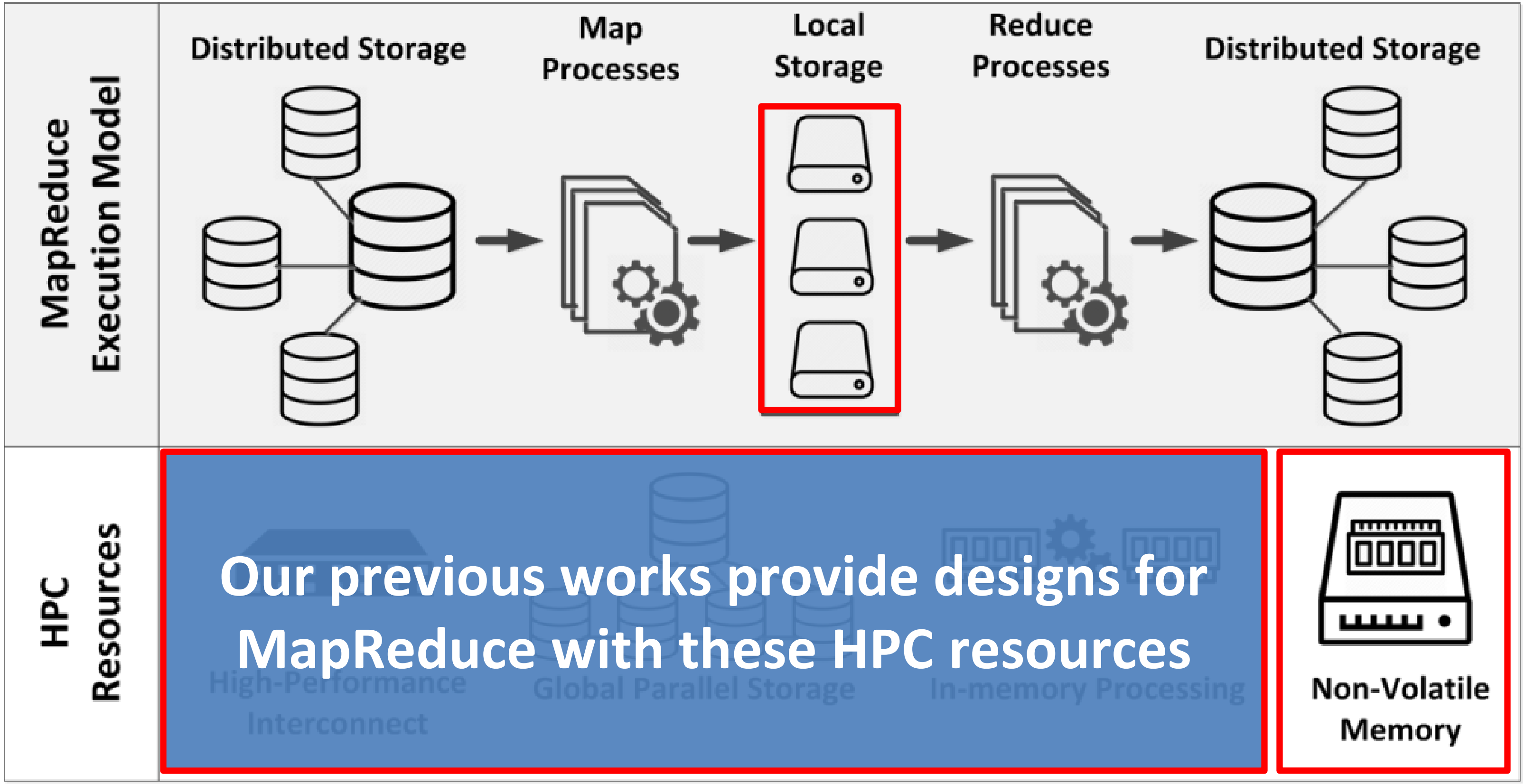
- NVM devices offer DRAM-like performance characteristics with persistence; suitable for data processing middleware
- Number of NVM applications are growing rapidly because of the byte-addressability and persistence features

NVM-aware HDFS

- Our previous work, NVFS provides NVRAM-based designs for HDFS
- Exploits byte-addressability of NVM for communication and I/O in HDFS
- MapReduce, Spark, HBase can obtain better performance for utilizing NVFS as input-output storage
- N. S. Islam, M. W. Rahman, X. Lu, D. K. Panda, *High Performance Design for HDFS with Byte-Addressability of NVM and RDMA*, 24th International Conference on Supercomputing (ICS '16), Jun 2016.



MapReduce on HPC Systems



Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
- Performance Evaluation
- Conclusion and Future Work

Problem Statement

- What are the possible choices for using NVRAM in the MapReduce execution pipeline?

- How can MapReduce execution frameworks take advantage of NVRAM in such use cases?

- Can MapReduce benchmarks and applications be benefitted through the usage of NVRAM in terms of performance and scalability?

Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
- Performance Evaluation
- Conclusion and Future Work

Key Contributions

- Proposed a novel NVRAM-assisted Map Output Spill Approach
- Applied our approach on top of RDMA-based Hadoop MapReduce to keep both map and reduce phase enhancements
- Proposed approach can significantly out-perform the current approaches proven by different sets of workloads

RDMA-enhanced MapReduce

- RDMA-based MapReduce
  - RDMA-based shuffle engine
  - Pre-fetching and caching of intermediate data
  - M. W. Rahman , N. S. Islam, X. Lu, J. Jose, H. Subramoni, H. Wang, and D. K. Panda, *High-Performance RDMA-based Design of Hadoop MapReduce over InfiniBand*, HPDIC, in conjunction with IPDPS, 2013
- Hybrid Overlapping among Phases (HOMR)
  - Overlapping among map, shuffle, and merge phases as well as shuffle, merge, and reduce phases
  - Advanced shuffle algorithms with dynamic adjustments in shuffle volume
  - 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, 2014

These designs are incorporated into the public release of "RDMA for Apache Hadoop" package under HiBD project

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, and HBase Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 195 organizations from 26 countries
- More than 18,600 downloads from the project site
- RDMA for Impala (upcoming)

Available for InfiniBand and RoCE

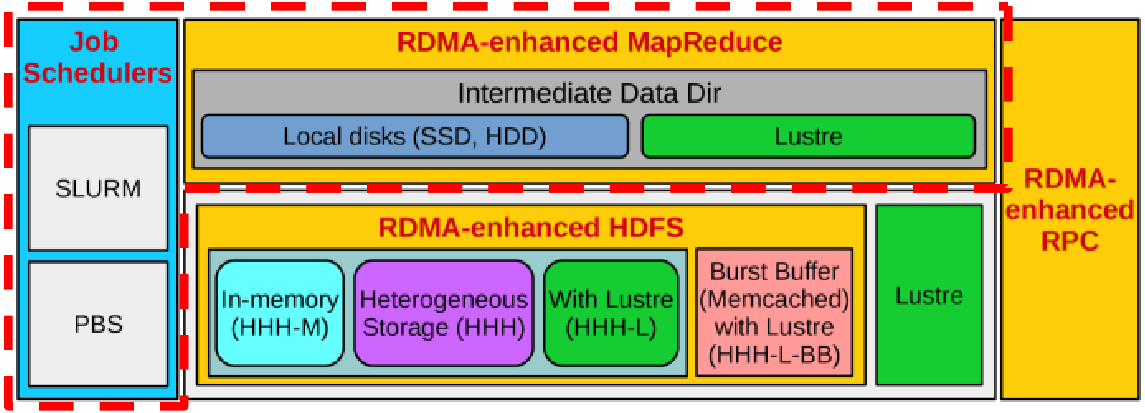
HiBD High-Performance Big Data

Network Based Computing Laboratory

THE OHIO STATE UNIVERSITY

RDMA for Apache Hadoop 2.x

- High-Performance Design of Hadoop over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand and RoCE support at the verbs-level for HDFS, MapReduce, and RPC components
  - Enhanced HDFS with in-memory and heterogeneous storage
  - High performance design of MapReduce over Lustre
  - Plugin-based architecture supporting RDMA-based designs for Apache Hadoop, HDP, and CDH



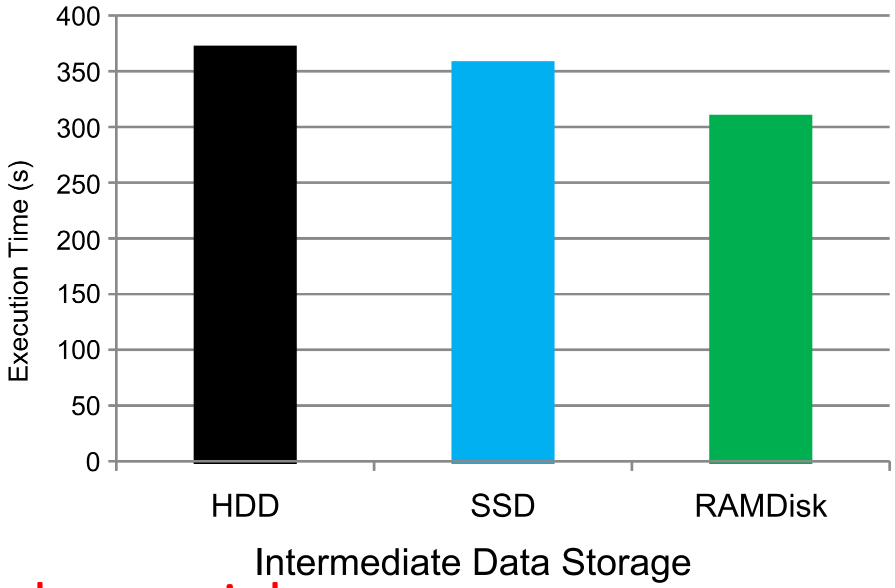
- Current release: 1.1.0
  - Based on Apache Hadoop 2.7.3
  - Compliant with Apache Hadoop 2.7.3, HDP 2.5.0.3, CDH 5.8.2 APIs and applications
  - http://hibd.cse.ohio-state.edu

Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
  - Optimization Opportunities
  - NVRAM-Assisted Map Spilling
- Performance Evaluation
- Conclusion and Future Work

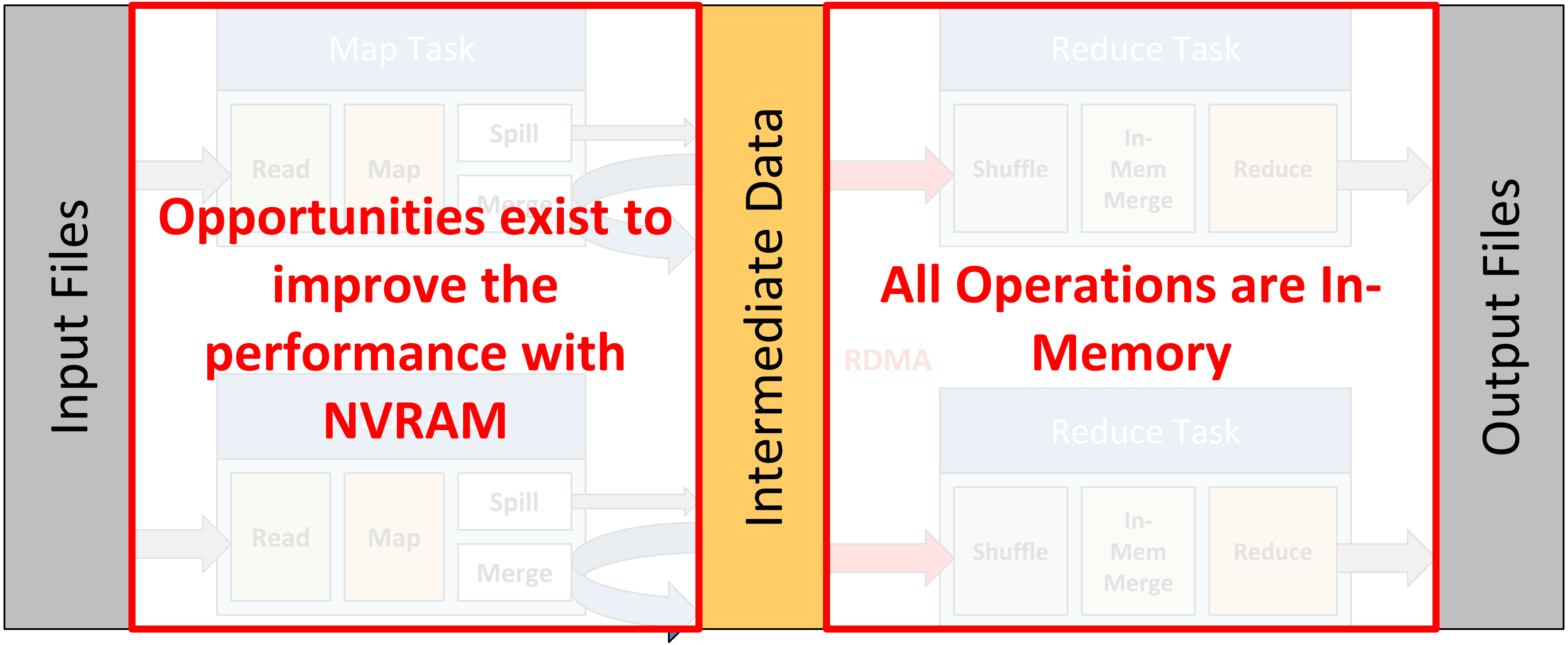
Optimization Opportunities

- Utilizing NVMs as PCIe SSD devices would be straight-forward
  - Configuring the Hadoop local dirs with the NVMe SSD locations
  - No design changes required
- Performance improvement potential with such configuration changes is not high
  - Only improves by **16%** for RAMDisk over HDD as intermediate data storage



- Utilizing NVMs as NVRAM can be crucial

HOMR Design and Execution Flow



Profiling Map Phase

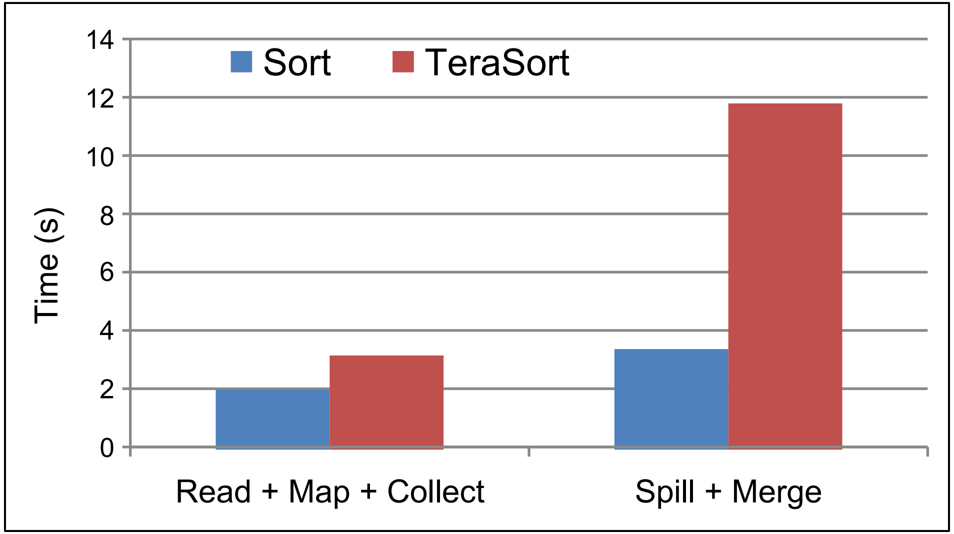
- Map execution performance can be estimated from five different stages

$$t_{Map} = t_{read} + t_{map} + t_{collect} + t_{spill} + t_{merge}$$

- t_{read}: Reading input data from file system
- t_{map}: Applying map() function
- $t_{collect}$: Serialization and Partitioning
- t_{spill}: Spilling key-value pairs to files
- t_{merge}: Merge the spill files and write the data to intermediate storage

Involves disk operations on intermediate data storage (t_{spill} and t_{merge})

Profiling Map Phase

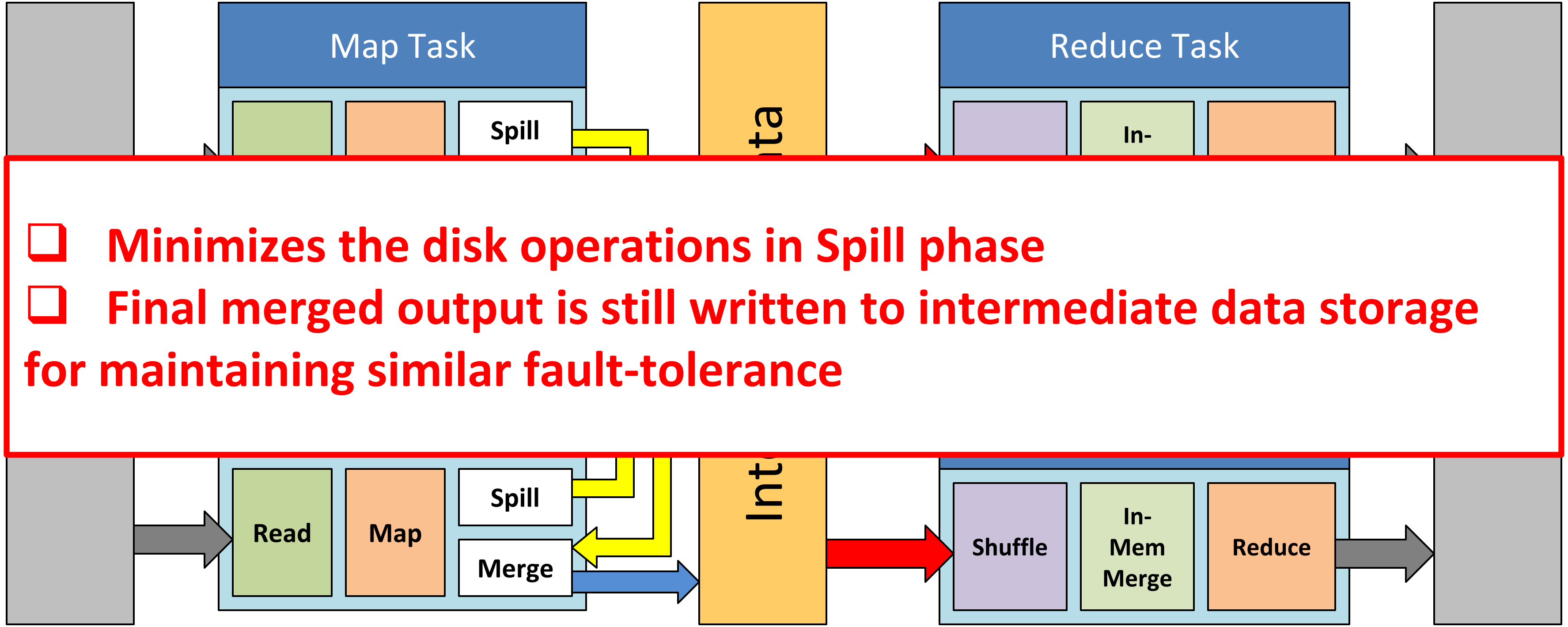


- Profiled 20GB Sort and TeraSort experiments on 8 nodes with default Hadoop
- Averaged over 3 executions
- Spill + Merge takes **1.71x** more time compared to Read + Map + Collect for Sort; for TeraSort, it takes **3.75x** more time

Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
  - Optimization Opportunities
  - NVRAM-Assisted Map Spilling
- Performance Evaluation
- Conclusion and Future Work

NVRAM-Assisted Map Spilling



- ❑ **Minimizes the disk operations in Spill phase**
- ❑ **Final merged output is still written to intermediate data storage for maintaining similar fault-tolerance**

Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
- Performance Evaluation
- Conclusion and Future Work

Experimental Setup

- We have used SDSC-Comet for our evaluation
  - 9 nodes
  - 12-core Intel Xeon E5-2680 v3 (Haswell) processors
  - 128 GB DDR4 DRAM
  - 320 GB local SATA SSD
  - 56 Gbps FDR InfiniBand
- Software and Libraries
  - Hadoop-2.6.0, JDK 1.7
  - RDMA-based Apache Hadoop 0.9.7

Configurations and Notations

- Hadoop configurations used throughout the experiments

Parameter	Value
HDFS Block Size	256 MB
HDFS Data Directory	<SSD Location>
Intermediate Data Directory	<SSD Location>
YARN Concurrent Containers	12

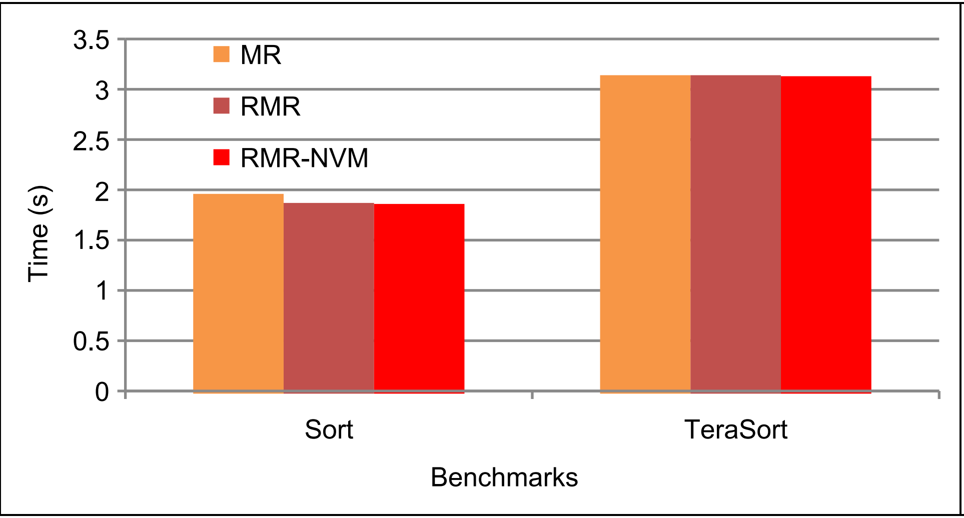
- Notations used in the graphs

Hadoop Repo	Notation Used
Apache Hadoop	MR
RDMA Hadoop	RMR
RDMA Hadoop with NVRAM-Assisted Map Spill (this paper)	RMR-NVM

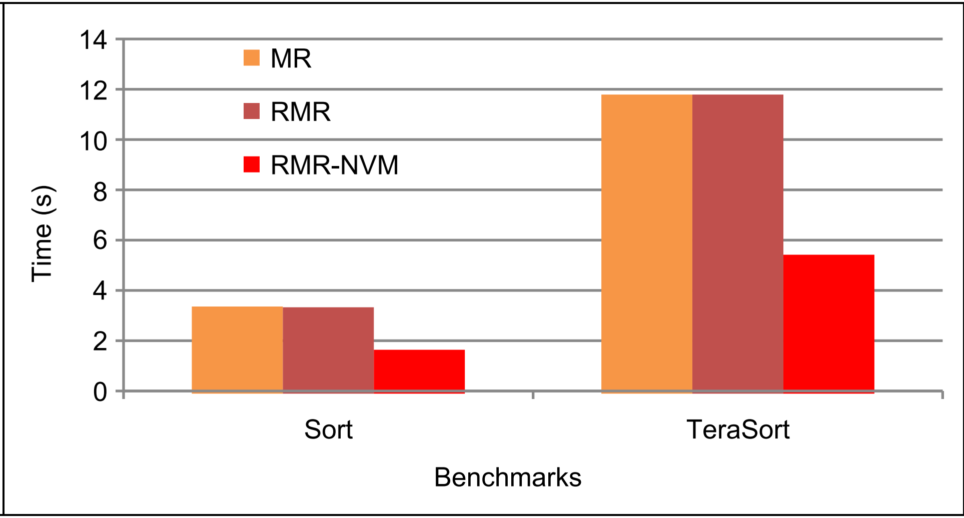
Simulating NVRAM performance

- Because of hardware limitation, we perform simulation to predict NVRAM performance using DRAM
- Assumption: NVRAM write is 10x slower compared to DRAM write; NVRAM read performs similar to DRAM Read
  - NVRAM. http://www.enterprisetech.com/2014/08/06/flashtec-nvram-15-million-iops-sub-microsecondlatency
  - S. Pelley, T. F. Wenisch, B. T. Gold, and B. Bridge. Storage Management in the NVRAM Era. *Proc. VLDB Endow.*, 2013.
- We simulate NVRAM performance by adding a delay (δ) after DRAM write operations
- We utilize `System.nanoTime()` for adding a sleep to simulate δ

Benefits in Map Phase



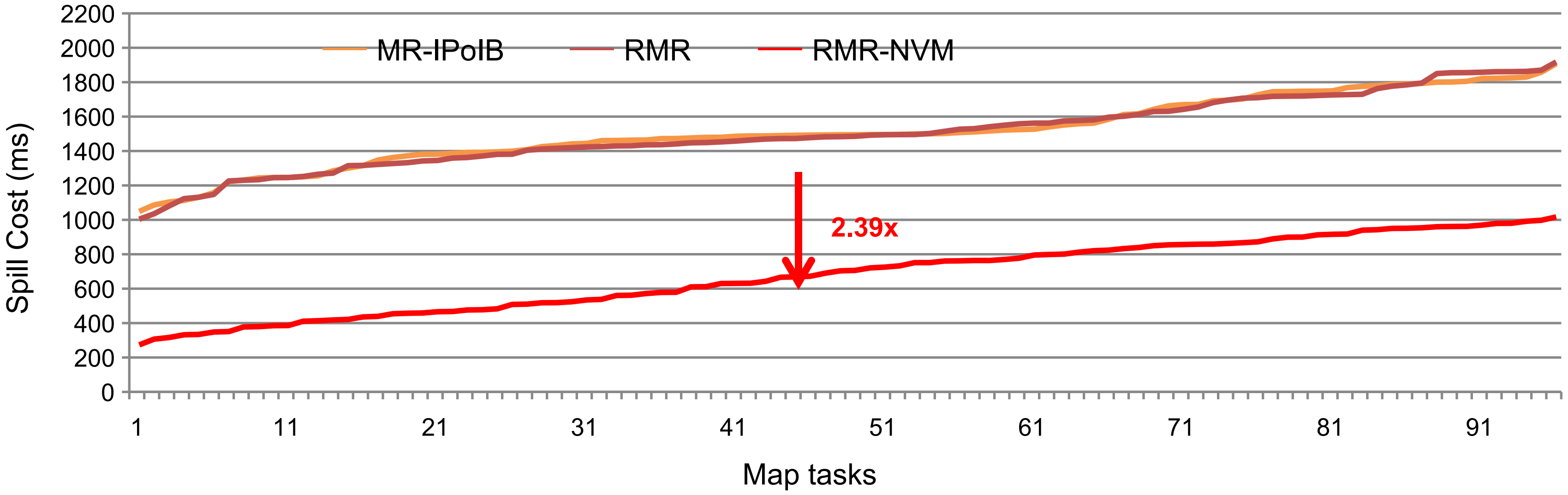
Read + Map + Collect



Spill + Merge

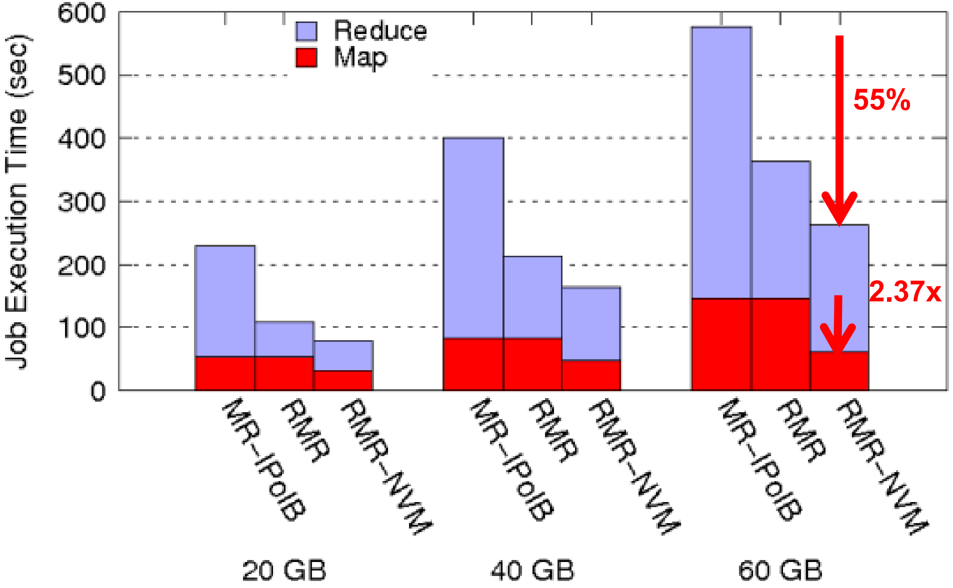
- Read + Map + Collect performs similarly across different MR designs
- Spill + Merge performs significantly better compared to both MR and RMR
- 20 GB Sort and TeraSort experiments on 8 nodes; RMR-NVM Map phase performs at least **2x** better compared to RMR

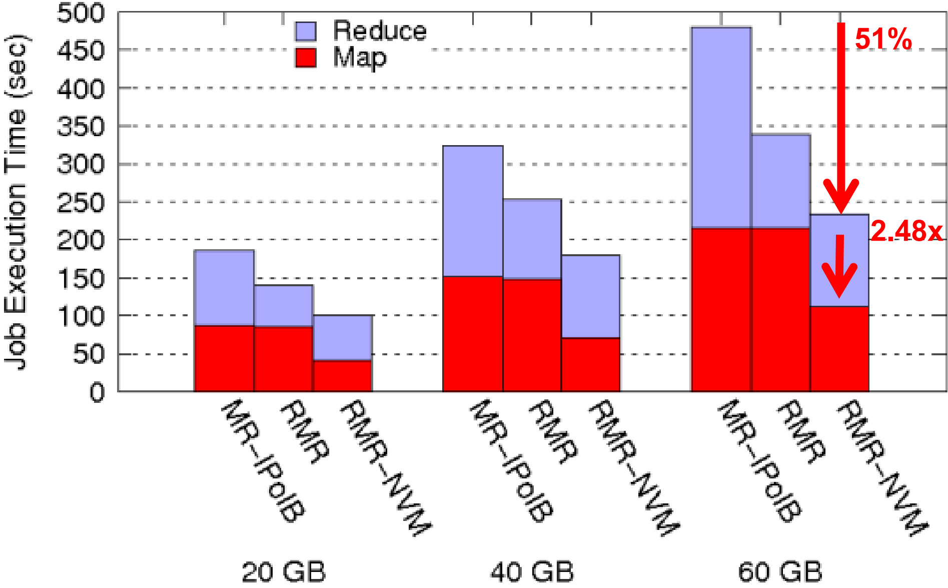
Benefits in Map Phase (Contd.)



- Profiling Map Spill Cost for different MR frameworks
- Sort experiment with 96 maps on 8 nodes
- Sorted spill costs for all maps; averaged over 3 iterations to minimize variation
- Average benefit of **2.39x** is achieved across all maps

Comparison with Sort and TeraSort

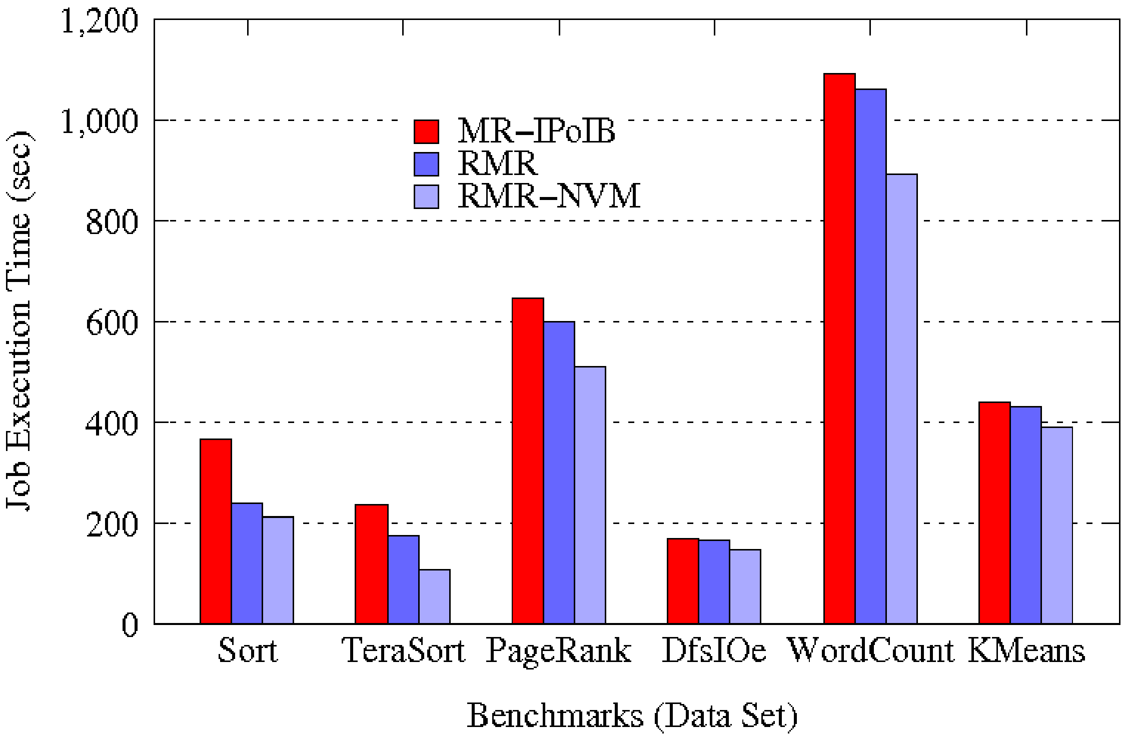




- RMR-NVM achieves **2.37x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **55%** compared to MR-IPoIB, **28%** compared to RMR
- RMR-NVM achieves **2.48x** benefit for Map phase compared to RMR and MR-IPoIB; overall benefit **51%** compared to MR-IPoIB, **31%** compared to RMR

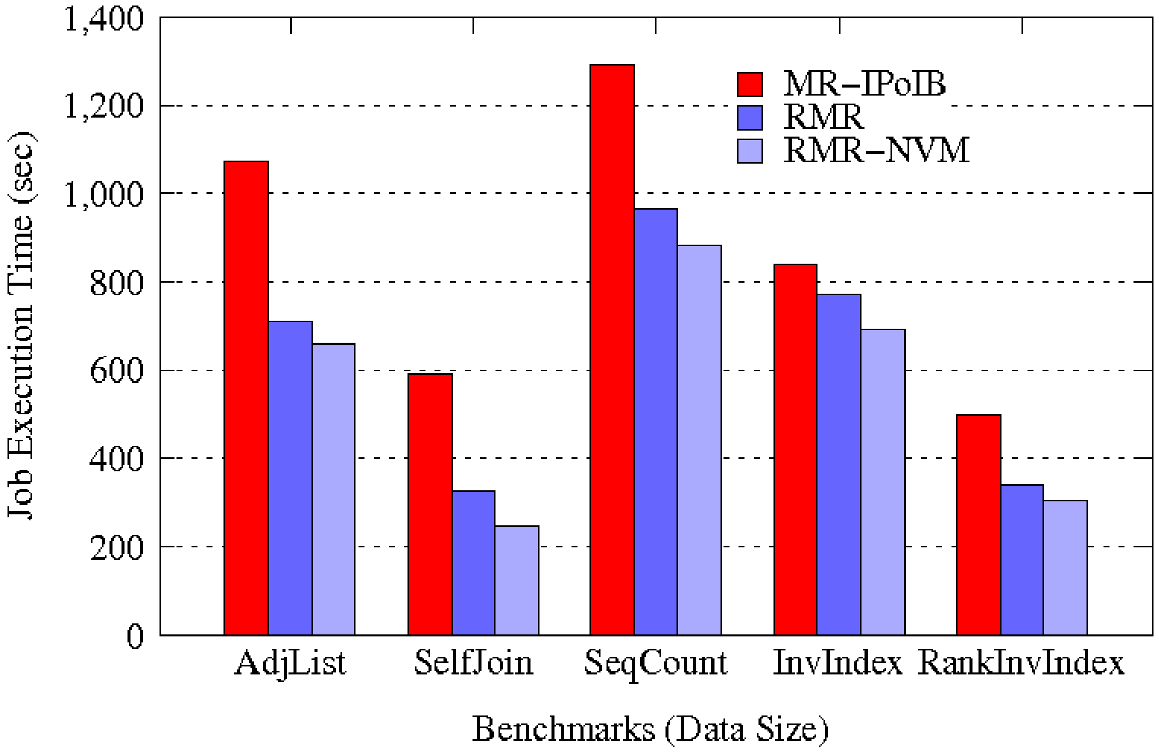
Evaluation of Intel HiBench Workloads

- 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%**



Outline

- Introduction
- Problem Statement
- Key Contributions
- Opportunities and Design
- Performance Evaluation
- Conclusion and Future Work

Conclusion and Future Work

- We propose an enhanced design of MapReduce with NVRAM
- NVRAM-assisted Map Spilling provides significant performance benefits (2.73x) in Map phase compared to previous designs
- Overall, it achieves 55% performance benefits for Sort, 58% for SelfJoin

- This design will be made available in the public release of "RDMA for Apache Hadoop" package under HiBD (http://hibd.cse.ohio-state.edu) project
- In the future, we plan to extend other MapReduce execution frameworks (e.g. Spark, Tez) by leveraging similar design choices with NVRAM

Thank You!

{rahmanmd, islamn, luxi, panda}@cse.ohio-state.edu

Network Based Computing Laboratory

HiBD
High-Performance Big Data

High Performance Big Data

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

Network-Based Computing Laboratory

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