NETWORK-BASED COMPUTING LABORATORY

High Performance File System and I/O Middleware Design for Big Data on HPC Clusters

by

Nusrat Sharmin Islam

Advisor: Dhabaleswar K. (DK) Panda

Department of Computer Science and Engineering

The Ohio State University

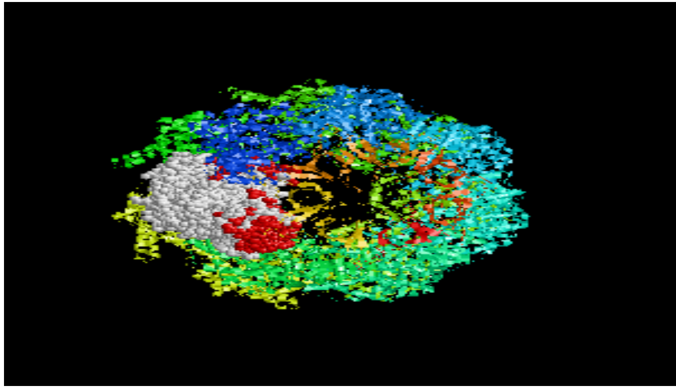
Columbus, OH, USA

OHIO STATE

Introduction

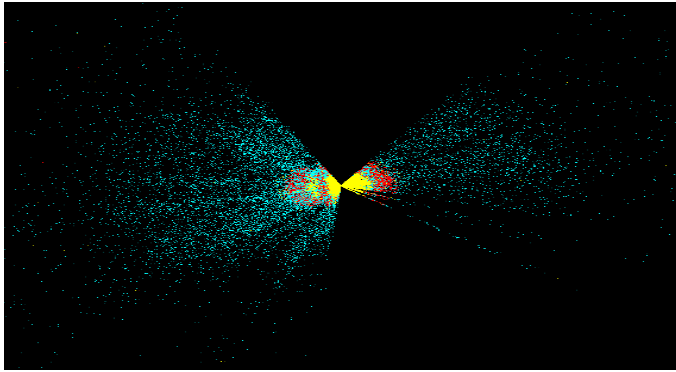
- Big Data provides groundbreaking opportunities for information management and decision making
- The amount of data is exploding; production of data in diverse fields is increasing at an astonishing rate
- IDC claims, digital universe is doubling in size every two years; will multiply 10-fold between 2013 and 2020 [*]
- Not only in internet services, scientific applications in diverse domains like Bioinformatics, Astrophysics, etc. are dealing with Big Data problems

http://sppider.cchmc.org/sppider_doc.html



Bioinformatics

http://complex.elte.hu/astro.html

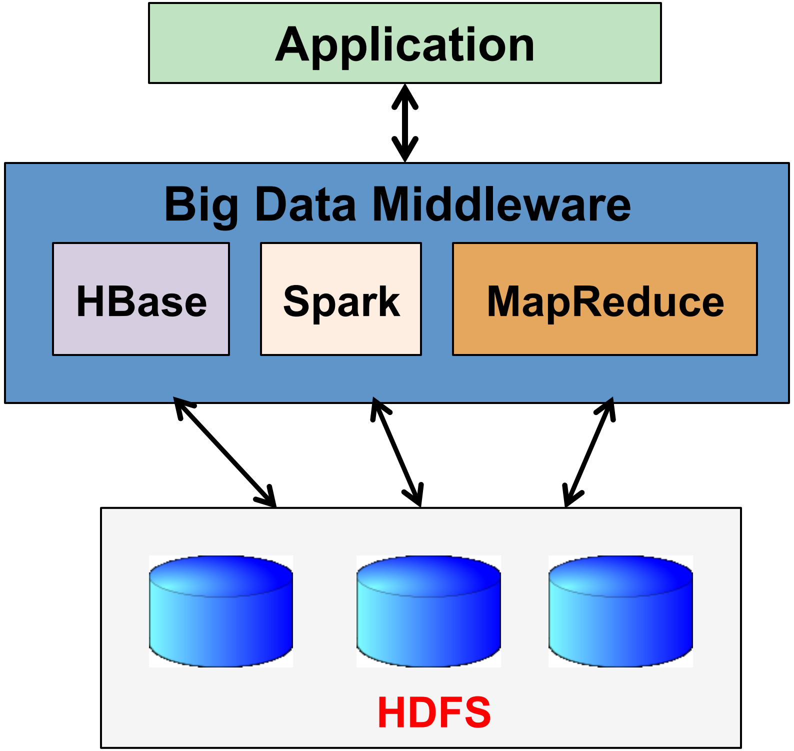


Astrophysics

[*] http://www.csc. com/insights/flxwd/78931-big data universe beginning to explode

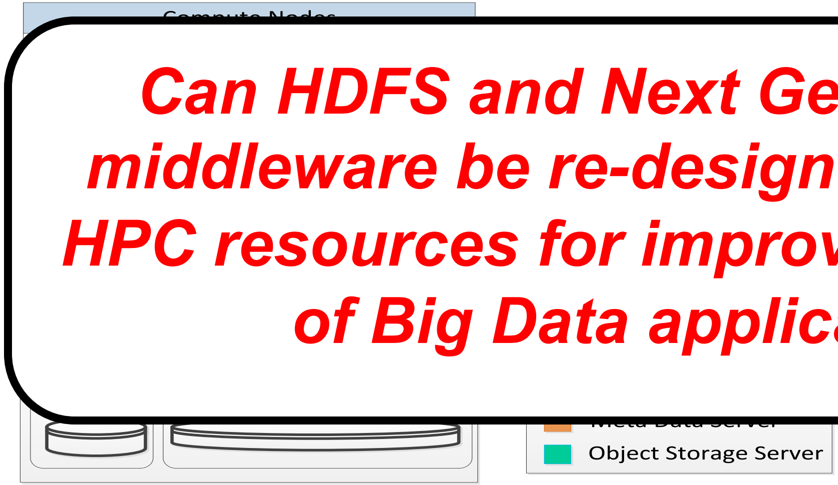
Big Data and Distributed File System

- Hadoop MapReduce and Spark are two popular processing frameworks for Big Data
- Hadoop Distributed File System (HDFS) is the underlying file system of Hadoop, Spark, and Hadoop database HBase
- Adopted by many reputed organizations, e.g. Facebook, Yahoo!
- HDFS, along with the upper-level middleware are being increasingly used on HPC clusters



Deployment and Limitations of HDFS

Interconnect | **Parallel File System** | **Heterogeneous Storage**



Can HDFS and Next Generation File Systems and I/O middleware be re-designed to fully exploit the advanced HPC resources for improving performance and scalability of Big Data applications on HPC systems?

- Requires high volume of local storage due to replication
  - Cannot utilize the parallel file system

- HDFS deployed on the compute cluster
- Big Data jobs co-located with DataNodes

Problem Statement

- Can we re-design HDFS to take advantage of RDMA (Remote Direct Memory Access) with maximized overlapping among different stages of HDFS operation?

- Is it possible to design HDFS with a hybrid architecture to take advantage of the heterogeneous storage devices on HPC clusters? Can we propose effective data placement policies for HDFS to reduce the I/O bottlenecks?

- Can we accelerate Big Data I/O through a key-value store-based burst buffer?

- How can we re-design HDFS to leverage the byte-addressability of NVM?

Research Framework

Big Data Applications, Workloads, Benchmarks

HBase | Hadoop MapReduce | Spark

High Performance File System and I/O Middleware

Hybrid HDFS with Heterogeneous Storage

- Advanced Data Placement
- Selective Caching for Iterative Jobs

KV-Store (Memcached) based Burst Buffer

Leveraging NVM for Big Data I/O

Enhanced Support for Fast Analytics

Maximized Stage Overlapping

RDMA-Enhanced HDFS

Parallel File System (Lustre) | Storage Technologies (HDD, SSD, RAMDisk, and NVM) | Networking Technologies/ Protocols (InfiniBand, 10/40/100 GigE, RDMA)

Major Publications

- RDMA-Enhanced HDFS with Maximized Overlapping
  - 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
  - N. S. Islam, X. Lu, M. W. Rahman, and D. K. Panda, SOR-HDFS: A SEDA-based Approach to Maximize Overlapping in RDMA-Enhanced HDFS, HPDC '14, Short Paper, June 2014
- Hybrid HDFS with Heterogeneous Storage
  - N. S. Islam, X. Lu, M. W. Rahman, D. Shankar, and D. K. Panda, Triple-H: A Hybrid Approach to Accelerate HDFS on HPC Clusters with Heterogeneous Storage Architecture, CCGrid '15, May 2015
  - N. S. Islam, M. W. Rahman, X. Lu, D. Shankar, and D. K. Panda, Performance Characterization and Acceleration of In-Memory File Systems for Hadoop and Spark Applications on HPC Clusters, IEEE BigData '15, October 2015
- Key-value store-based burst buffer for Big Data analytics
  - N. S. Islam, D. Shankar, X. Lu, M. W. Rahman, and D. K. Panda, Accelerating I/O Performance of Big Data Analytics with RDMA-based Key-Value Store, ICPP '15, September 2015
- Leveraging byte-addressability of NVM for HDFS over RDMA
  - N. S. Islam, M. W. Rahman, X. Lu, and D. K. Panda, High Performance Design of HDFS with Byte-Addressability of NVM and RDMA, (ICS '16)

Overview of the HiBD Project and Releases

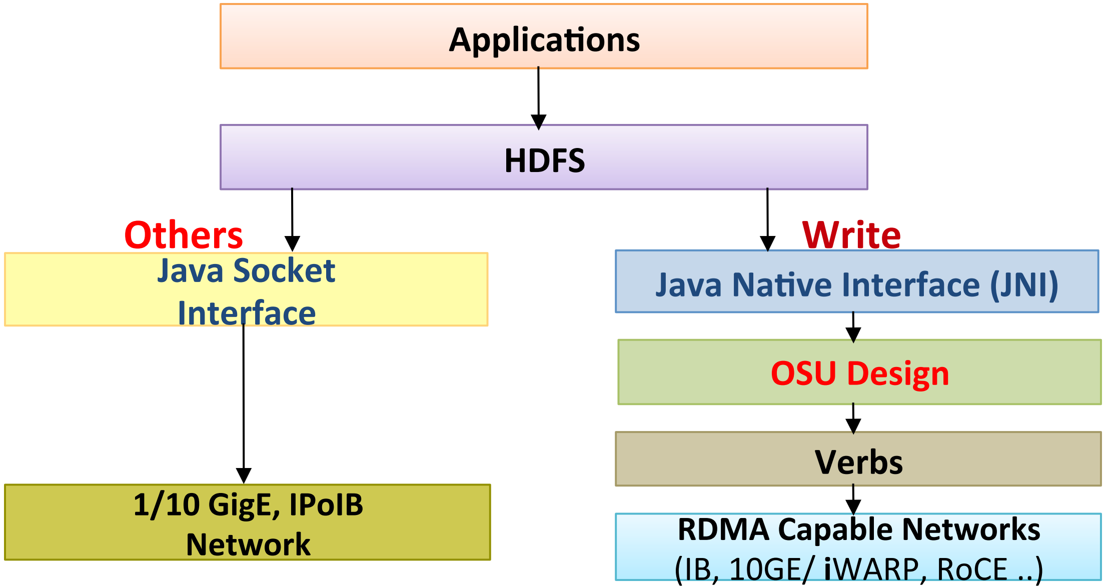
- RDMA for Apache Spark (RDMA-Spark)
- RDMA for Apache HBase (RDMA-HBase)
- **RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)**
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- RDMA for Memcached (RDMA-Memcached)
- OSU HiBD-Benchmarks (OHB)
- **http://hibd.cse.ohio-state.edu**
- Users Base: 190 organizations from 26 countries
- More than 18,200 downloads from project site

High Performance File System and I/O Middleware

- Detailed Designs and Results
  - RDMA-Enhanced HDFS with Maximized Overlapping
  - Hybrid HDFS with Heterogeneous Storage
  - Key-value store-based burst buffer for Big Data analytics
  - Leveraging byte-addressability of NVM for HDFS over RDMA

Design Overview of RDMA-Enhanced HDFS

Enables high performance RDMA communication, while supporting traditional socket interface



HDFS Write involves replication; more network intensive

HDFS Read is mostly node-local

- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - InfiniBand/RoCE support

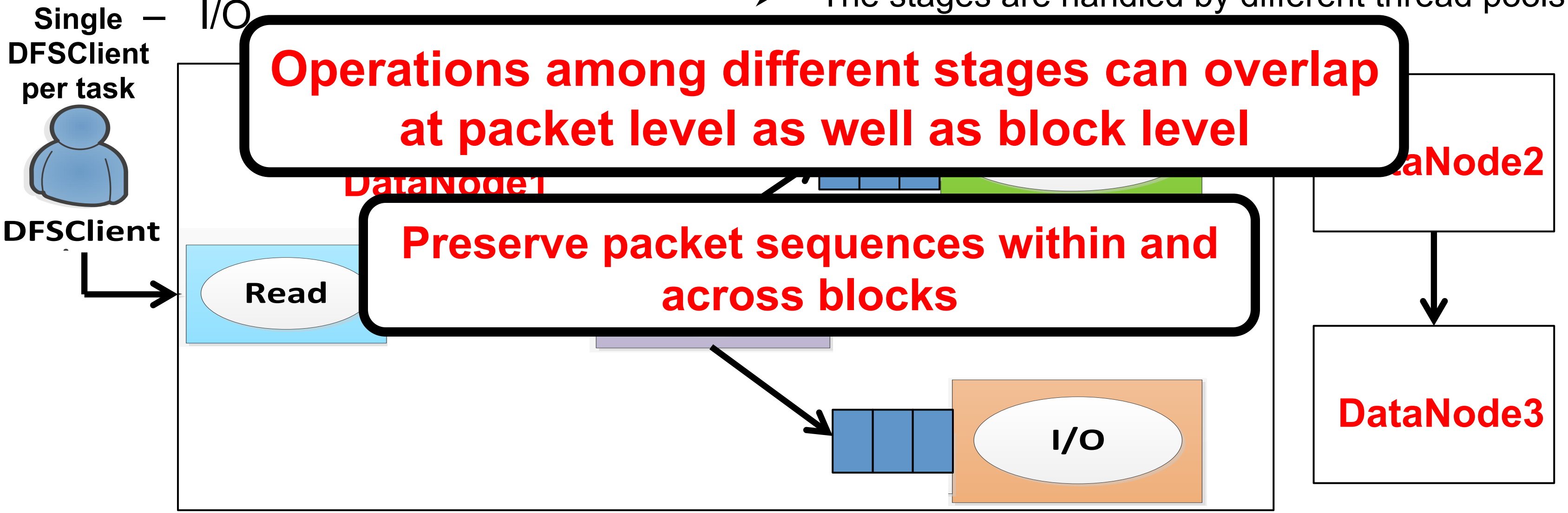
- **JNI Layer bridges Java based HDFS with communication library written in native code**
- Lightweight, high-performance communication library (Unified Communication Runtime (UCR)) to provide advanced network technologies

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

Architectural Overview of SOR-HDFS

- HDFS Write operation go through into four stages in the DataNode side:
  - Read
  - Packet Processing
  - Replication
  - I/O

- **Default (OBOT) architecture:**
  - ➢ Each stage handled sequentially by a single thread per block (no overlapping)
- **Proposed design (SOR-HDFS):**
  - ➢ The stages are handled by different thread pools

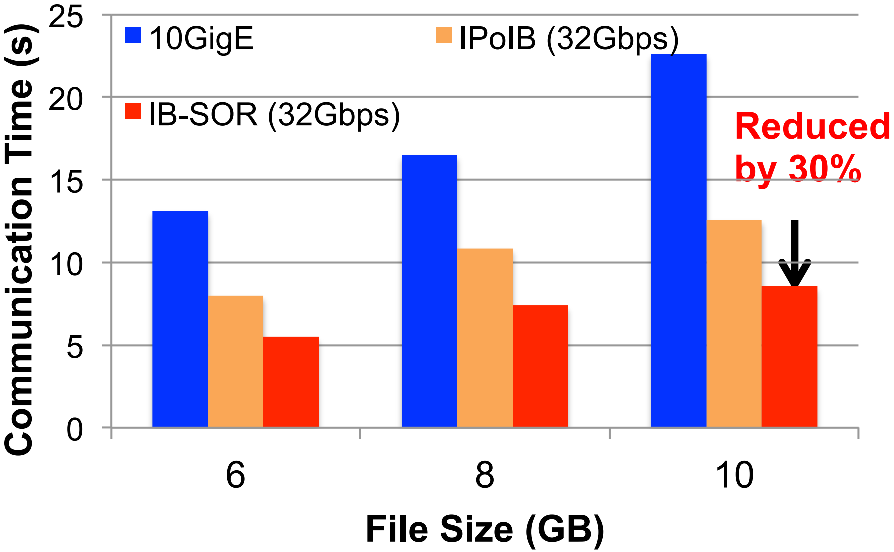


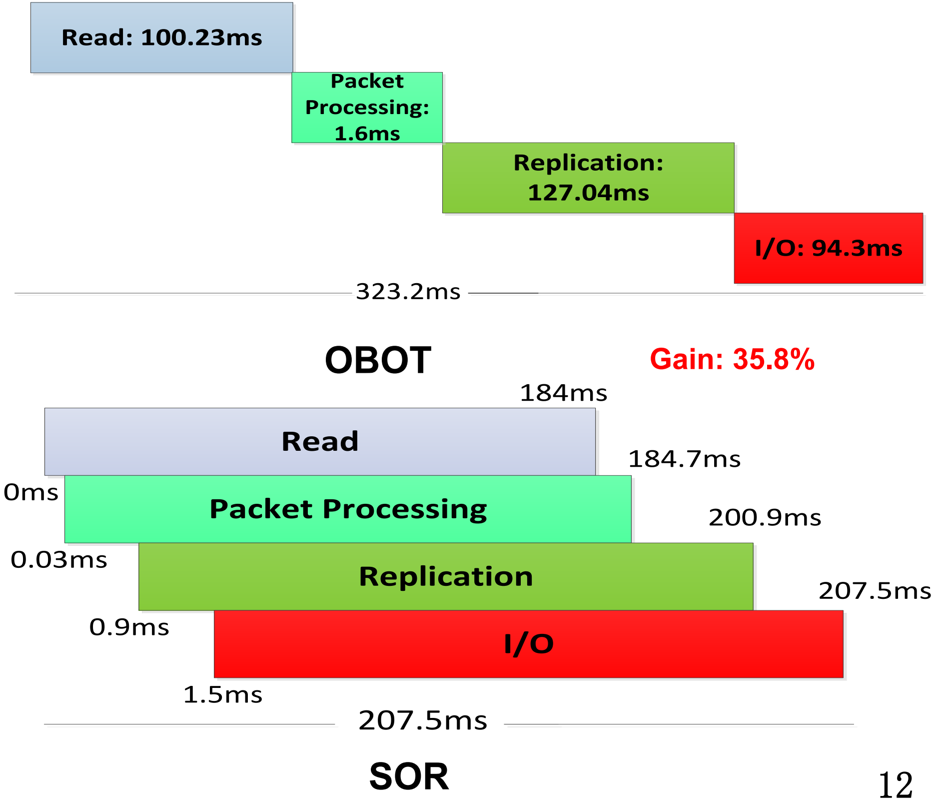
Operations among different stages can overlap at packet level as well as block level

Preserve packet sequences within and across blocks

N. S. Islam, X. Lu, M. W. Rahman, and D. K. Panda, SOR-HDFS: A SEDA-based Approach to Maximize Overlapping in RDMA-Enhanced HDFS, HPDC '14, Short Paper, June 2014

Communication Time and Overlapping Efficiency





- Cluster with 32 DataNodes
  - 30% improvement over IPoIB (QDR)
  - 56% improvement over 10GigE

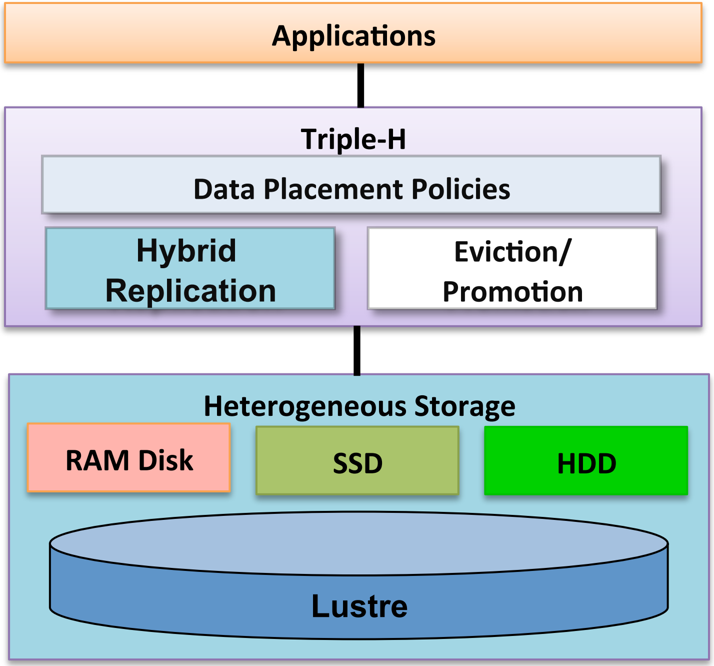
High Performance File System and I/O Middleware

- Detailed Designs and Results
  - RDMA-Enhanced HDFS with Maximized Overlapping
  - Hybrid HDFS with Heterogeneous Storage
  - Key-value store-based burst buffer for Big Data analytics
  - Leveraging byte-addressability of NVM for HDFS over RDMA

Architecture of Triple-H

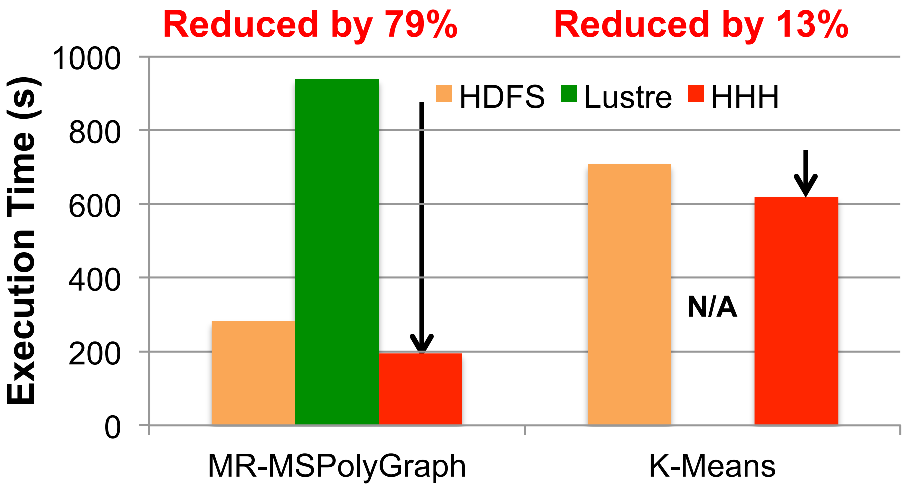
HDFS cannot efficiently utilize the heterogeneous storage devices available on HPC clusters; Limitation comes from the existing placement policies and ignorance of data usage patterns

- A hybrid approach to utilize the heterogeneous storage devices efficiently
- Two modes: Default (HHH), Lustre-Integrated (HHH-L)
- Placement policies to efficiently utilize the heterogeneous storage devices
  - Reduce I/O bottlenecks
  - Save local storage space
- Selective caching for iterative applications



N. S. Islam, X. Lu, M. W. Rahman, D. Shankar, and D. K. Panda, Triple-H: A Hybrid Approach to Accelerate HDFS on HPC Clusters with Heterogeneous Storage Architecture, CCGrid '15, May 2015

Evaluation with Applications

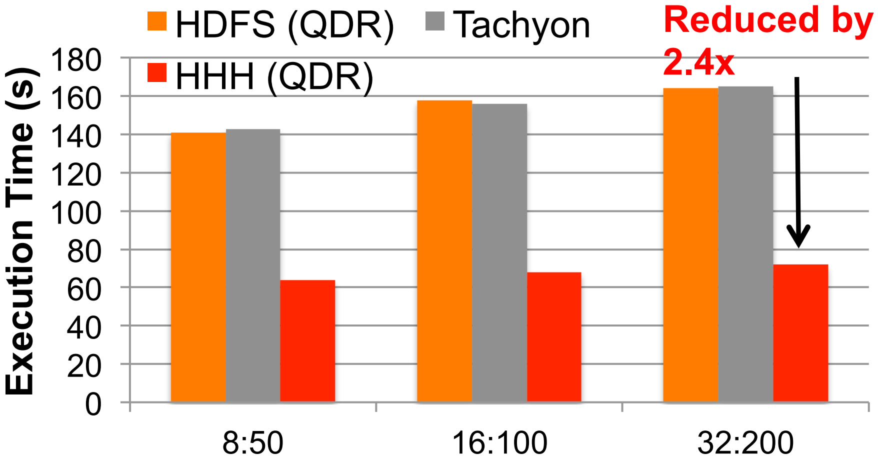


HDFS (FDR)	HHH (FDR)
60.24 s	48.3 s

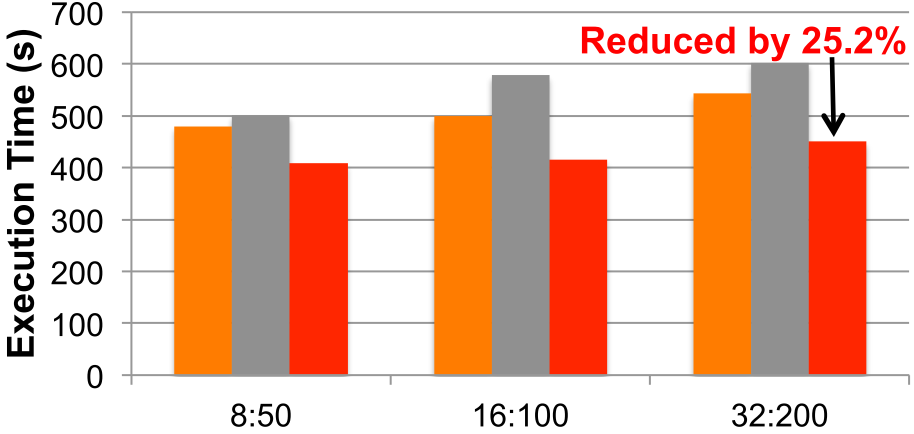
CloudBurst

- MR-MSPolygraph on OSU RI with 1000 maps
  - HHH reduces the execution time by **79%** over Lustre, **30%** over HDFS
- K-Means on 8 nodes on OSU RI with 100 million records
  - HHH reduces the execution time by **13%** over HDFS
- CloudBurst on 16 nodes onTACC Stampede
  - HHH: **19%** improvement over HDFS

Evaluation with Spark and Comparison with Tachyon



TeraGen



TeraSort

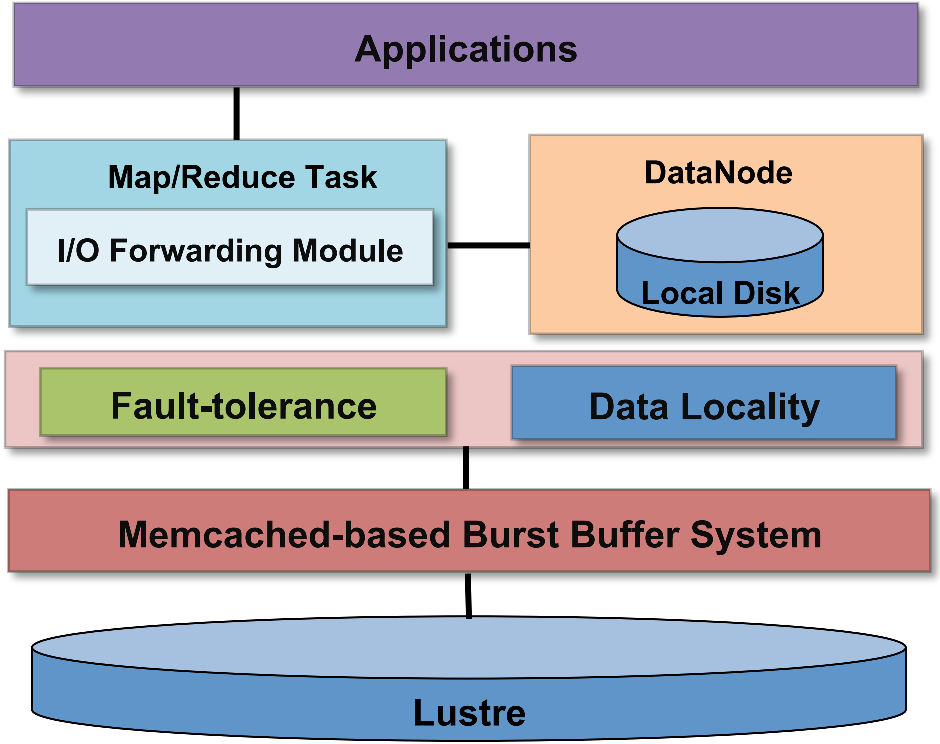
- **For 200GB TeraGen on 32 nodes on SDSC Gordon**
  - Spark-TeraGen: HHH has 2.4x improvement over Tachyon; 2.3x over HDFS (QDR)
  - Spark-TeraSort: HHH has 25.2% improvement over Tachyon; 17% over HDFS (QDR)

N. S. Islam, M. W. Rahman, X. Lu, D. Shankar, and D. K. Panda, Performance Characterization and Acceleration of In-Memory File Systems for Hadoop and Spark Applications on HPC Clusters, IEEE BigData '15, October 2015

High Performance File System and I/O Middleware

- Detailed Designs and Results
  - RDMA-Enhanced HDFS with Maximized Overlapping
  - Hybrid HDFS with Heterogeneous Storage
  - Key-value store-based burst buffer for Big Data analytics
  - Leveraging byte-addressability of NVM for HDFS over RDMA

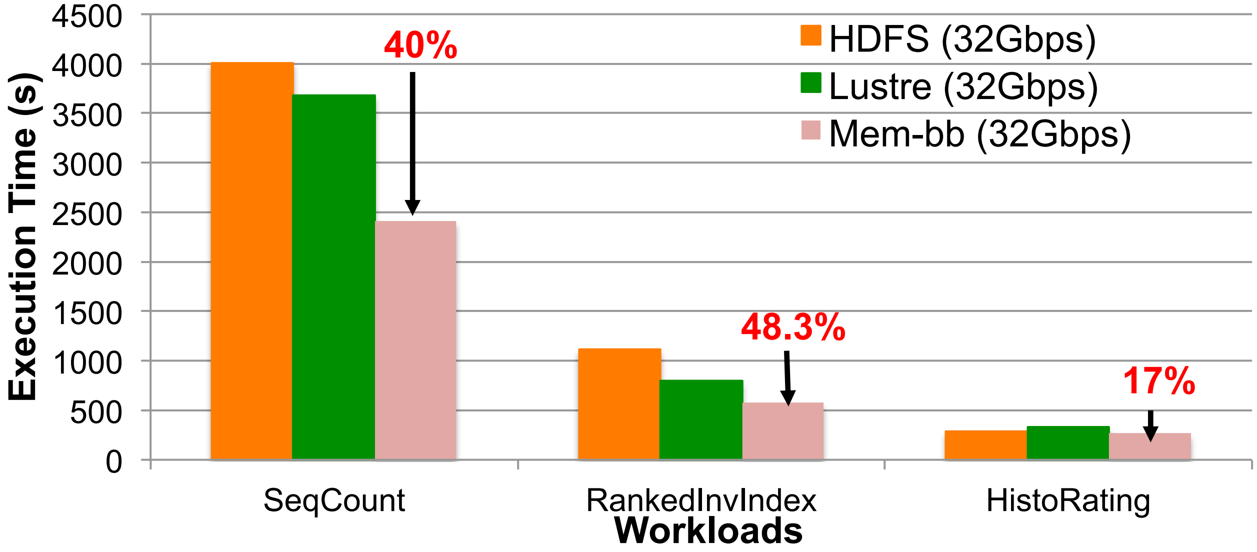
Key-Value Store-based Burst Buffer



- Design Features
  - Memcached-based burst-buffer system
    - Hides latency of parallel file system access
    - Read from local storage and Memcached
  - Data locality achieved by writing data to local storage
  - Different approaches of integration of Hadoop with parallel file system to guarantee fault-tolerance

N. S. Islam, D. Shankar, X. Lu, M. W. Rahman, and D. K. Panda, Accelerating I/O Performance of Big Data Analytics with RDMA-based Key-Value Store, ICPP '15, September 2015

Evaluation with PUMA Workloads



Gains on OSU RI with our approach (Mem-bb) on 24 nodes

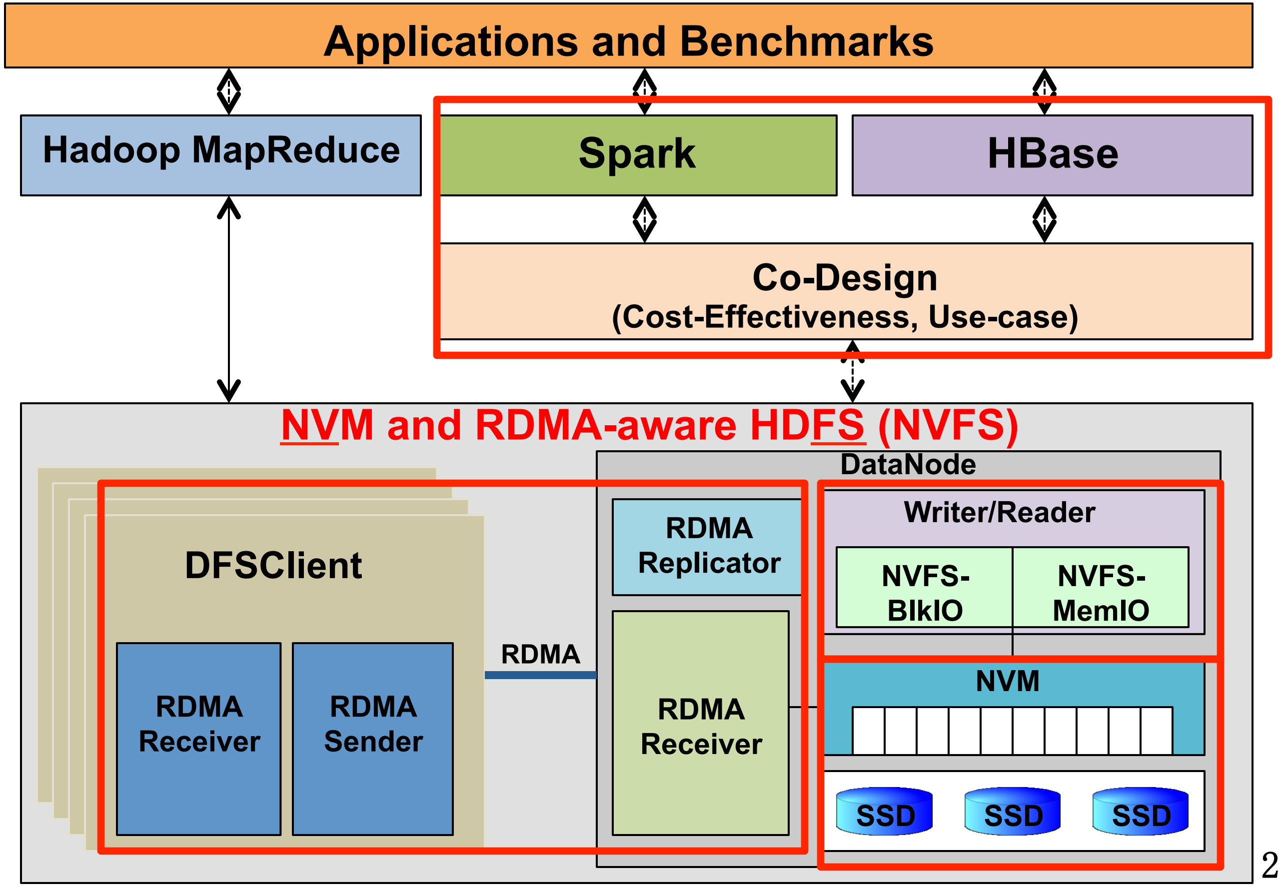
- SequenceCount: 34.5% over Lustre, 40% over HDFS
- RankedInvertedIndex: 27.3% over Lustre, 48.3% over HDFS
- HistogramRating: 17% over Lustre, 7% over HDFS

High Performance File System and I/O Middleware

- Detailed Designs and Results
  - RDMA-Enhanced HDFS with Maximized Overlapping
  - Hybrid HDFS with Heterogeneous Storage
  - Key-value store-based burst buffer for Big Data analytics
  - Leveraging byte-addressability of NVM for HDFS over RDMA

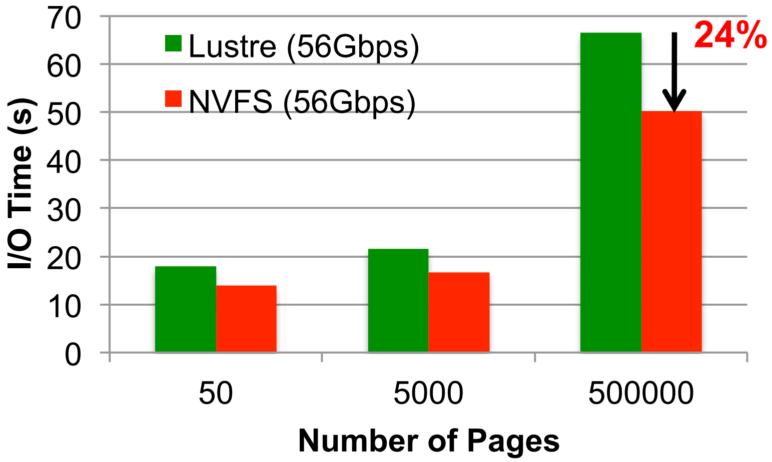
Design Overview of NVM and RDMA-aware HDFS (NVFS)

- RDMA over NVM
- HDFS I/O with NVM
  - NVFS-BlkIO
  - NVFS-MemIO
- Hybrid design
  - NVM with SSD
- Co-Design
  - Cost-effectiveness
  - Use-case (Burst Buffer)

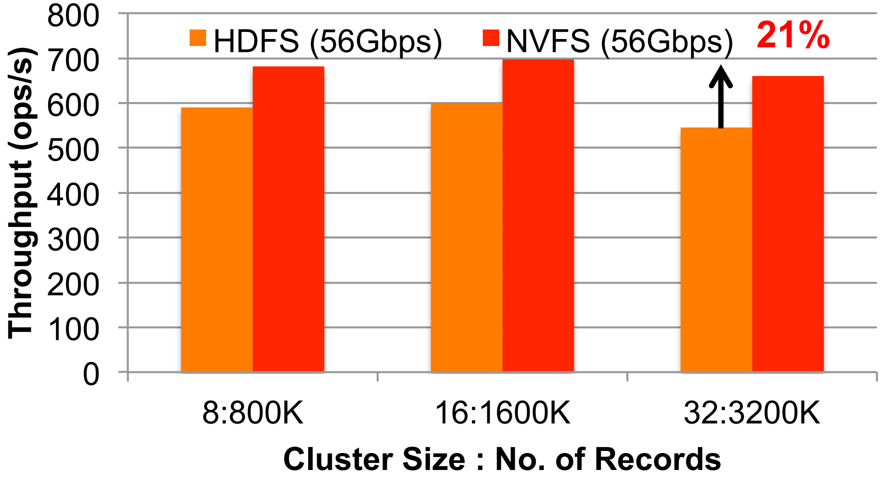


N. S. Islam, M. W. Rahman, X. Lu, and D. K. Panda, High Performance Design for HDFS with Byte-Addressability of NVM and RDMA, ICS '16, June 2016

Evaluation with Spark and HBase



Spark PageRank



HBase 100% insert

- Spark PageRank on SDSC Comet (Burst Buffer)
  - NVFS gains by **24%** over Lustre in I/O time
- HBase 100% Insert on SDSC Comet (32 nodes)
  - NVFS gains by **21%** by storing only WALs to NVM

On-going and Future Work

- Efficient data access strategies for Hadoop and Spark in the presence of high performance interconnects and heterogeneous storage
  - Locality and storage type aware data access
- High performance design of other storage engines (e.g. Kudu) to exploit HPC resources
  - Improve performance of replication over RDMA
  - Utilizing NVM and other heterogeneous storage devices to accelerate random access
- Enhanced computation and I/O subsystem design for deep and machine learning or bioinformatics applications

Conclusion

- Critical to design advanced file system and I/O middleware for Big Data applications on HPC platforms
- Proposed designs address several challenges
  - RDMA-Enhanced HDFS with maximized overlapping
    - Enhances communication performance of HDFS write and replication
  - Hybrid HDFS with in-memory and heterogeneous storage
    - Enhances I/O performance with reduced local storage requirements
  - Key-value store-based burst buffer for Big Data analytics
    - Reduces bottlenecks of shared file system access
  - High performance HDFS design with NVM and RDMA
    - Exploits byte-addressability of NVM for communication and I/O
- Research shows the impact of high performance file system and I/O middleware on upper layer frameworks and end applications
- Designs available in RDMA for Apache Hadoop and RDMA for Memcached software packages from HiBD (http://hibd.cse.ohio-state.edu)

Thank You!

islamn@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/