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

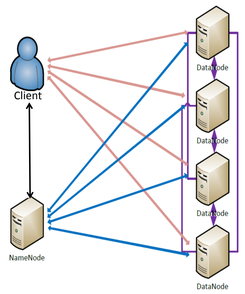
Nusrat Sharmin Islam

Advisor: Dhabaleswar K. (DK) Panda

THE OHIO STATE UNIVERSITY

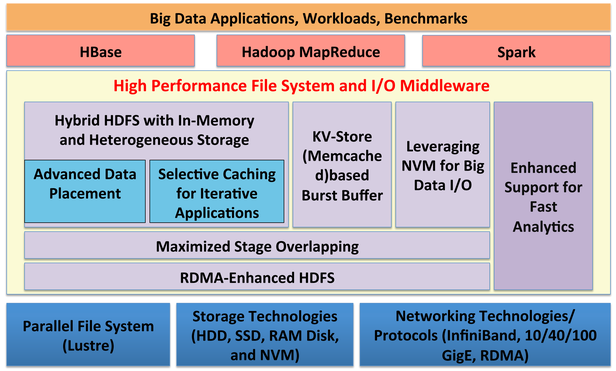
Introduction

- ❖ HDFS is the underlying storage for many Big Data processing framework like, Hadoop MapReduce, HBase, Hive and Spark
- ❖ HDFS has been widely adopted by reputed organizations like Facebook and Yahoo to store petabytes of data
- ❖ Replication is the primary fault tolerance mechanism for HDFS; replication enhances data-locality and read throughput and makes recovery process faster in case of failure
- ❖ HDFS uses Sockets for communication and cannot fully leverage the benefits of high performance interconnects like infiniBand
- ❖ I/O performance in HDFS as well as the requirement of large amount of local disk space due to the tri-replicated data blocks is of major concern for HDFS deployment
- ❖ HDFS cannot take advantage of the heterogeneous storage devices available on HPC clusters

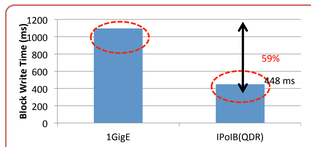


(HDFS Architecture)

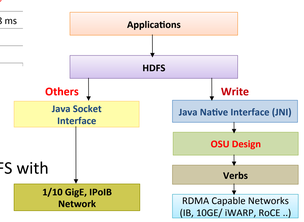
Research Framework

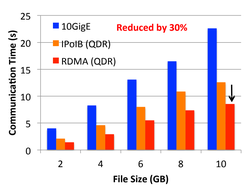


HDFS over RDMA



Enables high performance RDMA communication, while supporting traditional socket interface [1]

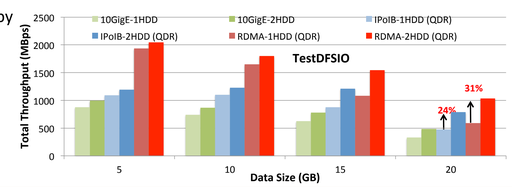


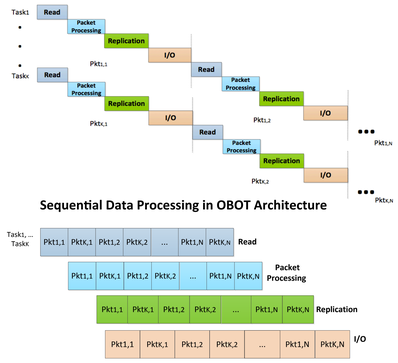


Design Features:

- RDMA-based HDFS write and replication
- JNI Layer bridges Java-based HDFS with native communication library
- On-demand connection setup
- Java Direct Buffer ensures zero copy data transfer

As I/O bottlenecks are reduced, RDMA offers larger benefits than IPoIB and 10GigE





Sequential Data Processing in OBOT Architecture

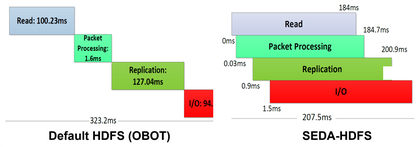
Overlapping in SEDA-based Approach

❖ SEDA-based approach for HDFS Write [2]

- ➢ Re-designs the internal software architecture from OBOT to SEDA
- ➢ Maximizes overlapping among different stages
- ➢ Incorporates RDMA-based communication

❖Four stages: (1) Read (2) Packet Processing (3) Replication (4) I/O

❖No. of threads in each stage is appropriately tuned so that RDMA-enhanced HDFS can make maximum utilization of the system resources



❖Default HDFS adopts One-Block-One-Thread (OBOT) architecture to process data sequentially; good trade-off between simplicity and performance

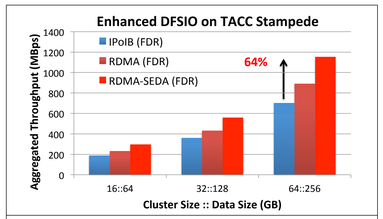
❖In OBOT architecture, incoming data-packets should wait for the I/O stage of the previous packet to be completed before they are read and processed

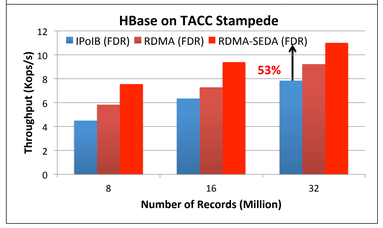
➢ Limits HDFS from fully utilizing the hardware capabilities

❖ In RDMA-Enhanced HDFS, bottleneck moves from data-transmission to data-persistence

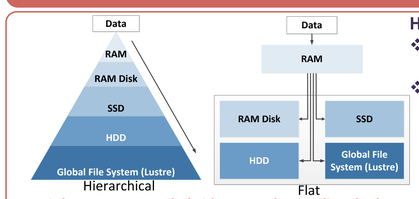
❖ Staged Event-Driven Architecture (SEDA): a high-throughput design approach for Internet services

➢ Decomposes complex processing logic into a set of stages connected by queues





Hybrid HDFS with Heterogeneous Storage

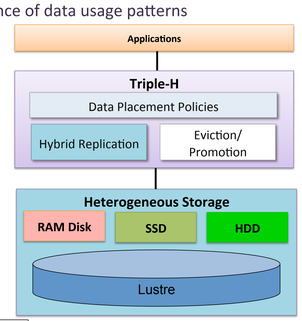


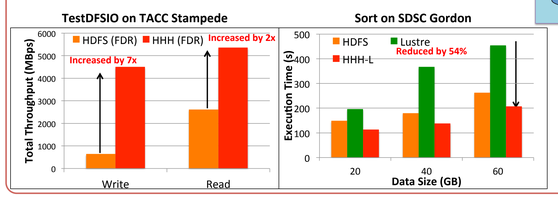
HDFS with Heterogeneous Storage:

- ❖ 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

Triple-H proposes a hybrid approach to utilize the heterogeneous storage devices efficiently [3, 4, 5]

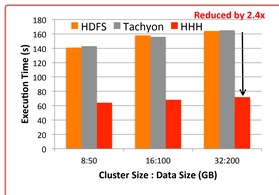
- Two modes: (1) Default (HHH) (2) Lustre-Integrated (HHH-L)
- RAM Disk and SSD –based buffer-cache
- Placement policies to efficiently utilize the heterogeneous storage devices, Hybrid Replication
- Eviction/Promotion based on data usage pattern
- Lustre-Integrated mode: Lustre-based fault-tolerance
- Caching hierarchical, placement hybrid
- Key-Value Store-based burst buffer





HHH-L reduces local storage requirement

Storage Used (GB)	
HDFS	360
Lustre	120
HHH-L	240



Spark TeraGen on SDSC Gordon

Batch Applications		
CloudBurst	HDFS 60.24s	HHH 48.3s
MR-MS Polygraph	Lustre 939s	HHH-L 196s

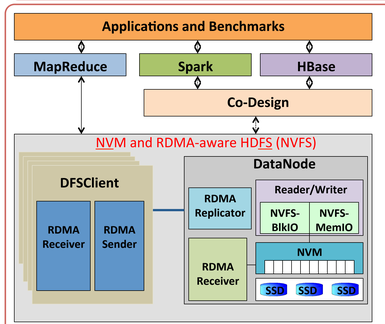
Iterative Application (KMeans)		
HDFS 708s	Tachyon 672s	HHH 618s

Selective caching: Cache output of intermediate iterations

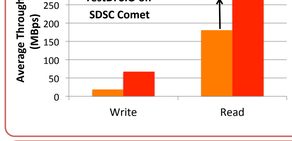
Gain for CloudBurst: 19% over HDFS
Gain for MR-MSPolygraph: 79% over Lustre
HHH-L offers better data locality over Lustre

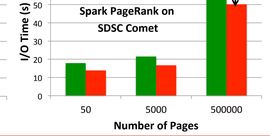
Gain over HDFS: 13%
Gain over Tachyon: 9%

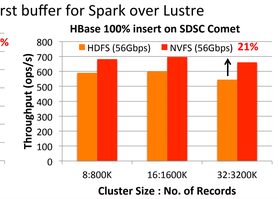
NVM-based HDFS



- For in-memory storage in HDFS, persistence is challenging; competition for memory between computation and I/O
- NVM (byte-addressable and non-volatile) emerging in HPC systems; Critical to rethink HDFS architecture to exploit NVM along with RDMA
- **Exploit byte-addressability of NVM for HDFS communication and I/O [6]**
- **Re-design HDFS storage architecture with memory semantics**
- RDMA over NVM: (D (DRAM)-to-N (NVM), N-to-D, N-to-N)
- NVM-based I/O (Block Access, Memory Access)
- Hybrid Design (NVM with SSD)
- Co-Design (Cost-effectiveness, Use-case)
  - Stores only Write Ahead logs (WALs) to NVM
  - Stores only job output of Spark to NVM
  - NVFS as a burst buffer for Spark over Lustre







Conclusion and Future Work

- ❖ The proposed framework improves communication and I/O performance of HDFS and leads to maximized overlapping and storage space savings on modern clusters
- ❖ In the future, we would like to propose efficient data access strategies for Hadoop and Spark applications on HPC systems
- ❖ We will also evaluate our proposed designs using different Big Data applications

Publications

[1] 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. In Proceedings of SC, November 2012

[2] 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

[3] 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

[4] 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

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

[6] 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

Software Distribution

- ❖ The designs proposed in this research are available for the community in RDMA for Apache Hadoop package from the High Performance Big Data (HiBD) project (http://hibd.cse.ohio-state.edu)
- ❖ As of September '16, more than 18,200 downloads (more than 190 organizations in 26 countries) have taken place from the project's website

Acknowledgements

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

HiBD High-Performance Big Data
High-Performance Big Data (HiBD)
http://hibd.cse.ohio-state.edu

The Ohio State University
College of Engineering

This research is supported in part by National Science Foundation grant #IIS-1447804.