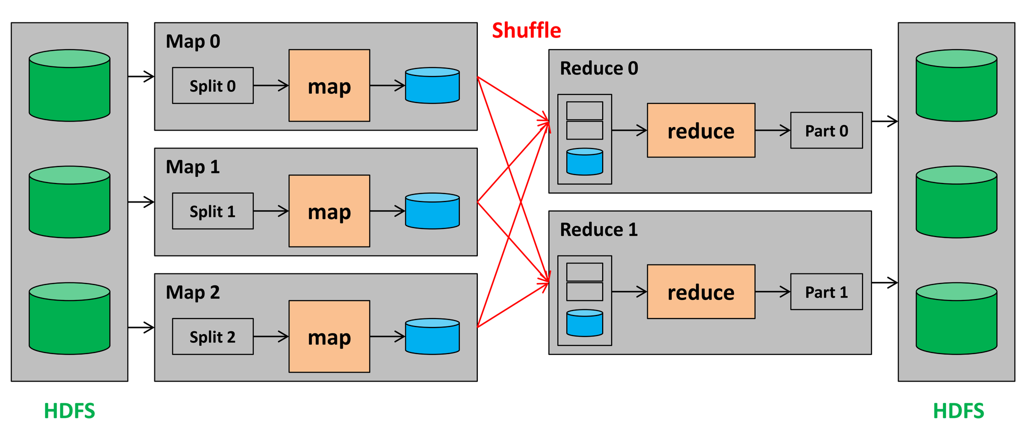
Designing and Modeling High-Performance MapReduce and DAG Execution Framework on Modern HPC Systems

Md. Wasi-ur- Rahman, Advisor: Dhabaleswar K. (DK) Panda

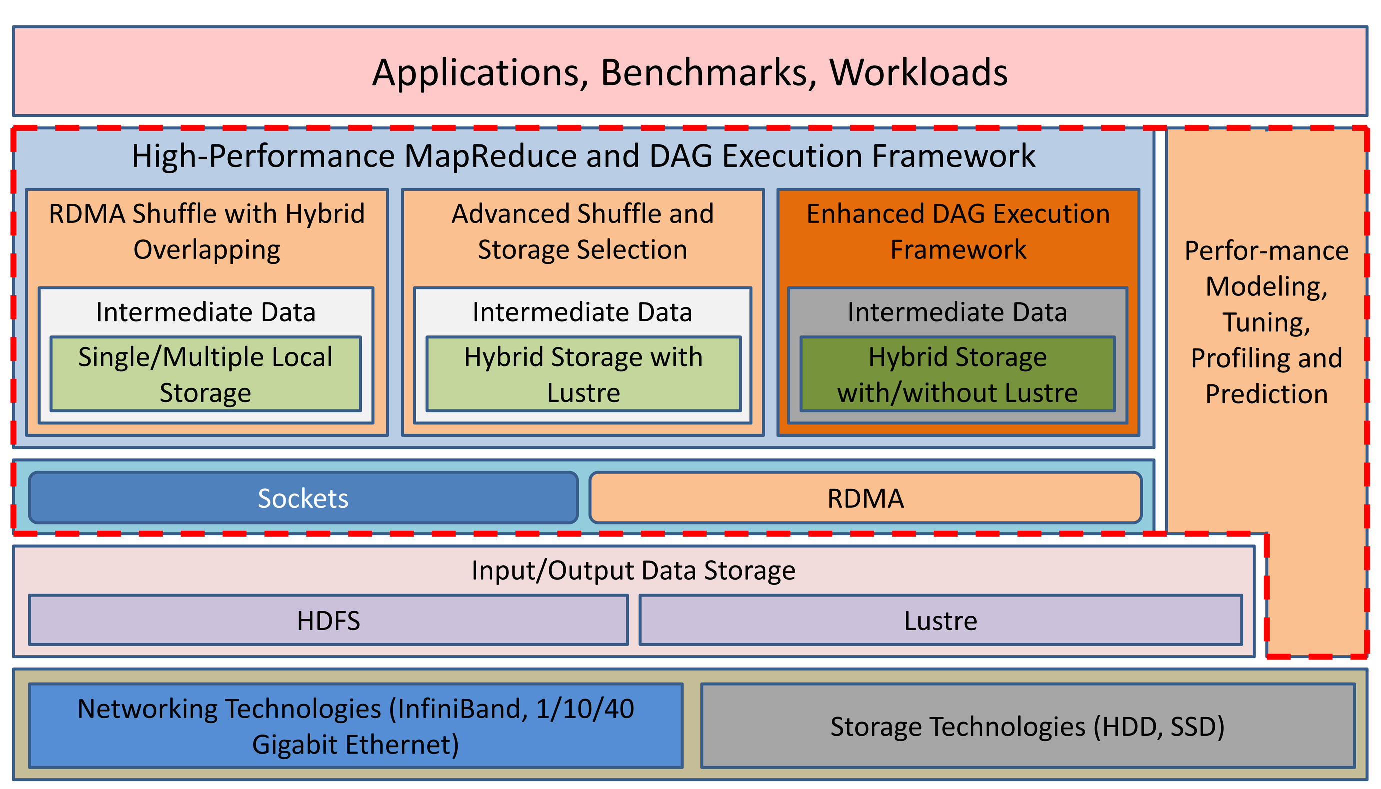
Introduction

- ❖ MapReduce is the de-facto parallel programming model for big data processing
- ❖ Open-source implementations from Apache (Hadoop, Spark, Tez) are the most popular frameworks because of proven scalability and fault-tolerance
- ❖ Java sockets based communication model for bulk data transfer in shuffle
- ❖ Costly frequent disk operations in the job execution workflow
- ❖ Cannot take advantage of global file systems because of shared-nothing based architecture

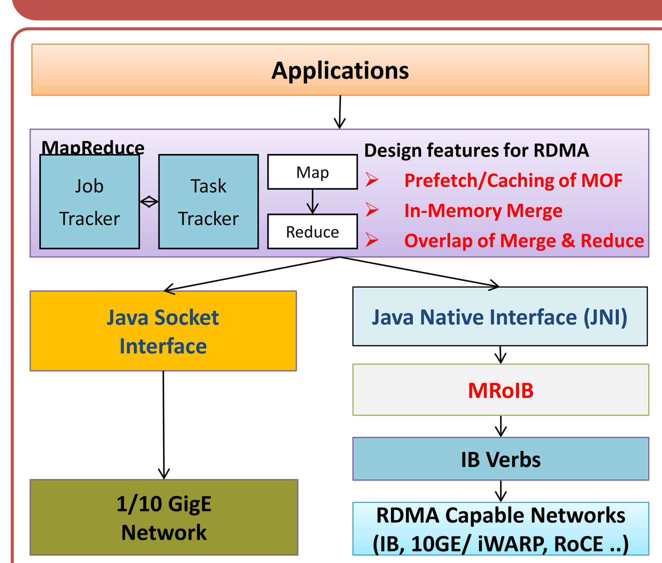


map (k_1, v_1) -> list (k_2, v_2)   reduce (k_2, list(v_2)) -> list (k_3, v_3)

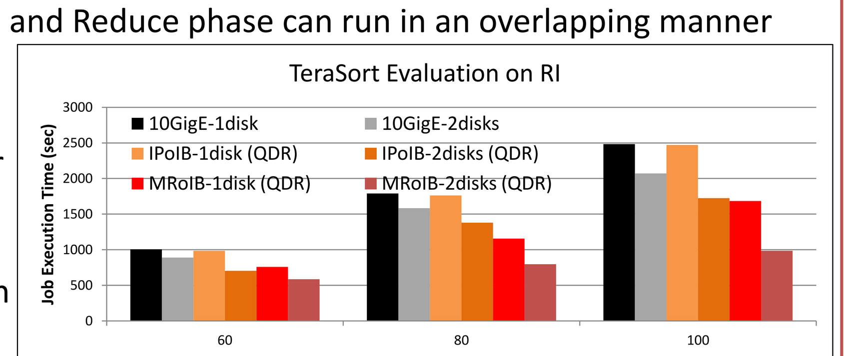
Research Framework



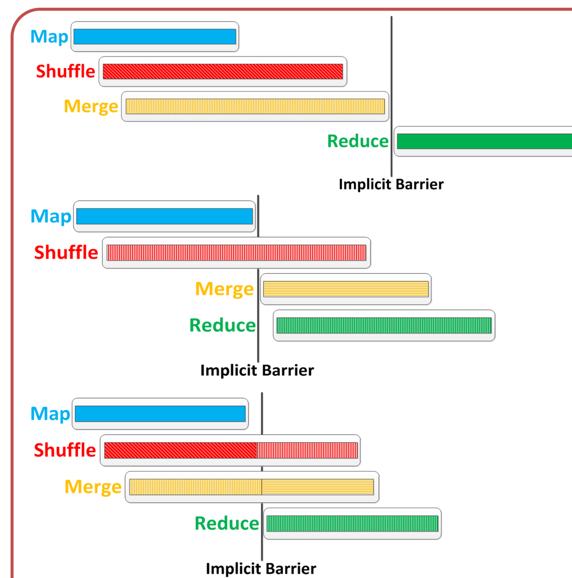
RDMA-based MapReduce



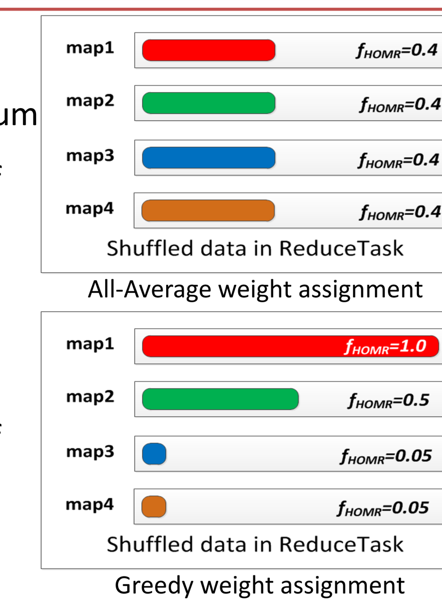
- ❖ MRoIB [1] introduces RDMA-based shuffle, replacing the slower HTTP-based request response messages
- ❖ MOFs are divided into small packets and are shuffled instead of shuffling the entire data at once as in default framework
- ❖ No on-disk merge. Initially, small packets of data are required to create the Priority Queue (PQ); subsequent packets are inserted in this PQ for sorting operation
- ❖ Merge and Reduce phase can run in an overlapping manner
- ❖ Pre-fetching and caching of Map Output Files are introduced to accelerate the response from TaskTracker for each request of ReduceTasks
- ❖ Performance evaluation shows 39% (31%) reduction in time with 2 HDD/node (1 HDD/node) for HDFS



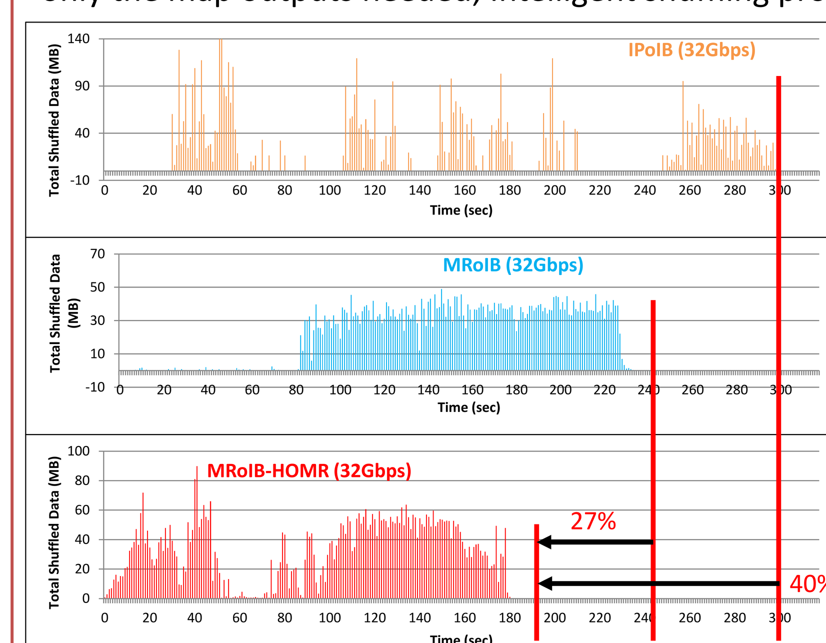
Hybrid Overlapping in MapReduce

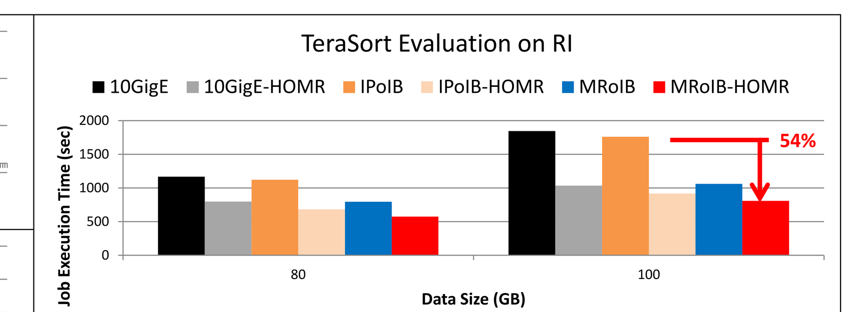


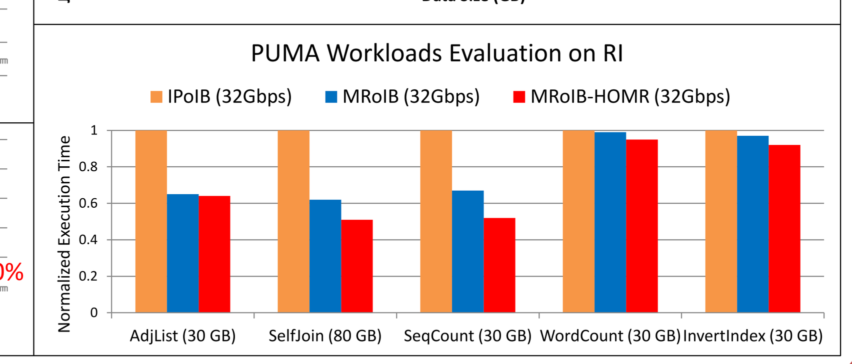
- ❖ HOMR [3] (Hybrid Overlapping in MapReduce) is designed to have maximum possible overlapping across all phases of MapReduce
- ❖ HOMR also ensures faster job execution over other high performance interconnects (10GigE, IPoIB) because of its new shuffle algorithms; provides the fastest execution over RDMA



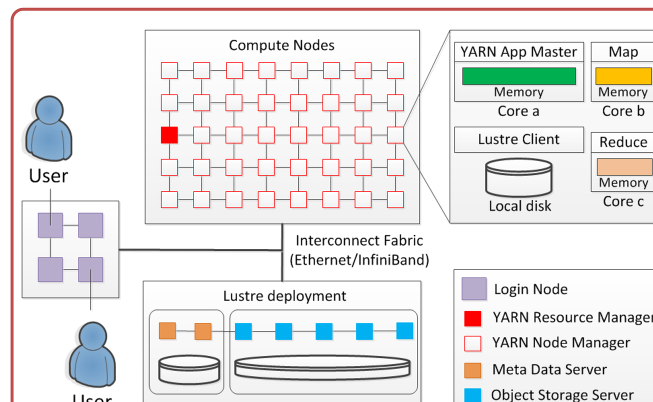
- ❖ HOMR assigns weights to different maps to signify how much data to shuffle on each request; this assignment can be greedy / all-average
- ❖ Initial static weight assignment is updated by on-demand adjustment which makes each shuffle to bring only the map outputs needed; Intelligent shuffling provides faster job execution pipeline



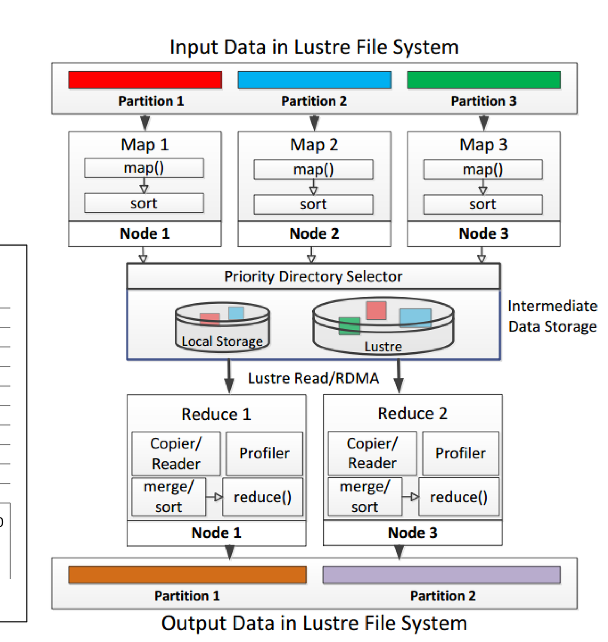


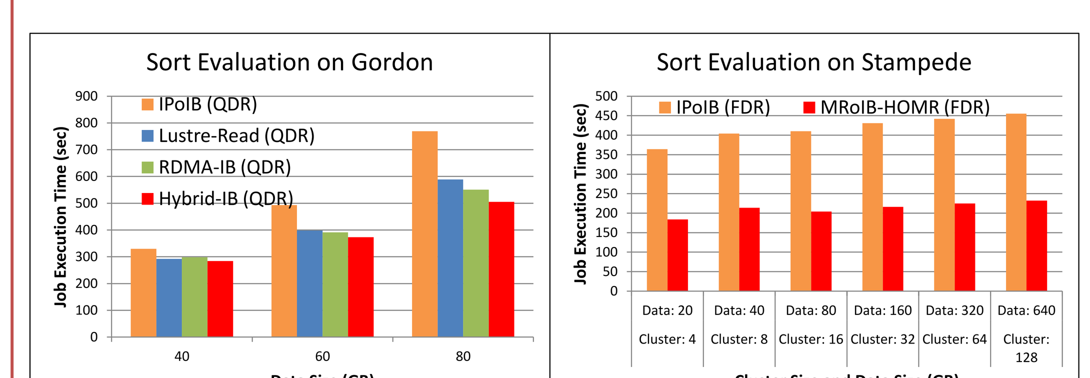


MapReduce over Lustre



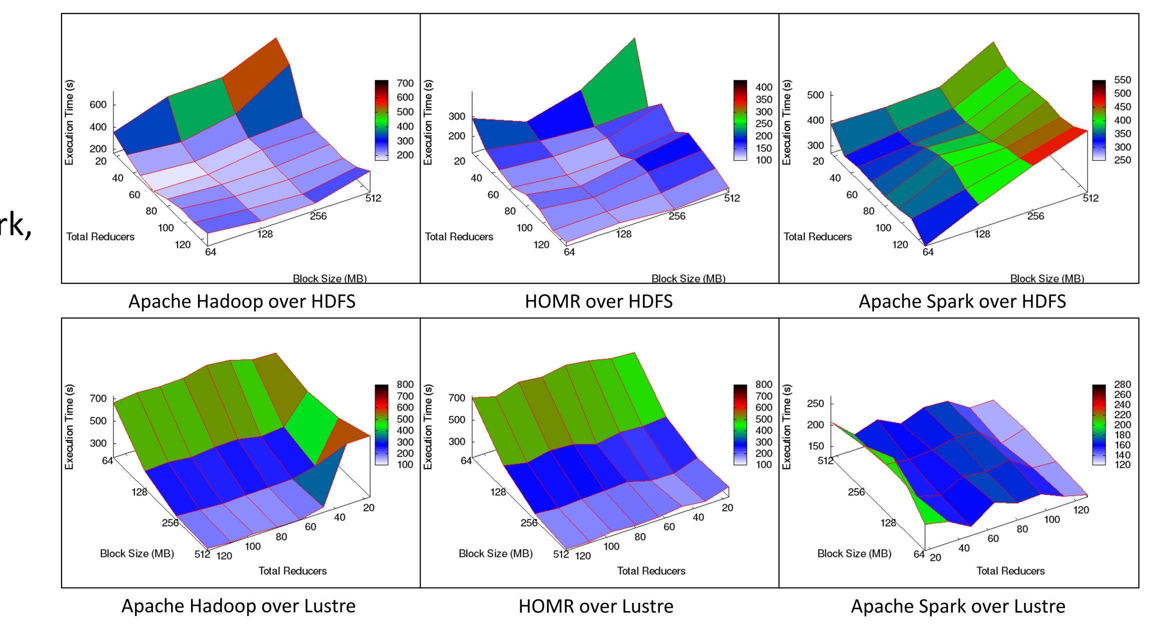
- ❖ Default MapReduce cannot take advantage of the underlying global file system in HPC clusters, such as Lustre
- ❖ We propose an advanced design of HOMR, that can utilize Lustre and extract further benefits
- ❖ The intermediate data directory can be configured to the local disks [4] or Lustre [6] or a combination of both [7]
- ❖ Advanced design of HOMR can dynamically detect which shuffle policy is efficient for a particular job execution and switch from one to another during runtime



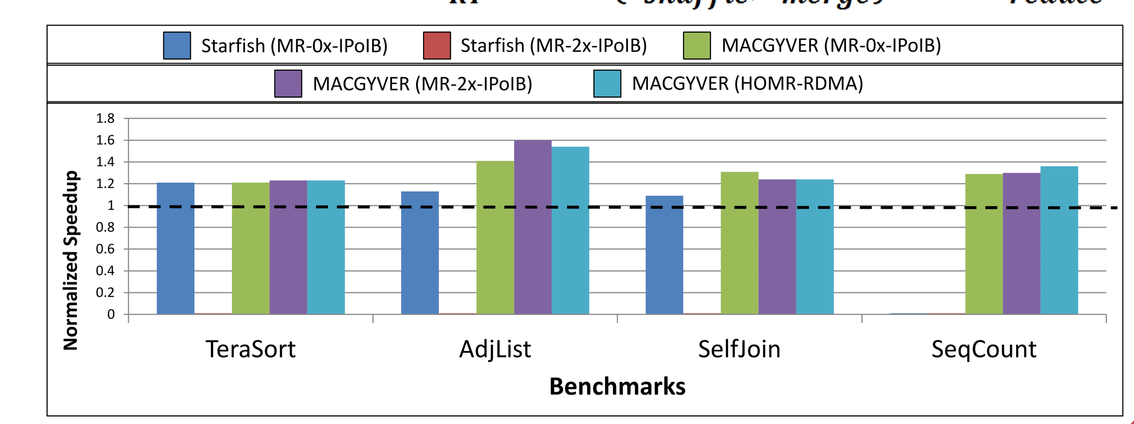


Tuning, Profiling, and Prediction

- ❖ We design a generalized parameter tuning and prediction framework (MACGYVER) for any MapReduce implementation [8]
- ❖ Automatic tuning, profiling is performed for MapReduce implementations in Hadoop, Spark, and HOMR with file systems – HDFS, Lustre, and Tachyon
- ❖ Generalized configuration parameter space is devised to facilitate different MapReduce implementations



- ❖ MACGYVER can also perform profiling and performance prediction using performance analytical models
- ❖ Performance of map and reduce tasks are modeled from execution times of each phase in these tasks. For example, execution time for a single Reduce task can be modeled as $t_{RT} = t_{shuffle} + t_{merge} + t_{reduce}$
- ❖ For RDMA-based MR, execution time can be re-modeled [2] $t_{RT} = \max\{t_{shuffle}, t_{merge}\} + \alpha * t_{reduce}$
- ❖ Simplified prediction model [5] is empirically derived from the detailed performance model
- ❖ Compared to Starfish, MACGYVER can achieve better speedup for different applications



Conclusion and Future Work

- ❖ For large scale data processing, HOMR achieves significant performance benefits compared to default Hadoop MapReduce; leverages benefit from modern HPC resources (RDMA and Lustre)
- ❖ Future plan is to design advanced DAG execution framework (e.g. Tez) with modern HPC resources

Software Distribution

- ❖ HOMR is publicly available in "RDMA for Apache Hadoop" public release (http://hibd.cse.ohio-state.edu)
- ❖ As of Sep '16, more than 17,850 downloads (190 different organizations) have taken place from this site

References

[1] **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*, Int'l Workshop on High Performance Data Intensive Computing (HPDIC), held in conjunction with Int'l Parallel and Distributed Processing Symposium (IPDPS), May 2013

[2] **M. W. Rahman**, X. Lu, N. S. Islam, and D. K. Panda, *Does RDMA-based Enhanced Hadoop MapReduce Need a New Performance Model?,* ACM Symposium on Cloud Computing (SoCC), October 2013 (Poster Paper)

[3] **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*, International Conference on Supercomputing (ICS), June 2014

[4] **M. W. Rahman**, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, *MapReduce over Lustre: Can RDMA-based Approach Benefit?,* 20th International European Conference on Parallel Processing (Euro-Par), August, 2014

[5] **M. W. Rahman**, X. Lu, N. S. Islam, and D. K. Panda, *Performance Modeling for RDMA-Enhanced Hadoop MapReduce*, 43rd International Conference on Parallel Processing (ICPP), September 2014

[6] **M. W. Rahman**, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, *High-Performance Design of YARN MapReduce on Modern HPC Clusters with Lustre and RDMA*, 29th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 2015

[7] **M. W. Rahman**, N. S. Islam, X. Lu, and D. K. Panda, *A Comprehensive Study of MapReduce over Lustre for Intermediate Data Placement and Shuffle Strategies on HPC Clusters,* Under Review

[8] **M. W. Rahman**, N. S. Islam, X. Lu, D. Shankar, and D. K. Panda, *MACGYVER: A MapReduce-centric Gray-Box Versatile Tuning and Prediction Framework for Hadoop and Spark*, Under Review

Acknowledgements

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

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

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