Does RDMA-based Enhanced Hadoop MapReduce Need a New Performance Model?

Md. Wasi-ur-Rahman, Xiaoyi Lu, Nusrat S. Islam, and Dhabaleswar K. (DK) Panda
The Ohio State University, {rahmanmd,luxi,islamn,panda}@cse.ohio-state.edu

Abstract

Recent studies [17, 12] show that leveraging benefits of high performance interconnects like InfiniBand, MapReduce performance in terms of job execution time can be greatly enhanced by using additional features like in-memory merge, pipelined merge and reduce, and prefetching and caching of map outputs. In this paper, we validate that it is time to have a new performance model for the RDMA-based design of MapReduce over high performance interconnects. Our initial results derived from the proposed analytical model matches the experimental results within a 3-5% range.

1 Motivation

Authors in [17, 12] present enhanced designs and algorithms for the RDMA-based MapReduce framework. With these design changes, MapReduce job execution can be greatly accelerated by leveraging the benefits of high-performance interconnects. The high performance design of Hadoop (Hadoop-RDMA) [3] also shows significant performance benefits achievable through RDMA-capable interconnects using enhanced designs of various components (HDFS [6], MapReduce [12], RPC [9]) inside Hadoop. On the other hand, much performance modeling research [4, 8, 2, 1, 13, 5, 7, 10, 11] has been carried out to deeply analyze the default MapReduce framework. But, because of the inherent architectural changes, these models are not appropriate for performance prediction of RDMA-based enhanced MapReduce. For example, Table 1 captures the performance evaluation for the Sort benchmark using default Hadoop [16] and enhanced MapReduce with RDMA [12] and compares these with the performance model in [4]. This clearly illustrates the necessity of a new model for the enhanced design of MapReduce.

SoCC'13, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1.
http://dx.doi.org/10.1145/2523616.2525953

Benchmark	Hadoop	Model [4]	RDMA [12]
Sort (20GB)	707 sec	691.78 sec	324 sec

Table 1: Comparison using Sort

2 Our Approach

For the RDMA-based enhanced design of MapReduce, all of the new features are added inside the ReduceTask. Thus, to predict the performance correctly for this design, we approach to model the performance of the ReduceTask from scratch. In the default MapReduce framework, execution time for a single ReduceTask, t_{RT} is calculated from the execution times of different phases in the ReduceTask.

$$t_{RT} = t_{shuffle} + t_{merge} + t_{reduce} \quad (1)$$

For the RDMA-based design, on the other hand, t_{RT}, will not be as simple as the default one. Because of the fully overlapping feature among these three phases, t_{RT} can be rewritten as:

$$t_{RT} = \max(t_{shuffle}, t_{merge}) + \alpha * t_{reduce} \quad (2)$$

α represents the fraction of the total data that resides in memory yet to be reduced, while both shuffle and merge phases have completed their execution. Also, because of the architectural changes in the enhanced design, all of the parameters $t_{shuffle}$, t_{merge}, and t_{reduce} need to be re-modeled to incorporate all of the new design enhancements.

3 Contribution

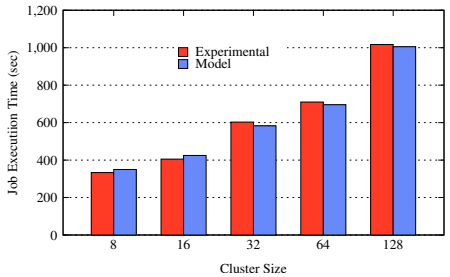

Figure 1: Model validation in Stampede Cluster

We validate our model for enhanced MapReduce using terasort [15] on Stampede [14]. We vary the cluster size from 8 to 128, while increasing the data size exponentially from 40 GB to 640 GB. As shown in Figure 1, we observe that the model successfully validates the experimental results with a difference of 3-5% range.

References

[1] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. MapReduce Online. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.

[2] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[3] Hadoop-RDMA: High-Performance Design of Hadoop over RDMA-enabled Interconnects. http://hadoop-rdma.cse.ohio-state.edu/.

[4] H. Herodotou. Hadoop Performance Models. Technical Report CS-2011-05, Computer Science Department, Duke University.

[5] Y. Huai, R. Lee, S. Zhang, C. H. Xia, and X. Zhang. DOT: A Matrix Model for Analyzing, Optimizing and Deploying Software for Big Data Analytics in Distributed Systems. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, SOCC '11, pages 4:1–4:14, New York, NY, USA, 2011. ACM.

[6] 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 *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, November 2012.

[7] H. Karloff, S. Suri, and S. Vassilvitskii. A Model of Computation for MapReduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 938–948, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[8] X. Lin, Z. Meng, C. Xu, and M. Wang. A Practical Performance Model for Hadoop MapReduce. In *Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on*, pages 231–239, 2012.

[9] X. Lu, N. S. Islam, M. W. Rahman, J. Jose, H. Subramoni, H. Wang, and D. K. Panda. High-Performance Design of Hadoop RPC with RDMA over InfiniBand. In *IEEE 42nd International Conference on Parallel Processing (ICPP)*, 2013.

[10] K. Morton, M. Balazinska, and D. Grossman. ParaTimer: A Progress Indicator for MapReduce DAGs. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 507–518, New York, NY, USA, 2010. ACM.

[11] Mumak: Map-Reduce Simulator. https://issues.apache.org/jira/browse/MAPREDUCE-728.

[12] 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. In *International Workshop on High Performance Data Intensive Computing(HPDIC), in conjunction with IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2013.

[13] N. Rapolu, K. Kambatla, S. Jagannathan, and A. Grama. TransMR: Data-centric Programming beyond Data Parallelism. In *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, HotCloud'11, pages 19–19, Berkeley, CA, USA, 2011. USENIX Association.

[14] Stampede at Texas Advanced Computing Center. http://www.tacc.utexas.edu/resources/hpc/stampede.

[15] TeraSort. http://hadoop.apache.org/docs/r0.20.0/api/org/apache/hadoop/examples/terasort/TeraSort.html.

[16] The Apache Software Foundation. The Apache Hadoop Project. http://hadoop.apache.org/.

[17] Y. Wang, X. Que, W. Yu, D. Goldenberg, and D. Sehgal. Hadoop Acceleration through Network Levitated Merge. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, 2011.