CS 5204 Project Proposal
MPI on HDFS

Luna Xu (xuluna@cs.vt.edu) Adam Binford (adamq@vt.edu)

September 28, 2014

1 Team member

Luna Xu (xuluna)
Adam Binford (adamq)

2 Project Background and Objective

MPI [3] is the de facto standard for scientific computing. Additionally, Hadoop has become the dominant framework for the distributed processing of large data sets across clusters due to its high reliability and scalability. A typical rich data analytic workflow usually involves multiple jobs, some of which are computation and communication intensive. With the YARN [6] framework now included in Hadoop, non-MapReduce jobs can be implemented on top of Hadoops distributed file system, HDFS [4]. Some of these jobs might not be able to exploit the locality used by MapReduce, such as matrix operations, including matrix multiplication and matrix inversion, widely used in data mining algorithms. Using MapReduce to run all the jobs may not be an efficient solution, especially when different tasks involve complex communication patterns and are computation intensive. Enabling workflows to embrace various frameworks would greatly improve performance [5].

However, it is not straightforward to enable these workflow interactions. One of the biggest challenges is the data transfer. In a typical HPC cluster setup, a shared file system is, such as Hadoop which uses HDFS which has a unique architecture for typical MapReduce jobs. Especially for large data IO, MPI-IO relies on a parallel file system support. For jobs that are unable to exploit locality, an efficient means of transferring data is needed. This is where MPI can come in handy. With the addition of YARN in Hadoop now, MPI applications can be run on top of HDFS more easily. Traditionally, however, data files need to be converted and transferred between HDFS and the shared file system that MPI is using. We want to develop a more seamless approach that better integrates MPI applications with HDFS.

3 Project Plan

1. Investigate all the ways we could enable MPI applications to run on HDFS. These includes options such as using Filesystem in Userspace (FUSE) [2], HDFS-NFS [1], which exports HDFS as a NFS without the use of FUSE, and libhdfs, to manipulate HDFS using C libraries and the Java Native Interface;

2. Evaluate each possible method and see which provides the best way to implement MPI application;

3. Develop an interface for running MPI applications using YARN;

4. Evaluate our implementation against current methods and hope to achieve better performance with our framework

4 Schedule

Sep. 29 - Oct. 6
Work with instructor to revise and get approval for topic.

Oct. 6 - Oct. 10
Evaluate current methods and write progress report 1.

Oct. 13 - Oct. 17
Framework design.

Oct. 20 - Oct. 24
Framework implementation.

Oct. 27 - Oct. 31
Meet formally with instructor to discuss progress and prepare for Midterm presentations.

Nov. 3 - Nov. 7
Refine design and implementation.

Nov. 10 - Nov. 14
Test our implementation and write progress report 2.

Nov. 17 - Nov. 21
Evaluate our framework and discuss with instructor.

Nov. 24 - Nov. 28
Write final project report and prepare for presentation.

References

[1] Hdfs nfs gateway. `http://hadoop.apache.org/docs/r2.3.0/hadoop-project-dist/hadoop-hdfs/HdfsNfsGateway.html`. Accessed: 2014-09-24.

[2] Mountablehdfs. `https://wiki.apache.org/hadoop/MountableHDFS`. Accessed: 2014-09-24.

[3] Mpich. `www.mpich.org`. Accessed: 2014-08-11.

[4] Hadoop Distributed File System (HDFS). `http://hortonworks.com/hadoop/hdfs/`, 2014. [Online; accessed 13-Feburuary-2014].

[5] Jha, S., Qiu, J., Luckow, A., Mantha, P. K., and Fox, G. C. A tale of two data-intensive paradigms: Applications, abstractions, and architectures. *CoRR* (2014), –1–1.

[6] Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S., et al. Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing* (2013), ACM, p. 5.