Generalizing MapReduce as a Unified Cloud and HPC Runtime

Judy Qiu
School of Informatics and Computing
Indiana University
150 S. Woodlawn Ave., Bloomington IN 47405, USA
xqiu@cs.indiana.edu

ABSTRACT

Computational simulation and analysis were one of the keys to the future in data-intensive science as a “fourth paradigm” of scientific discovery but facing a major challenge as handling the incredible increases in dataset sizes. This requires attractive powerful programming models that address issues of portability with scaling performance and fault tolerance. Further, one must meet these challenges for both computation and storage. We build on the success of our research on Iterative MapReduce with successful prototypes Twister (on HPC) and Twister4Azure (on clouds). We have designed a novel Map Collective runtime which generalizes previous work in both HPC and MapReduce communities, which we hypothesize can be used as the runtime for data analysis (mining) interoperably between clouds and clusters.

Categories and Subject Descriptors

D.1.3 [Software Programming Techniques]: Concurrent Programming

General Terms

Design, Experimentation, Performance

Keywords

Iterative Mapreduce, Map Collective, data analysis, HPC, Cloud, scaling performance, fault tolerance, interoperability

1. INTRODUCTION

It is well understood that data is increasingly important in scientific, industrial and societal domains with dataset sizes often expanding faster than predicted by Moore’s law and demanding cost-effective parallel-data analysis techniques. We are soon dealing with Petabytes if not Exabytes of data. Our research is based on the hypothesis that clouds and cloud-like environments including MapReduce will prove very important for science – data analysis in particular – by providing both cost effectiveness (from megascale data centers) and powerful new programming paradigms. Further HPC clusters spanning up to exascale capability will continue to be critical and so a key challenge is portability not just as systems scale up in size but also between HPC and Cloud systems. HPC systems are important not just as hosts for observational data analysis but because of the growing challenge of analyzing results of supercomputer simulations. Another synergy between HPC and clouds is common challenges in fault tolerance and storage.

This motived us to develop a unified framework that is interoperable between HPC and clouds. The proposed work will extend a prototype environment developed by my research team (called “Twister”) on HPC clusters, Amazon and Azure for several bioinformatics data analysis applications. This work also contributed to Microsoft’s Iterative MapReduce runtime for their recently announced Daytona project.

2. PROBLEM OUTLINE

Traditional HPC architecture in Grid separates computation nodes and storage nodes. Meanwhile MPI, the widely used parallel programming model in HPC, does not support fault tolerance natively and has a steep learning curve. New parallel programming models such as MapReduce and Hadoop were successful in processing massive data sets in a distributed environment. Data locality can be implicitly integrated into the runtime platform as the same set of nodes are used for both computation and storage, which instantiates the paradigm of “moving the computation to data”. MapReduce works well for pleasingly parallel applications. However, MapReduce framework has Map only or Map and Reduce phases with disk access which is not suitable for supporting more complicated data analysis or data mining that typically has many iterations.

Intel’s RMS (recognition, mining and synthesis) taxonomy [1] offers a way to describe a class of emerging applications. The technology underlying these applications is likely to have broad applicability from computer vision, rendering, physical simulation, (financial) analysis and data mining. There are common computing kernels at the core of these applications, which require iterative solvers and basic matrix primitives.

These observations suggest that iterative MapReduce will be a runtime important to a spectrum of eScience or eResearch applications in biology, chemistry, physics, social science, and the humanities as the kernel framework for large scale data processing.

3. ARCHITECTURE OVERVIEW

Open source Java Twister [2-3] and Twister4Azure [4-5] have been released as Iterative MapReduce framework based on our initial research on data-intensive programming models and their runtime. Twister interpolates between MPI and MapReduce and, suitably configured, can mimic their characteristics, and, more interestingly, can be positioned as a programming model that has the performance of MPI and the fault tolerance and dynamic flexibility of the original MapReduce. Figure 2 compares Hadoop, DryadLINQ, Twister and MPI showing a simple K-means clustering with MPI giving best performance while Hadoop and Dryad are at least two orders of magnitude slower. The Twister system illustrated in Figure 1 is much faster than conventional

PDAC’11, November 14, 2011, Seattle, Washington, USA.
ACM 978-1-4503-1130-4/11/11.

MapReduce on this problem and is competitive with MPI at large problem sizes.

In a broader context, our research is a new programming and runtime environment that greatly extends the scope of MapReduce and it will broadly support data analysis/mining/analytics, which is an area whose importance is growing as the data deluge is seen in so many areas. Further it will allow clouds to achieve high performance on linear algebra and related parallel algorithms with a framework interoperable with clusters. This is one of challenges driving the innovative research thrusts for run-time environment, fault tolerance, data positioning and high-level languages. We have started investigation in these areas.

The **run-time** research is built on early successes with iterative MapReduce and proposes an innovative new Map-Collective model where the familiar reduction operations are generalized and link earlier research on MPI and MapReduce. Figure 1 shows that each collective (such as reduce, gather, and multicast) offers a universal programmatic interface but has different polymorphic implementations on each supported platform. This is illustrated by early research on Twister with Twister4Azure implementing collectives with Azure tables and queues while the Java HPC cluster version uses publish subscribe technology.

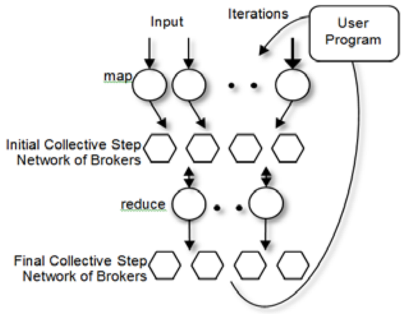

Figure 1: Twister architecture

The **fault tolerance** research exploits the iterative nature of the problem class and the collective abstraction to allow backup and recovery supported by the collective run time. The collective controller allows either an automatic or user specified application dependent time interval between backups. This time trades off overhead in performing the back up against recovery time taken to return to the earlier iteration where backup taken. This idea includes concepts now in MPI and MapReduce and takes advantage of the clean innovative collective architecture.

The **storage** research is motivated by both analysis of observational data and that produced by simulations where one must support the well-known principle of bringing the computing to the data, which is natural but has several challenges. These include difficult data-computing co-scheduling especially in a multi-user environment with data parallel file systems like HDFS. Further in many multi-disciplinary applications the associated data is naturally stored in a distributed fashion. This requires innovation in both integration of wide area and data parallel file systems/object stores and in the automatic transfer and caching of data after computing location chosen. We will explore Twister for in situ analysis and post processing for knowledge discovery. In particular integrating Twister’s streaming model within the ADIOS framework [6] allows our programming model to be applied to analysis of data produced by large scale simulations.

Data parallel languages for science built on an iterative Map Collective runtime and aiming at applications suitable for this runtime is an innovative important concept leveraging the success of languages such as Apache Pig but greatly enhancing their applicability.

4. CONCLUSIONS

In this paper, we provide a glimpse of the issues on interoperability between clouds and HPC clusters and conclude that the iterative Map Collective runtime is important for science as we can expect HPC clusters (becoming exascale) to continue to be important as they seem essential for applications such as large-scale particle dynamics and partial differential equation based simulations. The goal of our research is to clarify which applications are best suited for clouds; which require HPC and which can use both effectively. This will enable thoughtful planning of national Cyberinfrastructure. Particular ideas such as fault tolerance building on the collective abstraction have potential for broad impact on cloud and exascale systems.

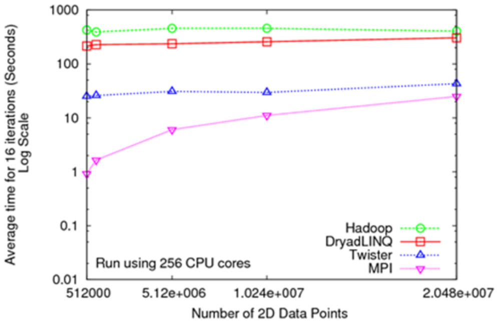

Figure 2: Comparison of Hadoop, DryadLINQ, Twister and MPI on Kmeans Clustering as a function of problem size

5. ACKNOWLEDGMENTS

Our work is partially supported by Microsoft, Indiana University's Faculty Research Support Program, and NSF OCI-1032677 Grant.

6. REFERENCES

[1] Dubey, Pradeep. A Platform 2015 Model: Recognition, Mining and Synthesis Moves Computers to the Era of Tera. Compute-Intensive, Highly Parallel Applications and Uses. Volume 09 Issue 02. ISSN 1535-864X. February 2005.

[2] SALSA Group. Iterative MapReduce. Twister Home Page. Available from: http://www.iterativemapreduce.org/

[3] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox Twister: A Runtime for Iterative MapReduce, in *Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference*, Chicago, Illinois, June 20-25, 2010

[4] SALSA Group. Twister4Azure. Home Page. Available from: http://salsahpc.indiana.edu/twister4azure/

[5] Thilina Gunarathne, Bingjing Zhang, Tak-Lon Wu, Judy Qiu, Portable Parallel Programming on Cloud and HPC: Scientific Applications of Twister4Azure, in *Proceedings of Fourth IEEE International Conference on Utility and Cloud Computing* (UCC 2011), Melbourne, Australia, December 5-8, 2011

[6] Jay Lofstead, Fang Zheng, Scott Klasky, and Karsten Schwan, Adaptable, metadata rich IO methods for portable high performance IO, in *Proceedings of the 2009 IEEE International Symposium on Parallel and Distributed Processing*. 2009, IEEE Computer Society.