Architecture and Performance of Runtime Environments for Data Intensive Scalable Computing

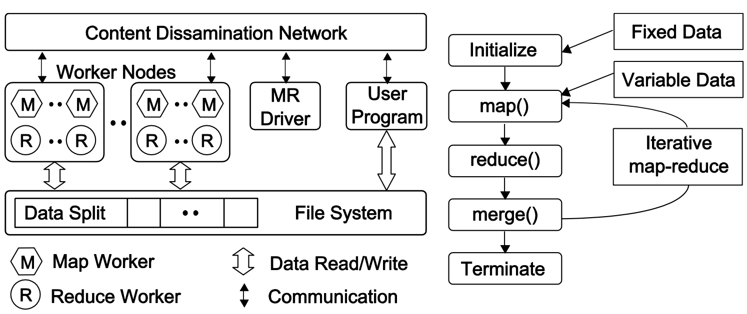
Student: Jaliya Ekanayake {jekanaya@cs.indiana.edu}
Advisor: Prof. Geoffrey Fox {gcf@indiana.edu}
School of Informatics and Computing
Indiana University 919 E. 10th Street Bloomington, IN 47404 USA

Large scale data/compute intensive applications found in many domains such as particle physics, biology, chemistry, and information retrieval are composed from multiple “filters”. The deluge of data and the highly compute intensive nature of these filters mandate the use of large computing infrastructures and parallel processing to achieve considerable performance gains. Cloud technologies such as Google MapReduce, Hadoop, and Dryad can be used to parallelize filters that perform “pleasingly parallel” operations such as calculating Smith Waterman or converting a collection of documents to a different format. Some filters, such as those performing principal component analyses require specialized frameworks such as R, while filters that perform operations such as clustering or multi dimensional scaling require MPI style functionalities to parallelize.

Cloud technologies have added new dimensions to parallel computing. Their support for handling large data sets, the concept of moving computation to data, and the better quality of services provided, simplify the implementation of some problems over the traditional systems. However, applicability of these technologies to the diverse field of parallel computing was not studied well. What filters are best handled by cloud technologies? How can we extend their explicabilities to more classes of filters? How would they operate on Cloud? These are some of the questions that lead to the formulation of the following problem statement for my Ph.D.

“Try to identify MapReduce enhancements that can be applied to large classes of filter pipelines in data/compute intensive applications, and derive a performance model for the enhanced programming model by comparing it with other cloud technologies on both virtualized and non virtualized hardware platforms.”

In pursuing the above goal, I have designed and developed CGL-MapReduce - a light-weight MapReduce runtime that incorporates several improvements to the MapReduce programming model wiz; (i) faster intermediate data transfer via pub/sub broker network, (ii) support for long running map/reduce tasks, (iii) efficient support for iterative MapReduce computations. Figure 1 shows the architecture and the programming flow of CGL-MapReduce, and a comparison with other runtimes. To derive a performance model, I have selected a series of data analysis applications with various filters and identified their mapping to parallel architectures of different types and looked at the performances. During the course of the research, I have implemented these applications using Hadoop, Dryad, and CGL-MapReduce and compared their performances.



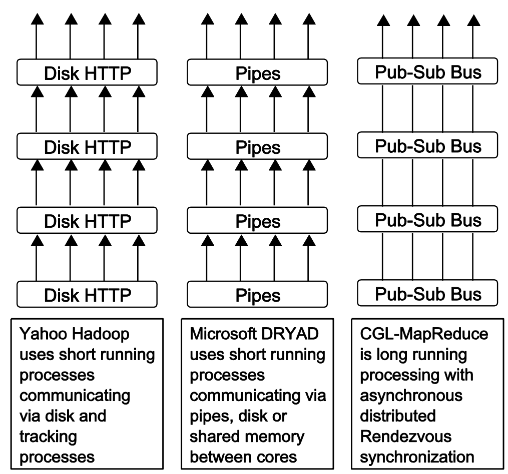


Figure 1.(left) Architecture of CGL-MapReduce, (middle) Programming flow and its support for iterative computations, (right) Long running map/reduce tasks in CGL-MapReduce compared to short running tasks in Hadoop and Dryad.

The results revealed that the enhanced MapReduce runtime is applicable to more classes of filters including the iterative MapReduce computations, and how it outperforms other runtimes for most of the applications. My current research involves applying cloud technologies to different data/compute intensive applications, evaluating them on cloud infrastructures, and using the findings to understand how best we can integrate these technologies.

More information: http://www.cs.indiana.edu/~jekanaya/

Thesis Related Publications

[1]. **Jaliya Ekanayake**, Geoffrey Fox, High Performance Parallel Computing with Clouds and Cloud Technologies, Technical Report June 12, 2009. (Submitted for CloudComp 2009)

[2]. Geoffrey Fox, Seung-Hee Bae, **Jaliya Ekanayake**, Xiaohong Qiu, and Huapeng Yuan, Parallel Data Mining from Multicore to Cloudy Grids, High Performance Computing and Grids workshop, 2008.

[3]. **Jaliya Ekanayake**, Shrideep Pallickara, and Geoffrey Fox, MapReduce for Data Intensive Scientific Analyses, Fourth IEEE International Conference on eScience, 2008, pp.277-284.

[4]. Shrideep Pallickara, **Jaliya Ekanayake**, Geoffrey Fox, An Overview of the Granules Runtime for Cloud Computing, Fourth IEEE International Conference on eScience, 2008, pp.412-413.

[5]. **Jaliya Ekanayake**, Shrideep Pallickara, and Geoffrey Fox, A collaborative framework for scientific data analysis and visualization, Collaborative Technologies and Systems(CTS), 2008,pp. 339-346.

[6]. Shrideep Pallickara, **Jaliya Ekanayake** and Geoffrey Fox, A Scalable Approach for the Secure and Authorized Tracking of the Availability of Entities in Distributed Systems , IEEE International Parallel & Distributed Processing Symposium (IPDPS 2007).

Other publications

[1]. Eran Chinthaka, **Jaliya Ekanayake**, David Leake, CBR Based Workflow Composition Assistant, Accepted for publication, IEEE 2009 Third International Workshop on Scientific Workflows (SWF 2009).

[2]. Srinath Perera, Chathura Herath, **Jaliya Ekanayake**, Eran Chinthaka, Ajith Ranabahu, Deepal Jayasinghe, Sanjiva Weerawarana,Glen Daniels Axis, Middleware for Next Generation Web Services on IEEE International Conference on Web Services (ICWS'06)

[3]. Ajay Smitha and **Jaliya Ekanayake**, Analysis of the Usage Statistics of Robots Exclusion Standard. In proceedings of the IADIS WWW/Internet 2006 Murcia, Spain 5-8 October 2006.