Towards Efficient Dataflow Frameworks for Big Data Applications: Integrating Parallel and Distributed Computing Runtimes

Ph.D. Thesis Proposal

Supun Kamburugamuve

Advisor: Prof. Geoffrey Fox

Committee: Prof. Judy Qiu, Prof. Martin Swany, Prof. Minje Kim, Prof. Ryan Newton,

School of Informatics & Computing

Indiana University, Bloomington, IN, USA;

Big data movement has been driven by the ever increasing velocity, volume and veracity of the data generated from various sources ranging from web users to IoT devices to large scientific equipment. Among the three characteristics of big data, velocity and volume are the key factors driving the capabilities of the big data systems designed to process the data. The data has to be processed in real time as it is generated and as a whole after storing to disks. In traditional HPC applications the data is comparatively small and uniformly distributed allowing a carefully designed algorithm to load balance the work to a higher level. On the other hand big data applications deal with large data sets coming from heterogeneous sources with varying velocities. Traditional control flow parallel applications developed using Message Passing Interface (MPI) are not suitable for such data processing due to this load imbalance and velocity of the data. It is widely accepted that dataflow execution frameworks are suitable to build better data pipelines in such environments due to their asynchronous execution and ease of programming. According to dataflow, an application is modeled as a graph with nodes processing data and edges indicating communications between nodes. A well designed dataflow framework hides the low level details such as communications, concurrency and disk IO from users making the task of application developer easier; which is one of the key driving forces behind such frameworks.

There are many aspects to efficient dataflow engines such as, communication frameworks, scheduling of tasks, use of threads and processes, placement of data and fault tolerance mechanisms. It is fair to say that communication is a vital part of any big data application, because typically they are required to transfer large data sets among the parallel workers. Communication operations involving large number of nodes is fundamental to parallel computing and they are termed collective communications. The naive implementations of such communication operations do not work efficiently while handling the asynchronous nature of dataflow. To overcome these limitations, we propose optimized dataflow collective operations to operate in a data driven fashion while handling the load imbalance and velocity of data. The collectives are made efficient by modeling them as an optimization of the dataflow graph. Because the collectives are embedded into the graph it preserves the dataflow model and the advantages associated with it. Further we investigate the semantics of dataflow engines and their applicability to big data applications in comparison to MPI style applications.

Even though there is a debate about whether dataflow is the right choice for batch applications requiring tight synchronized parallel operations, it is generally accepted that dataflow model is ideal for large scale streaming applications. The optimized collectives will be implemented on a streaming data processing framework and their performance and semantics will be evaluated in that context. Big data pipelines and streaming applications share common processing requirements and execution semantics. Streaming applications work on temporal data while data pipelines work on complete data sets executing loosely synchronized transformations on the data. Because of the commonality, the same collectives can be extended to batch applications. Hence we would like to explore how to extend the concepts to big data batch analytic frameworks.

Research Goals

1. Research into the applicability and semantics of various parallel communication patterns involving many tasks in dataflow programs
2. Research into algorithms that can make such communications efficient in a dataflow setting, especially focused on streaming applications
3. Implement and measure the performance characteristics of such algorithms to improve dataflow engine performance

Research Contributions

1. Introduce efficient dataflow collectives as one of the important areas of dataflow computing
2. Identify the implications of control flow and dataflow programming for data intensive computations
3. Integrate the findings to existing big data projects or possibly develop new frameworks based on the findings