Reading Assignment

*Spark: Cluster Computing with Working Sets* by Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Schenker, and Ion Stoica is about a new framework that supports applications that reuse data across multiple parallel operations while maintaining the scalability and fault tolerance of MapReduce. Spark does this by using resilient distributed datasets (RDDs). RDDs are read-only collections of objects partitioned across sets of machines. These datasets can rebuild itself if a partition is lost.

Cluster computing applications like MapReduce have become very popular. These frameworks do data-parallel computations on clusters of machines that come with local scheduling, fault tolerance, and load balancing. However, these applications use an acyclic data flow that is not useful for applications that reuse a working set of data for multiple parallel operations.

Spark is a new cluster computing framework that supports applications that reuse a working set of data for multiple parallel operations while maintaining similar scalability and fault tolerance compared to MapReduce.

The advantage of using Spark is that subsequent iterations require less and less runtime because they reuse cached data. A logistic regression is an iterative classification algorithm that tries to find a hyperplane that best separates two sets of points. This algorithm works by picking a random hyperplane, and moving it to improve the regression. Cached data in memory across iterations greatly benefits the algorithm. Spark can run logistic regressions up to 10x faster than Hadoop. An Alternating Least Squares (ALS) algorithm is used for collaborative filtering problems. Like predicting ratings for a movie a user has not seen based on their movie rating history. Use a partially filled matrix R containing known ratings for some user-movie pairs. ALS models R as a product of matrices M (m x k) and U (k x u) where each movie and user has a k-dimensional feature vector describing characteristics. A user’s rating for a movie is the dot product of its feature vector and the movie’s. Caching R (a user’s rating for a movie) improved performance by 2.8x in an experiment with 5000 movies and 15000 users on a 30-node EC2 cluster. Additionally, Spark was used to interpret almost 40GB of wikipedia in memory across 15 EC2 machines. It then queried it interactively. Comparable to Hadoop, the first query took around 35sec. But additional queries only take .5-1 sec. Overall, Spark works much more efficiently for applications that reuse a working set of data for multiple parallel operations.

Spark provides resilient distributed datasets (RDDs), broadcasts variables, and accumulators. These three data abstractions are limited, but are powerful enough to support iterative and interactive applications that do not work well with existing cluster computing frameworks. Additionally, the core characteristic of RDDs having enough information to reconstruct itself from available data in storage could be useful in developing abstractions for other clusters. Yet, there are still improvements to be made. The properties of RDDs and other abstractions and their suitability for different classes of applications and workloads could be set, RDD abstraction could be enhanced to allow trade between storage and re-construction cost, operations to transform RDDs could be designed to allow implementation of group-bys and joins, and an interface on top of spark (like SQL and R shells) could be provided.