**Apache Spark: A Unified Engine for Big Data Processing**¹

As data volumes grew immensely, single machines were incapable to process. The computations needed to be scaled out to multiple nodes. Moving away from “one size fits all” systems, a system that combines and functions different processing types was found. In 2009, Apache Spark project designed a unified system for distributed data processing. It can be looked at as an enhanced version of MapReduce that is used as a data-sharing structure called “Resilient Distributed Datasets,” or RDD. Previously, separate engines were needed to process a range of distributed workloads. Now, they can be run as libraries with a common engine. Using a unified API for processing different tasks will provide efficiency. It can avoid writing the data to storage and passing it to another engine. A great example is smartphone: it combines the functions of camera, cellphone, and GPS so we can use the one device with 3 functions. The RDDs are created by “transformations”, such as map, filter, and groupBy, to the data and Spark creates an efficient execution plan from the transformations. Another useful trait it had is the fault tolerance, which allows the RDDs to automatically recover from failures by data replication or checkpoints. The dependencies between the RDDs are logged in a graph called “lineage” in Spark. The fault tolerance and the RDD lineage graph are used to reconstruct missing or damaged partitions, hence the name Resilient. That saves both time and storage memory, instead of replicating the data-intensive workloads. The four main libraries included are SQL and DataFrames, Spark Streaming, GraphX, and MLlib. SQL contains the relational queries of analytical databases, and DataFrame contains abstraction for tabular data, both of which use programmatic methods for filtering, manipulating columns, and aggregation. Discretized Stream (DStream) is the fundamental model of Spark Streaming, which is a stream of RDDs of input data being split into small batches. GraphX is an API for graphs and graph-parallel computation, implementing a multigraph with properties attached to vertices and edges. MLlib is Spark’s library to make machine learning more scalable and efficient with more than 50 common algorithms for distributed data. Batch processing is commonly used on large datasets for converting raw data to structured format and also offline ML training. The processing tasks in the libraries as the data abstractions can be combined in application. Because of Spark’s wide range of applications, companies from various different industries use it. Spark is used for interactive queries, real-time stream processing, and scientific applications. It is a unified data-processing engine that is deployed diverse environments. The Apache Spark libraries are open-source at http:// spark.apache.org¹ for more details.

[1] <http://spark.apache.org/>