Zhiao Zhou zz1749

Nowadays more and more data are produced every day. For example, the New York Stock Exchange generates about 4 to 5 terabytes of data per day. To store and process them in a single machine could be mighty time and resource consuming. Consequently, MapReduce was invented based on the concept “parallel computing” which is a programming model and associated implementation for processing and generating large data sets. Not only can it separate a single task into M small pieces, assign them to quite a few different machines and then aggregate the results into R pieces, it can also address fault and failure in the session and manage the required inter-machine communication to make sure that jobs won’t be interrupted halfway. Users should express the computation as two functions written by them: Map which takes an input pair and produces a set of intermediate key/value pairs and Reduce which accepts an intermediate key I and a set of values for that key. The machines used include Master, Mapper Worker and Reducer Worker. Master is responsible of scheduling works, splitting them to mappers and after completion, mappers will hand their results to mappers. In addition, besides those sufficient functions, MapReduce also support a few extensions which can either improve the performance or help users debug etc. Based on its simplicity and strong ability to handle lags and errors, MapReduce is widely used now in the field of big data management.