Map Reduce: Simplified Data Processing on Large Cluster
One Paragraph Summary

Heewon Kim(hk2874)

The paper introduces “easy to use”, “problems easily expressible”, “scalable” MapReduce programming model. It was created to resolve the complexity of parallelizing computation, distributing data, and handling failures, with the support of a powerful interface and its implementation. The map function processes input data and create intermediate key/value pairs. The reduce function, just how the name of the function expresses itself, reduces intermediate data sets by merging intermediate values that have the same intermediate keys, all created by the map function. The paper then explains the implementation process step by step which includes Input data getting split into m sets, how map workers and reduce workers deal passed input data, where do workers toss processed data to, how does master manage the whole process, and when the user code receives the output. It also talks about how the MapReduce tolerate machine failures whether it is due to master task failure, worker failure, or straggler. Even if map reduce operators are deterministic which allows programmers to easily predict program behavior, the paper talks about the semantics in failures. Furthermore, the paper lists refinements to support better usage. It includes the combiner function, function that partition data other than hashing, implementation of a simple reader interface for new input type, the utility of master’s status page, and many others.