Abstract

Currently, data traffic is massive that cause a crucial problem for storing and processing large volume of data. Hadoop framework can tackle those two problems. Hadoop consists of two main components, the HDFS (Hadoop Distributed File System) and the MapReduce. HDFS capable to handle large data even in petabyte size distributedly on a cluster. A cluster consists of a large number of computers. MapReduce capable to process distributed data paralelly.

This final project aims to implement distributed MapReduce program on a Hadoop cluster. This implementation performing text indexing problem in two distributed indexing strategies. The first strategy requires an intense communication between the map and the reduce process, and the second one perform less communication but process bigger data to be transferred between the map and the reduce process.

The experiment is performed by setting variation of corpus size and the number of used reducers. The result shows that the per-term distributed indexing requires less running time due to the minimum number of the intermediate <key, value> pair that is created in the map process. Although the intermediate <key, value> pairs that are produced in the map process of per-token indexing have less size, but the transfer time requires longer time than the processing time of those the intermediate <key, value> pairs. This intense of communication increases the running time.
Keywords: distributed programming, Hadoop, MapReduce, distributed indexing