Abstract

The development of information technology in recent decades so rapidly, it can be seen from the information traffic between nodes through information technology to produce very large data. With so large amount of data, it is not possible to conventional processing. With the algorithm introduced by Google, the algorithm map / reduce expected large data processing problem is resolved. Algorithms map / reduce is built based on the components *map* and *reduce* derived from functional programming languages. Algorithms map / reduce built in this thesis using a pure functional language, Haskell language. Some built-framework system testing stressed to some of the parameters that influence the process of parallelization occurs. From the results of the implementation of a system built MapReduce could be concluded that the system is built into the Haskell environment for handling very large files still have problems with setting the garbage collector, especially a long execution time. However, for a small file that is less than the size of main memory, built-MapReduce system produced a very good performance. Configuration optimization is also important in minimizing the execution time. MapReduce systems that are built have a better performance compared with the imperative program especially execution time is faster.

Keywords: *Large-scale Data, Chunk data, Map/Reduce Algorithm, Functional programming, Parallelization, Garbage Collector, Haskell.*