УДК 681.518
Emmanuel Akokhia, B.B. Mlynko Ph.D., Assoc.Prof.
Ternopil Ivan Pul'uj National Technical University, Ukraine

MAPREDUCE AND ITS APPLICATION IN DATA CLUSTERING USING NETFLIX MOVIE DATA

Еммануел Акокхіа, Б.Б.Млинко канд. техн. наук, доц.

ЗАСТОСУВАННЯ MAPREDUCE ДЛЯ КЛАСТЕРИЗАЦІЇ БАЗИ ДАНИХ ФІЛЬМІВ NETFLIX

Data clustering is the partitioning of object into groups (called clusters) such that the similarity between members of the same group is maximized and similarity between members of different groups is minimized. Often some form of distance measure issued to determine similarity of objects. MAPREDUCE is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

Over the past years, the authors and many others at Google have implemented hundreds of special-purpose computations that process large amounts of raw data, such as crawled documents, web request logs, etc., to compute various kinds of derived data, such as inverted indices, various representations of the graph structure of web documents, summaries of the number of pages crawled per host, the set of most frequent queries in a given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order tarnish in a reasonable amount of time.

The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues. As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. My abstractions are inspired by the map and reduce primitives present in Lisp and many other functional languages. I realized that most of our computations involved applying a map operation to each Logical record in our input in order to compute a set of intermediate key/value pairs, and then applying a reduce operation to all the values that shared the same key, in order to combine the derived data appropriately. My use of a functional model with user–specified map and reduce operations allows us to parallelize large computations easily and to use re-execution as the primary mechanism for fault tolerance. The major contributions of this work are a simple and powerful interface that enables automatic parallelization and distribution of large-scale computations, combined with an implementation of this interface that achieves high performance on large clusters of commodity PCs.

In conclusion using MapReduce which is a feasible solution for processing problems involving large amounts of data. Especially for problems that can easily be partitioned into independent sub tasks that can be solved in parallel. Hadoop is an open source MapReduce implementation featured in this study. Hadoop will be considered only for non-time sensitive tasks that can be batch processed. An example is certain types of data clustering, such as the Netflix data clustering presented in this report.