Chapter 2
Applying the K-Means Algorithm in Big Raw Data Sets with Hadoop and MapReduce

Ilias K. Savvas
TEI of Larissa, Greece

Georgia N. Sofianidou
TEI of Larissa, Greece

M-Tahar Kechadi
University College Dublin, Ireland

ABSTRACT

Big data refers to data sets whose size is beyond the capabilities of most current hardware and software technologies. The Apache Hadoop software library is a framework for distributed processing of large data sets, while HDFS is a distributed file system that provides high-throughput access to data-driven applications, and MapReduce is software framework for distributed computing of large data sets. Huge collections of raw data require fast and accurate mining processes in order to extract useful knowledge. One of the most popular techniques of data mining is the K-means clustering algorithm. In this study, the authors develop a distributed version of the K-means algorithm using the MapReduce framework on the Hadoop Distributed File System. The theoretical and experimental results of the technique prove its efficiency; thus, HDFS and MapReduce can apply to big data with very promising results.

INTRODUCTION

Big data refers to data sets whose size is beyond the capabilities of most current hardware and software technologies in order to be managed and processed within a reasonable response time. In addition, data may have different structures, heterogeneous, or may be completely unstructured (e.g., multimedia and text documents). The management of extremely large and always growing volumes of data has been since many years a challenge for all fields of science. For example, by 2014, the

DOI: 10.4018/978-1-4666-4699-5.ch002

Large Synoptic Survey Telescope (LSST, 2011) will produce 20 terabytes each night while by 2019 it is anticipated that the Square Kilometre Array radio telescope is planned to produce 7 petabytes of raw data per second (SKA, 2011). Facebook uploads six billion photos per month for a total of about 72 petabytes per annum. In addition, the vast amount of digital information that is now available should make easier the investigation of criminal activities, however the enormous size of the available data created a new challenge of how can we extract evidence within a reasonable processing time.

With Data Intensive Computing, organizations can progressively filter, transform and mine massive data volumes into information that can help the users make better decisions quicker. Data Mining is the process for extracting useful information from large datasets. As the datasets are very large, the time complexity of most Data Mining techniques is very high. One common method to overcome the complexity problem is to reduce the initial dataset size by using a representative sample and then use this small sample to extract the knowledge. The challenge here lies in identifying the representative sample, as its choice impacts directly on the final results. Another method is to distribute the dataset among a set of processing nodes and perform the calculation in Single Program Multiple Data (SPMD) paradigm (in parallel). It can be implemented by using threads, MPI or MapReduce. The choice of an appropriate implementation strategy is based on the size of the dataset, the complexity of the computational requirements, synchronisation, and the hardware profile.

The Apache Hadoop software library provides useful and efficient tools for the distributed computing of large datasets. HDFS is a distributed file system that provides high-throughput access to data-driven applications and MapReduce is a programming model for distributed processing of large datasets. Typical MapReduce computations involve many terabytes of data on thousands of machines. MapReduce usually divides the input dataset into disjoint subsets (chunks). The number of subsets depends on the size of the dataset and the number of processing nodes available. The users may specify a mapping function that processes (key, value) pairs to generate a set of intermediate (key, value) pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

The purpose of this chapter is to explore the possibility of using Hadoop's MapReduce framework and Hadoop Distributed File System to implement a popular clustering technique in a distributed fashion. The experimental results obtained are very promising and showed good performance of the proposed technique. In addition, the theoretical analysis of the algorithm's complexity is in line with the experimental results, and the approach scales very well and outperforms the sequential version.

RELATED WORK

There have been extensive studies on various clustering methods; and especially the k-means clustering has been given a great attention. However, there is very little on the application of k-means to the MapReduce. Since its early development, the k-means clustering (Lloyd, 1982) has been identified to have a very high complexity and significant effort has been spent to tune the algorithm and improve its performance. While k-means is very simple and straightforward algorithm, it has two main issues: 1) the choice of the number of clusters and of the initial centroids. 2) the iterative nature of the algorithm which impacts heavily on its scalability as the size of the dataset increases. Many researchers have come up with various algorithms that:

- Improve the accuracy of the final clusters;
- Help in choosing appropriate initial centroids;