Performance Evaluation of Cluster Algorithms for Big Data Analysis on Cloud

Chu-Hsing Lin, Jung-Chun Liu, Tsung-Chi Peng

Department of Computer Science
Tunghai University
Taichung 40704, Taiwan
{chlin, jcliu, g043500011}@thu.edu.tw

Abstract

In this study, based on clustering algorithms, we perform data mining on land price data in Taichung City during past ten years. For big data analysis, we combine Hadoop HDFS and MapReduce with R language and visualize results on Google Maps. We also study performances of K-means and Fuzzy C-means clustering algorithms, executed in the Hadoop cloud and a stand-alone PC. The experimental results show that with a cloud of 9 compute nodes, about 3.5 times of acceleration are attainable; hence Hadoop cloud with R can be applied to solving insufficient memory issues in big data applications.

Key words: cloud computing, R language, K-means, Fuzzy C-means

Introduction

Along with the fast advancement of information technology and networks, Big Data and cloud computing technologies have become increasingly important; how to effectively and efficiently obtain intelligence from the huge amount of data has become the focus of attention [1].

In this study, we perform data mining on open data of the land price in Taichung City, Taiwan, during the past ten years. From analysis of the yearly variations of land prices, useful information can be acquired, such as land price factors, city development trends, and popular land investment locations.

The use of the R statistical programming language in Big Data analysis faces with issues of rapid data storage expansion demands and performance bottlenecks in analyzing data. Hence, based on the Hadoop Distributed File System (HDFS) and MapReduce, we propose a scheme to combine R with the Hadoop distributed computing platform, and implemented with K-means and Fuzzy C-means clustering algorithms. For data visualization, the processed data are plotted on top of Google Maps. Besides, we compare the performance of clustering algorithms with R implemented on the cloud clusters with 1 to 9 compute nodes and a single PC.

Background

A. Apache Hadoop

The Apache Hadoop, a distributed computing framework, mainly consists of Hadoop Distributed File System (HDFS), a distributed and scalable file system, which stores data on commodity machines to offer high collective bandwidth; and MapReduce, a parallel programming framework, which implements the MapReduce programming model for processing large amounts of data [2].

Hadoop organizes computers in master/slave architecture that helps attain great scalability in processing. Every HDFS cluster has two types of nodes: a single master node called NameNode, and a large number of slave worker nodes called DataNodes [3-5].

NameNode manages the overall file system, its namespace, and controls the access to files by clients. The DataNodes store and serve blocks of data over the network using a block protocol, under the direction of the NameNode. A block of data, ranging from 16 to 128 MB in size, is the storage unit in HDFS. Files are divided into blocks, and if possible, each block of data will be saved on a different DataNode.

Any piece of data is stored typically as replications on three nodes. NameNode needs to record every file location. When there is a demand for file access, NameNode coordinates the DataNode responsible for responding; moreover, when a node is damaged, NameNode automatically executes data relocation and replication.

B. MapReduce

MapReduce, a software architecture proposed by Google, can be used for parallel computing for large scale data [6, 7]. As a key technique for cloud computing, it executes a task by a Map and Reduce approach to achieve the effect of distributed computing. Developers can write simple programs to use the massive amount of computing resources and efficiently process Big Data.

A MapReduce operation is divided into two parts: Map and Reduce. In the start of operation, the big data set is converted by the computing system into lists of (key, value) pairs and automatically divided into many parts, and respectively sent to different Mappers to process. After completing processing of the assigned work, Mappers also arrange computation results into lists of (key, value) pairs, and then send them to Reducers to consolidate results from all Mappers and output the final results.

C. R Language

R language is integrated data processing and statistics software; it supports data analytics with powerful array and matrix operations capabilities and graphic tools. As free and open source software, plus numerous packets written by R users can be found in its official website, in recent years R has become very popular and been used by many professionals such as risk analysts, researchers, and statisticians [8]. The rapid expansion of R is due to its object-oriented capabilities, ability to execute user-defined functions and packages, flexibility in syntax, and easy to edit features. However, since R

 ISBN 978-1-5090-4897-7

processes the data loaded in the main memory, R alone cannot easily handle huge data sets (TB or PB). Hence, we combine R with Hadoop for Big Data analysis.

Experimental Designs

A. RHadoop Design

Although R language itself has provided a lot of data analysis and graphic packages, but it lacks distributed processing ability to effectively deal with huge amounts of data, and is weak in handling big data. The introduction of RHadoop, which combines easy to analyze data characteristics of R language, with distributed computing and storage capacity of Hadoop, can be quite effective to solve the problems related to Big Data analysis.

As shown in Fig. 1, we successfully combine R and Hadoop by using RMR2 and RHDFS packages. The RMR2 package enables R to run MapReduce programs; whereas the RHDFS package provides R with HDFS functions. In this study, data are stored in MySQL database, and can be accessed by R via the RMySQL package.

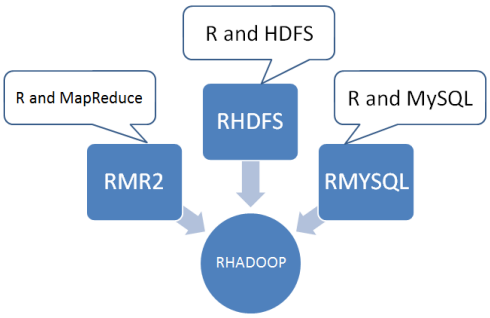

Fig. 1 The RHadoop architecture.

B. Open Data Retrieval

The open land price data is selected as the target dataset due to the following reasons: first, its size is huge and fast expanding; second, variations of it are keys to understand trends of land price; third, the range and changes of data are great enough to see performance differences in clustering algorithms.

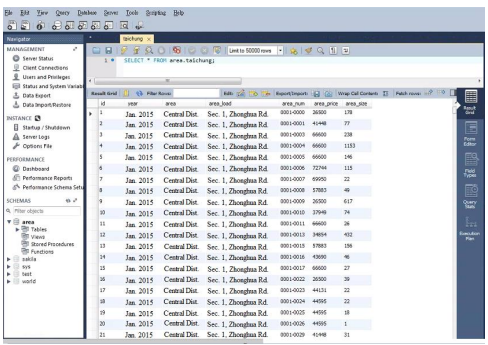

Fig. 2 Data storage format.

We used the Jsoup parser, a Java library for extracting and manipulating HTML data, to retrieve land price data in Taichung city from 2006 to 2015 and stored the retrieved data into MySQL database. As shown in Fig. 2, the data storage format is divided into time (year), area, land price (area_price), and land size (area_size).

C. Clustering Algorithms

The K-Means and Fuzzy C-Means clustering algorithm packets can be used in R [9, 10]. In order to have a clear display of results on Google Maps, we filtered the raw data and selected data with land price greater than New Taiwan dollar (NTD) 100,000 per Taiwanese ping (or 3.306 square meters) for clustering.

For K-means clustering algorithm, the number of clusters, k, is set to be 7, and the maximum number of iterations allowed is set to be 200:

kmeans(data, centers, iter.max = 200 , algorithm = c("Hartigan-Wong", "Lloyd" , "Forgy", "MacQueen "))

Where “centers” is the number of clusters, i.e, k, and the default Hartigan-Wong algorithm is chosen.

For Fuzzy C-Means clustering algorithm, similar parameters setting are used:

cmeans(data, centers, iter.max = 200,verbose = FALSE, method = "cmeans", m = 2)

Where m is the degree of fuzzification.

Experimental Results and Analyses

Specifications of the experiment are listed as follows:

Hardware: three physical machines with Intel Core i7-4790 CPU with 8GB memory are used; inside each physical machine, 3 virtual machines allocated with 1Core and 2GB memory are created to have 9 nodes. The master node consists of a physical machine with G860 CPU and 8G memory.

Software: Hadoop 2.6.1.

Dataset: open data from Taichung City government.

A. Trends of Annual Land Price

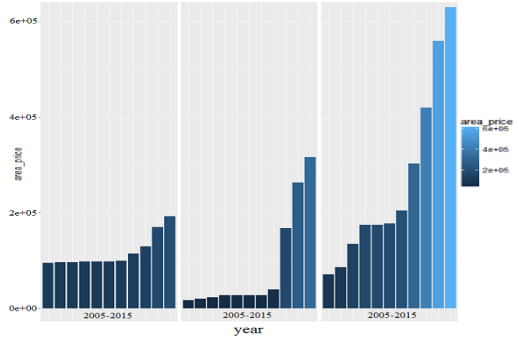

Fig. 3 Annual land prices in three areas in Taichung city from 2006 to 2015. From left to right, the three areas are old downtown, developing commercial and trade parks, and new City hall and culture centers.

It is difficult to directly attain information from the massive

raw land price data retrieved by Jsoup; hence, we used R scripts, which can speedily access huge data and offer powerful plotting features to elegantly depict data. Fig. 3 shows plots of annual land prices during 2006 to 2015 in three city areas in Taichung: Yizhong street (old downtown neighborhood), Fengjia night-market (developing commercial and trade parks area), and the 7th Re-planning District (new City hall and culture centers). The land prices have been increasing for all three areas during the past 10 years; especially, great jumps of land prices are witnessed in the last 3 years.

B. Land Price via Clustering Analysis

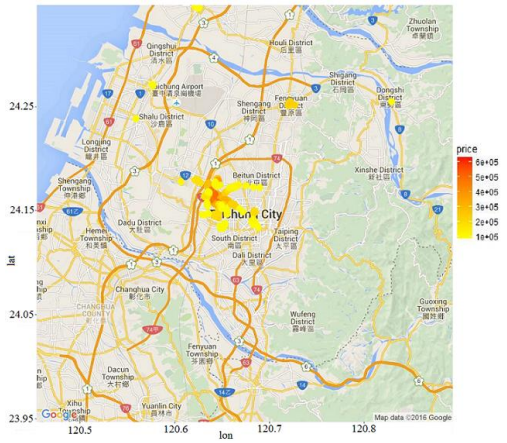

Fig. 4 Plot of clustering analysis results for land price in Taichung city from 2006 to 2015.

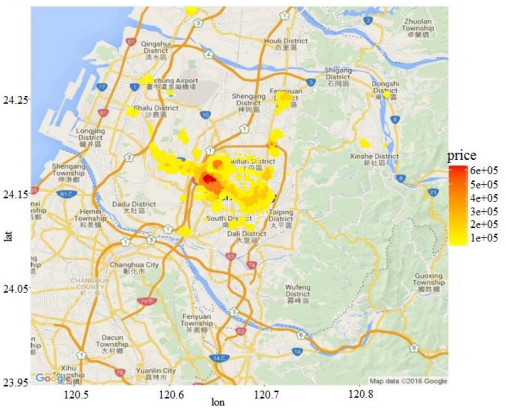

Fig. 5 Plot of clustering analysis results for land price variations in Taichung city from 2006 to 2015.

After clustering analysis, the land price data are displayed over Google Maps. To do this, we used the R mapping tool, the ggmaps package with ggplot2, to visualize spatial data on top of Google Map.

Fig. 4 shows the results of clustering analysis of land price in Taichung city. From the clustering results, we observe that locations of higher land prices (more than NTD 100,000 per Taiwanese ping) are around train stations (which are covered in yellow on the map); as one expected, starting from those traffic centers, cities and towns grow, expand, and connect one another.

C. Land Price Variations via Clustering Analysis

Fig. 5 shows results of clustering analysis for land price variations in Taichung city from 2006 to 2015. In addition to urban areas of Taichung city, the locations with higher land price changes in the past decade include areas that are near to train stations and revolve around the two mass rapid transit (MRT) routes. It shows that transport infrastructure has significant impacts on land prices.

D. Computation time with K-Means

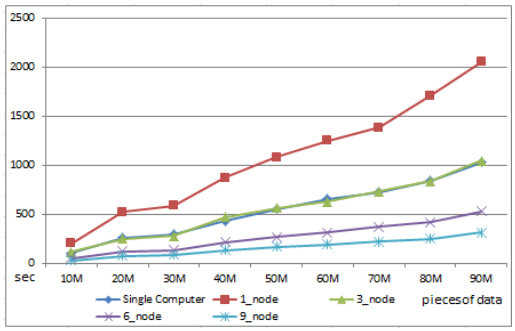

Fig. 6 Computation time of K-Means for varying pieces of data executed on PC and clouds of different nodes.

Fig. 6 shows computation times spent in performing the K-Means clustering algorithm for 10 to 90 million pieces of data executed on PC and clouds of different node numbers. The cloud with one node is slowest (slower than the PC), since it takes time to deploy data in Hadoop; however, the computation time decreases as the node number of the Hadoop cloud increases. We observe that the speedup rate of the cloud with 9 nodes over the PC is about 3.3 (1000/300) times when performing clustering analysis on 90 million pieces of data.

E. Computation time with Fuzzy C-Means

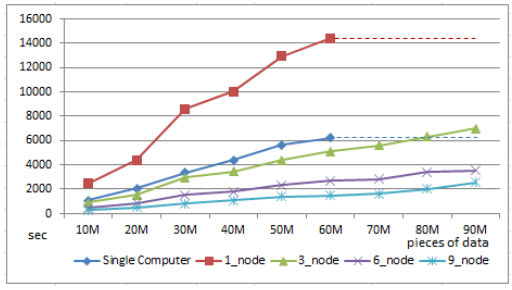

Fig. 7 Computation time of Fuzzy C-Means for varying pieces of data executed on PC and clouds of different nodes.

Fig. 7 shows computation times spent in performing the Fuzzy C-Means clustering algorithm for 10 to 90 million pieces

 ISBN 978-1-5090-4897-7

of data executed on the PC and clouds of different node numbers. We observe that performance of R codes is influenced more significantly by complexity of computation than the amount of data, since Fuzzy C-Means uses many iteration operations and takes longer computation time (2500 seconds for the cloud with 9 nodes in processing 90 million pieces of data, in Fig. 7) than K-Means (300 seconds, in Fig. 6).

Besides, two dotted lines are shown in Fig. 7, since the Fuzzy C-Means algorithm cannot be completed on the PC and the cloud with 1 node, due to insufficient memory when the size of data grows too large (greater than 70 million pieces). Therefore, in face of the arrival of the big data era, combining the R language with distributed computing platform such as Apache Hadoop is imperative in order to significantly reduce the operation time and cope with insufficient memory issues.

Conclusions

In this study, we perform K-Means and Fuzzy C-Means clustering algorithms on the past 10 years land price data of Taichung city and display clustering results on top of Google Maps. We observe that transport infrastructure and city development planning have significant effect on the land price. We also perform performance evaluation of executing the clustering algorithms on the PC and Hadoop cloud systems with various compute nodes. We find that by combining R with Hadoop distributing computing, the computation time can be greatly reduced and the inefficient memory issue can be solved with adequate number of compute nodes.

Acknowledgments: This research was partly supported by Ministry of Science and Technology, Taiwan, under grant number MOST 105-2221-E-029-012-MY2.

References

[1] M. Thangavel, P. Varalakshmi, S. Sridhar, "An analysis of privacy preservation schemes in cloud computing," *The second IEEE International Conference on Engineering and Technology*, 2016.

[2] The Apache Software Foundation, Apache POI. https://poi.apache.org/.

[3] K. Shvachko, H. Kuang, S. Radia, R. Chansler, "The Hadoop Distributed File System," *IEEE 26th Symposium on Mass Storage Systems and Technologies*, 2010.

[4] K. Pandey, A. Gadwal, P. Lakkadwala, "Hadoop multi node cluster resource analysis," *Symposium on Colossal Data Analysis and Networking*, 2016.

[5] The Apache Software Foundation, *HDFS*: *Permissions User and Administrator Guide*, 2007.

[6] S. Perera and T. Gunarathne, *Hadoop MapReduce Cookbook, Recipes for analyzing large and complex datasets with Hadoop MapReduce*, Packt Publishing, 2013.

[7] J. Dean and S. Ghemawat, *MapReduce*: *Simplified Data Processing on Large Clusters*, Communications of the ACM, Vol. 51, 2008, pp. 107-113.

[8] R Core Team (2016). *R*: *A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

[9] J. A. Hartigan, M. A. Wong, "*A K-means clustering algorithm*," Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28, No. 1, 1979, pp. 100-108.

[10] J. C. Bezdek, R. Ehrlich, W. Full, "*FCM: the fuzzy c-means clustering algorithm*," Computers & Geosciences Vol. 10, No. 2-3, 1984, pp. 191-203.