Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 24 (2016) 1393 – 1399

Procedia Technology

International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015)

A Scalable Product Recommendations using Collaborative Filtering in Hadoop for Bigdata

Riyaz P A, Surekha Mariam Varghese

M A College of Engineering, Kothamangalam, Kerala, India
M A College of Engineering, Kothamangalam, Kerala, India

Abstract

The growth of data and information causes the need of next-generation databases and data science tools. Most of the business needs a service recommendation system which have been used by millions of users. Day by day, the amount of customers, products and information has grown rapidly, yielding the big data analysis problem for service recommender systems. Consequently, conventional recommender service systems often suffer from lack of scalability and efficiency problems when processing or analysis of this data on a large scale. To avoid these problems, a novel recommendations system using collaborative filtering algorithm is implemented in Apache Hadoop leveraging MapReduce paradigm for Bigdata. Apache Hadoop is an open framework for Distributed processing systems can process large volumes of data. It can be used for offline processing and not suitable for low latency analytics. Port data onto the next generation databases like HBase and optimize the performance of it. For the product recommendations the Amazon dataset is used. Proposed Framework have significant improvement in performance compared to conventional tools.

Peer-review under responsibility of the organizing committee of ICETEST – 2015

Keywords: Bigdata; Recommendations; Product; Hadoop; MapReduce; HBase; Collaborative;

1. Introduction

Ecommerce build their services and business to quickly improve, but the data produced by them are still maintain some inherent features and complexities that are difficult to address. Product datasets are continuously becoming larger, thus making it increasingly difficult for standalone systems to process products data. The large amount of data available on the web in the form of ratings, ranks, reviews, opinions, complain, remarks, feedback, and comments about any item (product, event, individual and services) can be used for making correct decision.

Peer-review under responsibility of the organizing committee of ICETEST – 2015
doi:10.1016/j.protcy.2016.05.159

Moreover lots of blog forums are available on the web where web users can give their opinion, reviews, and comments about the items. The recommendation based on the ratings and summary of relevant text about the items can be used for decision making. The growth of e-commerce sites and online businesses are enhancing the requirements of a robust recommendation system. Now a day the millions of users buy products from online shopping websites. Understanding the logic behind product big data (PBD) has great significance for designing Ecommerce Applications (EA) that can be used for recommending services.

Fig. 1 shows the features of Bigdata. The bigdata, the massive volume as well as rapid rates of data will makes conventional systems difficult to handle. The open-source softwares are pretty to use now a day mainly Apache Hadoop provide a framework for terabyte-scale data warehouses on multiple node of clusters, thus enabling scalable and distributed analysis of PBD using the MapReduce programing model.

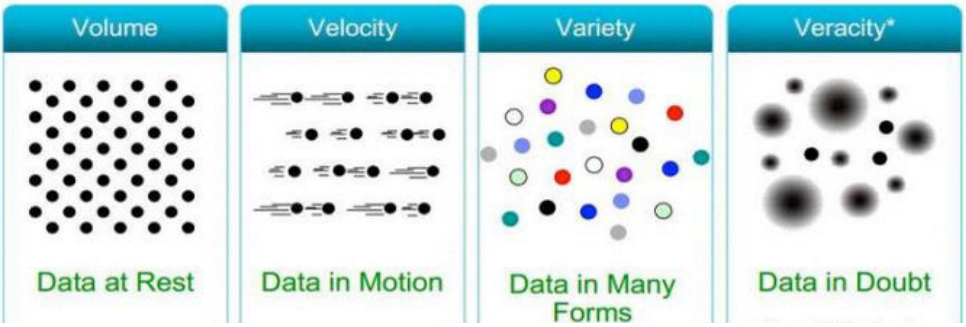


Fig. 1. Four V's of Bigdata

In this paper, recommend the products or services in a real-time manner, prediction should be accurate and provide scalability, these are the main objectives. We developed Hadoop-based [1] applications named as Scalable Product Recommendations using Collaborative Filtering in Hadoop for Bigdata, to intelligently process PBD with three-node Hadoop nodes to execute distributed MapReduce algorithms. Compared with single-node algorithms, our multiple node or distributed algorithms show promise for facilitating efficient Product Big Data processing. Moreover, with PBD analytics, we can optimize the HBase which is a NoSQL database to provide better read performance, which may lead to better low latency analytics.

2. Related Works

Zhiyang Jia and Wei Gao proposed the recommender system is constructed as an online application which is capable of generating a personalized list of preference attractions for the tourist [2]. Modern technologies of classical recommender system, such as collaborative filtering are considered to be effectively adopted in the tourism domain.

On the basis of the collaborative filtering principle, the recommendation process of tourist attractions divided into three steps. The first step is representation of user (tourist) information (the visiting history of attractions by tourist need to be analyzed and modeled).

Jyoti Gupta proposed a system that predict using item based collaborative filtering is combined with prediction using demographics based user clusters in a weighted scheme [3]. The proposed solution is scalable while successfully addressing user cold start. In this item based collaborative filtering (IBCF) is combined with demographics based collaborative filtering (DBCF) in a hybrid weighted approach.

Shunmei Meng and Wanchun Dou developed a system aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively [4]. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate

recommendations. To improve its scalability and efficiency in big data environment, KASR is implemented on Hadoop, a widely-adopted distributed computing platform using the MapReduce parallel processing paradigm.

A lot of works are use Hadoop for the scalable applications. But there is a still room for improvement in many areas and corners. Processing in MapReduce makes fast process, but cannot use with low latency analytics. The recommendation should be done in real-time.

Another important thing is the number of users who are use the recommendation system. Now it is millions of users, in future it may be billion. Provide recommendation or any other services in low latency in major issue in all the applications in coming days. Leveraging HBase database, distributed and column-oriented which will provide low latency analytics.

3. System Architecture

Our Apache Hadoop based product recommendations system has three components, as shown in Fig. 2. The components are Hadoop nodes, distributed recommendation engine and Hbase Storage. Combined with the applications that produce a distributed recommender interface. The amazon product dataset is used to recommend the products. Product Bigdata can be stored in HDFS. The functions of each component are described below.

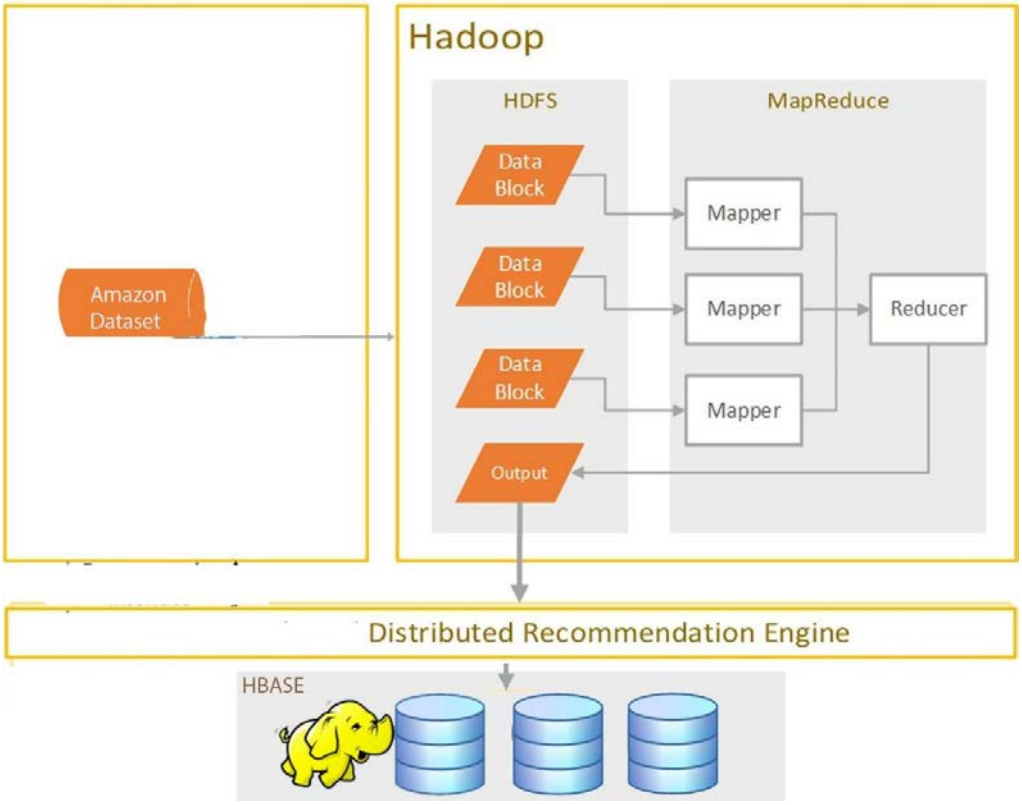


Fig. 2. The architecture of Proposed Framework

3.1. Data Extraction

The amazon product dataset consists of reviews. Each review will have rating associated with it. Every customer have a customer Id. The amazon product data is loaded into the Hadoop clusters with MapReduce paradigm. The data is changed according to the customer rather than the product. The user tastes are compared with others. So the data is arranged as a customer with list of products he had purchased. PBD leverages MapReduce technique for fast loading.

Data extraction problems framed as key-value pairs can be efficiently distributed with Hadoop and HDFS [5]. Created a custom Input Format to read the amazon dataset.

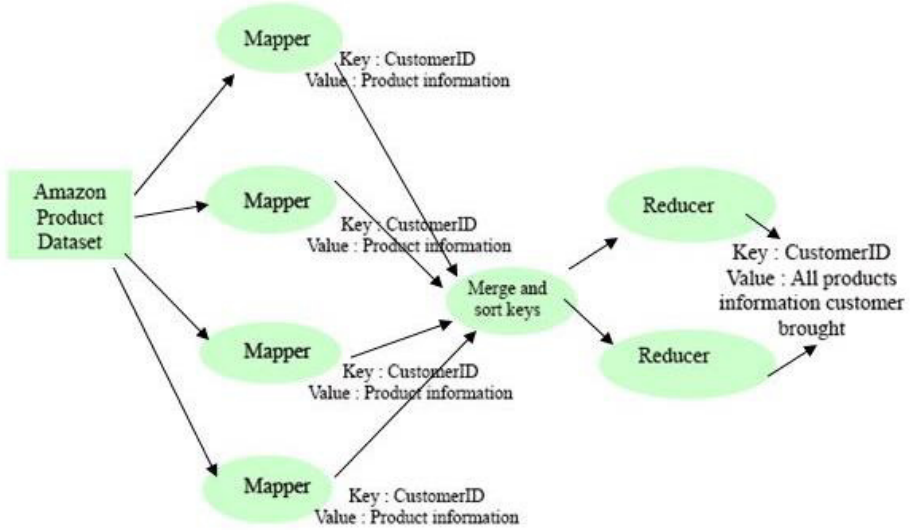


Fig. 3. Data Extraction by MapReduce

The amazon data formatter will parse the dataset and emit the data about each Amazon product as key-value pairs. It emits the key-value pairs to the map function. When the map task receives the product data, it emitted the CustomerID as the key and product data as the value for each customer who has brought the product.

Then, Hadoop collects all values for the key and invokes the Reducer once for each key. There will be a reduce function invocation for each customer, and each of those invocations will receive all products that have been bought by a customer. The Reducer emits the list of items bought by each customer, thus building a customer profile. Each of the items contains product rating as well.

3.2. Data Analysis

The objective of this part is to build a scalable big data analysis system with recommendation-related algorithms implemented on top of the Hadoop. For the data science part, used python with MapReduce that is scalable to large data sets. Python is powerful language which is used for machine learning recent times. In our system, we primarily use the Collobartaive Filtering algorithm to analyze Amazon Product Bigdata [6] which is stored in the Hadoop cluster to make recommendations for external applications.

3.3. Algorithm Realization

To realize Collaborative Filtering, we need to perform the following steps: (1) collect the user-preferences, (2) find the similar items based on the user tastes and (3) calculate the recommendations. First, we use the data extraction module to get the data from amazon product dataset. The data need to groups into user basis [7]. Then we can collect user preferences from the amazon product dataset with historical information about user preferences is transformed into a simple triple:

$$< CustomerID, product\_Title, Rating > \tag{1}$$

Then, we use Pearson correlation coefficient (PCC) measure to calculate the similarity. Compared to cosine similarity and Euclidean distance the PCC is better. It first finds the items rated by both users. Then calculates the sums and the sum of the squares of the product ratings for both the users and calculates the sum of the products of their ratings. Finally, it uses these results to calculate the Pearson correlation coefficient.

$$\begin{aligned} S_{xx} &= \sum x^2 - \left(\sum x\right)^2 / n \\ S_{yy} &= \sum y^2 - \left(\sum y\right)^2 / n \\ S_{xy} &= \sum xy \ - \left(\sum x\right)\left(\sum y\right) / n \\ r &= \frac{S_{xy}}{\sqrt{S_{xx} S_{yy}}} \end{aligned} \tag{2}$$

Item to item Collaborative Filtering has proved as more scalable than user to user Collaborative Filtering [8] and is able to handle large user bases. It uses the similarities between items for making recommendations. It is based on past behaviour of user and recommend items that are similar to that were liked by user in past. The basic idea behind Item to Item Collaborative Filtering [9] is that, if two items have same rating from some users, or have same features it means they are similar items and next time when a user like one item of those two then he may like the other item as well. Let's say item P, Q and R are rated similarly by user A so now they are similar item, when user B liked item Q in past then he will get suggestions for item P and R.

3.4. Data Storage

Fig. 4. HBase Architecture

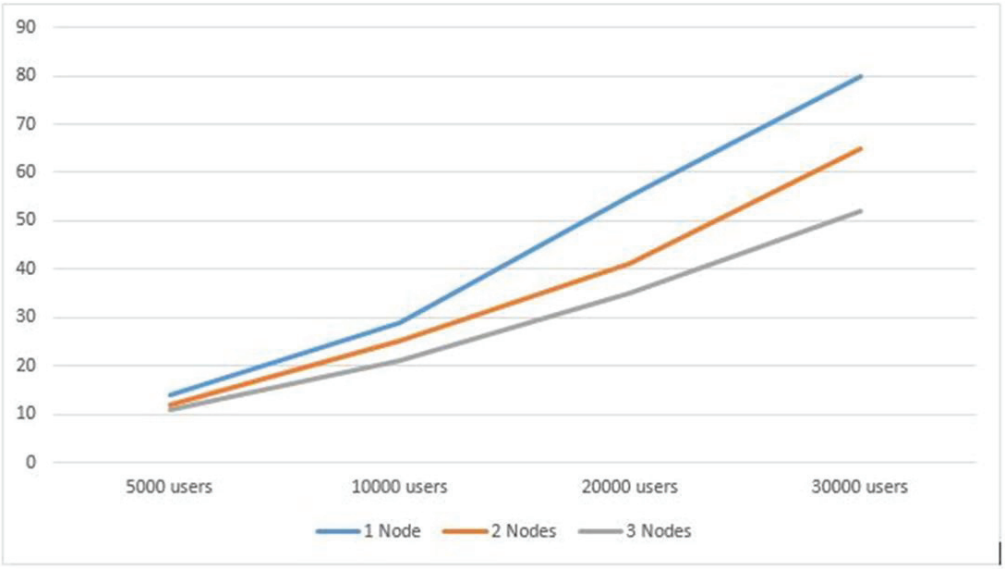
Such stores are used by interactive, user facing portions of applications. The second type is used for analytical workloads and emphasizes write throughput and sequential reads over latency or random access. This forces applications to be broken into "fastpath" processing, and asynchronous analytical tasks.

HBase store the recommendations. HBase used Bloom Filters which reduce the extra seek on the disk. The architecture of HBase as shown in Fig 4, consists of multiple HRegionservers managed by the HMaster.

Each region store the tables. Basically HBase is a distributed database. The all activities in HBase is coordinated by the Zookeeper.

With the increment of the number of the recommendations in the table, one table will be spilt into multiple slices which called the regions. Different regions will be assigned to the appropriate HRegionserver for management, and ultimately the data is written to the HDFS(one distributed file system). HBase provides low latency to the ecommerce applications.

4. Performance Evaluation



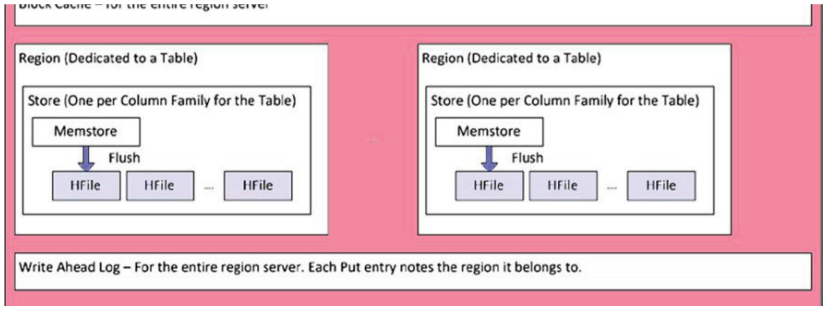


Fig. 5. Performance Graph – Users vs Datanodes

For the performance evaluation purposes, the amazon product dataset is used. The hardware configuration of used systems are Intel Pentium Dual core 2 GHz processors with 4 GB 1066 MHz RAM. The operating system used is Ubuntu 15.10. The version of Hadoop is 2.6.0. In Fig 5, x-axis represents the number of users and y-axis represents the time in seconds. When the number of datanodes increases then it gradually improve the parallel processing capability of the system. For a small size data, small number of data nodes is enough. For a large size data, the higher number of datanode significantly improve the performance of the system.

5. Conclusion

This paper gives a scalable product recommendations collaborative filtering for Bigdata on a Hadoop-based processing system. An optimized HBase gives better performance. For low latency applications HBase is highly preferred because of distributed architecture and leverage the power of Apache Hadoop. As the size of data increases the Hadoop performs well by adding more datanodes into the processing. Collaborative Filtering is one of the best algorithm for the product recommendations.

References

[1] Apache Hadoop, https://hadoop.apache.org/

[2] Zhiyang Jia, Wei Gao, Yuting Yang, Xu Chen, User-based Collaborative Filtering for Tourist Attraction Recommendations, 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 978-1-4799-6023-1/15 © 2015 IEEE DOI 10.1109/CICT.2015.20

[3] Jyoti Gupta, Jayant Gadge, Performance Analysis of Recommendation System Based On Collaborative Filtering and Demographics, IEEE International Conference on Communication, Information & Computing Technology (ICCICT), 978-1-4799-5521-3, 2015 DOI: 10.1109/ICCICT.2015.7045675

[4] Shunmei Meng, Wanchun Dou, Xuyun Zhang, and Jinjun Chen, KASR: A Keyword Aware Service Recommendation Method on MapReduce for Big Data Applications, IEEE 2014

[5] Riyaz P.A., Surekha Mariam Varghese, Leveraging MapReduce with Hadoop for Weather Data Analytics , IOSR - Journal for Computer Science, 2015, volume 17, issue 3, pp 6-12.

[6] Wu Yueping and Zheng Jianguo, "A research of recommendation algorithm based on cloud model", IEEE 2010.

[7] Yanhong Guo et al., "An improved collaborative filtering algorithm based on trust in e-commerce recommendation system", IEEE 2010.

[8] B.M. Sarwar et al., "Item-item Collaborative Filtering Recommendation Algorithms," 10th Int'l World Wide Web Conference, ACM Press, 2001, pp. 285-295.

[9] B.M. Sarwarm et al., "Analysis of Recommendation Algorithms for E-Commerce," ACM Conf. Electronic Commerce, ACM Press, 2000, pp.158-167.