Distributed processing using cosine similarity for mapping Big Data in Hadoop

Andres Felipe Rojas Hernandez and Nancy Yaneth Gelvez Garcia

Universidad Distrital Francisco Jose de Caldas

Abstract. The analysis of big data is an issue that has become very important in recent years. The use of algorithms for processing big data that have generated as a result valuable information to organizations can be considered one of the biggest developments and most important lines of work today. This paper aims to show results in implementing cosine similarity for mapping big data in a flat database. For this purpose the information from movie ratings will be used, so it will result in a recommendation of a movie highly related to another. If the information used for testing is considered real, these results could be useful for the development of a recommendation system for products and services from an organization which has as well the records of their customers' ratings.

Keywords: Big Data, Hadoop, Cluster, Cosine similarity.

1 Introduction

This paper presents the results and analysis of the use of a recommendation system that implements the cosine similarity algorithm for mapping records from a flat database. The records include information from one hundred thousand movies from the MovieLens database (https://movielens.org/) and represent what can be understood as big data with an open and free distribution character. In order to process this information it is necessary the use of distributed computing due to the high computational costs. EMR (Elastic Map Reduce) is a solution of Amazon Web Services that provides a user-friendly environment for cloud distributed computing. Together, there is an environment for data processing in a distributed manner which is intended to analyze the performance and reliability, also the mapping is to analyze the results and what this information can mean for an organization.

2 Big Data

Currently, technologies are capable of storing and processing an increasing amount of data. Data of this type is what is known as Big Data. Big data is the term for a collection of large and complex data sets that are difficult to process with traditional data processing tools. Big data can be characterized by three V's: Volume (large amounts of data), Variety (includes different types of data) and Velocity (constant accumulation of new data) [1]. Data becomes big data when its volume, variety of velocity exceed the capabilities of computer systems to store, analyze and process them. Recently the understanding of big data has been expanded by adding two more V components. So, you can characterize big data in five V's: Volume, Velocity, Variety, Veracity and Value [2]. Big data is not only large amount of data, it is actually a new concept that provides an opportunity to find a new vision of existing data.

There are many applications of big data: business, technology, communications, medicine, health, bioinformatics (genetics), science, e-commerce, finances, Internet (information search, social networks), etc. Big data can come not only from computers but also from billions of mobile phones, social media messages, different car installed sensors, utility meters, transportation and many more. In many cases data is being generated even faster than it can be analyzed.

Big data can include structured and unstructured information. Unstructured data is the one that either does not have a pre-defined data model or is not organized. Structured data is relatively simple and easy to analyze because, usually this data resides in databases as columns and rows. The challenge for scientists is to develop tools to transform unstructured to structured data [3].

When it comes to big data, grouping it becomes a problem. Often data sets, especially large data sets consist of some groups (clusters) that need to be processed simultaneously. The Cluster method has been applied to many important problems [5] due to the potential of computing distribution. For example, to discover health trends in patient records, to eliminate duplicate address lists entries, to identify new classes of stars in astronomical data, to divide data in meaningful and useful groups and to group millions of documents or websites. To address these applications and many others a variety of clustering algorithms have been developed. There are some limitations in the existing clustering methods. Most algorithms require exploring data sets several times, so they are not suitable to be processed in a single node. Data cluster provides a more reliable and plausible solution for processing big data.

2.1 Hadoop

When it comes to big data, it is about challenges such as data rate, data volume and variety of data. EDW and Hadoop technologies can be useful for managing these challenges. Hadoop is an open source framework and is the source of technology that preceded almost every data storage and analysis tools that have been labeled as 'Big Data' (http://hadoop.apache.org). With Hadoop it is possible to build easily, economically and effectively in a very large scale a data storage and processing solutions system. Hadoop file system (HDFS) allows to send data in Hadoop and then works with them simultaneously on all disks and all servers in the cluster. In the cluster there are multiple computers so Hadoop provides a new approach to distributed computing by implementing an idea called MapReduce (Reduced Mapping). MapReduce is essentially a programming model for processing big data under a parallel distributed algorithm that allows the separation, processing and aggregation of information. Compared with traditional relational database management systems (RDBMS), Hadoop provides improved query response times and integrity with other data analysis products [3].

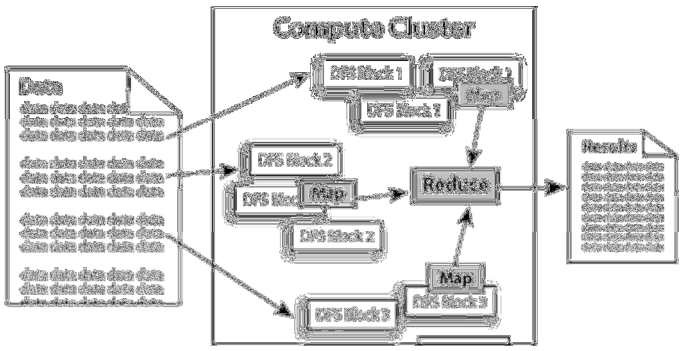

Fig.1. Information flow in a Hadoop Cluster (Source:http://sentidoweb.com/2007/11/21/hadoop-plataforma-para-trabajar-con-gran-cantidad-de-datos.php).

2.2 Amazon Elastic Map Reduce

Elastic Map Reduce (EMR) is a web service from Amazon Web Services (AWS) for fast and cost-effective processing of big data. It simplifies big data processing by providing a self-managed Hadoop framework that facilitates the distribution and processing of big data between instances dynamically scalable called Elastic Cluster (EC2). EMR manages safely and reliably their big data use cases including log analysis, web indexing, data storage, machine learning, financial analysis, scientific simulation and bioinformatics.

EMR uses Hadoop processing for tasks such as web indexing, data mining, log file analysis, machine learning, scientific simulation and data storage [7].

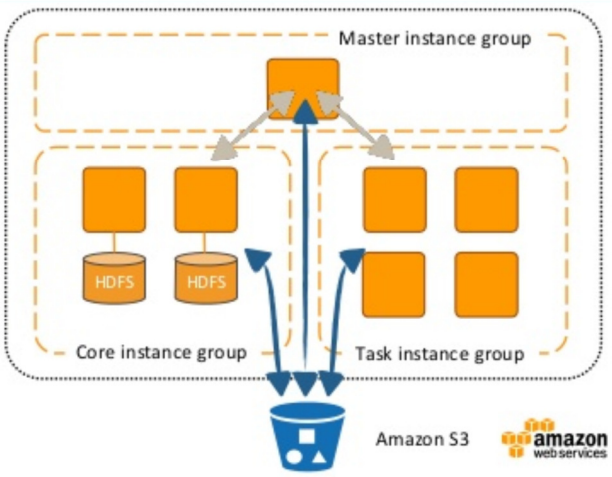

Fig.2. EMR Architecture
(Source: http://www.slideshare.net/AmazonWebServices/clickstream-analytics-amazon-kinesis-and-apache-storm)

3 Cosine Similarity

Cosine similarity erroneously called cosine distance is really a mapping similarity measurement. Cosine similarity is a similarity measurement between two vectors in a space that has an inner product which is used to evaluate the cosine value of the angle between them. This trigonometric function provides a value equal to one if the angle is zero. For any angle between the vectors the cosine would result in a value less than one. If the vectors were orthogonal the cosine would be nullified and if they were pointing in the opposite direction its value would be minus one. Thus, the value of this metric is between minus one and one, i.e. in the closed interval [-1, 1]. This is the formula to calculate the cosine for two vectors with Sij features (1).

$$Cosine\ (a,b) = \frac{\sum_{i,j}^{N} S_{ij} a_i b_j}{\sqrt{\sum_{i,j}^{N} S_{ij} a_i a_j}\sqrt{\sum_{i,j}^{N} S_{ij} a_i a_j}} \qquad (1)$$

This distance is often used in information retrieval representing words (or documents) in a vector space. Once represented the documents and queries as vectors we can measure their similarity. An alternative is to use the Euclidean distance but the difference in length between documents would affect the metric. What is most often used is the cosine of the angle between the vectors as a similarity measurement. In text mining cosine similarity is applied in order to establish a resemblance metric between texts. It is often used as a cohesion

indicator of text clusters. The complexity of this measurement is quadratic, which makes it completely applicable to real-world problems. This complexity can even be transformed into a linear one [6].

4 Mapping Application

To conduct an analysis of information in big data we proceed with the following steps: preprocessing of the information, distribution, processing, results and analysis. The preprocessing of the information is the identification of variables that are necessary and will be included in the analysis, as well as the collation and digitalization of information that can come from any source, it also cleans data of unnecessary records. In the application this is done during mapping. The distribution is the establishment and distribution of information in a cluster where distributed computing can be done. In this case, it corresponds to the EMR instances. The processing of the information is the core of the entire operation. In this case, it is the cosine similarity mapping. At the end there are the results which provide information that can be analyzed. In this paper all those steps are not discussed since the information used was collected externally by the organization MovieLens.

4.1 Preprocessing of the information

The data that will be used for the application correspond to one hundred thousand movie records whose information is in digital format in a plain text document. This means that it is not necessary to collect, classify nor organize information. What is necessary is to identify the variables to be mapped or more precisely, the variables that are necessary to establish a correlation. If you consider that each record in the database is given of the form [user ID, movie ID, rating, time tag], then when you run the mapping the data will have the form (key: movie ID, [ratings]) discarding the other data because it is not important for the correlation. Previously, all the names and movie IDs are taken from another file in order to use them after the correlation to make the results more legible. The latter is included as another mapping step and only influences the processing time. Below it is displayed a portion of the file containing the movie ratings and the one containing the names. (See Fig. 4 and 5)

```
22    377   1     878887116
244   51    2     880606923
166   346   1     886397596
298   474   4     884182806
115   265   2     881171488
```

Fig.4. Ratings file sample

```
1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?
Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?
GoldenEye%20(1995)|0|1|1|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-
exact?Four%20Rooms%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-
exact?Get%20Shorty%20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?
Copycat%20(1995)|0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|1|0|0
```

Fig.5. Names file sample

4.2 Distribution and processing

For the distribution it is precise the use of the Amazon S3 scalable cloud storage, other of the AWS services. Initially data is uploaded to the cloud, then the mapping and reduction in EMR is executed and finally the response is received on the local computer. The response could be stored in S3 but it is a dispensable procedure in this case. It should be remembered that the purpose of this paper is the performance analysis of the correlation in different configurations of a cluster and then the mapping procedure is the same for all tests.

An initial test is performed as a control sample in a local machine with depreciable features. The same procedure is later done in a slaveless cluster or an EMR machine. Consecutively then the same procedure is performed with more complex cluster configurations as it follows: one master machine and a slave; one master machine and two slaves, and finally one master machine and three slaves. In all cases the same procedure is followed step by step. Mapping for extracting variables, reducing to group movies and ratings for each user, mapping to link paired films with paired ratings, reducing by executing the cosine based similarity correlation, mapping to change the movie IDs for their names and finally reducing to generate an output file. Below is a chart where the whole process becomes clearer. (See Fig. 6)

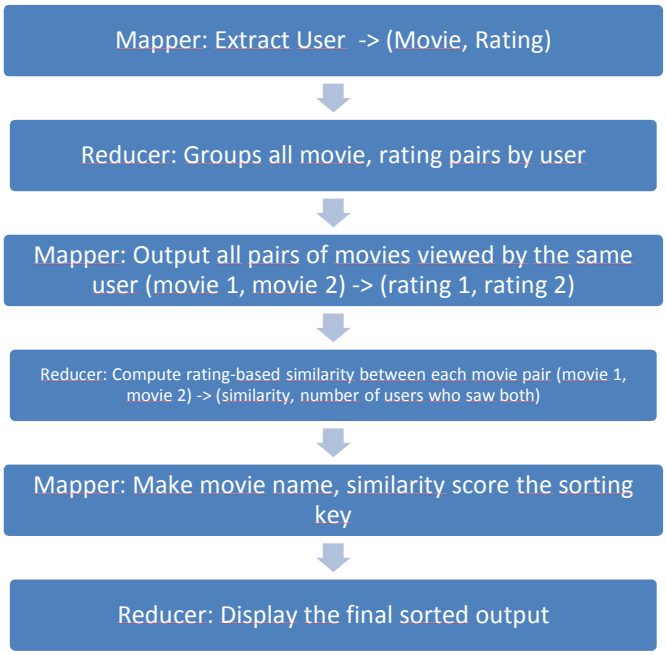

Fig.6. Steps for the correlation test

4.3 Results

For this paper the results can be classified into two types: data processing results and cluster performance metrics.

The results of data processing correspond to the result of the correlation of movie records and their ratings. The aim in using the cosine similarity is to provide information that is easily readable and say how related regarding their rating are some movies from others. At the end the result is a set of records of the form [movie

name one, movie name two, correlation level, number of ratings]. Movie name one and two is the filter after merging the name record (See Fig. 2) with the correlation result. The correlation level is a number between zero and one that represents the correlation percentage between movie one and two. The number of ratings corresponds to the number of times movie two was rated. Thus, it can be interpreted that the movie that has more correlation with other is the one that has a higher level of correlation and has been greatly qualified. (See Fig. 7)

```
"Star Wars (1977)"    ["Nell (1994)", 0.95440149533124929, 75]
"Star Wars (1977)"    ["Jumanji (1995)", 0.95454005841416234, 91]
"Star Wars (1977)"    ["Shine (1996)", 0.95478403680312518, 104]
"Star Wars (1977)"    ["Mary Poppins (1964)",
```

Fig.7. Final correlation file sample

The metric results for the tests corresponding to four cluster configurations each one adding a node to the control test are presented below. The identification of each step is mapping (Step 1), the cosine similarity correlation (Step 2) and the organization and result display (Step 3). All the above steps correspond to a MapReduce process (Mapping and
Reduction).

Table 1. Runtime Steps for each cluster configuration

Step (time) / Cluster Conf.	Local Machine	(1) Control Machine	(2) Cluster with 1 slave	(3) Cluster with 2 slaves	(4) Cluster with 3 slaves
Step 1	--	2 min	2 min	2 min	3 min
Step 2	--	37 min	35 min	21 min	14 min
Step 3	--	2 min	2 min	2 min	2 min
Total	50 -60 min	50 min	48 min	34 min	27 min

In addition to the relationship between steps and processing time, AWS provides an analytical section wherein the following graphs are generated (See Fig. 8 to 10). These are the ratio of consumption of the clusters HDFS storage system over time given in bytes. It is important the analysis of this data as in a production environment since the ability to improve the performance of a process is a critical procedure.

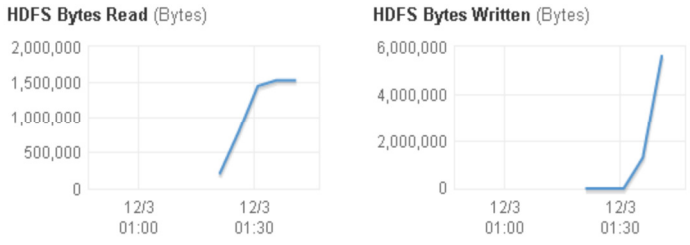

Fig. 8. HDFS file system performance for control machine (1)

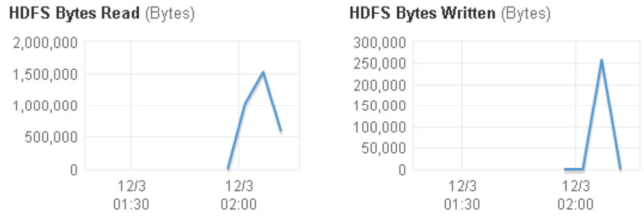

Fig.9. HDFS file system performance for cluster (2)

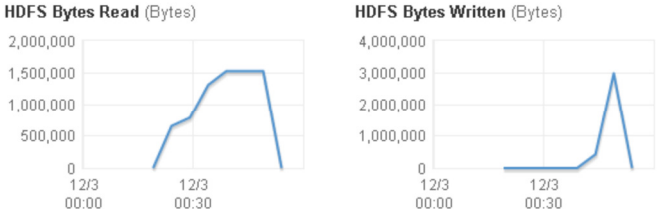

Fig.10. HDFS file system performance for cluster (3)

5 Analysis

Product and service recommendation systems are currently on the web a matter of primary concern. The test of an algorithm than can correlate ratings not only of movies but from any type of product then can take even more relevance. From the results of Table 1 it can be concluded that the use of distributed computing significantly reduces the information processing time in critical processes. However, and contrary to common sense, adding more nodes to the cluster does not make the processing time divide into the number of nodes compared to a single machine. Time is indeed reduced but not in direct proportion to the number of nodes in the cluster.

Regarding the clusters HDFS file system performance it can be said that adding more nodes will affect dramatically the speed and amount of data input and output. It can pass the order of six million byte write (See Fig 8) on a single node cluster to two hundred fifty thousand bytes (See Fig 11) in a four node cluster.

To improve the performance when processing the data there are some changes proposed such as: discard the bad ratings before making the correlation i.e. at the time of mapping, change the correlation type (Pearson correlation, Jaccard coefficient, conditional probability). Although it is not the purpose of this paper, it is a good way to improve the patch. Other changes are adjust the threshold of minimum amount of ratings or minimum ratings, suggest a new similarity that can be correlated as well and take into account the number of ratings. All of the above is not studied in this paper but since these factors directly affect the performance it is worth proposing them.

6 Conclusions

Cloud distributed computing by cluster is a very good alternative to traditional computing especially when it is intended to process and analyze big data. Some companies offer services for cloud computing. The Elastic Map Reduce service based on Hadoop is very easy to use, it requires no maintenance or configuration and still, has the processing potential from any other distributed computing environment. It also has the ability to scale according to the processing needs. Using correlation based on cosine similarity is a procedure that can be used as a base for a product recommendation system. Although it is not the fastest correlation it has good reliability and an acceptable computational complexity performance. Finally, although cluster distributed computing is better in volume, variety and velocity this does not mean that the work and process runtime scale linearly compared to the number of nodes in the cluster.

References

[1] S. Schmidt, Data is exploding the 3V's of big data, Business, Computing World, 2012.

[2] KY. Zhai, Y-S. Ong, and IW. Tsang, the Emerging "Big Dimensionality". In Proceedings of the 22nd International on World Wide Web Companion. Computational. Intelligence Magazine IEE, Vol 9, no. 3, pp. 14-26, 2014

[3] Olga Kurasova, Virginijus Marcinkevicius, Viktor Medvedev, Aurimas Rapecka, and Pavel Stefanovic. Strategies for Big Data Clustering. Computational. IEEE 26th International Conference on Tools with Artificial Intelligence, 2014

[4] Kannan Govindarajan1, Thamarai Selvi Somasundaram1, Vivekanandan S Kumar, Kinshuk, Clustering in Big Data Learning Analytics, IEEE Fifth International Conference on Technology for Education, 2013

[5] McCallum, K. Nigam, and L. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching", in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178, 2000.

[6] Felipe Bravo Márquez, Procesamiento de texto y modelo vectorial, http://www.cs.waikato.ac.nz/~fjb11/clases/irintro.pdf

[7] Elastic Map Reduce from Official AWS web, https://aws.amazon.com/es/elasticmapreduce/

[8] MovieLens web, Base de datos para la recomendación de Películas, https://movielens.org/

[9] Frank Kane, Data Scientists, Taming Bid Data with MapReduce and Hadoop, Online Course, https://www.udemy.com/taming-big-data-with-mapreduce-and-hadoop