Data Mining Engine based on Big Data

Guo Song
NanChang Institute of Science & Technology

Abstract- In the environment of billions of data, aspect mining, real-time processing, ad hoc analysis, and offline computation ask higher requirement on calculation and storage performance. However, the data mining platform realized based on traditional relational database, distributed Hadoop platform cannot satisfy all tasks. This paper completes two traditional data mining algorithms-parallel transformation of Apriori and PageRank based on the in-memory computing module of Spark and its several actions as well as conversion operators. The conducted experiment validates the implementation efficiency as well as parallel effect of these two algorithms. This platform does not only take full advantage of in-memory computing, improve iteration speed but also support various distributed computing and storage scenes with strong expandability, which can perfectly deal with various scenario issues in big data environment.

Keywords-Big data; Data mining; Spark

I. INTRODUCTION

In recent years, with the rapid development of information technology, Internet, intelligent devices, and Internet of Things all develop quickly and therefore people become more capable of collecting and producing data. As a result, the data scale, dimension and types increase while big data emerges as the times require.

Big data continually emerges around us and the digital process greatly advances the growth of data. The data collected by each intelligent device, user's Web log and upload data in social media are resources of bid data. The blog ignited by the search engine of Baidu may reach to hundred GB per hour while the accessing data created in 11th November in Taobao is about dozens TB. However, traditional data storage and analysis technology cannot analyze the data scale of IB or even PB. In the past, such data scale is beyond imagination let alone analysis on these data so big data is only theoretical which cannot be adopted in reality.

The data scale has increased from GB to IB or even PB when people accumulate growing amounts of data. In order to find out the potential value of data, the common approach is to flexibly adopt various data mining algorithm based on reality. Even though data mining has been fully adopted and developed in traditional small database with its value and guidance meaning being confirmed its implementation efficiency, algorithm parallelization and platform accessibility are facing grand challenge in terms of big data.

This paper aims to come up with a storage and search engine for big data based on the above problems, trying to improve the implementation efficiency of big data's data mining algorithm and realize the transparency of underlying specific processing. Therefore, it is not necessary for data analysts to deeply grasp the distributed parallel computing and establish complicated computing environment. They can analyze the big data with the approach of dealing with small database, which greatly lowers the bar for mining big data.

II. THE BASIC PLATFORM FOR BIG DATA

At present, the big database platform prevails with the Hadoop ecosystem as the core and Hadoop can be beckoned as the synonym of big data. In Hadoop ecosystem, various open source application frameworks are realized which do not only lower the bar for users but also help countless users complete big data calculation. Figure 1 draws the main tool in Hadoop ecosystem.

HBase is a dynamic databse for structured data which is telescopic, reliable, distributed and column-oriented. It adopts strengthened sparse sorted map data module and the key is made up of row key, column Key and timestamp. The data stored in HBase can be processed with MapReduce which perfectly integrates data storage and parallel computing and can realize random access for data. Pigt6 defines Pig Latin, a kind of data-flow language, which transfers script into MapReduce and implements based on Hadoop. Hive defines a query language HQL which is similar to SQL, it transfers SQL into MapReduce and implements based on Hadoop and it is often adopted in off-line analysis. Zookeeper refers to data management problems in Chubby Clone distributed environment, including uniform naming, state synchronization, cluster management, and configuring synchronization etc. Sqoop is mainly used in data transmitting between traditional data and Hadoop. Mahout realizes widely used data mining approaches such as cluster, classification, recommendation engine (collaborative filtering) and frequent set mining. Flume refers to floristic characteristics of Cloudera's open source log gathering.

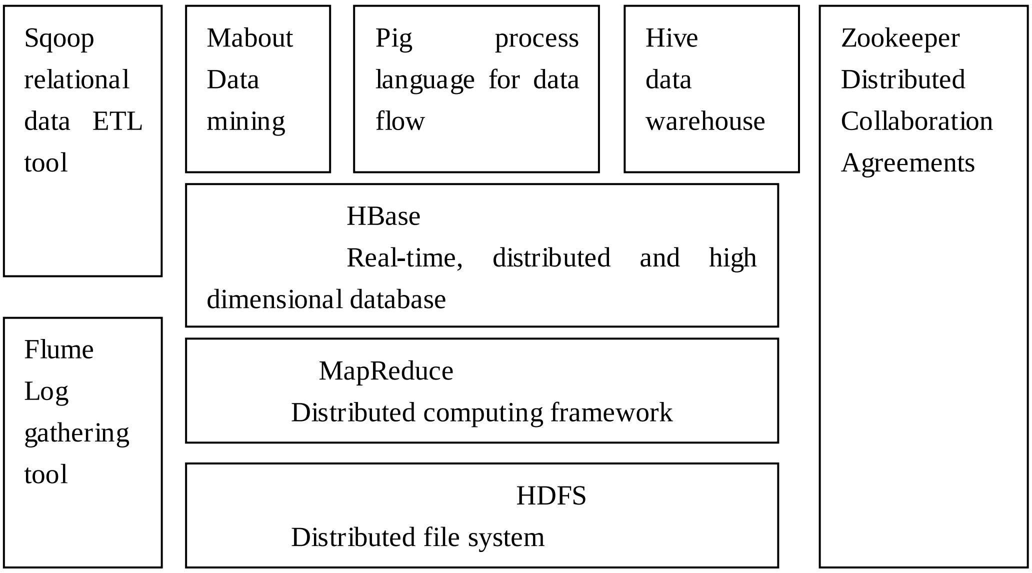

Sqoop relational data ETL tool
Mabout Data mining
Pig process language for data flow
Hive data warehouse
Zookeeper Distributed Collaboration Agreements
HBase Real-time, distributed and high dimensional database
Flume Log gathering tool
MapReduce Distributed computing framework
HDFS Distributed file system

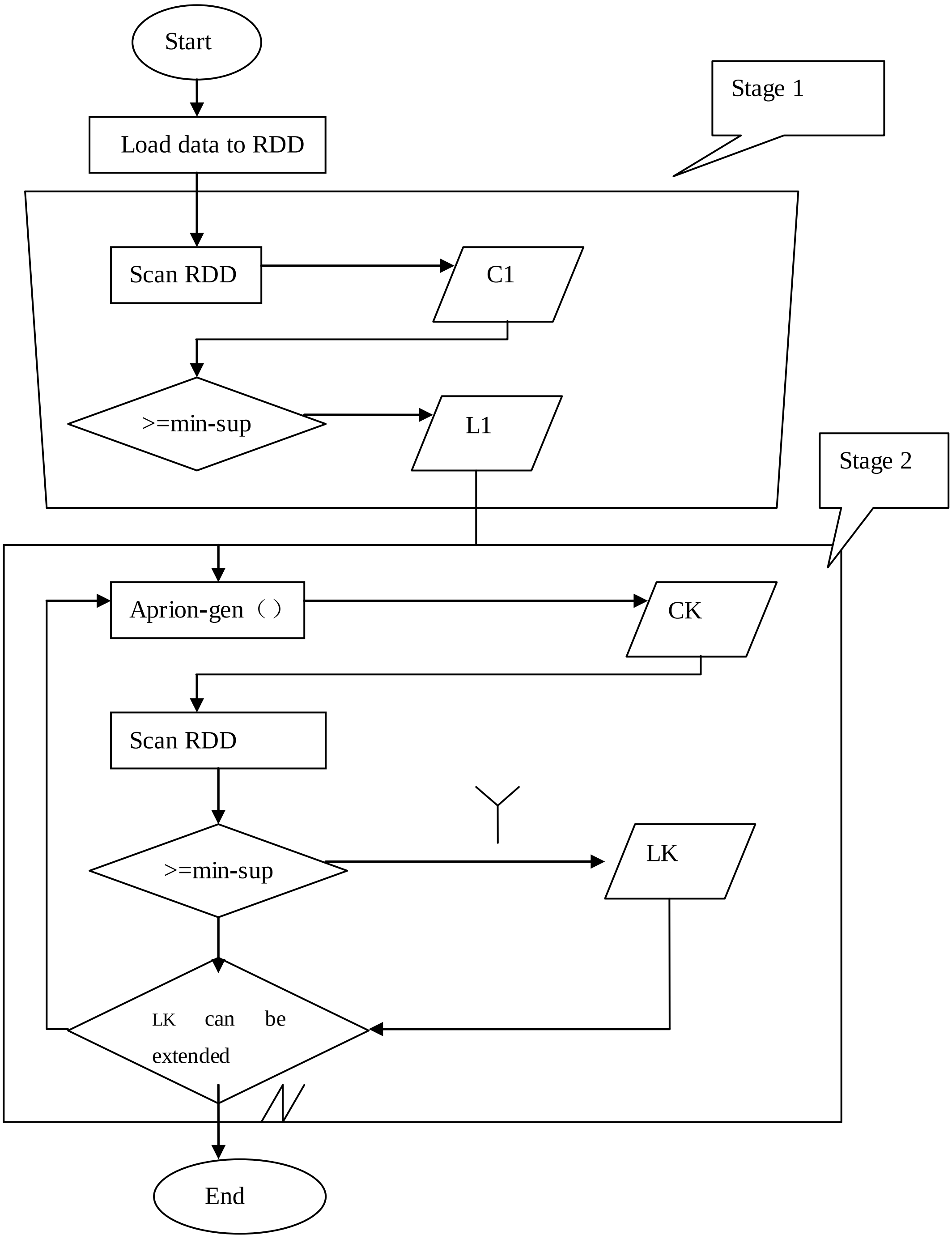
Figure 1. Hadoop ecosystem

Even though Hadoop has gradually established its cornerstone position in big data ecosystem, there are still various substitutes such as Hydra and Spark. Among which, the open source framework Spark based on Hadoop's distributed file system shines out because Spark makes up for the demerits of Hadoop such as improve the speed with better programming interface. Besides, real-time computing and in-memory computing has always been the hot topic in bid data field.

III. THE REALIZATION OF PARALLEL DATA MINING ALGORITHM BASED ON SPARK

The core abstract conception of Spark is Resilient distributed datasets, which integrates all node memories and makes parallel workflow possible. RDD can be established based on loading the folder of HDFS or converting the current RDD. Users can cache the RDD in memory, namely a certain RDD won't be recycled by the memory. As a result, while using this RDD, we don't have to re-establish it so the process will be quicker. Besides, RDD can be recorded based on the descent means so as to automatically recover the RDD contents if a certain node breaks down.

Another abstract conception of Spark is shared variable. Shared variables can be adopted in parallel computing and every task implemented in different nodes can be granted with a copy of shared variable. Spark supports two types shared variables-broadcast variable and accumulator variable such as count And sum. The Apriori algorithm based on Spark can be implemented with two stages. Stage 1: load the transaction data set to the flexible data set of Spark to generate frequent 1 term; Stage 2: iterate frequent K term to k+1 term. The execution flow diagram is shown as follows:



IV. Realization of PageRank algorithm parallelization

PageRank algorithm is the patent of Google which is adopt to know the web search ranking and is one of the ten data mining algorithm. The core idea of PageRank is to take the number of links as the support for a certain page. So if a page is linked for various pages, it wins popular support and ranks top among the search result. The support rating formula of PageRank is shown as follows:

$$R(i) = \sum_{j \in B(i)} R(j)$$

Among which, R(i) refers to the PageRank of I while B(i) refers to the Webbei of i. That is to say, the PageRank of i is the sum of links. However, in reality, if we directly use this formula we cannot identify the page quality. For example, if there are two webs A and B, A is linked with various unknown pages while B is linked with few famous pages we shall conclude that the PageRank of B is better than A with the above formula and, which is not in accordance with the expectation of people and should be improved.

A common approach for improvement is to set weight for each link. For example, if J is linked to I and N links are linked to J, the importance for each link is

R(j)/N. In order to standardize the result, we introduce a constant C and get the following formula:

$$R(i) = C \sum_{j \in B(i)} \frac{R(j)}{N(j)}$$

From the above formula we can find out that if the PageRank of A is better than B, A has to be linked with various important webs or a large amount of nameless, both situations are in accordance with people's expectation.

The concrete calculation process of PageRank algorithm: in the course of calculation, we take one page as a node of directed side while the link of a certain page is the directed edge. Therefore, a large sum of web pages as well as links can be demonstrated based on directed graph and the save graph can be demonstrated in adjacent matrix. If I is linked to J, then 1 will be shown in matrix or 0. That is to day, m[i][j]=l or m[i][j]=0. The adjacent matrix is shown as follows:

$$M\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

By observing matrix M we can find out that the nodes with value 1 in I line are linked pages of I while the nodes with value 1 in J line are linked pages of J. By observing the matrix and taking the calculation formula of PageRank into consideration we can conclude that if every element of matrix M is divided by the sum of the elements in this line we can have matrix N. Then, we change N to N^T and the sum of all elements of all lines is PageRank. In this example, the N^T formula is shown as follows:

$$N^T = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Here we can take N^T as the transition probability matrix of a user searching for different pages.

V. CONCLUSION

With the rapid development of Internet, people gradually enhanced their capacity to collect data, and the historical data is becoming larger and larger. What's more, big data is a hot topic of computer science while big data mining is the core to explore the value of big data. Although data mining algorithms have been fully developed and used, there are not enough convenient and effective tools for big data mining and as a result it is hard to obtain the value of data. The big data mining system designed in this paper takes advantage of open source technologies and make cross-platform, efficient, and convenient data mining tools come true.

REFERENCES

[1] Clifford L. Big data: How do your data grow?[J]. Nature, 2008, 455(7209):28-29.

[2] Cohen J, Dolan B, Dunlap M, et al. MAD skills: new analysis practices for big data[J]. Proceedings of the Vldb Endowment, 2009, 2(2):1481-1492.

[3] Jacobs A. The Pathologies of Big Data.[J]. Queue, 2009, 7(8):36-44.

[4] Wu X, Zhu X, Wu G Q, et al. Data Mining with Big Data[J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(1):97-107.

[5] Lohr S. The Age of Big Data[J]. SR1, online: http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html, 2012, 16(4):10-15.

[6] Times T N Y. The Age of Big Data[J]. SR1, online: http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html, 2012.

[7] Vivien M. Biology: The big challenges of big data[J]. Nature, 2013, 498(7453):255-260.

[8] Labrinidis A, Jagadish H V. Challenges and opportunities with big data[J]. Proceedings of the Vldb Endowment, 2012, 5(12):2032-2033.

[9] Chen Y, Alspaugh S, Katz R. Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads[J]. Proceedings of the Vldb Endowment, 2012, 5:1802-1813.

[10] Marx V. The big challenges of big data[J]. Nature, 2013.