Survey on Implementation of Market Basket Analysis using Hadoop Framework

Rupali S. Vairagade
Department of Computer Engineering, SITS
Vadgaon-Budruk, Pune-411041

Tejas Shah
Department of Computer Engineering, SITS
D803, Gloria Bavdhan, Pune-411021

Tejas Chavan
Department of Computer Engineering, SITS
Dhayari, Pune-411041

Rohan Bhatt
Department of Computer Engineering, SITS
Flat No.12, Jaibha Landmark,
Narhe, Pune-411041

ABSTRACT

Market Basket Analysis is a technique to identify items likely to be purchased together. A predictive market basket analysis is used to identify sets of products/services purchased or events that occur generally in sequence. The basic approach is to find the associated pairs of items in a store when there are transaction data sets. Hence, our proposed system performing ‘Market Basket Analysis’ will help the retailers to make better decisions throughout the entire company which will help in increasing the profits and effectiveness of the organization. Also, by controlling the order of products and marketing visits or the transactions of the customers could be increased. The system will take the large transactional data sets from the retailers and find the associations between different items from the item sets. These associations of the items purchased frequently and the items that are purchased together will be presented in graphical formats such as tables, pie-charts, bar graphs etc. There are different functionalities or patterns providing for performing analysis such as weekend -weekday sales analysis, month-end sales analysis analysis on different customer profiles etc. The system will be built in ‘Apache SPARK’ framework using Scala and processed on Amazon AWS and the data will be stored at its HDFS on the cluster.

General Terms

- Information systems →Parallel and Distributed DBMSs
- Information systems → Information Retrieval
- Computing methodologies →Parallel algorithms

Keywords

Hadoop Distributed File System, Customer relationship management, Big Data, Interactive Data Mining.

1. INTRODUCTION

Market Basket Algorithm is to analyze associates transaction items in order to utilize store stocks, product displays, and item discounts at store. As the data, over last decade is increasing exponentially, the processing speed to equate to that much amount of data has become a daunting task. To achieve faster processing we need to connect multiple nodes or processors without causing bottleneck. The retailers and malls have large number of customer base due to the variety in the produce that they distribute or sell every day. However, there is a requirement of a system that will help them understand the relevance between the customer activities and purchase patterns that occur frequently and that occur together. Hence, there needs to be a system that will help the retailers to better understand the customers so that is satisfaction increases which will lead to increase in the popularity and profitability in the business. If the customer requirements are well understood, the retailers or the organizations can make better decisions to increase their sells thus having healthy increase in their profits.

Hadoop is the most popular parallel programming platform which is built on Hadoop Distributed File Systems (HDFS) for Map/Reduce computation that processes data in the <key, value> pairs. It has been receiving highlights for the enterprise computing because business enterprises always have the big data sets such as log files for web transactions.

Recently, Spark is gaining popularity for large scale data analysis as it is much faster using In-Memory processing than the traditional MapReduce while it maintains data locality on HDFS. Apache Spark is a platform that provides a user friendly programming interface with aim reduce coding efforts and provide better performance with problems related to big data. Spark code can be built in Scala, Java, Python which are popular programming languages.

2. RELATED WORK

In this section we discuss a few previous implementations of Market Basket Analysis Algorithm using other possible technology like Hadoop MapReduce , HBase .We discuss about the advantages of Apache Spark over other platforms and their limitations.

2.1 Map/Reduce in Hadoop

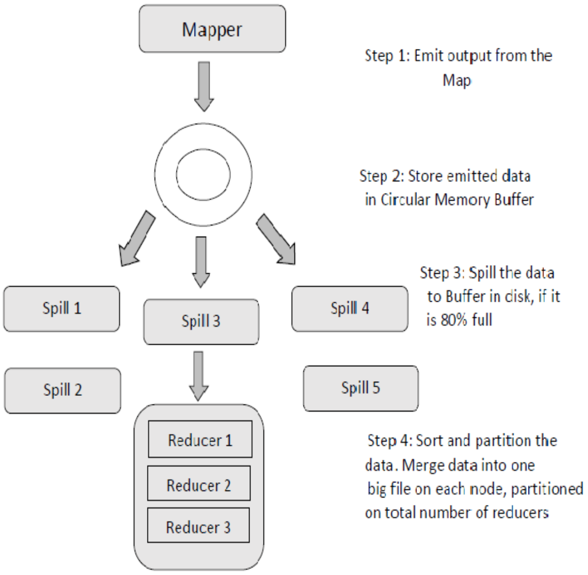
Mapper
Step 1: Emit output from the Map
Step 2: Store emitted data in Circular Memory Buffer
Spill 1
Spill 3
Spill 4
Step 3: Spill the data to Buffer in disk, if it is 80% full
Spill 2
Spill 5
Reducer 1
Reducer 2
Reducer 3
Step 4: Sort and partition the data. Merge data into one big file on each node, partitioned on total number of reducers

Figure 1. Mapper in MapReduce

Hadoop Map/Reduce encourages the need to introduce new parallel algorithms instead of sequential algorithms for computations for the existing applications.
Hadoop is the parallel programming platform built on Hadoop Distributed File Systems for MapReduce computation that processes data as <key, value> pairs.

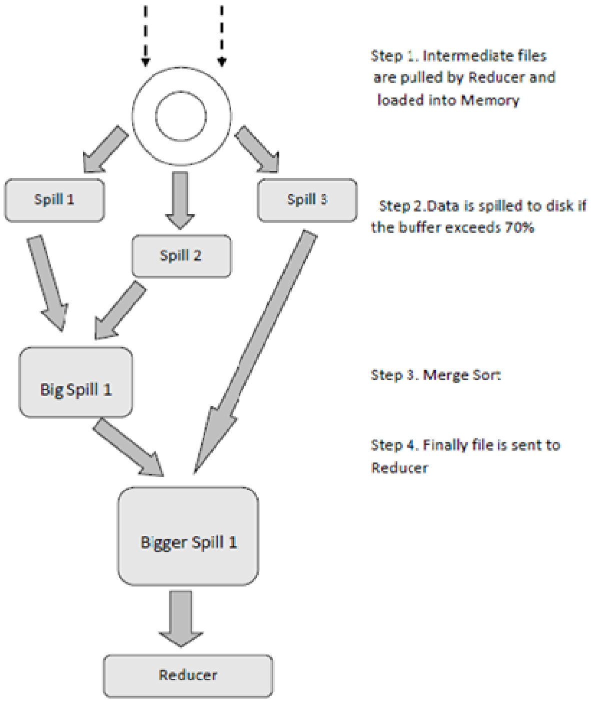
Step 1. Intermediate files are pulled by Reducer and loaded into Memory
Spill 1
Spill 3
Spill 2
Step 2.Data is spilled to disk if the buffer exceeds 70%
Big Spill 1
Step 3. Merge Sort
Step 4. Finally file is sent to Reducer
Bigger Spill 1
Reducer

Figure 2. Reducer in MapReduce

2.2 Map/Reduce in Parallel Computing

The map and reduce functions of Map/Reduce run on distributed nodes in parallel. Each map operation can be processed independently on each node and all the operations can be performed in parallel. But practically, it is limited by the number of CPUs for storing the data. The experiments show that the execution times of the proposed algorithm gets much better performance while running on larger number of nodes. However, at a certain point, though we add more nodes, Hadoop Map/Reduce does not guarantee to increase the performance because there is a limitation to enhance the parallelism, which is negatively affected by the time spent for distribution and aggregation of data set and then reducing it among nodes against computing powers of additional nodes.

The following is the diagrammatic presentation of Mapper and Reducer in Apache SPARK:

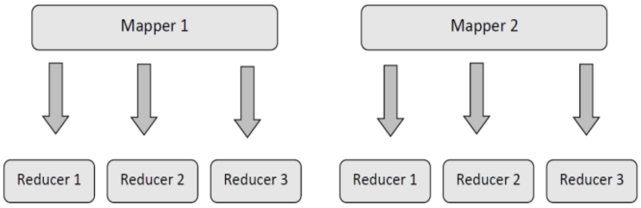

Figure 3. Mapper in SPARK

Step 1: Emit the Output from Map

Step 2: Create Reducer shuffle files per Mapper. Data is usually stored in O S Buffer Cache and some written to disk if buffer fills. Hence Writes/Reads are at memory speed

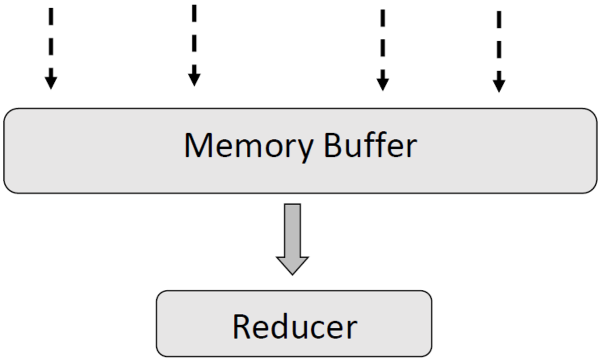

Figure 4. Reducer in SPARK

Step 1: Data is pushed by Mapper to Reducer. Data is spilled to disk, if it does not fit in memory.

Step 2: Finally, file is sent to Reducer

2.3 Issues of Map/Reduce

- Need tens-, hundreds-, or thousands-of-nodes to compose Hadoop Map/Reduce platform.
- If using services of cloud computing, for example, AWS EC2, the overheads mainly come from I/O. That is, it takes long to upload big data to AWS EC2 platform or AWS S3, which is more than computing time.
- There is a compulsion to convert data to the format of (key, value) pair for Map/Reduce, which misses most of the features that are routinely included in current DBMS.
- Incompatible with all of the tools or algorithms that have been built.

2.4 Survey Table

The following is the table of study on various implementations of Market Basket Analysis.

Table 1. Survey Table

Sr. No.	Title	Pros/Cons
1.	Market Basket Analysis on Map/Reduce in AWS EC2	Map/Reduce is restricted parallel computing paradigm. From a certain number of nodes, Map/Reduce does not guarantee to increase the performance.
2.	Spark: Cluster Computing with Working Sets	Spark introduces an abstraction called Resilient Distributed Datasets(RDDs). Spark outperforms Hadoop by 10x in iterative machine learning jobs, and can be used to interactively query a 39 GB dataset with sub-second response time.
3.	Apriori-Map/Reduce Algorithm	The Apriori-Map/Reduce algorithm gains much higher performance than the sequential apriori algorithm as the map and reduce nodes get added. The item sets produced by the algorithm can be adopted to compute and produce Association Rule for market analysis.
4.	Market Basket Analysis using Spark	Spark's in –memory processing achieves much better high performance than the legacy Map/Reduce processing.

3. EXPERIMENTAL RESULT

The comparison between Apache Spark and MapReduce on working on the K-Means algorithm on the described data set shows the following achieved results for comparison (shown in the tables below).

To gain a varied analysis, the experiment considered 64MB, 1240 MB with a single node and 1240MB with two nodes and monitored the performance in terms of the time taken for clustering as per our requirements using K-Means algorithm. The machines used had a configuration as follows:

- 4GB RAM
- Linux Ubuntu
- 500 GB Hard Drive

The results clearly showed that the performance of Spark turn out to be considerably higher in terms of time, where each of the dataset size results in a decrease in the processing time of up to three times as compared to that of Map Reduce. Although there exists a minor fluctuation in this result, this is due to the random nature of the K-Means algorithm and does not affect the analysis to a large extent.

Table 2. Results for K-Means using Spark (MLib)

Dataset Size	Nodes	Time(s)
62MB	1	18
1240MB	1	149
1240MB	2	85

Table 3. Results for K-Means using Map Reduce (Mahout)

Dataset Size	Nodes	Time(s)
62MB	1	44
1240MB	1	291
1240MB	2	163

4. ALGORITHM

1. Take an input transaction text T such that, T={t1,t2,t3….tn}
2. Read every transaction from the input file and generate the data set D such that, D={d1,d2,d3…..dn}
3. Generate ngram list for each line
    a. Data is alphabetically sorted.
    b. Remove duplicate item pair from transaction
4. Generate set of elements E such that, E={e1,e2………en} where

   e1= (item, frequency) pair
5. All pairs are reducing by each key and values are summed.
6. Resultant key value pair generated (item, 256)
7. The Top Item sets are displayed using visual effects like bar charts etc.

5. CONCLUSION

Spark leads Big Data computing community as it is In-Memory processing that achieves much better high performance than the legacy MapReduce processing. The paper presents a Market Basket Analysis algorithm on functional programming. And, the Market Basket Analysis code is written in Scala Spark and processed on AWS EC2. Our results show that Spark is a very strong contender and would definitely bring about a change by using in-memory processing. The experimental results also show that the more nodes the better performance is achieved. Besides, more the nodes or processors, faster is the performance especially in time-bound operations.

6. REFERENCES

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

[2] Apache Spark, http://spark.apache.org/

[3] Jongwook Woo .*Market Basket Analysis Algorithms with MapReduce*, DMKD-00150,Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, Oct 28 2013, Volume 3, Issue 6, pp445-452,ISSN 1942-4795

[4] Jongwook Woo, Science & Engineering Research Support Society (SERSC), Sept 2012 .*Market Basket Analysis Algorithm on Map/Reduce in AWS EC2", in International Journal of Advanced Science and Technology (IJAST)*, Volume 46, No 3, pp25-38, ISSN 2005-4238

[5] Jongwook Woo, Siddharth Basopia, Yuhang Xu, Seon Ho Kim, The Third International Conference on Emerging Databases (EDB 2011). *Market Basket Analysis Algorithm with NoSQL DB HBase and Hadoop*, Songdo Park Hotel, Incheon, Korea, Aug. 25-27,2011

[6] Jongwook Woo, *Apriori-Map/Reduce Algorithm* ,The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2012), Las Vegas (July 16-19, 2012)

[7] Bradford Stephens .*Building a business on an open source distributed computing*, Oreilly Open Source Convention (OSCON) 2009, July 20-24, 2009, San Jose, CA

[8] Woohyun Kim. *MapReduce Debates and Schema-Free* .Coord, March 3 2010

[9] Jimmy Lin and Chris Dyer, *Data-Intensive Text Processing with MapReduce* ,Tutorial at the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), June 2010, Los Angeles, California

[10] Jongwook Woo, *Introduction to Cloud Computing* ,the 10th KOCSEA 2009 Symposium, UNLV, Dec 18-19, 2009

[11] Jongwook Woo**,** *The Technical Demand of Cloud Computing* ,Korean Technical Report of KISTI (Korea Institute of Science and Technical Information), Feb 2011

[12] Apache HBase, "http://hbase.apache.org/"

[13] Jimmy Lin and Chris Dyer, Morgan & Claypool Publishers,2010.

[14] GNU Coord, http://www.coordguru.com/

[15] Jongwook Woo, Dong-Yon Kim, Wonhong Cho, MinSeok Jang, *Integrated Information Systems Architecture in e-Business* The 2007 international Conference on e-Learning, e-Business, Enterprise Information Systems, e- Government, and Outsourcing, Las Vegas (June 26-29,2007)