Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance Improvement of Apriori Algorithm Using Hadoop

Sonali Satija[1], Dr. Rajender Nath[2]
[1] M .Tech Scholar, [2] Professor,
Dept. of Computer Science and Application
Kurukshetra University, Kurukshetra,
Haryana, India

Abstract: - Data mining is a process of extracting meaningful pattern from the large pool of data. Association is one of the techniques for Data Mining. Apriori algorithm is useful in finding out the frequent itemsets from the transactional dataset, however it needs to scan dataset many times to generate many candidate itemsets. So, in order to reduce the response time and increase the efficiency, Parallel and Distributed computing are effective strategies. In this paper, implementation of an efficient Map-Reduce Apriori algorithm based on Hadoop Map-Reduce model is shown. Experimental results showed that Map-Reduce Apriori algorithm outperforms the traditional Apriori algorithm.

Keywords: - Data Mining, Association Rules, Apriori algorithm, Hadoop, Map-Reduce. Abbreviations: - Data Mining (DM), Association Rule (AR), Map-Reduce (MR).

I. INTRODUCTION

Data Mining is a process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. It is a non-trivial extraction of implicit, previously unknown and potentially useful information from data. It is a search for relationships and global patterns that exists in the large databases but that are hidden among vast amount of data. DM has various techniques and one such is Association. AR's are if/then statements that help to discover relationships among unrelated data in a data repository.

Apriori Algorithm is an algorithm for frequent itemsets mining and association rule learning over transactional database. This algorithm helps in finding out the frequent itemsets and thus deriving the AR between the itemsets. In order to find the frequent itemsets there is a need to scan the database again and again. The main limitation of Apriori algorithm is costly wasting of time to hold a vast number of candidate sets. In addition, single processor's memory and CPU resources are very limited, which make the algorithm performance inefficient. Furthermore, because of growth of information, enterprises have to deal with growing amount of data. So, the solution to this problem is parallel and distributed computing. This can be achieved by Hadoop Map-Reduce model.

II. RELATED WORK

The Classical Apriori Algorithm was proposed which help in finding out the frequent itemsets in the database (Chai, Yang, cheng, 2007) [1].The AR mining is a two-step process: - first finding out the frequent itemsets in the transaction and then generating the strong AR from frequent itemsets. The principle of improvement thus proposed an algorithm that mined frequent itemsets without new candidate generation.

In order to improve the performance of the association rules mining different ways have been proposed (lei, Zhang, Jianhua, 2006) [2]. The High-Dimension Oriented Apriori Algorithm was proposed which cut down the redundant generation of identical sub-itemsets from candidate itemsets by means of pruning. The improvement was mainly in pruning function.

Mining of AR is one of the important and powerful aspect of Data Mining (Prasad, Mourya, 2013) [3].The criteria proposed for ARM is to find relationship among various items in the database. The various approaches of Data Mining like Apriori have many drawbacks that it scans the database lot of times and also the time complexity is high. This can be solved up by grouping the items into higher conceptual groups and also reducing the number of scans of the entire database.

Hadoop is an open-source implementation of map-Reduce and the most popular Map-Reduce variant currently in use in an increasing number of prominent companies with large user bases (Doulkeridis et.al, 2013) [4]. Hadoop consists of two main parts-HDFS and Map-Reduce. According to Doulkeridis, Hadoop consisted of various servers-Namenode, Datanode, Secondary-namenode for managing HDFS and Job Tracker and Task Tracker for performing Map-Reduce.

Jeffrey and Ghemawat (2004) [5] described Map-Reduce as a programming model and an associated implementation for processing and generating large data sets. Users specified a map function that processed a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. According to Jeffrey et.al (2004), restricting the programming model made it easy to parallelize and distribute computations and to make such computations fault-tolerant

Yang et.al (2013) [6] also proposed an Apriori algorithm on the basis of map-reduce using Hadoop. According to Yang, datasets in files got split into smaller segments automatically after stored in HDFS and the map function is executed on each of these data segments.

Jongwook(2013) [7] proposed Apriori-Map/Reduce Algorithm and illustrated its time complexity, which theoretically showed that the algorithm gains much higher performance than the sequential algorithm as the map and reduce nodes get added.

Yahya et.al (2012) [8] proposed an efficient algorithm called MR Apriori algorithm. According to Yahya, it only needed two MR phases to find all the frequent k-itemsets.

III. BACKGROUND

A. Association Rules(AR)

An AR expressed as A→B, where A and B are itemsets, states that if the customer purchases A itemset then he is likely to purchase B itemset [3]. An AR represents if-then statement. AR uses two criteria- support and confidence to identify the relationships and rules, which are generated by analyzing data for frequent if-then pattern. Support is the indication of items how frequently it appears in the database. The AR described as A□ B, where A is antecedent itemset, B is consequent itemset and P is the percentage, support is the percentage of transaction in the database that contains AUB.

Support=P (AU B)

Confidence indicates the number of times the statement is found true. The AR described as A→B, where A is antecedent itemset, B is consequent itemset and P is the percentage, confidence is the percentage of transaction in the database containing A that also contains B.

Confidence=P (B/A)

AR's are generally needed to satisfy a user specified minimum support and a user specified minimum confidence at the same time.

B. Apriori Algorithm

Apriori Algorithm is one of the DM algorithm which is used to find the frequent itemsets from a given data repository. The key idea of Apriori algorithm is to make multiple passes over the database. The working of Apriori algorithm is fairly depends upon the Apriori property which states that" All nonempty subsets of a frequent itemsets must be frequent". It also described the anti-monotonic property which says if the system cannot pass the minimum support test; all its supersets will fail to pass the test. Therefore if the one set is infrequent then all its supersets are also infrequent and vice versa. This property is used to prune the infrequent candidate elements.

C. Hadoop

Hadoop is an open source framework used for implementation of MR. Hadoop consists of two main parts-Hadoop Distributed File system and MR for distributed processing [4]. HDFS is designed for the storage of large files. HDFS is not the primary storage rather data is copied over HDFS for the purpose of performing MR. HDFS is optimized for streaming access of large files, random access to parts of files is significantly more expensive. The typical scenario of applications using HDFS follows write-once read-many access model.

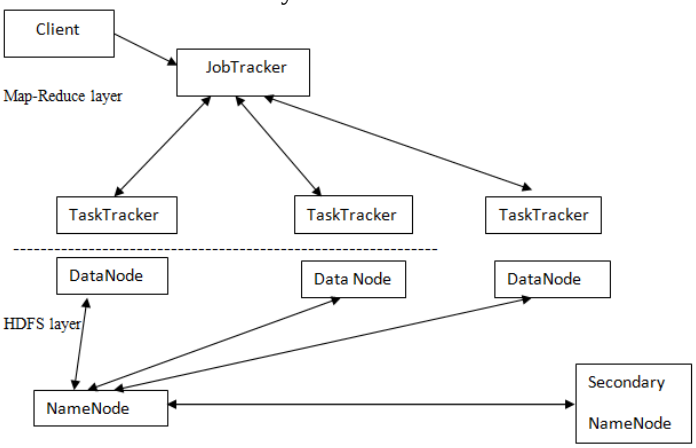

Fig3.1: Hadoop Architecture

D. Map-Reduce(MR) Model

MR is the best model in parallel and distributed computing areas [5]. A MR application consists of two functions-Map and Reduce functions. Map function takes a key/value pair and generate a set of intermediate key/value pairs and the reduce function merges together these value to a possibly smallest set of values [6]. The intermediate values are supplied to the user's reduce function via iterator. This allows handling a list of values that are too large to fit in memory.

IV. PERFORMANCE IMPROVEMENT OF APRIORI ALGORITHM USING HADOOP

Though, Apriori algorithm is helpful in finding out the frequent itemsets from the large pool of data yet it has some limitations of multiple scans of transaction database, huge number of candidates and tedious workload of support

counting for candidates. The main limitation is wastage of time in holding a large number of candidate sets. So improvements are needed in order to reduce the time complexity. In addition, single processor's memory and CPU resources are very limited, which makes the algorithm performance inefficient [8]. Furthermore, because of growth of information, enterprises have to deal with growing amount of data.

To address these problems, a parallel and distributed solution is needed, which is achieved by using Hadoop MapReduce model. An Apriori algorithm is implemented using the Hadoop which includes the mapper and reducer which help in overcoming the limitations of Apriori algorithm and also helps in reduction of response time.

Fig 4.1 shows the data flow of Apriori Algorithm on Hadoop. Hadoop map-reduce basically works on mapper and reducer. Mapper count the items in the candidate set. Reducer function prunes the candidate and summary the same items to get frequent itemsets and finally the output is of frequent k-length itemsets. This parallel and distributed computation of Apriori algorithm helps in the reduction of response time for finding out the frequent itemsets from the large datasets

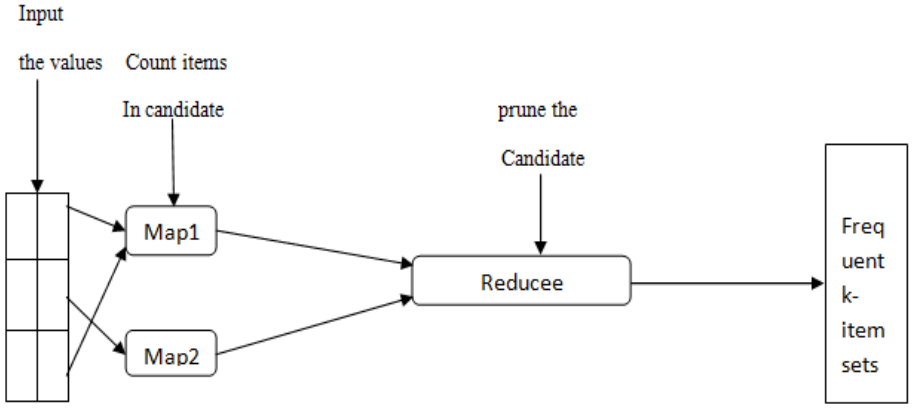

Fig: 4.1 Data Flow for Parallel Apriori

V. EXPERIMENT AND RESULTS

The two experimental setups are being used to compare the implementation of Apriori algorithm. Firstly, the Apriori algorithm is implemented using java platform and secondly using Hadoop single node cluster. When Apriori algorithm runs on Java, it gives result based on centralized technique. On the other hand, Hadoop works on mapper and reducer which help in parallel and distributed computing.

When the algorithm is run on dataset (kosarak.dat), which contains click-stream data of a Hungarian online news portal, using Java it takes 861ms.The same algorithm when run using Hadoop single node cluster it takes 360ms. It shows the speedup of the Apriori algorithm. Thus, it can be concluded that Hadoop has reduced the response time of Apriori algorithm.

Here described below in Fig: 5.1 is a graph which shows the comparison of response time of Apriori algorithm when implemented on java with the response time of Apriori algorithm when implemented on the Hadoop single node cluster.

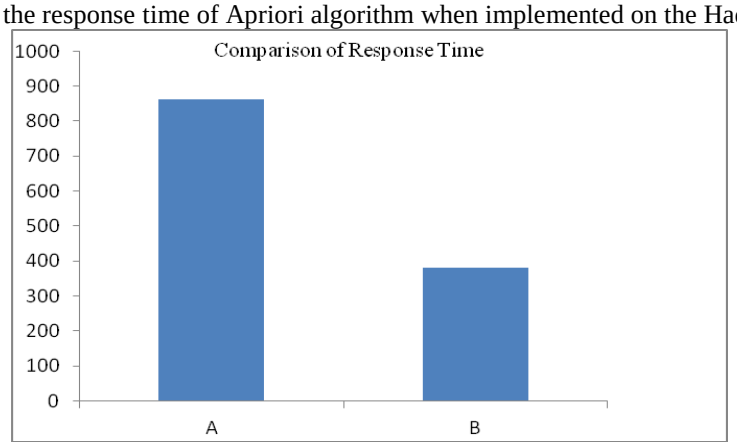

Fig: 5.1 Comparison of response time of Apriori algorithm on java with Hadoop

In the above graph A shows the response time of Apriori algorithm when implemented on java platform and B shows the response time when implemented on Hadoop MapReduce model. The comparison shows that the Hadoop platform decreases the response time of Apriori algorithm and speedup the algorithm.

VI. CONCLUSION

Performance of Apriori algorithm has been improved by using Hadoop technology. The proposed work has been tested on click-stream data set of a Hungarian online news portal by implementing it on Java platform and on Hadoop technology on single node cluster. Experimentally it has been found that Hadoop MapReduce model has improved the performance of Apriori algorithm significantly.

REFERENCES

[1] Sheng Chai, Jia Yang and Yang Cheng, "The Research of Improved Apriori Algorithm for Mining Association Rules", IEEE 2007.
[2] Lei Ji, Baowen Zhang and Jianhua Li, "A New Improvement on Apriori Algorithm", IEEE 2006.
[3] Phani Prasad J and Murlidher Mourya, " A Study of Market Basket Analysis Using a Data Mining Algorithm", In Proceedings of International Journal of Emerging Technology and Advanced Engineering, Volume 3, June 2013.
[4] Christos Doulkeridis and Kjetil Norvag, "A Survey on large-scale analytical query processing in MapReduce", In Proceedings of The VLDB Journal Springer (2013).
[5] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified data Pro*cessing on Large Clusters", In Proceedings of OSDI, 2004.
[6] Xin Yue Yang, Zhen Liu and Yan Fu, "MapReduce as a Programming Model for Association Rules Algorithm on Hadoop", In Proceedings of National Natural Science Foundation of China (2013).
[7] Jongwook Woo, "Apriori-Map/Reduce Algorithm",(2013).
[8] Othman Yahya, Osman Hegazy and Ehab Ezat, "An Efficient Implementation of Apriori algorithm based on Hadoop-MapReduce model", International sjournal of Reviews in Computing, 2012.
[9] ApacheHadoop.http://hadoop.apache.org/
[10] http://www.michaelnoll.com/tutorials/running-hadoop-onubuntulinux-single-node-cluster

