IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. VIII (Mar-Apr. 2014), PP 112-118
www.iosrjournals.org

Association Rule Mining using Apriori Algorithm for Distributed System: a Survey

Mr. Uday K. Kakkad[1], Prof. Rajanikanth Aluvalu[2]
[1](CE/IT Department, School of Engineering/ R.K. University, India)
[2](CE/IT Department, School of Engineering/ R.K. University, India)

Abstract : *Data mining technologies provided through Cloud computing is an absolutely necessary characteristic for today's businesses to make proactive, knowledge driven decisions, as it helps them have future trends and behaviors predicted. By implementation of data mining techniques in Cloud will allow users to retrieve meaningful information from virtually integrated data repository that reduces the costs of resources. Research in data mining continues growing in business and in learning organization over coming decades. Association rule mining is a most important area in data mining domain. In association rule mining Apriori algorithm is a very basic and important algorithm as a research point of view. It has some disadvantages it becomes expensive because of frequently scanning of database and it did not support a large amount of raw data and also we have limited resources to implement scalable algorithm. For implementation of scalable Apriori algorithm Map Reduce programming model will be used. Map reduce is a programming model which used to implement and process a scalable raw data. Hadoop provides an open source platform to the map reduce for implementation. But Hadoop have some limited and default task scheduler. So In this paper we have made a survey to implement Apriori algorithm for huge raw data and also overcome Hadoop limitation by using enhanced scheduling algorithm.*
Keywords: *Association rule mining, Hadoop, Map Reduce.*

I. Introduction

EMC news [15] declare in December 2012 that

- From 2005 to 2020, the digital universe will grow by a factor of 300, from 130 exabytes to 40,000 exabytes, or 40 trillion gigabytes (more than 5,200 gigabytes for every man, woman, and child in 2020).
- From now until 2020, the digital universes will double every two years.
- 68% of the data created in 2012 was created and consumed by consumers-- watching digital TV, interacting with social media, sending camera phone images and videos between devices and around the Internet, and so on.

Cloud computing [17] provides a powerful, scalable and flexible infrastructure into which one can integrate, previously known, techniques and methods of Data Mining. The result of such integration should be strong and capacitive platform that will be able to deal with the increasing production of data, or that will create the conditions for the efficient mining of massive amounts of data from various data warehouses with the aim of creating (useful) information or the production of new knowledge.

In this paper Section 2, 3 describes data mining and Apriori algorithm for association rule mining, Section 4 and 5 describes Cloud computing and its file system and Distributed System, In Section 6 we gives some issues by doing literature survey and then existing system describes in Section 6.

II. Data Mining

Data mining is a process to extract hidden or unknown information from the raw data [1]. Data mining produces useful patterns or models from data. It is an essential step in the knowledge discovery in databases (KDD) process [2]. The terms of KDD [16] and data mining are different. KDD refers to the overall process of discovering useful knowledge from data. Data mining refers to discover new patterns from a wealth of data in databases by focusing on the algorithms to extract useful knowledge

2.1 Association Rule Mining

Association Rule mining, one of the most important and well researched techniques of data mining was first introduced in [3]. ARM extract interesting patterns, correlations, associations structures among sets of items in the data repositories.

- Finding Frequent item set from the raw data
- Generate association rules for the frequent item set [1]

2.1.1 Apriori Algorithm

The frequent item sets determined by Apriori [18] can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis [24]. Mining association rule consists of the following two steps:

2.1.1.1 Finding the item sets that are frequent in the dataset:

The frequent item sets are set of those items whose support (sup(item))in the data set is greater than the minimum required support(min_sup). Considering the above example all of the three items A, B and C belongs to the frequent item set and sup(A, B) and sup(C) would be greater than min_sup. Proportion of transactions that contains the item set is defined as a support of an item set.

2.1.1.2 Generating interesting rules from the frequent item sets on the basis of confidence (conf).

The confidence of the above rule will be sup (A, B) divided by sup(C). If the confidence of the rule is greater than the required confidence, then the rule can be considered as an interesting one. The first step is very important in association rule mining because it affect to the whole performance of ARM. As is evident, the algorithm does not require the details to be specified, such as like the number of dimensions for the tables or the number of categories for each dimension, as each item of transaction is considered.[4]

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation

Input:

- D, a database of transactions:
- min_sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

1. L_1 = find_frequent_1-itemsets(D);
2. for(k=2;$L_{k-1} \neq \emptyset$;k++){
3. C_k=apriori_gen(L_{k-1});
4. for each transaction t ϵ D { // scan D for counts
5. C_t = subset(C_k,t); // get the subsets of t that are candidates
6. for each candidate c ϵ C_t
7. c.count++;
8. }
9. L_k= {c ϵ C_k | c.count ≥ min_sup }
10. }
11. return L= $U_k L_k$;

procedure apriori_gen(L_{k-1};frequent (k-1)-itemsets)

1. for each itemset l_1 ϵ L_{k-1}
2. for each itemset l_2 ϵ L_{k-1}
3. if(l_1[1] – l_2[1])^ (l_1[2] – l_2[2])^… ^(l_1[k-2] – l_2[k-2])^(l_1[k-1] – l_2[k-1])then {
4. C =l_1 JOIN l_2;
5. if has_infrequent_subset(c,L_{k-1}) then
6. delete c; // prune step:remove unfruitful candidate
7. else
8. add c to C_k;
9. }
10. return C_k;

procedure has_infrequent_subset(c:candidate k-itemset; L_{k-1}: frequent (k-1) = itemsets); // use prior knowledge

1. for each (k-1) = subset s of c
2. if s ϵ L_{k-1} then
3. return TRUE;
4. return FALSE;

Fig. 1. Apriori Algorithm (From [1])

Many times, an association rule mining algorithm generates extremely large number of association rules [19]. In most cases, it is impossible for users to comprehend and validate a large number of complex association rules. Therefore, it is important to generate only "interesting" rules, "no redundant" rules, or rules satisfying

certain criteria, such as coverage, leverage, lift, or strength. In Association rule mining apriori algorithm is very useful algorithms to generate association rules. Apriori algorithm was designed to run on databases containing transactions. It is a "bottom up" approach as candidate items are first generated and then the database is scanned to count the support for candidate items exceeding minimum support [23]. The number of items in candidate subsets is increased one at a time with iteration. These candidate sets are converted to frequent subsets once their support count is matched with the required minimum support. The iteration would stop when no frequent subset can be generated. In the candidate generation phase, an additional pruning step is used, to check that all the subsets of the candidate are frequent [21]. This helps in reducing the size of candidate set before scanning the data base. In this method, the number of times the input file will be read will depend on the number of iteration is required to get the maximum items in the frequent item set. For example if the maximum number of items in the frequent item set is 15, then the whole input file will be read a minimum of 15 times. The Apriori Algorithm is based on the Apriori property:

Apriori is the most classic and most widely used algorithm for mining frequent item sets for Boolean association rules, proposed by R. Agrawal and R. Srikant in 1994 in [4]. The pseudo-code given below is of Apriori algorithm.

Step 1 of Apriori finds the frequent 1-itemsets, L1. In steps 2 to 10, Lk-1 is used to generate candidates Ck in order to find Lk for $k \geq 2$. The apriori_gen procedure generates the candidates and then uses the Apriori property i.e. Prune property to eliminate a subset that is not frequent (step 3). Once all of the candidates have been generated, the database is scanned (step 4). Now a subset function find all the possible subset of the transaction that are candidates for each transaction (step 5), and by using counter count for each of these candidates is collected (steps 6 and 7). Finally, all of those candidates satisfying minimum support (step 9) form the set of frequent itemsets, L (step 11). In this step generate a frequent itemset so procedure of this step is called to generate association rules from the frequent itemsets.

The apriori_gen procedure is used to perform join and prune action on candidate set, as described below. In the join component, L_{k-1} is joined with L_{k-1} to generate potential candidates (steps 1 to 4). The prune component (steps 5 to 7) employs the Apriori property to remove candidates that have a subset that is not frequent. The test for infrequent subsets is shown in procedure has infrequent subset.

Apriori algorithm [4] is a most suitable algorithm for the association rule mining. It have some disadvantages like frequently scanning database to generate candidate set but it is very basic and important algorithm for research experiment. It also not supports huge raw data. But in today's world data are exponentially increasing. So now it is difficult to analyze the huge amount of data and also we don't have that much efficient resource which processes it. After few years it is impossible to process those data which move from Gigabytes data to Terabytes data very quickly. So we required to find efficient techniques which analyze and also manage scalable data.

Concept of parallel processing [22] and distributed computing [23] are helpful to solve above problems so we can combine both the concepts. We can divide the data then generate processing node and then we can process that distributed data separately in each node after processing we combine the results. To implement this we required scalable cloud mechanism which can easily implement this. So we can use Hadoop-Map Reduce model from cloud. [5]

III. Cloud Computing

Cloud computing [25] term is used to describe a variety of computing concepts that involve a large number of computers connected through a real-time communication network such as the Internet. In science, cloud computing is a synonym for distributed computing over a network, and means the ability to run a program or application on many connected computers at the same time.

3.1 GFS

As a consequence of the services Google provides, Google faces the requirement to manage large amounts of data including but not being limited to the crawled web content to be processed by the indexing system. GFS is designed as a distributed file system to be run on nodes up to thousands of machines. [26]. Google File System is a proprietary distributed file system which is developed by the Google so here we use HDFS. [27] To provide efficient, reliable access to data using large clusters of commodity hardware. The codename of the new version of the Google File System is Colossus. [28]

3.2 Hadoop

Apache Hadoop [6] is an framework for storage and large scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users. Hadoop is an open source and it licensed under the Apache License 2.0. [7]

Hadoop provides a Platform on which Map Reduce can be modeled. Map reduce is a programming model in which we can done scalable implementation. Author is going to implement Apriori algorithm on Map Reduce Programming model using Hadoop Platform. [8].

3.3 File system in Hadoop (HDFS)

The Hadoop Distributed File System (HDFS) [12, 13] is a distributed file system designed to run on clusters. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS have features like highly fault-tolerant and also deployed on low-cost hardware.

3.4 Map Reduce

Map Reduce is a programming model controlling a large number of nodes to handle a huge amount of data by combining node. A MapReduce application that needs to be run on the MapReduce system is called a job. The input file of a job, which resides on a distributed file system throughout the cluster, is split into even-sized block replicated for fault tolerance.

A job can be divided into a series of tasks. After splitting input into block they first processed by a map task, which generate a list in the form of key-value pairs. Map outputs are split into buckets based on the key. After Mapping a reduce function is apply on the list of key generated by map function. [5, 9]

In a cluster which runs MapReduce, there is only one Name Node also called master, which records information about the location of data chunks. There are lots of Data Nodes, also called workers, which store data in individual nodes. There is only one Job Tracker and a series of TaskTrackers. Job Tracker is a process which manages jobs. Task Tracker is a process which manages tasks on a node. Before explaining the steps involved in a MapReduce job, let us clarify the terminology that will be used from this point on in this paper.

3.4.1 JobTracker[32]

Master node is controlling the distribution of a Hadoop (Map Reduce) job across free nodes on the cluster. It is responsible for scheduling jobs on Task Tracker nodes. In case of a node failure, the Job Tracker starts the work scheduled on the failed node on another free node. The simplicity of Map Reduce tasks ensures that such restarts can be achieved easily.

3.4.2 NameNode[33]

Master node is controlling the HDFS. It is responsible to serve any component that needs access to files on the HDFS. It is also responsible for ensuring fault tolerance on HDFS. Usually, fault tolerance is achieved by replicating data blocks over three different nodes with one of the nodes being an off-rack node.

3.4.3 TaskTracker (TT)[34]

Node is running the Hadoop tasks. It requests work from the JobTracker and reports back the progress of the work allocated to it. The TaskTracker does not run tasks on its own, but forks a separate daemon for each task. This ensures that if the user code is malicious, it does not bring down the TaskTracker.

3.4.4 DataNode[35]

This node is part of the HDFS and holds the files that are put on the HDFS. Usually, these nodes also work as TaskTrackers. The JobTracker often tries to allocate work to nodes, where file accesses can be done locally.

3.4.5 ProgressScore (PS)

A progress score of a task in the range [0,1], based on how much of a task's key/value pairs have been finished.

3.4.6 ProgressRate (PR)

A progress rate of a task is calculated based on how much a task's key/value pairs have been finished per second.

3.4.7 TimeToEnd (TTE)

TimeToEnd estimates the time left for a task based on the progress rate provided by Hadoop.

3.4.8 Weights of map function stage (M1) and order stage (M2) in map tasks

M1 and M2 in the range [0, 1] record the stage weights in a map task. The sum of M1 and M2 is 1.

3.4.9 Weight of shuffle stage (R1), order stage (R2), and merge stage (R3) in reduce tasks

R1, R2 and R3 in the range [0,1] record the stage weights in a reduce task. The sum of R1, R2 and R3 is 1.

Map Reduce scheduling system has six steps when executing a Map Reduce job, as illustrated in Figure given below:

1. The Map Reduce framework first splits an input data file into G pieces of fixed size which passed on to the participating machines in the cluster. To achieve fault tolerance 3 copies of each piece are maintain.

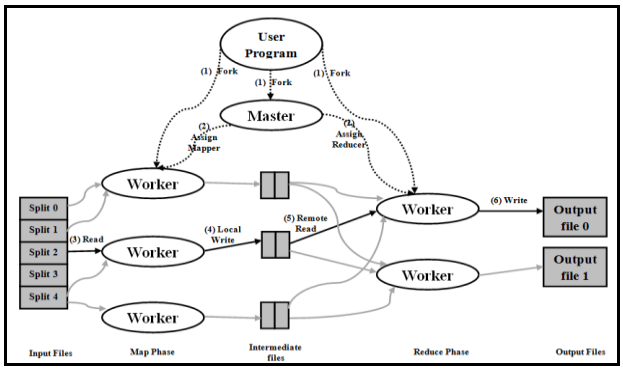

Fig. 2. Map Reduce scheduling system (From [5])

2. Now there are Master Nodes and worker node are assign and then map tasks and reduces tasks assign. This Node configuration decided by the user program. Master node assign tasks to the worker node which called as map tasks. After generating a output of the map task master node assign reduce tasks to the worker node.
3. After assigning tasks to the worker it start reading input files. It produces the intermediate key/value pairs which stored in memory of the corresponding machines that are executing them.
4. All intermediate key/value pairs generated by the worker node stored to a local disk periodically and by using partitioning function. The Location address of these intermediate pairs sends to the master. The master then forwards these locations to the reduce workers.
5. Now reduce worker sorts it by the intermediate key so that all occurrences of the same key are grouped together. Next, the reduce worker generate unique intermediate key by iterating over the sorted intermediate data. This key passes to the corresponding set of intermediate values to the reduce function. The output of the each reduce function is combined and generate final output file for this reduce partition.
6. After completing all map tasks and reduce tasks, the master returns back to the user code.

IV. Distributed System

A distributed system is collection of independent, networked, communicating, and physically separate computational nodes [29]. Distributed computing [30] is domain of computer science in which studies distributed systems. A distributed system is a message passing software system in which components located on networked computers communicate and coordinate with each other. [31]

V. Literature Survey

Paper Information	Content and Proposed System	Conclude from the paper (Research gap)
Title : Apriori-Map/Reduce Algorithm **Author**: Jongwook Woo **Publication Year**: 2012 **Publication** : WORLDCOMP'12 - The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing	In this paper Author presents Map/Reduce algorithm of the legacy Apriori algorithm that has been popular to collect the item sets frequently occurred in order to compose Association Rule in Data Mining. The paper proposes Apriori-Map/Reduce Algorithm and illustrates its time complexity, which theoretically shows that the algorithm gains much higher performance than the sequential algorithm as the map and reduce nodes get added	In this paper author gives only theoretical base so here build the code of following algorithm on Hadoop frame and generate experimental data by executing the code with the sample transaction data, which practically proves that the proposed algorithm works. Besides, the algorithm should be extended to produce association rule.
Title: Data Mining in Cloud Computing **Authors**: Bhagyashree Ambulkar, Vaishali Borkar **Publication**: International Journal of Computer Applications (IJCA)	Authors studied about popular data mining algorithm that is Apriori and found it can improve wise using MapReduce technique. MapReduce support parallelism and is also fault tolerance	In particular, we studied the improved Apriori Algorithm on MapReduce programming model on the Hadoop platform. The improved algorithm can scale up to large data set with comparatively less cost.

Title: A Survey on Apriori algorithm using MapReduce Technique **Authors**: Mr. Kiran C. Kulkarni, Mr.R.S.Jagale, Prof.S.M.Rokade **Publication Year**: 2013 **Publication**: IJETAE	In this paper authors gives different idea to improve algorithm in map reduce programming model by Hadoop platform which can be use in implementation phase.	Here Authors describes implementation of Apriori algorithm using map reduce which implementation is not done. Just suggest a proposed algorithm.
Title: An Efficient Implementation Of Apriori Algorithm Based On Hadoop-Mapreduce Mode **Authors**: Othman Yahya, Osman Hegazy, Ehab Ezat **Publication Year:** 2012 **Publication**: IJRIC	Authors implemented an efficient MapReduce Apriori algorithm (MRApriori) based on Hadoop-MapReduce model which needs only two phases (MapReduce Jobs) to find all frequent k-itemsets, and compared our proposed MRApriori algorithm with current two existed algorithms which need either one or k phases (k is maximum length of frequent itemsets) to find the same frequent k-itemsets.	In this paper author implement Apriori algorithm on a single machine or can say a stand-alone mode so there are some chance to implement on multiple node.

Table: 1 Summary of Literature Survey

VI. Conclusion

In today's world a data is increasing exponentially. Data is widely distributed. so for we required a system that handle huge amount of data The above study reveals that the data is huge and distributed so we need to improve the performance of an Apriori algorithm and the said algorithm should work for a distributed model. We found that implementation of algorithm on Map-Reduce Programming model using a default Scheduling algorithm by improving a scheduling algorithm the performance of Apriori algorithm may increase. In future, we are interested to do experimental analyses. we have plan to implement stand alone system then convert the same into the Pseudo distributed or fully distributed mode once fully distributed mode is implemented we work on improving map reduce scheduling algorithm.

References

[1] Han J. & Kamber M. (2006). Data Mining Concepts and Techniques. San Francisco, CA, Elsevier Inc.

[2] Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), 37-54.

[3] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207{216.

[4] R. Agrawal and R. Srikant. (1994) Fast algorithms for mining association rules. In Proc. 1994 Int. Conf.Very Large Data Bases, pages 487-499, Santiago, Chile, September 1994.

[5] Jeffrey Dean and Sanjay Ghemawat (2008). Mapreduce: simplified data processing on large clusters. Commun. ACM, 51:107{113, January.

[6] Apache Hadoop" http://hadoop.apache.org/ , Date Updated [11/02/2014]

[7] Hadoop" http://en.wikipedia.org/wiki/Hadoop, Date Updated [11/02/2014]

[8] Mr. Kiran C. Kulkarni, Mr.R.S.Jagale, Prof.S.M.Rokade (2013), "A Survey on Apriori algorithm using MapReduce Technique", IJETAE, pages 24-32

[9] Xiaoyu Sun (2012), An Enhanced Self-Adaptive Mapreduce Scheduling Algorithm, Pages 56-104, University of Nebraska, Lincoln, Nebraska

[10] Jongwook Woo (2012), ": Apriori-Map/Reduce Algorithm", Pages 27-36, WORLDCOMP'12 - The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing,LasVegas,USA

[11] Othman Yahya, Osman Hegazy, Ehab Ezat (2012), "An Efficient Implementation Of Apriori Algorithm Based On Hadoop-Mapreduce Model". IJRIC , Pages 59:67, Cairo University, Cairo, Egypt

[12] Dhruba Borthakur (2007), The Hadoop Distributed File System: Architecture and Design, The Apache Software Foundation, pages 35-36

[13] HDFS" http://en.wikipedia.org/wiki/HDFS#Hadoop_distributed_file_system , Date Updated [12/01/2014]

[14] MapReduce",http://en.wikipedia.org/wiki/Mapreduce , Date Update [02/12/2013]

[15] EMC News, Press Release, http://www.emc.com/about/news /press/2012 /20121211-01.htm, Date Updated [11/12/2013]

[16] Introduction to Data Mining and Knowledge Discovery, 3rd Edition ISBN: 1-892095-02-5, Two Crows Corporation, 10500 Falls Road, Potomac, MD 20854 (U.S.A.), 1999.Ch.1

[17] Anthony T.Velte, Cloud Computing: A Practical Approach, TATA McGraw Hill Edition.

[18] AprioriAlgorithm"http://en.wikipedia.org/wiki/Apriori_algorithm, Date Updated [12/01/2014]

[19] Ven Katadri .M, Dr. Lokaanaathaa C.Reddy, A review on data mining from past to future, International Journal of Computer Applications (0975 – 8887) Volume 15– No.7, February 2011.

[20] Bhagyashree Ambulkar, Vaishali Borkar, Data Mining in Cloud Computing, Proceedings published by International Journal of Computer Applications (IJCA) ISSN: 0975 – 8887.

[21] Charanjeet Kaur(2013), Association Rule Mining using Apriori Algorithm: A Survey, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 6, June 2013,ISSN: 2278 – 1323,pg no. 2081-2084

[22] R. Agrawal and J. Shafer, "Parallel Mining of Association Rules," IEEE Trans. Knowledge and Data Eng., Vol. 8, No. 6, Dec. 1996, pp. 962–969.

[23] D. Cheung et al., "A Fast Distributed Algorithm for Mining Association Rules," Proc. 4th Int'l Conf. Parallel and Distributed Information Systems, IEEE Computer Soc. Press, Los Alamitos, Calif., 1996, pp. 31–42.

[24] Troy Raeder, Nitesh V. Chawla (2010) , "Market Basket Analysis with Networks", Social Networks Analysis and Modeling Journal.pp 1-30.

[25] Cloud Computing" http://en.wikipedia.org/wiki/Cloud_compu ting, Date Updated [20/01/2014]
[26] InformationWeek, Google Revealed: The IT Strategy That Makes It Work, 08/28/2006
[27] Carr, David F (2006-07-06), "How Google Works" http://www.baselinemag.com/c/a/Infrastructure/How-Google-Works-1/[20/01/2014]
[28] Google's Colossus Makes Search Real-Time By Dumping MapReduce",http://highscalability.com/blog/2010/9/11/ googles-colossus-makes-search-real-time-by-dumping-mapreduce.html[20/01/2014]
[29] Tanenbaum, Andrew S (September 1993). "Distributed operating systems anno 1992. What have we learned so far?". Distributed Systems Engineering. pp. 3–10.
[30] Distributed Computing", http://en.wikipedia.org/wiki/Distrib-uted_computing, Date Updated [11/02/2014]
[31] Coulouris, George; Jean Dollimore, Tim Kindberg, Gordon Blair (2011). Distributed Systems: Concepts and Design (5th Edition). Boston: Addison-Wesley.
[32] Job tracker", http://wiki.apache.org/hadoop/ JobTracker, Date Updated [11/02/2014]
[33] Name Node", http://wiki.apache.org/hadoop/ NameNode, Date Updated [11/02/2014]
[34] Task Tracker", http://wiki.apache.org/ hadoop/TaskTracker, Date Updated [11/02/2014]
[35] Data Node", http://wiki.apache.org/hadoop/ DataNode, Date Updated [11/02/2014]