International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 8, August - 2014

# A Novel Algorithm PDA (Parallel And Distributed Apriori) for Frequent Pattern Mining

Prachi S. Bhokare
Computer Engineering Dept
Thakur College of Engg
Mumbai, India

Dr. Rekha Sharma
Computer Engineering Dept
Thakur College of Engg
Mumbai, India

***Abstract*- Frequent itemset mining is the highly researchable field of data mining. Apriori and FP Growth algorithms are most traditional algorithms for it. Developing fast and efficient algorithm for frequent pattern mining is challenging task. In this paper, we are improving the efficiency of Apriori algorithm using transaction reduction concept to handle big data problem which can partition the data into the clusters and perform data mining operation in parallel as well as distributed environment. Implementation is being done in Hadoop. This method does not require redundant communication or computation, but can achieve load balancing so as to fully utilize the computing resources.**

***Keywords- Association rules, Apriori, Hadoop, KDD, MapReduce.***

## I. INTRODUCTION

Data Mining is a nontrivial process, extracting potential, useful, novel, ultimately understandable knowledge from a large database or a large number of original data in data warehouse [14]. Data mining is the core of KDD and one of the most international front line in Database, Data warehouse and the domain of Information Decision.

In the age of big data, complex statistical analysis such as market basket analysis and data association analysis has become an urgent need for enterprises with an effective way to analyze the large scale data deeply. From the analysis of abstracted patterns, decision-making process can be done very easily. So, KDD has drawn attention from industry as well as research communities. Association rule is most essential concept of data mining which can discover the relationship between itemset from the database that often has hundreds of attributes and records, contains complex relationship between data tables, and remains a time-consuming process.

A traditional mining algorithm faces many problems while handling massive data. With rapid growth of internet, internet of things and sensor network, data are increasing exponentially. Many proficient algorithms are proposed based on two main algorithms: Apriori and Frequent pattern growth for association rule mining in the large scale data will cause to "high memory consumption, low computing performance, poor scalability and reliability" and other problems.

In this paper, we have implemented an efficient approach for frequent pattern mining using parallel Apriori algorithm which is based on MapReduce. We also try the same algorithm in distributed environment. The rest of the paper is organized as follows: Section II contains basic information of frequent pattern mining. Section III Introduces the Hadoop. Detail implementation for PDA algorithm is given in section IV. Section V contains performance evaluation of PDA algorithm. Conclusion and future work is discussed in section VI.



## II. FREQUENT PATTERN MINING

Central themes of data mining is association rule mining which was first invented by the Agrawal [1][2].

The association rule has the following basic concepts:
Support S is the percentage of transactions in D that contain AUB.
Support (A=>B) = P (AUB)
Confidence c is the percentage of transactions in D containing A that also contain B.
Confidence (A=>B)=P(B/A)
This association rule-mining task is classified into two steps:
Step1. Find out the frequent itemset which can fulfill the minimum support criteria of user specified minimum support. Step2. Generate the rule by applying the user defined minimum confidence and frequent itemsets which fulfill the minimum support criteria.
The Support and Confidence are the common factors to measure the strength of the association rule.The rules that have a support and confidence greater than thresholds are called as strong rules.
There are many types of association rules. Some of them are listed below:

1. Generalized Association Rule.
2. Quantitative Association Rule.
3. Interval Association Rule.
4. Sequential pattern mining.
5. Maximal Association Rule.

Researchers developed many algorithms based on association rules. Each algorithm has its own advantages and disadvantages. These are some traditional algorithms use in data mining to find frequent itemsets.

- Apriori Algorithm
- FP-Growth Algorithm
- Eclat Algorithm
- LORE etc.

Ariori is one of the traditional algorithms, which is a seminal algorithm proposed by R.Agrawal and R.Srikant in 1994 for mining frequent itemsets for Boolean association rule [1][2]. Even though it is most simplex algorithm but it is costly in terms of space and time complexity since it requires repeated scan of database. With rapid growth of internet of things and sensor network, data are increasing exponentially which often cause non negligence problems of memory overflow and huge delay in communication.
Researchers proposes many algorithms to overcome above problems but the performance of every algorithm is different on different databases [3][9]. They also endeavor to parallelize these frequent itemset mining algorithms to speed up the mining of the ever-increasing sized databases. To overcome above problems, the Hadoop is introduced.

## III. INTRODUCTION TO HADOOP

Now a days data deluge is the most frequent challenge faced by the many customers. In 2010 digital data universe was 1.2 Zetta Byte($10^{21}$). According to the scientist, in 2020 it will be around 35 Zetta Byte. Actual problem is not only it's size but the most problematic thing is that 90% of digital universe is unstructured. Hadoop provide the solution for this problem. Using Hadoop we can handle many problems related to the big data mining. It also provides the solution for increasing the efficiency of many sequential algorithms. Hadoop adaption in industry increases day by day. It is one of the scalable fault tolerant distributed systems for data storage and processing [4] [5].
There are two components of Hadoop:

1. HDFS
2. MapReduce.

Over the years, many distributed and parallel computing technologies and frameworks were proposed. In particular, MapReduce has gained immense amount of interest in recent years and has rapidly become a popular platform for distributed computing due to its simplicity and scalability at low cost. Map Reduce provides fault tolerant distributed processing. It is a recent programming framework for processing and generating large datasets. The main important feature of Map Reduce is that it can operate on both structured and unstructured data. The two primitive functions that MapReduce provided are: Map and Reduce [12]. The Map function is applied on the input data and produces a list of intermediate <key, value> pairs.
Map:: (Key1) ➡ list (key2, value2)
The Reduce function is then applied to the list of intermediate values that has the same key. It typically performs some form of merging operation, and produces zero or more output <key, value> pairs.
Reduce:: (Key2, list (value2)) ➡ list (value2)

A key benefit in MapReduce is that the programmer does not need to deal with the complicated code parallelism, and focuses on the required computation. The MapReduce runtime is responsible for parallelization, distributed computing and concurrency control.

## IV. IMPLEMENTATION OF PDA- ALGORITHM

In this section, we are going to redesign the Apriori algorithm to work parallel using the MapReduce. MapReduce is a parallel/distributed programming model. This new design improves the efficiency of sequential Apriori in many aspects. Here we are using Hadoop environment, with the machine confirmation is CPU Pentium I-7, Ram 4.0 GB, Hard Disk 1.0 TB.
Parallel program based on the MapReduce, doesn`t need to consider the problem of task allocation. The MapReduce can dynamically assign tasks to the idle cores. A key benefit in MapReduce is that the programmer does not need to deal with the complicated code parallelism, and focuses on the required computation. The MapReduce runtime is responsible for parallelization and concurrency control. The runtime splits the input automatically into parts that are processed concurrently on multiple nodes. Each node is then assigned to the Map function. Each Map function then process, localize input data to find out candidate 1 item set in form of <Key, Value > pair. Here the Key is individual item which is present in input dataset and Value is nothing but the number of occurrences of it. After computation of this, output of each Map function is passes to the data aggregation layer. Data aggregation layer combines all the data according to the key and generate global <Key, Value> pairs. In this stage, all the intermediate results are stored in the temporary file. Later data stored on temporary file again split and passes to the Reduce function. Reduce function prunes the Key items which don't satisfy the minimum support criteria which are previously mention by the user.
In both Map and Reduce stages, the runtime must dynamically decide the size of the data partitions, the number of computing nodes, the assignment of data partitions to nodes, and the allocation of memory buffer space. These decisions (including the workload partitions) can either be implicit (automatically decided by the runtime based on some default settings) or explicitly specified by the programmer via APIs or configuration files. As per the property of an Apriori Algorithm, all nonempty subset of a non-frequent itemsetmust also be a non-frequent.The output of this stage is further provided as an input to the next iteration. This algorithm stops when there is no output file is present. The new algorithm (PDA) helps in reducing the size of candidate itemsetas, it removes those itemsets whose subset were absent in the previous iteration's output file. Once the frequent itemsets are generated, association rules are developed.
Figure 4.1 shows the single iteration data flow of MapReduce.

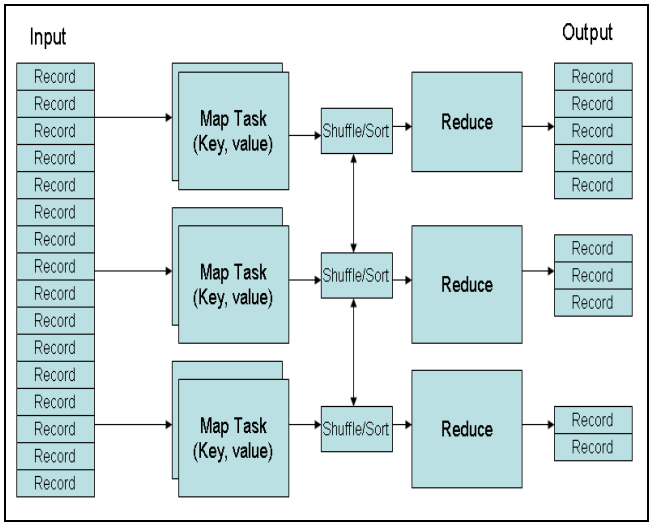


Figure 4.1 Dataflow of MapReduce[13].

Another key aspect of Hadoop is its failure model. Hadoop is designed to have high degree of fault tolerance. In comparison to many available parallel/distributed systems, it is able to complete the assigned tasks failures in the cluster. The primary way that Hadoop achieves fault tolerance is through restarting tasks. In distributed environment, the slave nodes (*TaskTrackers)* involved in the computation are in constant communication with the master node (*JobTracker).* If a TaskTracker failed to communicate with the JobTracker for a period of time (by default, 1 minute), the JobTracker will assume failure on that TaskTracker. It will then assign another active TaskTracker to re-execute all the tasks that were in progress on the failed Tasktracker. In Hadoop MapReduce, a computing node has no knowledge of its peers and knows only its own set of inputs and its own outputs, this enable a very simple and reliable task restart procedure upon any failure.

## V. PERFORMANCE EVALUATION

To evaluate the performance of the algorithms over a large range of data, the result of varying minimum support and number of transactions are shown in table 5.1.

Table 5.1 Variation of execution time with minimum support

| support | Old Apriori | Parallel Apriori |
|---|---|---|
| **0.25** | 12.33 | 7.23 |
| **0.50** | 5.24 | 2.665 |
| **0.75** | 4.93 | 2.484 |
| **1** | 4.01 | 2.41 |

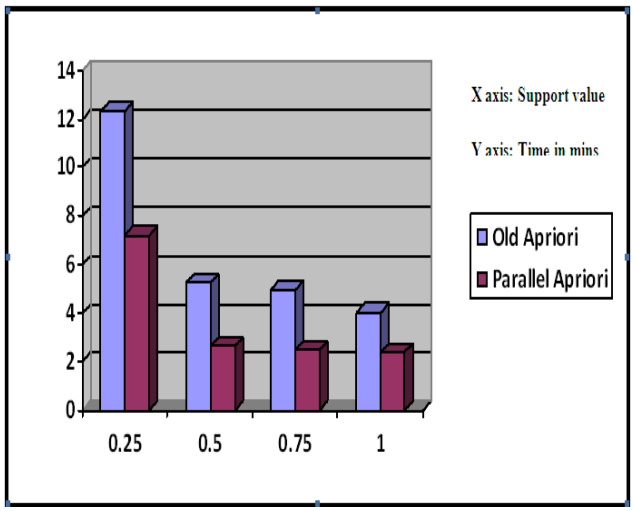


Figure 5.1: Comparison of Old Apriori with Parallel Apriori [9]

In above graph, we are comparing old Apriori Algorithm with Parallel Apriori Algorithm. Result shows that, Parallel Apriori performs well as compare to old Apriori algorithm. It will show nearby 50% improvement in performance [9]

Table 5.2 Comparison of old Apriori and Parallel Apriori with different number of transactions [9]

| No. of Transactions | Old Apriori | Parallel Apriori |
|---|---|---|
| **25000** | 2.10 | 3.39 |
| **50000** | 2.89 | 2.67 |
| **75000** | 3.86 | 2.64 |
| **98980** | 4.01 | 2.41 |

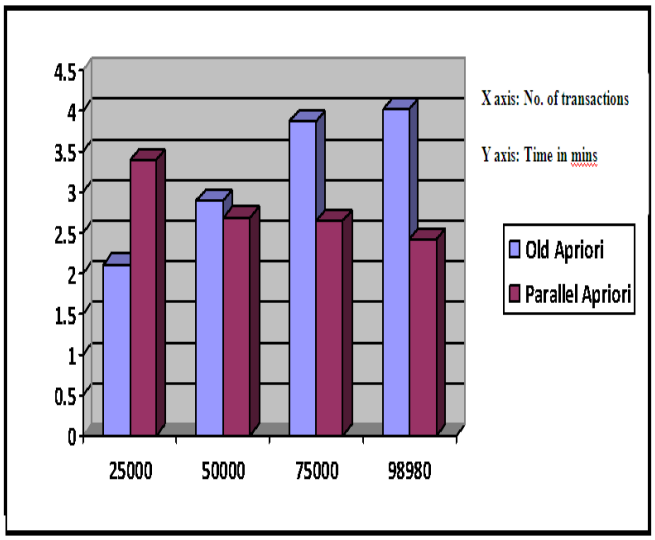


Figure 5.2: Comparison of old Apriori and parallel Apriori with different number of transactions [9].

In this, Comparison of old Apriori and parallel Apriori with different number of transactions i.e. 25000, 50000, 75000, 98980 is shown. As we all knows that old Apriori

algorithm performs well for small number of transactions. As soon as transactions get increased its performance gets decreased [9]

Table 5.3: Different support on different nodes.[9]

| support | Old Apriori | PDA on 1 node | PDA on 2 nodes | PDA on 3 nodes | PDA on 4 nodes |
|---|---|---|---|---|---|
| **0.25** | 12.33 | 7.23 | 6.70 | 3.48 | 3.65 |
| **0.50** | 5.24 | 2.66 | 2.70 | 2.73 | 2.82 |
| **0.75** | 4.93 | 2.48 | 2.50 | 2.57 | 2.80 |
| **1** | 4.01 | 2.41 | 2.36 | 2.40 | 2.42 |

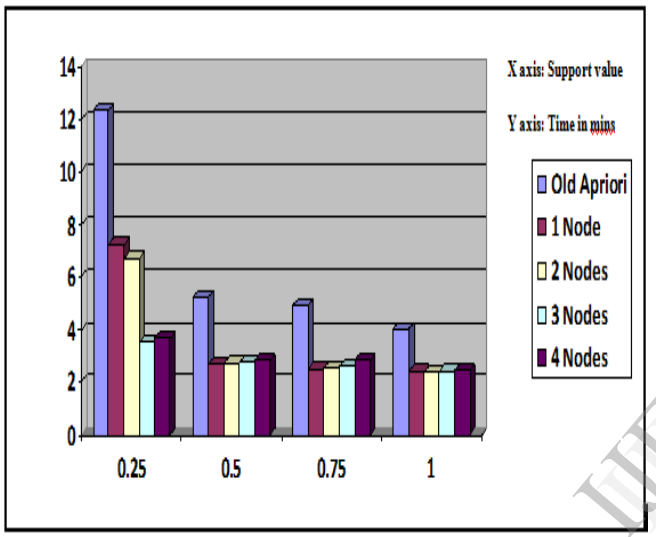


Figure 5.3: Different support on different nodes.

In this, we are trying to execute parallel Apriori in distributed environment. For that, we are using Hadoop's MapReduce framework which will itself perform all the tasks in parallel with distributed environment. As we all knows that to implement the Hadoop, we required powerful machine (at least I7 processor with 4GB Ram). Due to the resource problem here, we are using virtual machine concept for distributed environment.

In this, we are comparing traditional Apriori algorithm with PDA algorithm which will run on single machine, 2 machines, 3 machines and 4 machines (Virtual machines). Above graph shows that Traditional Apriori algorithm takes more time as compare to the PDA algorithm. We are executing this experiment on virtual machines so it will not show much difference when performing on 1 machine, 2 machines, 3 machines and 4 machines. It is due the single processor's extensive use but still all the readings takes about to half time of traditional Apriori algorithm. Results will definitely improve, if we perform same operation in actual distributed environment.

Table 2. Same support, Different no. of transactions and no. of nodes. (Support value 100%)

| Transaction | Old Apriori | PDA on 1 node | PDA on 2 nodes | PDA on 3 nodes | PDA on 4 nodes |
|---|---|---|---|---|---|
| **25000** | 2.10 | 3.39 | 3.43 | 3.58 | 5.85 |
| **50000** | 2.89 | 2.67 | 2.70 | 2.63 | 2.62 |
| **75000** | 3.86 | 2.64 | 2.58 | 2.57 | 2.54 |
| **98980** | 4.01 | 2.41 | 2.36 | 2.30 | 2.22 |

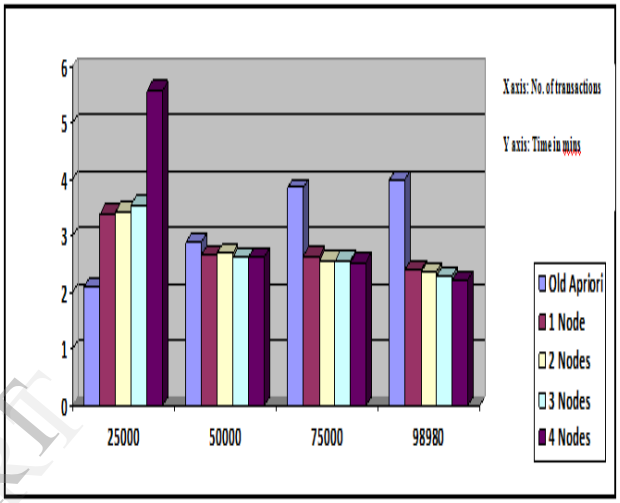


**Figure 5.4:** Same support, Different no. of transactions and number of nodes. (Support value 100%)

In this, comparison of old Apriori and PDA algorithm with different number of transactions i.e. 25000, 50000, 75000, 98980 on different number of machines (virtual machines) i.e. on 1, 2, 3 and 4 machines. Old Apriori algorithm performs well for small number of transactions. As soon as transactions get increased its performance gets decreased but at the same time if we try to execute same algorithm on multiple machines to achieve distributed computing, we will get better performance than parallel computing.

From graph, we get some idea about parallel with distributed nature of the PDA algorithm. For evaluating the performance, we used virtual machine technique (For configuring the Hadoop Minimum requirement is I7 Processor with 4GB Ram). As soon as number of transactions gets increased performance of old Apriori starting decreased. As stated earlier, here we are using virtual machine so it doesn't show the drastic changes in performance when comparisons are done with multiple machines. It is simply because of exhaustive use of single processor but performance is still better than old Apriori. From the above results, it is seen that PDA algorithm perform well as compare to the old Apriori.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we mainly try to resolve the problems which are faced by the Sequential Apriori algorithm using parallel computing. We used improved MapReduce function of Hadoop for implementation. Map/Reduce paradigm is a clearly parallel and/or distributed system which takes full advantage of each machine with some parallel/distributed computing environment. Experiments show that Parallel Apriori Algorithm performs well as compare to old Apriori algorithm. It will show nearby 50% improvement in performance [9]

It also have inbuilt ability to handle many difficult problems, including parallelization, concurrency control, network communication and fault tolerance [13].

We performed all these experiments on single high performance, I7 Processor having 4 GB ram. If we will execute PDA algorithm on multiple machines with actual distributed environment, it will defiantly give the better performance than virtual machine.

## VII. REFERENCES

[1] Agrawal R, Imielinski T, Swami A, "Mining association rules between sets of items in large databases". In: Proc. of the l993ACM on Management of Data, Washington, D.C, May 1993. 207-216

[2] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, in Proc. 20th Int. Conf. Very Large Data Bases, VLDB , edited by J.B. Bocca, M. Jarke, and C. Zaniolo, Morgan Kaufmann 12(1994) 487–499.

[3] W. Fang, K. K. Lau, M. Lu, X. Xiao, C. K. Lam, Y. Yang, B.He, Q.Luo, P. V. Sander, and K. Yang.Parallel data mining on graphics processors. Technical Report 07, The Hong Kong University of Science & Technology, 2008

[4] http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/Introduct ion_EMRArch.html.

[5] Mr.Kiran C. Kulkarni1, Mr.R.S.Jagale2, Prof.S.M.Rokade3, A Survey on Apriori algorithm using MapReduce Technique, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008Certified Journal,, Volume 3, Special Issue 4, March 2013)

[6] Data Mining Using Clouds: An Experimental Implementation of Apriori over MapReduce, Juan Li1, Pallavi Roy1, Samee U. Khan1, Lizhe Wang2, Yan Bai3

[7] Available from: http://hadoop.apache.org/hdfs/ [Last cited on 2011 Oct 15].

[8] Karim M, Hossain M, Rashid M, Jeong BS, Choi HJ. A MapReduce Framework for Mining Maximal Contiguous Frequent Patterns in Large DNA Sequence Datasets. IETE Tech Rev 2012;29:162-8

[9] Prachi Bhokare, Dr. Rekha Sharma "An Efficient Approach for Frequent Pattern Mining Using Parallel Computing" International Journal of Engineering Research & Technology, Vol. 3 - Issue 7 (July - 2014)

[10] Han Jiawei, KamberMiceline. Fan Ming, MengXiaofeng translation, "Data mining concepts and technologies". Beijing: Machinery Industry Press. 2001

[11] R. Agrawal and J.C. Shafer , "Parallel Mining of Association Rules,"IEEE Tran. Knowledge and Data Eng. , vol. 8, no. 6, 1996,pp.962-96

[12] Zhuobo Rong Sch. of Comput. & Inf. Sci., Southwest Univ., Chongqing, China Dawen Xia ; Zili Zhang "Complex statistical analysis of big data: Implementation and application of Apriori and FP-Growth algorithm based on MapReduce".

[13] MapReduce Tutorial http://pages.cs.wisc.edu/~gibson/mapReduceTutorial.html

[14] Map Reduce Programming Tutorial - Manjrasoft www.manjrasoft.com/download/MapReduceModel.pdf

[15] Ming Zhu. Data Mining[M]. Press of University of Science And Technology of China,2002:5-7.