International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

Frequent Data Partitioning using Parallel Mining Item Sets and MapReduce

Chenna Venkata Suneel[1], Dr. K. Prasanna[2], Dr. M. Rudra Kumar[3]
[1]M.Tech.,(PG Scholar), Dept of CSE,Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa, Andhra Pradesh, India
[2]Assocaite Professor, Dept of CSE,Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa, Andhra Pradesh, India
[3]Professor, Dept of CSE, Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa, Andhra Pradesh, India

ABSTRACT
For mining frequent Itemsets parallel traditional algorithms are used. Existing parallel Frequent Itemsets mining algorithm partition the data equally among the nodes. These parallel Frequent Itemsets mining algorithms have high communication and mining overheads. We resolve this problem by using data partitioning strategy. It is based on Hadoop. The core of Apache Hadoop consists of a storage part, called as Hadoop Distributed File System (HDFS), and a processing part called Map Reduce. Hadoop divides files into large blocks. It distributes them across nodes in a cluster. By using this strategy the performance of existing parallel frequent-pattern increases. This paper shows the various parallel mining algorithms for frequent itemsets mining. We summarize the various algorithms that were developed for the frequent itemsets mining, like candidate key generation algorithm, such as Apriori algorithm and without candidate key generation algorithm, such as FP-growth algorithm. These algorithms lacks mechanisms like load balancing, data distribution I/O overhead, and fault tolerance. The most efficient the recent method is the FiDoop using ultrametric tree (FIUT) and Mapreduce programming model. FIUT scans the database only twice. FIUT has four advantages. First: I reduces the I/O overhead as it scans the database only twice. Second: only frequent itemsets in each transaction are inserted as nodes for compressed storage. Third: FIU is improved way to partition database, which significantly reduces the search space. Fourth: frequent itemsets are generated by checking only leaves of tree rather than traversing entire tree, which reduces the computing time.
Keywords : Data Mining, Recommender Systems, Social Network

I. INTRODUCTION

Parallel Frequent Itemset mining is looking for sequence of actions and load balancing of dataset. Creating Hadoop cluster is especially for storage and analyzing data. Through frequent Itemset mining extracting knowledge from data. Example of this technique is Market Basket Algorithm. It also affect on load balancing. It helps to increase the speed of performance. This parallel Frequent Itemset mining is done using map reduce programming model. Partitioning of data in dataset through algorithm making data more efficient. This data partitioning is carried out on Hadoop clusters. Data partitioning necessary for scalability and high efficiency in cluster. In Frequent Itemsets Mining data partition affects to computing nodes and the traffic in network. Data partition may be spread over multiple nodes, and users at the node can perform local transactions on the partition. This increases performance for sites that have regular transactions involving certain views of data, whilst maintaining availability and security. By using Fidoop-DP concept, performance of parallel Frequent Itemset Mining on Hadoop clusters increases.Fidoop-DP is voronoi diagram. It is conceptualized on data partition strategy. Data mining is a process of discovering the pattern from the huge amount of data. There are many data mining technics like clustering, classification and association rule. The most popular one is the association rule that is

divided into two parts i) generating the frequent itemset ii) generating association rule from all itemsets. Frequent itemset mining (FIM) is the core problem in the association rue mining. Sequential FIM algorithm suffers from performance deterioration when it operated on huge amount of data on a single machine.to address this problem parallel FIM algorithms were proposed. There are two types of algorithms that can be used for mining the frequent itemsets first method is the candidate itemset generation approach and without candidate itemset generation algorithm. The example for candidate itemset generation approach is the Apriori algorithm and for, without candidate itemsets generation is the FPgrowth algorithm. The important data-mining problem is discovering the association rule between the frequent itemset.in order to find best method for mining in parallel, we explore a spectrum for trade-off between computation, synchronization, communication, memory usage. Count distribution, data distribution, candidate distribution are three algorithms for discovering the associate rule between frequent itemsets. Minimizing communication is the focus of the count distribution algorithm.it will thus even at the expense of winding up redundant duplication computation in parallel. The data distribution effectively utilizes the main memory of the system.it is communication-happy algorithm. Here nodes to all other nodes broadcast the local data. The candidate distribution algorithm for both, to segment the database upon the different transaction support and the patterns, exploits linguistics of a particular problem. Load balancing is also incorporated by this algorithm.[1]

II. LITERATURE SURVEY

Yaling Xun, Jifu Zhang, Xiao Qin, "FiDoop-DP: Data Partitioning in Frequent Itemset Mining on Hadoop Clusters", 2016.It describes, A data partitioning approach called FiDoop-DP using the Map Reduce programming model. The overarching goal of FiDoop-DP is to boost the performance. A similarity metric to facilitate data-aware partitioning. As a future research direction, we will apply this metric to investigate advanced load balancing strategies on a heterogeneous Hadoop cluster. I.Pramudiono &M.Kitsuregwa," Fp-tax: Tree structure based generalized association rule mining",2004.This paper describes, Investigation of data partioning issues in parallel FIM.Main focus is on map-reduce. Future work is development of Fidoop which exploits correlation among traction to partition large datasets in Hadoop. X.Lin," Mr-apriori:Association rules algorithm based on mapreduce",2014.It explain, Main focus on classical Algorithm connecting and pruning step using prefix Itemset based storage using has table. It points some limitations of Apriori algorithm. S. Hong, Z.Huaxuan, C. Shiping, and H.Chunyan," The study of improved fp-growth algorithm in mapreduce",2013.This describes, Build cloud platform to implement the parallel FPgrowth algorithm based on linked list and PLFPG.PLFPG algorithm compared higher efficiency and scalability. M. Liroz-Gistau, R. Akbarinia, D. Agrawal, E. Pacitti, and P. Valduriez," Data partitioning for minimizing transferred data in mapreduce",2013.It state that, Map Reduce jobs are executed over distributed system composed of a master and set of workers. Input is dividing into several splits and assigned to map tasks. Future work is evasion to perform the repartitioning in parallel

Sandy moen's at al,[2] proposed two new method for mining frequent itemset in parallel on the Mapreduce framework First method is the Dist-Eclat. This method distributes the search space evenly as possible among mapper. This technique mines large dataset but not massive datasets. This algorithm operates in three steps: We use vertical database rather than transaction database.in the first step the vertical database is divided into equal sized blocks called shards and distributed to available mappers. Each mapper extracts the frequent singletons from each block and gives to the reducer. The reducer collects all the frequent tested. In the second step the set of frequent itemsets of size K ae generated (Pk). Frequent singleton itemsets are distributed to the mappers. Each mapper runs Éclat [3] to find frequent K-sized superset of items. The reducer collects all the frequent K-sized supersets of items and distributes it to the next batch of mappers. Round Robin is used for the distribution of the frequent itemset. The third step is the mining the prefix tree.

The mutual information between the mappers are independent, so mapper complete each step independently Every mapper takes the database and gives itemsets for which, we want to know the support .the reducer takes all itemsets and returns only the global frequent itemsets. These itemsets are considered as candidates and distributed to the mappers for breath-first search.

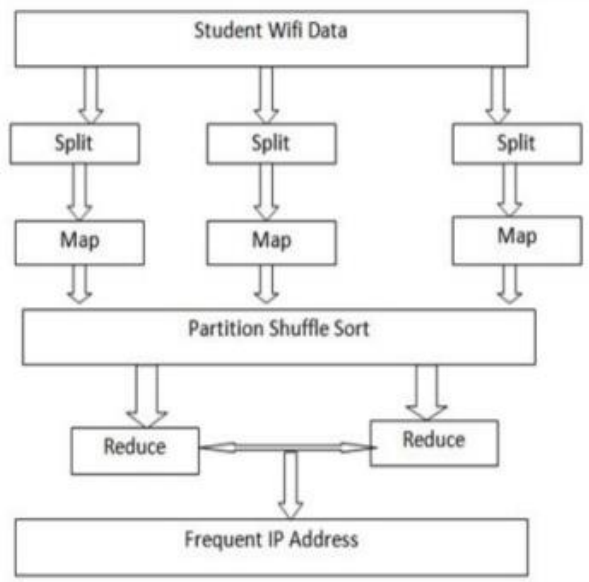
This process continues K-times to generate K-FI's.next step is computing the possible extension. The mapper gives local Tid-list to the reducer the reducer combines the local Tid-lists, to one Tid-list, and assigns prefix to

mappers. The mapper in the final step works on individual prefix group. A prefix group fits in the memory as a conditional database. The diffsets are used to mine the frequent itemsets in the conditional database.

Enhilvathani et al [4] have used the Apriori algorithm for frequent item set generation on mapreduce programing model. For implementation of algorithm is given in five steps In the first step the transaction dataset is partitioned that is Divided into n subsets done that are of map phase.in the second step the data subsets are formatted as <key1, value1>pair, key is Tid(Transaction id). The mapreduce task is executed in third phase. The record of input item subsets are scanned by the Map function and candidate item sets input are generated by the map function

III. PROPOSED WORK

This section is about objective of project, which tells detail of system. It reduces the complexity of data access and retrieval. When we have to dealing with big data i.e. huge amount of data traditional exisisting system seems inefficient. The alternative to this is apache Hadoop, which deals with big data with efficiency. Hadoop itself consists of Map Reduce and HDFS.We use Hadoop with concept called frequent Itemsets which makes it FiDoop(Frequent item Hadoop).It runs on Hadoop cluster Job of map reduce is partitioning of data, it splits the local input data to generate local 1-itemset and it further reduce to specific reduced data. It sorts the data in decreasing order of frequency. In second step job of map reduce is FP-Growth based partitions and last job is last task to aggregate the result from previous stages to generate output. LSH based partitioning boost the performance of system by avoiding large number of comparisons. It uses bucket to keep similar transaction together.



System Architecture

IV. METHODOLOGY

To deals with migration of high communication, and reduce computing cost in map reduce. We use frequent item data partitioning which establishes correlation among transaction for data partitioning. It is based on: Similarity in data and transaction

- Group this highly correlated data.
- Group this highly correlated datacontent comes here Conclusion content comes here . Conclusion content comes here The existing references tell that frequent item mining improves the output up to 31% with18% average. We are working to develop system that investigates the detail of students. [Uses Wi-Fi].It allows generating result based on various parameters.

V. CONCLUSION

Mapreduce programming model is applied for existing parallel mining algorithm for mining frequent itemsets from database and solves the load balancing and scalability. This paper gives the overview of algorithms designed for parallel mining of frequent itemsets .The Apriori and FP tree algorithm were used for mining frequent itemsets. Main drawback of Apriori algorithm is that the database has to be scanned many number of times and huge candidate keys needs to be exchanged between the processor. I/O and synchronization are the other problems in the Apriori algorithm. The disadvantage of FP-growth, however, lies within the impracticableness to construct in-memory FP trees to accommodate large-scale databases. This drawback becomes a lot of pronounced once it comes to huge and

two-dimensional databases. To overcome these problems, FiDoop, an parallel frequent itemset mining algorithm is developed. FiDoop incorporates the ultrametric tree (FIU) rather than Apriori or FP-growth algorithm. The FIU tree achieves compressed storage. FiDoop runs three MapReduce jobs. The third MapReduce job is important. in third job the mapper independently decomposes itemsets and reducer built the ultrametric trees.

VI. REFERENCES

[1]. "Parallel Mining of Association rule." Rakesh Agarwal ,John C Safer

[2]. "Frequent Itemset Mining for Big Data Sandy Moens, Emin Aksehirli and Bart Goethals Universiteit Antwerpen, Belgium

[3]. "ECLAT Algorithm for Frequent Itemsets Generation "Manjit kaur , Urvashi Grag Computer Science and Technology, Lovely Professional University Phagwara, Punjab, India . International Journal of Computer Systems (ISSN: 2394-1065), Volume 01– Issue 03, December, 2014 Available at http://www.ijcsonline.com/

[4]. "Implementation Of Parallel Apriori Algorithm On Hadoop Cluster" A. Ezhilvathani1, Dr. K. Raja. International Journal of Computer Science and Mobile Computing

[5]. "Frequent Itemsets Parallel Mining Algorithms " Suraj Ghadge, Pravin Durge, Vishal Bhosale,Sumit Mishra. Department of Computer Engineering, JSPM's ICOER. International Engineering Research Journal (IERJ) Volume 1 Issue 8 Page 599-604, 2015, ISSN 2395-1621

[6]. "FiDoop: Parallel Mining of Frequent Itemsets Using MapReduce" Yaling Xun, Jifu Zhang, and Xiao Qin, Senior Member, IEEE

[7]. Yaling Xun, Jifu Zhang, Xiao Qin,FiDoop-Dp Data Partitioning in Frequent Itemset Mining on Hadoop clusters,2016.

[8]. S. Sakr, A. Liu, and A. G. Fayoumi, âœThe family of mapreduce and large-scale data processing systems, ACM Computing Surveys (CSUR), vol. 46, no. 1, p. 11, 2013.

[9]. X. Lin, Mr-apriori: Association rules algorithm based on mapreduce,â in Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on. IEEE, 2014, pp. 141"144.

[10]. S. Hong, Z. Huaxuan, C. Shiping, and H. Chunyan, âœThe study of improved fp-growth algorithm in mapreduce, in 1st International Workshop on Cloud Computing and Information Security. Atlantis Press, 2013. 11P. Uthayopas and N. Benjamas, Impact of i/o and execution scheduling strategies on large scale parallel data mining, Journal of Next Generation Information Technology (JNIT), vol. 5, no. 1, p. 78, 2014.

[11]. Y. Xun, J. Zhang, and X. Qin, Fidoop: Parallel mining of frequent itemsets using mapreduce, IEEE Transactions on Systems, Man, and Cybernetics: Systems, doi: 10.1109/TSMC.2015.2437327, 2015.

[12]. W. Lu, Y. Shen, S. Chen, and B. C. Ooi, Efficient processing of k nearest neighbor joins using mapreduce,â Proceedings of the VLDB Endowment, vol. 5, no. 10, pp. 1016â"1027, 2012.

[13]. J. Leskovec, A. Rajaraman, and J. D. Ullman, Mining of massive datasets. Cambridge University Press, 2014. 9B. Bahmani, A. Goel, and R. Shinde, Efficient distributed locality sensitive hashing,â in Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012, pp.2174â"2178.

[14]. M. Liroz-Gistau, R. Akbarinia, D. Agrawal, E. Pacitti, and P. Valduriez, âœData partitioning for minimizing transferred data in mapre- duce,â in Data Management in Cloud, Grid and P2P Systems. Springer,2013, pp. 1"12