# Parallel Mining of Frequent Item sets using Map Reduce Technique: A Survey

**Prajakta G. Kulkarni, Rubeena A. Khan**

***Abstract: In the task of data mining, the most important job is to find out frequent itemsets. Frequent itemsets are useful in various applications like Association rules and correlations. These systems are using some algorithms to find out frequent itemsets. But these parallel mining algorithms lack some features like automatic parallelization, well balancing the load, distribution of data on large number of clusters. So there is a need to study the parallel algorithms which will overcome the disadvantages of the existing system. In this paper a technique called fidoop is implemented, In this technique the mappers work independently as well as concurrently. This is done by decomposing the data across the mappers. Reducers work is to combine these jobs by developing small ultra metric trees. To show this fidoop technique on the various clusters is very delicate in distribution of data because different datasets are with different partition of data. This fidoop technique is also useful in heterogeneous clusters[16].***

***Index Terms: Frequent item sets, mappers, reducers, Ultrametric trees, FiDoop.***

## I. INTRODUCTION

For better utilization of frequent item sets using large size database, speed is very important. But this is a critical issue to speed up computation of frequent itemsets. In today's fast computing era there are excessive databases that are generated from different applications. So only sequential process of FIM is not sufficient to compute the frequent itemsets as it suffers from low performance. Hence there should have been a strategy to handle this issue. MapReduce is the solution which can handle the large number of databases across number of clusters. This distributed approach is integrated with FIM to overcome the disadvantages of sequential FIM and hence performance can be increased[16]. This Map Reduce with FIM is called FiDoop. In this strategy we are focusing less on the traditional techniques like FP growth by using the FIUT with parallel approach. The working of mappers and reducers is done concurrently to optimize the speed, well balancing the load across various clusters[16].

By using the basic principle of MapReduce will distribute the data among all mappers and get the result from the reducers. Here the reducers integrate the result by developing the small ultrametric trees parallely [16].

## II. PRILIMINARY

In this segment, we will have a look over Association rules, Frequent itemsets , And then we will see the basics of FIUT. FiDoop implementation is done by MapReduce programming model and Hadoop Framework.

### *A. Association Rules [13][14]*

To find out the meaningful and important information from the excessive database which is in the scattered form is called data mining process, such as data warehouse, XML etc. [17]Data mining is likewise a procedure of Knowledge Discovery in Database(KDD). Association Rule Mining(ARM) is a kind of information mining, was initially presented by Agrawal et. Al.1993. ARM gives a method to separate the interesting pattern or frequent pattern from extensive database or database archives. Association Rule Mining(ARM) is characterized by tenets, for example, Support and confidence.

- Support: This rule defines the support *sup* in transaction *T* if sup% of transactions *T* contain *A*□ B.
  □*up* = Pr(*A* □ B).
- Confidence: This rule defines the confidence *conf* if *conf* % of transactions T that contain *A* also contain B.
  conf = Pr(B | A)

### *B. Frequent Itemset Mining*

Frequent itemsets are very important in data mining. It tries to find out interesting pattern from large database, such as association rules, data warehouse etc.

The procedure of extracting the frequrnt itemsets are dependent on strong association rules. These Association rules were first presented by Rakesh Agrawal in 1993. Association rules are important in many applications such as product selling in supermarkets. The products which are purchased often by the customers can be identified as frequent itemsets and these are based on strong Association rules. For example ,if a customer buys butter and bread together, then he could buy milk also. And hence this type of information is useful to take important decisions regarding product sale, pricing of items etc.[18]

### *C. FIUT*

FIUT is a frequent Itemset Ultrametric tree. In which all branches of the tree have common ancestor and all are equal in length. FIUT has following structure-

1. First root is labeled as null and all frequent itemsets are inserted as branches of tree , without repeating the nodes. Each branch will start from same root.
2. Each leaf has two fields that are - Named item ,Name and count. Count is used to define the number of transactions.

FIUT has main two phases-

The first phase calculates the support for each item or frequent one itemset and then pruning technique is applied

**Revised Version Manuscript Received on November 28, 2016.**

**Prajakta G. Kulkarni,** Department of Computer Engineering, Modern Education Society's College of Engineering, Savitribai Phule Pune University, Pune (Maharashtra) - 411001, India.

**Prof. Rubeena A. Khan,** Department of Computer Engineering, Modern Education Society's College of Engineering, Savitribai Phule Pune University, Pune (Maharashtra) - 411001, India.

on large dataset. Second phase does, the repetitive construction of small ultrametric trees. These trees show actual output. User can get the idea of Ultrametric tree by following diagram.

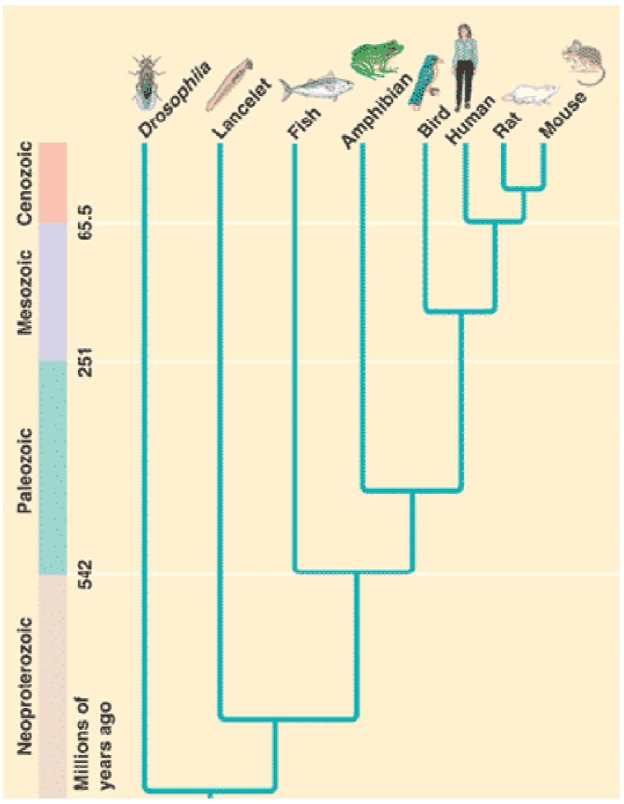

**Fig. 1. Ultrametric Tree [15]**

### *D. HADOOP [12]*

Hadoop is an open-source Java based system, which permits to store and process the extensive number of information in a distributed domain[12].

The Apache Hadoop framework has following modules:

*Hadoop Common* – This module provides the libraries and utilities which are requested by other modules.

*Hadoop Distributed File System (HDFS)* – This module is used to store huge amount of data across the data nodes. This is a master and slave structure in which master node is Name node and rest are data nodes. Master node is the one which controls the data processing. It has one or more data nodes as slave which is used to store the partitioned data. Name Node maps the input data into number of data nodes.

*Hadoop YARN* – This is a resource-manager and it is responsible for managing computing resources in distributed system.

*Hadoop MapReduce* – This is a programming model and it processes excessively large data.

Above four components are shown in pictorial form in Fig.2

### *E. Advantages of Hadoop*

Hadoop framework is very efficient for excessively large data which allows automatic data processing, data partition. It is most suited for distributed systems. It partitions the data across all data nodes and gets the results in reduced form very quickly. Hadoop uses automatic fault-tolerant technique. It shows high availability. Removal and addition of clusters are done dynamically without any interruption. It is compatible with any platform.

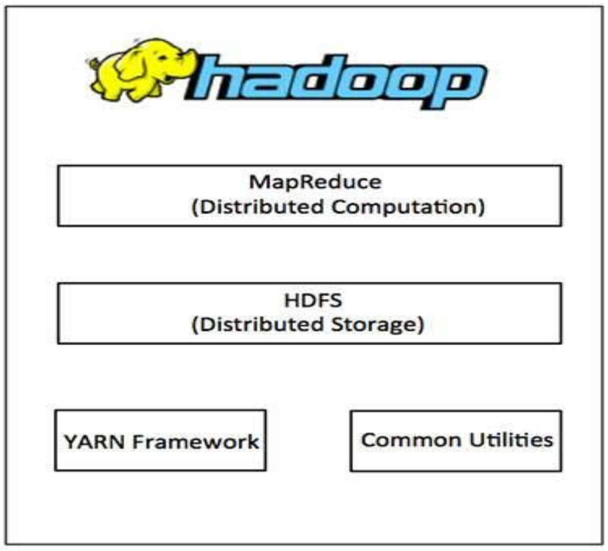

**Fig. 2. Hadoop Componants [12]**

### *F. Map Reduce [11]*

MapReduce is a programming model that runs in the background of Hadoop and provides an easy way to process excessively large data.

Working of MapReduce is as follows:

The MapReduce defines two important tasks, that are Map and Reduce.

•The Map task accepts one group of dataset and transforms it into another group of dataset, where each elements are broken down into partitions called tuples (key value pairs).

•The Reduce task accepts the output from the Mappper as an input and adds another function in data tuples (key value pairs) of mapper.[19]

The reduce task is performed after the map task. There are some important phases in MapReduce; let us have a look in it.[19]

•Input Phase

− In this phase, a Record Reader is used for transforming each record as an input set. Then it will send the translated datasets to the mapper.

•Map

− This module is characterized by user. It accepts pairs of key values for computing and forms more than one pairs of key values.

•Intermediate Keys

−The mapper generates key value pairs is known as intermediate keys.

• Combiner

− This section is a type of local Reducer that gathers same data from the previous phase into similar sets. It also takes the intermediate keys from the map phase as input and applies a user defined code to gather the values in a small scope of one mapper. This is not a part of the main MapReduce form; it is an alternative.

• Shuffle and Sort

− The Reducer task uses Shuffle and Sort technique. It gathers pairs of key value onto the similar machine . Each key values are sorted by their respective key into a larger data set. This larger data set gathers the similar keys together and stores them in a reducer.

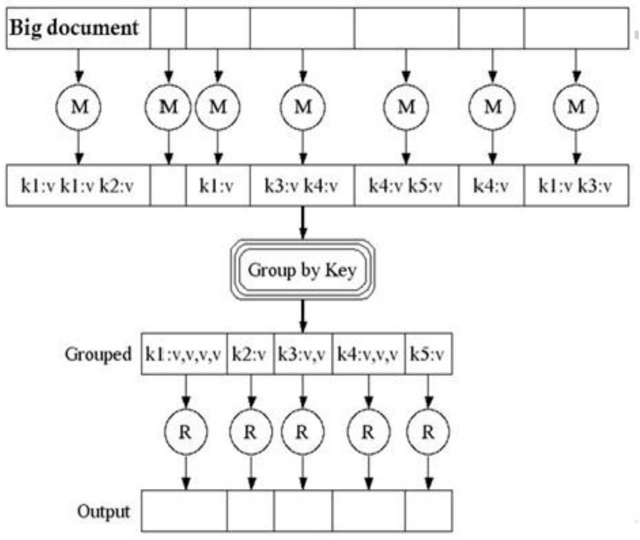

**Fig. 3. Major components of MapReduce [11][22]**

• Reducer − The Reducer takes the similar key valued data as input and applies reducer method on each key value.

## III. RELATED WORK

- T. Imielinski, R. Agrawal [1][20],Swami A., Introduces "**Association rules Mining between sets of items in large databases**". In this paper the authors have presented an issue of extracting the frequent items from very large number of database. The authors found out rules that have minimum transactional support and minimum confidence. They have proposed an algorithm that carefully estimates the itemsets for one pass. Likewise it will adjust between the number of passes over data and itemsets that are measured in a pass. This calculation utilizes pruning system for avoiding certain itemsets. Hence giving right Association itemsets from excessive databases. [20]Advantages of this algorithm are ,it uses buffer management technique which are not fit in the memory in one pass and so will shift to next pass. Also there is no redundancy[1].
- Lin M.-Y., Lee P.-Y., and Hsueh S.-C. proposes[2]," **Apriori-based frequent itemset mining algorithms on MapReduce**", This is an improved way to measure performance of Apriori like algorithm into MapReduce. MapReduce is the approach which is used for parallel mining of large size data in either homogeneous or heterogeneous groups. MapReduce distributes the excessive data between map and reduce functions and it allows total utilization of resources compared to existing systems. Therefore now a days MapReduce is the popular technique for parallel mining. By taking benefit of MapReduce the authors have suggested three algorithms that are SPC, FPC, and DPC. In these algorithms they have used Apriori algorithm with MapReduce function. DPC algorithm accept the different lengths of data dynamically,which is advantage of this algorithm. DPC shows great performance compared to other two algorithms that are SPC and FPC. Thus these three algorithm demonstrates that these calculations scale up straightly with dataset sizes.
- L. Zhou *et al[3] int*roduced "**Balanced parallel FP-growth with MapReduce**". To find out frequent itemset using Association rule is very difficult ,especially when the database is too large. With this reference in this paper authors had proposed a parallel FP growth algorithm using balanced partitioning(BPFP). In first phase of BPFP, it computes the given load based on conditional pattern. In second phase this load divides into several groups. For this it uses MapReduce programming model on FP algorithm. In mapper phase the database is changed into new group of database and then mapper will develop FP tree. This procedure works recursively.[3]
- Tsay Yuh-Jiuan , Hsu Tain-Jung , Yu Jing-Rung [4] proposes "**FIUT: A new method for mining frequent itemsets**" - a very efficient technique for frequent itemset mining(FIM) named as Frequent Itemset Ultrametric Trees (FIUT). It contains two main phases of scans of database. In the first phase it calculates the support count for all itemsets in a large database. In the second phase it applies pruning technique and gives only frequent itemsets. Meanwhile frequent one itemsets are calculated, phase two will construct small ultrametric trees. These results will be displayed in small ultrametric trees. Benefit of FIUT is that it expels K-FIU tree speedily after K-itemsets are created and every time just K-FIU tree will stay in main memory. FIUT has four fundamental points of interest. To start with, it decreases I/O overhead by examining the databases just twice. Second, the FIU-tree is an effective approach to break down a database, which comes due to clustering data exchanges. And hence FIUT decreases the searching space. Third, FIUT gives frequent itemsets as output for each large number of processing. So user can get only frequent itemsets by using this new method of FIUT as each leaf provides frequent itemsets for every data exchange within the cluster. Each data processing of FIUT leaves gives new frequent itemset and this is managed without navigating the tree over and over, which diminishes processing time productively.
- Moens Sandy, Aksehirli Emin and Bart Goethals[5] introduces "**Frequent Itemset Mining for Big Data** " FIM algorithm using MapReduce is implemented using two techniques :Dist-Eclat, BigFIM. Dist-Eclat concentrates on speed by load adjusting technique using k-FIS. BigFIM mainly focuses on mining the large number of database with the use of hybrid approch. Apriori algorithm used here to produce k th FIS. These K th FIS are used to search frequent itemsets which is based on the E-clat technique. Dist-Eclat, BigFIM and k-FIS are utilized with round robin scheduling approach which brings about a better data distribution[5].
- Riondato Matteo, DeBrabant Justin, Rodrigo, Eli Fonseca Upfal[6] presented **"PARMA: A Parallel Randomized Algorithm for Association Rule Mining in MapReduce**"[6] To find out the frequent itemsets from very large databse a randomized parallel approach is used by authors called PARMA. It is beneficial to use in many data mining applications such as Association rules. PARMA uses two functions ,first it

*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication Pvt. Ltd.*

collects the arbitrary data samples and another it applies parallel processing approach to increase the speed of mining procedure. PARMA stays away from replication of dataset which is exceptionally costly. PARMA does this, by making number of little arbitrary segments of the data exchanges and after that applies a mining algorithm on every segment autonomously. And applies parallel approach l by utilizing MapReduce programming paradigm. It gives better performance than previous algorithms and it is limited to find out frequent itemset. Authors have experimented this technique and confirmed that it gives the better output. It is experimented on a variety of datasets from large datasets to very large datasets and proved that it gives correct results[6].

- Wei Lu, Yanyan Shen, Su Chen, Beng Chin Ooi[7][21], presented "**Efficient Processing of k Nearest Neighbor Joins using MapReduce**". In this paper, the authors have presented a method using KNN join with the popular programming paradigm MapReduce for data related calculations. This large data is distributed on number of data nodes. In this the mappers distributes the data into number of data nodes and the reducers gives results in-terms of the KNN join.[21]
- Shekhar Gupta, Christian Fritz, Johan Kleer, and Witteveen Cees[8] presented "**Diagnosing Heterogeneous Hadoop Cluster**". The authors focus is to find small and primary faults in various Hadoop clusters. The faults which are distributed over number of machines that have slowed down as well as to determine the cause for degraded performance. The long term goal is to enable Hadoop schedulers to generate efficient schedules even if they are in heterogeneous clusters. In this, schedulers need to schedule fewer tasks on slower nodes, like CPU hogging, I/O hogging. [8]
- Yao Yi, Wang Jiayin, Bo Sheng2,Chiu C. Tan, Ningfang MI[9] proposes "**Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters**". The primary focus of this paper is to increase the resource utilization, to decrease the makespan of multiple tasks. Authors have suggested that a technique which utilizes the map slots proportion and Reduce slots proportion as a adjustable knob. This adjustable knob reduces the makespan of a given set. These slots are utilized to demonstrate limit of performing the task on every machine. It progressively allocates the slots to the map and reduce functions. Also they have invented a new Tumm system which manages the slots for the map and reduce task in Hadoop[9].

## IV. PROPOSED SYSTEM [16]

FiDoop suggests parallel frequent itemset mining algorithm which is implemented by using MapReduce programming model. This eliminates the disadvantages of existing system and allows automatic parallelization, balancing the load of large database, and effective distribution of given data. In this paper three MapReduce phases are used. The input database is given to the mapper of first phase of MapReduce. Its output is obtained from reducer's frequent one itemset that is the first phase of MapReduce, using decomposition technique[16]. Second phase of MapReduce scans all the data to give k-itemsets by removing infrequent itemsets. In this way second phase of MapReduce removes infrequent itemsets. The third phase of MapReduce is very essential task in which it makes Frequent Itemset Ultrametric tree and generates only frequent k- itemsets using FIUT algorithm. This Proposed System suggests a new method for partitioning the data that is used to well balance and to compute the load among the cluster nodes. [16]The objective of the Parallel frequent itemset mining algorithm using FiDoop is to improve energy efficiency running on clusters of Hadoop. For this we will use different algorithms that can achieve data partitioning ,parallel mining as well as to improve energy efficiency for excessively large database. We will use MapReduce programming paradigm to achieve the parallel mining . The clusters used in this are varying in specification.[16]

## V. CONCLUSION

To overcome the issues which are present in existing systems like parallel mining and load balancing algorithms for frequent itemsets, we used the MapReduce programming approach. It develops an algorithm which is capable to do parallel mining for frequent itemsets, called FiDoop. FiDoop is avoiding the FP tress method and it is using the frequent items ultrametric tree or FIU-tree. So it achieves compressed storage and also avoids the pattern base conditions. FiDoop is the combination of three MapReduce phases. Parallel mining and load balancing of frequent itemsets is done by these three MapReduce phases. [16]The first phase of MapReduce is used to identify the frequent itemsets. The second phase of MapReduce removes infrequent itemsets. The third phase of MapReduce is very essential in parallel mining, in this phase, the mappers divide frequent itemsets whereas reducers make small ultrametric trees parallely. As here we used the parallel mining so automatically it balances the load across the nodes and also it increases the speed.

## REFERENCES

1. R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22,no. 2, pp. 207–216, 1993.
2. M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, "Apriori-based frequent itemset mining algorithms on MapReduce," in Proc. 6th Int. Conf. Ubiquit. Inf. Manage. Commun. (ICUIMC), Danang, Vietnam, 2012, pp. 76:1–76:8. [Online]. Available: http://doi.acm.org/10.1145/2184751.2184842L. Zhou et al., "Balanced parallel FP-growth with MapReduce," in Proc. IEEE Youth Conf. Inf. Comput. Telecommun. (YC-ICT), Beijing, China, 2010, pp. 243–246
3. L. Zhou et al., "Balanced parallel FP-growth with MapReduce," in Proc. IEEE Youth Conf. Inf. Comput. Telecommun. (YC-ICT), Beijing, China, 2010, pp.243–246.
4. Y.-J. Tsay, T.-J. Hsu, and J.-R. Yu, "FIUT: A new method for mining frequent itemsets," Inf. Sci., vol. 179, no. 11, pp. 1724–1737, 2009.
5. Kiran Chavan, Priyanka Kulkarni, Pooja Ghodekar, S. N. Patil," Frequent itemset mining for Big data ", IEEE,Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on Year: 2015 ,Pages: 1365 - 1368, DOI: 10.1109/ICGCIoT.2015.7380679
6. M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, "PARMA:A parallel randomized algorithm for approximate association rules mining in MapReduce," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.,Maui, HI, USA, 2012, pp. 85–94.
7. Wei Lu,Yanyan Shen,Su Chen,Beng Chin Ooi,"Efficient Processing of k Nearest Neighbor Joins using MapReduce"2012 VLDB Endowment 2150-8097/12/06

8. Shekhar Gupta, Christian Fritz, Johan de Kleer, and Cees Witteveen, “Diagnosing Heterogeneous Hadoop Clusters”
9. Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, Ningfang Mi, “Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters ”
10. L. Cristofor. (2001). Artool Project[J]. [Online]. Available: http://www.cs.umb.edu/laur/ARtool/, accessed Oct. 19, 2012.
11. J. Dean and S. Ghemawat, “MapReduce: A flexible data processing tool,” Commun. ACM, vol. 53, no. 1, pp. 72–77, Jan. 2010.
12. https://www.tutorialspoint.com/map_reduce/map_reduce_tutorial.pdf
13. https://www.tutorialspoint.com/hadoop/hadoop_introduction.htm
14. https://en.wikipedia.org/wiki/Association_rule_learning
15. https://www.google.co.in/search?q=ultrametric+tree&client=ubuntu&channel=fs&biw=1315&bih=673&tbm=isch&imgil=CttqmLzrkUFM6M%253A%253BndBUpoGJ782mMM%253Bhttps%25253A%25252F%25252Fwww.quia.com%25252Fjg%25252F1581071list.html&source=iu&pf=m&fir=CttqmLzrkUFM6M%253A%252CndBUpoGJ782mMM%252C_&usg=__igYSfh9gBevb0wNhWTJzJ8R6PFA3D&ved=0ahUKEwjj5snptKLPAhVLPiYKHZrtARsQyjcISA&ei=-ILjVOKF8v8mAGa24fYAQ
16. Yaling Xun, Jifu Zhang, and Xiao Qin,” FiDoop: Parallel Mining of Frequent Itemsets Using MapReduce ” IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 46, NO. 3, MARCH 2016
17. Ramakrishnudu, T, and R B V Subramanyam."Mining Interesting Infrequent Itemsets from Very Large Data based on MapReduce Framework", International Journal of Intelligent Systems and Applications, 2015.
18. Bechini, Alessio, Francesco Marcelloni, and Armando Segatori. "A MapReduce solution for associative classification of big data",Information Sciences, 2016.
19. Yun Lu, , Mingjin Zhang, Shonda Witherspoon,Yelena Yesha, Yaacov Yesha, and Naphtali Rishe. "SksOpen: Efficient Indexing, Querying, and Visualization of Geo-spatial Big Data", 2013 12th International Conference on Machine Learning and Applications, 2013.
20. He Lijun. "Comparison and Analysis of Algorithms for Association Rules", 2009 First International Workshop on Database Technology and Applications, 04/2009vldb.org
21. http://slideplayer.com/slide/5769249/

*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication Pvt. Ltd.*