ISSN 2319 – 7595

Volume 8, No.2, March - April 2019

International Journal of Information Systems and Computer Sciences

Available Online at http://warse.org/IJISCS/static/pdf/file/ijiscs11822019.pdf

https://doi.org/10.30534/ijiscs/2019/11822019

Efficient Parallel Mining Of Frequent Itemset Using MapReduce

Anaswaravs[1], Dr. Mohamed Mubarak[2]

[1]College of engineering Perumon, India ,anuavs091@gmail.com

[2]College of engineering Perumon, India,Mubarak.fma@gmail.com

ABSTRACT

Big dataextremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interaction and the data mining used for dig deep into analyzing the patterns and relationships of data. Frequent item set mining is a data mining method that was developed for market basket analysis. In the project proposed to anefficient data processing using Lshfp growth algorithm and grouping similar objects as the clusters with group id. The traditional datamining is based on the fp growth algorithm focused on the load balancing, and distributed among the nodes of the clusters.The process is mainly based on mapreduce which highly supported by Hadoop.Hadoop is a efficient popular frame work which supports mapreduce and itemset mining .Map reduce is that which contains map phase and reduce phase.Map phase which results the pair of key values and reduce phase which results the reduced results. It aims to decrease network overhead and efficient processing.

Key words: Frequent item set, map reduce, Hadoop.

1. INTRODUCTION

Datamining is the process of extracting information from the data in a huge dataset. Frequent item set mining is implemented on Hadoop. Hadoop is a frame work using map reduce. Frequent item set mining is created by inappropriate data partitioning techniques. So that it results less correlation, network traffic and computing loads. FIM is to solve network traffic, reducing computing loads and also the correlation. Map reduce is an efficient programming model on Hadoop. It stores as the clusters. It is an open source model and it solves the problems of huge amount of data and computation. APACHE Software Foundation is the developer of Hadoop. Frequent item set mining on Hadoop is an efficient parallel programming using map reduce.

Frequent mining set algorithms are divided into apriori schema and fp growth schema .apriori schema contains generate phase and prune phase. Fp growth is frequent pattern mining method without using candidate generation than apriori schema. Which creates a conditional pattern with the support. Frequent mining is a major issue for the sequence mining and association rule mining. Because of the speed and more amount of mining time and high computation. Map reduce is adopt to overcome these problems. Map reduce is processing dataset parallel and load balancing. It is an efficient method for FIM. Map reduce have two phases, map phase and reduce phase. The map phase that divides the task into n number of fragments. The input data small number of fragments as the key value and gives an intermediate result. Reduce phase taken the result of map phase and which gives the output of the collection values.

In a particular set of transactions, association rule mining or frequent itemset mining leads to find the rules that can be enable businesses to predict the occurrence of a specific item based on the occurrences of the other items in the transaction.
This technique can used to analyze various types of datasets.

- **Basket data analysis** – To analyze the association of purchased items in a single basket or single purchase.
- **Cross marketing/Selling** – To work with other businesses that complement your business, but not your competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- **Catalog Design** – The selection of items in a business catalog are often designed to complement each other so that buying one item will lead to buying of another.
- **Medical Treatments** – Each patient is represented as a transaction containing the ordered set of diseases, and which diseases

are likely to occur simultaneously or sequentially and can therefore be predicted

2. RELATED WORKS

Large number of algorithms are introduced for the map reduce framework. Map reduce is the work that counting the data set parallel. Apriori based frequent itemset mining [3] they proposed three algorithms that are Single pass counting (SPC) is the map reduce includes with frequency counting steps and the candidate generation. Fixed pass counting (FPC) is counting the candidate set with n different lengths after p phase. They also count their frequencies. Dynamic pass counting(DPC) is like fpc that n and p generating dynamically in each phase with number of generated candidates.
The basic apriori based on bottom up approach and it quickly results the short frequent item sets. Apriori algorithm have two stages:

- Generate phase
- Prune phase

Disadvantage of this apriori algorithm is, It is a combinatorial explosion and it didn't manage with long frequent items set, and also algorithm is generate a large number of candidate set and also the candidate using repeatedly. It is costly for each transaction to obtain the support value for candidate item set.

Eclat algorithm: It is first set based algorithm used for the vertical database layout. The items are stored in the cover and it is used as the tid list. It is an intersection based approach to calculate the support of item set. It is only suitable for less number of item sets and short time for frequent pattern generation than apriori algorithm. Disadvantage of this is it have a large tid list and takes large space to store candidate set. It required more time when tid is large.

Parallel fp growth: I t is a parallel mining algorithm. It mainly focused for load balancing and by equally partitioning data by the results of the fp growth. Parallel fp growth groups the iems in the fpming result. Parallel fp mining is not efficient neither memory nor speed. But it scans the whole database in the memory. But prohibitive in the case of big data.

FP growth algorithm: Fp growth algorithm is a frequent item set mining technique without the candidate generation. It is a tree based structure with minimum support. It scans the data and discard infrequent items. It counts the support value and displays in the descending order. The disadvantages are it may not fit in main memory. Execution time is large when it is complex compact data structure.

Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello [6] That the paper decide to create a comprehensive survey of the vital recent developments of MOEAs for finding data processing issues. This survey focuses on the first data processing tasks, specifically feature choice, classification, clustering, and association rule mining, since most of the multiobjective algorithms that area unit applied to data processing have controlled these tasks. And also tendency to discuss the essential ideas of multiobjectiveoptimisation and MOEAs, followed by fundamentals of knowledge mining tasks and motivation for applying MOEAs for finding these data processing tasks. afterward,and have a tendency to review totally different MOEAs used for feature choice and classification tasks of knowledge mining. Totally different MOEAs used for agglomeration, association rule mining, and alternative data processing tasks area unit surveyed on the long run scope of analysis. Then, completely different MOEAs that are accustomed solve 2 major data processing tasks, namely, feature choice and classification are mentioned with a special specialize in problems like body cryptography, organic process operators, the sort of objective perform used for optimisation, and choice of final resolution from the nondominated set.

Mohamed Y. Eltabakh∗, Yuanyuan Tian∗, Fatma ¨Ozcan∗ 2Rainer Gemulla, AljoschaKrettek, John McPherson[7] states CoHadoop, a light-weight resolution for colocating connected files in HDFS. The approach to colocation is easy nonetheless flexible; it is exploited completely in several ways that by different applications. It have a tendency to identified 2 use cases—join and sessionization—in the context of log process and represented map-only algorithms that exploit colocated partitions. Pre-partitioning the info improves the performance by eliminating the high-priced data shuffling operation,but every clerk still should pay some network overhead. The default knowledge placement policy of HDFS randomly places partitions across the cluster in order that mappers typically got to scan the corresponding partitions from remote nodes. propose a flexible, dynamic, and lightweight approach to colocating connected knowledge files, that is enforced directly in HDFS.

ShlomiDolev, Patricia Florissi, Ehud Gudes, Shantanu Sharma[8]proposed Big data is used to describe the exponential growth and availability of data, having structured, unstructured and semistructured data, whose size (volume), complexity (variability), and rate of growth make them difficult

or even impossible to be managed and analyzed using conventional software tools and technologies. When the amount of data is to be increased than the time to produce results is also increased. Recover data from big data is still a complex and time consuming approach. Mapreduce frame work using Hadoop and stream frame work using spark and SQL style processing geo-distributed frameworks .It is a very efficient processing methods for the large amount of locally distributed data .i.e many organizations are in different locations and datacenters across the globe ,or even in the same country. Here map reduce and spark based different methods , challenges and advantages are present in the Hadoop and spark. In Hadoop the output stored in the disc and spark is stored in the main memory. So the spark supporting real time computation, fast and efficient processing. But for the efficient geo-distributed big data processing ,the widely used method is the map reduce because it can implement in an efficient and fault tolerant manner in private and public clouds. Geo-distributed Big data support for geo-distributed applications, enrich datasets and big performance, providing geo-security and geo privacy mechanisms, handling dc failures and natural disasters**.**

X.Lin proposed Mr. Apriori [4] algorithm which runs on a parallel Map/Reduce framework. Prune (Ck+1) function to remove the non frequentitemset from the transaction. Where this function eliminates redundancy execution and ck cannot be subset of frequent item sets. This algorithm first calculate frequent itemset for each map node as the time complexity with respect to transaction t, Number of transactions n , Number of item in the transactions m. In second task is to calculate frequent item set with an additional item by joining, sorting and eliminating the duplicated items in each map node. Finally similarity can be calculated at the reduce nodes and eliminate the frequencies that do not meet the minimum support.

S. Hong, Z. Huaxuan, C.Shiping, and H.Chunyan [5] proposed improved FP growth algorithm which combines the sub-tree with same patterns which has high support count. Further it combines with mapreduce computing model MR-IFP (mapreduce-improved FP). It uses the depth first method to mine the frequent itemsets and saves a great deal of space. Built cloud platform to implement the IFP based on linked list Therefore it achieves high efficiency and scalability.

3. PROPOSED WORK

3.1 Partitioning of Data

Data partitioning is based on voronoi diagram based partitioning . voronoidaigram technique is divided spaces into number of parts. For the selection of group select apivote(seed) at preprocessing phase. For each pivot

there is a corresponding transaction consisting of all data points closer to it .The regions for corresponding pivot are called voronoi cells. For a dataset Db set of k pivots are selected and each object in D is correspond to the nearest pivot.

3.2 Distance Metric

To calculate the correlation between the transactions is using Jaccardsimilarity . A Jaccard similarity value 1 indicates two transactions are highly correlated. The relation between of two transactions A and B is given as be

$J(A,B) = (A \cap B) / (A \cup B)$

J(A,B) is a value in between 0 to 1,if it is zero then row set is not compleatly equal, if it is 1 then the row set are same.

3.3 Selection of pivots

Pivot selection is the major process in partitioning the data.CLOPE algorithm, proposed by Yiling Yang, Xudong Guan and Jinyuan You yet in 2002 and adopted by Nikolay Paklin in 2017. principally new clustering CLOPE algorithm which is a more efficient alternative to the existing k-means and c-means algorithms to perform a better clustering, compared to the already existing algorithms used at the meanwhile.

3.4 Partitioning Strategies

The two data partitioning strategies are MinHash and LSH-Based partitioning. MinHash has major role in the Locality Sensitive Hashing.

3.4.1 MinHash

MinHash is helps to to obtaining the similarity between two sets.. MinHashing technique is majoritively using for dimension reduction huge sets to the smaller sets called “signatures”.Two phases to generate signatures.

1) Characteristic matrix
2) MinHash Signatures

Characteristic matrix

Characteristic matrix can be from FList (frequent List) and original dataset. Where rows represent transaction and column denotes items in the transaction. For a given Dataset D= {T1, T2, .Tk}, which contains m items.

3.4.2 MinHash Signatures

Signature matrix can be constructed by characteristic matrix in every item and generates hash function. Characteristic matrix and signature matrix consist of same number of columns but very less rows, thereby reducing the dimensions.

LSH-Based partitioning

Based on the banding method, the signature matrix are divided into (b x r) where b represents the number of bands. Each band consists of maximum rows. For each band a hash function is defined. The function takes column of its corresponding band and hashes them to large number of buckets. We can use the same hash function.

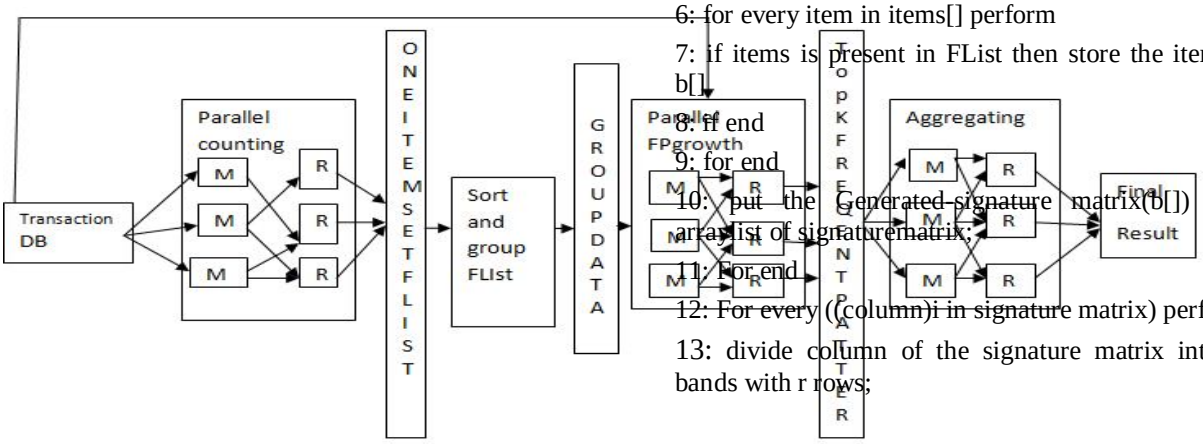

Figure 1. System architecture of pfp

4. IMPLEMENTATION

The implementation consist of three phases that selection of k pivotes and then it is the input of the coned phase map reduce. Frequent list item means the set of transactions (20.050) with 17 attribute list. Taking frequent list by using fp growth and they have a support value. Fp growth results with the combination of attribute list in the total transaction and its support value(number of transactions of x&y in the total transaction).

In the pivote selection stage first calculating the correlation by means of jaccard similarity. Jaccard similarity is the calculation of characteristic or pattern for selecting groups. Using k means creating pivote id. Using jaccard similarity there is four correlation values and connected to the pivote id(k means),corresponding frequent list(fp mining) and serial number(Db).The Db split to items and if it is in the first which is add to the array.

Algorithm 1: LSH-fp-growth

Input: FList, k pivots, DBi;
Output: Transactions for each Group id
1: MAP function contain(key as offset and values as Databasei)
2: load FrequentList(FList), m pivots;
3: Grouplist will be generated from the FList and m pivots
4: for each Transaction in Database perform
5: split the items in transaction and store it as items[]
6: for every item in items[] perform
7: if items is present in FList then store the item in b[].
8: If end
9: for end
10: put the Generated-signature matrix(b[]) into arraylist of signaturematrix;
11: For end
12: For every ((column)i in signature matrix) perform
13: divide column of the signature matrix into B bands with r rows;
14: Store the column in Hashbucket using hashmap
15: For end
16: if one band of column in signature matrix and pivot pk is mapped in to same bucket do
17: Assign j to the Gid;
18: Output contains Gid and new transaction tree of bi
19: If End
20: For each Grouplist (transaction≠i) do
21: if column in sigmatrix contains item in Grouplist then
22: Assign Gid to t;
23: Output contains Gid and new transaction tree of bi

24: If end
25: For end
26: Function end

Input: transactions corresponding to each Gid;
Output: frequent k-itemsets
28: Reduce contains key as Groupid and valuesas database
29: Load Grouplists;
30: perform Grouplist according to Gid
31: perform local fp growth
32: for all (Transaction in Database Gid) do
33: build FP tree for the transaction34: for end
35: for all element in b do
36: Define maxheap with the size K
37: perform TopKFpgrowth in maxheap
38: for all transaction in TopKFpgrowth
39: Output contains final transaction and support
40: for end
41: for end
42: function end

In the signature matrix take the array of items. Byte value for each item(>>24,>>16,>>8,value) and stores corresponding hash index 0,1,2,3 .it divides columns into b bands and r rows .Stored in the hash buckets.it results a new transaction tree with a groupid.

ALGORITHM 2 :Creation of SIGNATURE-MATRIX

Input: item transaction matrix of b[];
Output: generated signature matrix b[]
1: Function to generate signature matrix of b[]
2: For N number of hashfunction
3: Assign every value in MinHash to max integer;
4: For end
5: For each(j=0;j<numberOfhashfunction;j++) perform
6: For every element in the b[ ] perform
7: Convert element to integer and store in value

8: Convert byte to hash of value and perform left shift 24 <- bytetohash[1];
9: Convert byte to hash of value and perform left shift 16 <- bytetohash[2];
10: Convert byte to hash of value and perform left shift 8 <- bytetohash[3];
11: Convert byte to hash of value <-bytetohash[4];
12: hashindex←hashfunction[j]*hash(bytesToHash);
13: if(minhashvalue[j]) >hashindex then
14: minhashvalues[j]=hashindex;
15: if end
16: For end
17: For end
18: Function end

Reduce

Using groupid and database it build a local fp tree and using the maximum heap value by heapsort it generates a final result.(a: abc, adc, acd, b:abc ,bac, cab).

5. RESULTS AND DISCUSSION

Dataset name	No.of Transactions	No.of distinct items	Average transaction length
Banking	**20050**	**5109**	**17**

Figure 2. Data set Description

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The ouput obtaining that in a cluster by calles a query. For example seconed mapreduce is more complicated and it results abc, adc, bdc. Then the final output for each item is a:abc,adc,acd etc With the maximum support.Here taken low ,moderate and high which indicates 0-30,30-70 ,70-100 support values with the id.Less memory utilization compare than other methods because of using the hashing techniques, so it I an efficient method.The results are analyzed and evaluated in terms of Accuracy , Memory usage , Execution time. The proposed method is compared with the existing method and the proposed work is implemented in VISUAL STUDIO 2010.

K-means clustering[1] was actually the first algorithm that was considered to be the most applicable for producing users-to-items recommendations based solely on performing the partitioning of n - observations into k - standalone clusters. According to the following algorithm, each specific cluster normally consists of a pair of two datasets, one - for centroids and another one - for items belonging to a current cluster. Centroids typically represent central items of a cluster, having its features that are very common to each particular item that has been put into a cluster. Specifically, while using k-means clustering to produce users-to-

items recommendation,

dataset of centroids (i.e. cluster's central items) normally represents a set of users for which a list of recommended items is generated. Also, let's remind that the requirement for performing k-means cluster is that each cluster must contain at least one centroid (i.e. user) and a dataset of multiple items.

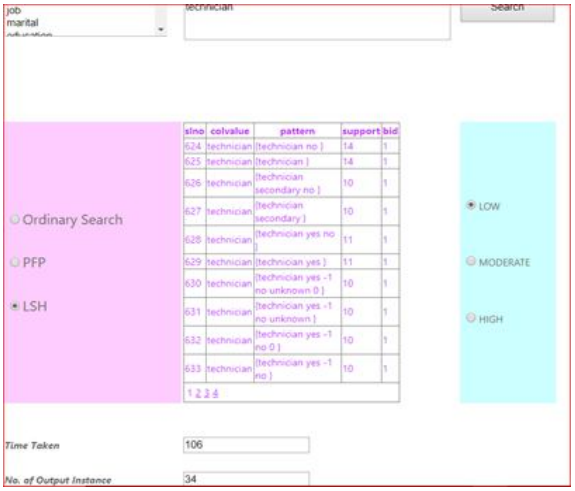

Figure 3. Result analysis

6. CONCLUSION

This paper, proposed parallel mining of frequent item set mining using map reduce. Initially, the direct marketing campaignsof data set is selected and uploaded . After that, the preprocessing technique is used to select the pivote and flist. . These data are classified depends on threshold values and the clustering algorithm is used to detect the support values among clusters to discover frequent item. Finally, these frequent item set is used to order the data and the output is merged as groups in the cluster of hadoop. The proposed method is provide the high accuracy, less executiontime and less memory usage. It provides the better result compared than existing. In future, the dimensionality of the nodes are increased in frequent item set mining on Hadoop.

REFERENCES

1.YalingXun, JifuZhang,Xiao Qin,**FiDoop-DP:Data Partitioning in frequent itemset mining on Hadoop Clusters** IEEE Transcations on Parallel and distributed system, vol28, jan.2017. https://doi.org/10.1109/TPDS.2016.2560176

2.M. J. Zaki, "**Parallel and distributed association mining**: A survey," IEEE Concurrency, vol. 7, no. 4, pp. 14–25, Oct. 1999. https://doi.org/10.1109/4434.806975

3.Pramudiono and M. Kitsuregawa, **Fp-tax: Tree structure based generalized association rule mining**, in Proc. 9th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery, 2004, pp. 60–63. https://doi.org/10.1145/1008694.1008704

4. M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, **Apriori-based frequent itemset mining algorithms on mapreduce,** in Proc. 6th Int. Conf. Ubiquitous Inform. Manag. Commun., 2012, pp. 76:1–76:8.

5. X. Lin, **Mr-apriori: Association rules algorithm based on mapreduce**, in Proc. IEEE 5th Int. Conf. Softw. Eng. Serv. Sci., 2014, pp. 141–144. https://doi.org/10.1109/ICSESS.2014.6933531

6.AnirbanMukhopadhyay, Senior Member, IEEE, UjjwalMaulik, Senior Member, IEEE, SanghamitraBandyopadhyay, Senior Member, IEEE, and Carlos ArtemioCoelloCoello, Fellow, IEEE "**A Survey of Multiobjective Evolutionary Algorithms for Data Mining**"ieee transactions on evolutionary computation, VOL. 18, NO. 1, FEBRUARY 2014

7.Mohamed Y. Eltabakh∗, Yuanyuan Tian∗, Fatma ¨Ozcan∗ 2Rainer Gemulla+, AljoschaKrettek#, John McPherson∗∗IBM Almaden Research Center, USA {myeltaba, ytian, fozcan, jmcphers}"**CoHadoop: Flexible Data Placement and Its Exploitation in Hadoop**"@us.ibm.com +Max Planck Institutf¨urInformatik, Germany rgemulla@mpi-inf.mpg.de #IBM Germany aljoscha.krettek@de.ibm.com

8.ShlomiDolev, Senior Member, IEEE, Patricia Florissi, Ehud Gudes, Member, IEEE Computer Society, Shantanu Sharma, Member, IEEE, and Ido Singer"**A Survey on Geographically Distributed ig-Data Processing using MapReduce**"DOI 10.1109/TBDATA.2017.2723473.