GENERATING FREQUENT ITEMSET IN BIG DATA USINGFIN ALGORITHM

R.PRAKASH, Dr.D.PRABHA.

Abstract-Emerging research area in Big data are handling issues like storing, searching, sorting, retrieving, securing, analyzing and visualizing are of immense importance. Customer behavior can be analyzed using Association rule mining which improves the organization thereby excelling business intelligence. To mine frequent itemset in big data we propose a novel FIN algorithm with map-reduce concept which helps in increase the performance by parallel processing. The parallel execution on identifying the frequent item set can performed by map-reducer function and using FIN algorithm. The performance of FIN is faster when compared with traditional algorithm such as Apriori, FP growth. This will also be applicable when it is perform in big data. The parallelization of the algorithm is increases the effectiveness. This proposed strategy can recommend the closely related products to the customers.

Index Terms-FIN, Parallelization, Frequent itemset, MapReduce.

1 INTRODUCTION

Discovering the patterns, associations and connections from the massive amount of information stored in datasets or commercial is known as Data mining. Big data refers to a collection of datasets which is so huge and complicated that it is infeasible to process by using traditional methods and available technologies. Even if some analytical approach can barely finish the work, it still takes a long time and the outcome might not be satisfactory. Data mining, using existing data to analyze the overall trend or predict a problem that may arise in the future, is undoubtedly the core area of big data research. It helps in finding patterns which are unseen in dataset. For extract interesting knowledge from big data, several mining algorithms are in implementations which are useful in different application. The association rules, identifying the frequent items is the key task of this area. Major Researcher focus on extraction of frequent patterns which can be used association rules. The rule which is not frequently occurring has more importance than the rule which occurs commonly.

Association rule learning is an efficient method for discovering interesting relations between variables in large databases. It is involved in identifying strong rules that can be discovered in databases using different measures of interestingness. The associations between the values of attribute in any item set are described by association rules [1]. Based on the concept of strong rules, association rules are introduced in discovering regularities between products which are in big data. An

Association rule is best expressed by means of the expression $X \rightarrow Y$. It means that for any occurrence of item X present in the big data there is relatively high probability item Y of occurring in the same tuple. Here antecedent is X and consequent is Y.

For example, {onions, potatoes} →{burger} rule found in the sales big data of a supermarket. It indicates that if a customer buys both onions and potatoes together, they are likely to buy burger. Such information can be effectively useful for decisions support in marketing activities such as, promotional pricing or product placements. The strength of such rule can be calculated using some measures such as support and confidence.

MapReduce is firstly introduced by Google and under map reduce programming framework one could easily implement parallel algorithm. It consists of two periods, map and reduce. The input data are split into small pieces and each small split of input data is delivered to a mapper for calculation, each mapper will shuffle its own intermediate output data (key, list (value)) pair to corresponding reducer. Once the reducer has collected all key-value pair, it will begin to run reduce function and calculate the output. Both map and reduce function here are specified and implemented by programmers.

Apache hadoop [4] is open source MapReduce

- *R.Prakash is currently pursuing masters degree program in Computer Science and Engineering in Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India, PH-7502958603. E-mail: 14MG015@skcet.ac.in.*
- *Dr.D.Prabha is Professor in Computer Science and Engineering in Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India, PH-7373350567. E-mail: prabha@skcet.ac.in.*

framework software firstly developed by Yahoo. Hadoop weakens the obstacles for programmers to implement MapReduce application. Based on the mechanism of MapReduce and Hadoop, application has strong fault tolerance and parallel calculation is guaranteed operating successfully.

2. LITERATURE SURVEY

2.1 PP Tree

This method used for generating frequent pattern using parallel and distributed algorithm [5] by PP Tree. This operates on distributed environment with advanced application using FP tree. The large database is partitioned into non overlapping blocks and each block is assigned to individual node in a distributed environment there by load is shared equally in all the nodes.

The Frequent patterns are constructed in five phases First phase each individual node scans the database once and generates a local PP through this it counts the support of each item distinctly. Second phase calculates the global count of each item. Third phase the local PP tree is reconstructed according to the global count of each item. Fourth phase potential global frequent item set is generated at each local site. Fifth phase the actual global frequent patterns are constructed. All the nodes are considered to be identical, so the load sharing is achieved by partitioning equally. The frequent pattern identified will be same as that are generated homogenous algorithm. Since the all local PP tree is send to the master processor all the data items are included in frequent pattern generation.

2.2Load balancing Frequent Pattern tree(LFP)

In the distributed environment the load are shared between the nodes. The database is equally partitioned and distributed to the nodes [3]. A local header table is generated at each node (Slave node SN) which scans the database once and counts the occurrence of each item. The mater node receives entire local tables from the slave nodes. A global table is prepared by master node. The global table has the number of occurrences of the all data items. The global table is transmitted to all the nodes. FP tree is constructed in each slave node based upon the local database and the global databases.

Each node of FP tree consists of item name, count and link. The item name is the item of the node represents, count represents the number of transaction occurred in the corresponding path. And link links to the next node. The depth and width of FP tree is calculated in each node. To avoid large database transfer SN preserves the data item in which loading is large. The load degree is decided at each node. Depending upon the load degree each SN is assigned a data item to mine. Then the FP tree is exchanged between the SN. Each node calculates frequent item sets in its FP tree. Finally MN collects all the frequent patterns.

2.3 Parallel FP-Growth (PFP)

In this method an algorithm to parallelize the FP-Growth algorithm on distributed machines [2]. At the core this algorithm partitions computation in such a way that each individual machine executes an independent group of mining tasks, through this it eliminates the computational dependencies each machine and thereby reducing the communication between the machines. This algorithm targets to overcome and reduce the challenges such as Storage Overheads, Complexity in distributing the computation & Communication complexities in the FP-growth algorithm. The algorithm achieves this through five phases, during the first phase sharing, the database is divided into equal parts and distributed into different machines, the second phase Parallel Counting, counts the support values of all items that appear in the database and stores it in list of frequent items, the third phase Grouping Items, divides the list of frequent items into groups, then each group is identified uniquely using unique id., the fourth phase Parallel FP-Growth, is the core phase which is sub divide into two sup-phases Mapper & Reducer phase, the Mapper phase identifies the group-dependent transactions using a mapper algorithm, followed by the reducer phase that generates FP-growth on the dependent transactions finally the aggregating phase aggregates the results arrived at the previous phase.

3. SYSTEM ARCHITECTURE

The architecture of the proposed system is described in fig: 1. Input data set D is divided into D1, D2, and D3 as a small chunk. Small chunk can be of required numbers which allow the big data to process parallel in MapReduce framework. The partitions are send to the mapper phase which is used to compute frequency of each items in D1, D2, and D3. Next involves in the reducer phase where the frequency of each itemset for D is calculated. Frequency items in D are used by the mapper for generating frequent itemset using FIN algorithm. The reduce phase combine the frequent itemset from D1, D2, D3.

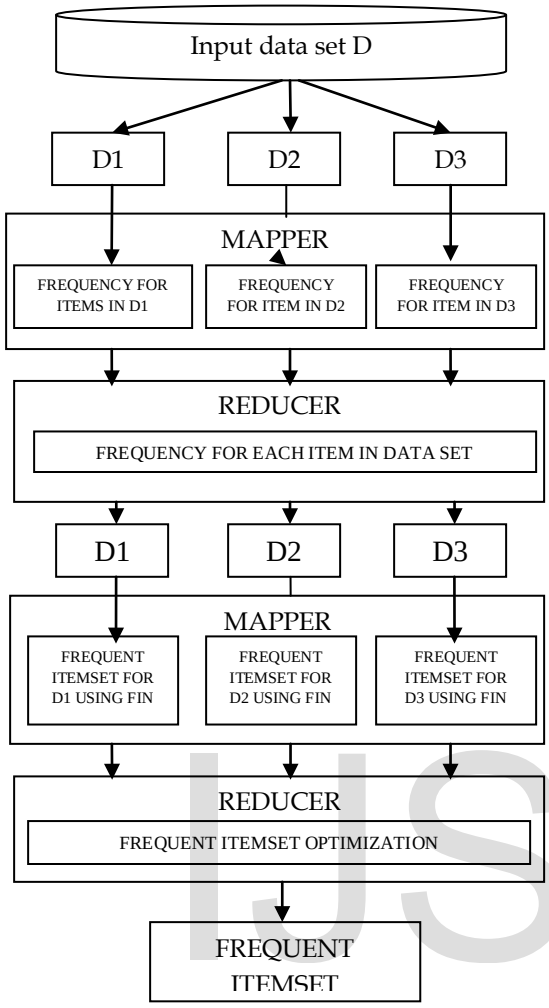

Fig. 1.Implementation of FIN on map reducer.

4. DESCRIPTION OF MODULES

In the proposed system the process involves three phases. The first stage involves collecting the input dataset and partitioning. The second stage involves implementing map reduce framework for counting the frequency of each item in the dataset and then third stage, implementation of FIN algorithm on map reduce.

4.1 Input dataset collection and partitioning

Frequent Itemset Mining Implementations (FIMI) repository provides several datasets. The datasets are Connect and Mushroom were often used in previous frequent itemset mining. The mushroom dataset contains various species and its characteristics of mushrooms and connect dataset is derived from game steps. These two real datasets are very dense.

Census dataset can also be used for planning public services. The planning includes health, transport, education, funds as well as in public business such as setup new shopping malls, factories or banks and marketing particular products. The census dataset attributes are age, work class, education, edu_ num, race, sex, marital, occupation, relationship, gain, loss, hours, country, and salary.

4.2 Implementation of map reduce for counting the frequency of item

The map function is implemented on each slave node. The slave nodes use the hadoop structure where data sets are stored. The mapper in slave nodes uses the local dataset D1, D2, D3 to count the item frequency. The frequency of each item in local dataset is calculated by slave nodes. The master reduce function combines the result set from all the slave nodes. The total count of each itemset is calculated by the master node. The infrequent itemsets are identified based on the support count. The items are removed from the result set by the reduce function. The removal of infrequent itemset based on the support count improves the performance.

4.3 Implementation of map reduce function for FIN algorithm

The frequency of each item for given dataset are obtained as a result set. The traditional FIN algorithm uses the result set for constructing the POC Tree [6]. By scanning the POC tree frequent itemset are found by constructing the conditional pattern base tree. Mapper function use the result set in generating the POC tree. The slave nodes generate the POC tree for each local database. The reducer function uses the local POC tree in constructing the frequent itemset.

5. CONCLUSION AND FUTURE WORK

The proposed algorithm aims in implementation of FIN algorithm on hadoop map-reduce function for big data. The main problem on using traditional algorithm on big data will take too much of time and space. The proposed system overcomes those problems by using parallelization based FIN algorithm on MapReduce framework. The experiment result on different dataset shows that the proposed algorithm is efficient. In future the FIN algorithm can be tested inthe incremental big data by using dynamic threshold value.

REFERENCES

[1] Agrawal.R,Imielinski.T, and Swami.A, "Mining association rules between sets of items in large database", ACM SIGMOD

Int'l Conference on Management of Data, pp. 207-216, 1993.

[2] Haoyuan.Li, Yi Wang, Dong Zhang, Ming Zhang, Edward Chang,"Pfp: parallel fp-growth for query recommendation", on Proceedings of the ACM conference on Recommender systems, pp. 107-114, 2008.

[3] Kun-Ming Yu, Jiayi Zhou, and Wei Chen Hsiao, "Load Balancing Approach Parallel Algorithm for Frequent Pattern Mining", Springer-Verlag Berlin Heidelberg, pp. 623–631, 2007.

[4] Shvachko.K, Kuang.H, Radia.S, &Chansler. R, "Thehadoop distributed file system. In Mass Storage Systems and Technologies (MSST)", IEEE 26th Symposium, pp. 1-10, 2010.

[5] Tanbeer.S.K, Ahmed.C.F, Jeong.B, "Parallel and Distributed Algorithms for Frequent Pattern Mining in Large Databases", IETE Tech Rev, pp. 55-65, 2009.

[6] Zhi-Hong Deng, Sheng-Long.Lv, "Fast mining frequent itemsets using Nodesets", Expert Systems with Applications, ELSEVIER, pp. 4505–4512, 2014