INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

CLUSTERING OF FREQUENT ITEMSET MINING OF BIG DATA WITH MAP REDUCED PLATFORM

MS. NEHA SHARAD BHAGWAT[1], PROF. DR. SAMEER S. PRABHUNE[2]

1. PG student Department of Computer Science and Engineering, Shri Sant Gajanan Maharaj College of Engineering, Shegaon.
2. PG student Department of Computer Science and Engineering, Shri Sant Gajanan Maharaj College of Engineering, Shegaon.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: FIM is most famous skill to extract chunk of data from the different cluster It is little bit of difficult in Big data, fortunately parallel programming already provide best tool to solve the problem. Apriori algorithm, have challenge to overcome the issue of chunk of data. FIM aim to enhanced size for handling more database it is not solved through traditional FIM . It is not possible to managed chunk of data in single machine for that we connect virtual machine as well as prepared clusters on different machine have different clusters. The output and runtime is under control by increasing clusters and render take little bit of time too parse the clusters .Map reduced solved the issue of database and its whole communication. Cluster generate huge dataset apriority processed to FIM through clusters and map reduced coding. In this paper, Dataset run with huge efficiency and higher speed clusters of big data comparatively.

Keywords: Hadoop, Frequent Item, Big Data, Clusters, Map Reduce, Nodes, HDFS

Corresponding Author: MS. NEHA SHARAD BHAGWAT

Access Online On:

www.ijpret.com

How to Cite This Article:

Neha Sharad Bhagwat, IJPRET, 2016; Volume 4 (9): 309-319

PAPER-QR CODE

INTRODUCTION

Big data used everywhere (in technology, science, businesses, marketing etc.) give rise to production and storage of huge amounts of data, not surprisingly, with knowledge and analysis of big data has become more important for both businesses and academics for all over world. Already, Frequent Item set Mining (FIM) has been an essential part of data analysis, apriori algorithm and data mining. Many new techniques have been invented on databases for frequent events. These techniques work well in practice on typical datasets, but they are not suitable for open dataset. Applying frequent item set mining to huge databases is problem itself. First of all, very huge databases do not stored into memory. In such cases, one solution is to use level wise market basket analyses based algorithms, such as the well-known Apriori algorithm. Parallel programming is becoming an essential to deal with the large amounts of data, which is produced and adopted more and more every day. Parallel programming have two main subcategories: shared memory and distributed (share nothing).On shared memory systems, all processing units can parallel use a shared memory area. Secondly, distributed systems are composed of processors that have their own internal main memories and communicate with each other by messages.

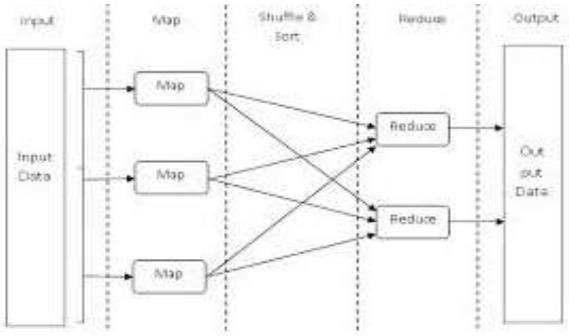
It is simple to accepted algorithms to shared memory parallelism in general, but they are usually not scalable enough for. Distributed item set systems, allow linear scalability for well adapted programs. However, it is hard to write or even adapt the programs for distributed systems, i.e., algorithmically solutions to common problems may have to be reinvented.

Moreover, current commercial and non-commercial systems and services improve the usability and availability for anyone. For example; the Mahout framework by Apache Foundation , which provides a straight forward usability for the most common machine learning techniques, can be set up and run on full-blown clusters on the cloud, having more than hundreds of processors in total, in less than 1 hour without prior experience with the system. We believe that, although the initial design principles of the Map Reduce framework do not fit well for the Frequent Item set Mining problem, it is important to provide Map Reduce with efficient and easy to use data mining methods, considering its availability and wide spread usage in industry.

Thanks to the scalability of distributed systems, not only in terms of technical availability but also cost, they are becoming more and more common.

It will get enhanced from the all high level programming like fortan and c. Data mining and Knowledge Discovery in Databases are important techniques to discover hidden information from huge datasets with lot's of property. Now a day Big Data has bloom in different areas such

as social networking, retail, web blogs, forums, online groups and many more. Frequent Itemset Mining is one of the important techniques of ARM. Aim of Frequent item set mining techniques is to open frequent itemsets from transactional databases. Apriori algorithm which generates frequent itemsets having frequency greater than minimum support and confidence given. It is not easy on single computer when dataset size increases. Enormous amount of work has been put forward to uncover frequent items. MapReduce has Map and Reduce functions; dataflow in MapReduce is shown in below figure.



MapReduce developed by Google along with Big data distributed file system is exploited to find out frequent itemsets from Big Data on large clusters. MapReduce uses parallel computing approach and HDFS is fault tolerant system.

Problem Statement

Big data tackeles with various task there are lot of relational database and statistic package it required thousands and more server. The various tackeling challenges faced in large data management include scalability, unstructured data, accessibility, real time analytics, shared, fault tolerance and many more. In addition to huge amount of data stored in different clusters, the types of data generated and stored it encode and decode the chunk of the documents. The chunk of data is growing day by day which is proportional too growth of the Market and details of Market. For the purpose of decision making in an organizations, the need of processing and scan large volume of data is increases data storage. As the Big data is the weak up technology that can be beneficial for the business, organizations, so it is important that various issues and challenges associated with this new technique should bring out into light. The two main problems regarding big data are the storage capacity and the processing of the data as well as the speed of scanning of huge data.

Let *I* be a set of items, *I* = {i1,i2,i3,…,in}, X is a set of items, X = {i1,i2,i3,…,ik} I called k -itemset.

A transaction T = {t1,t2, t3, …,tm}, denoted as T = (tid, I) where tid is transaction ID. TÎD, where D is a transactional database. The cover of itemset X in D is the set of transaction IDs containing items from X.

Cover(X, D) = {tid | (tid, I) ID, X I I}

The support of an itemset X in D is count of transactions containing items from X.

Support (X, D) = |Cover(X, D)|

An itemset is called frequent when its absolute minimum support threshold s abs, with 0 ≤ s abs ≤|D|.

Partitioning of transactions into set of groups is called clustering. Let s be the number of clusters then {C1, C2, C3… Cs} is a set of clusters from {t1,t2, t3, …,tm} , where m is number of transactions. Each transaction is assigned to only one clusters. After that map reduced the document and get the result it is simple statement of apriori algorithm related with mining rule. K-means algorithm starts with one cluster and assigns each transaction to clusters. The HDFS name space is a hierarchy of files and directories. Files and directories are represented on the Name Node by inodes, which loaded attributes like permissions, modification and access times, namespace and disk space quotas. Hadoop Distributed File System is a block structured file system where each file content is spread into large blocks sets(typically 128 megabytes, but user selectable file by file) and each block of the file is independently send at multiple DataNodes.

Literature Survey

Initially, there are three main frequent Itemset mining algorithms that run in single node. In Apriorialgorithm loop k produces frequent itemsets with length k. By using the property and input of k loop, loop k+1 calculate candidate itemsets. Property is: any subset in one frequent itemset must also be frequent. FP-Growth algorithm creates an FP-Tree by two scan of the whole dataset and then frequent itemsets are mined from frequent pattern tree. Eclat algorithm transposes the whole dataset into a new table. In this new table, every row contains list of sorted transaction ID of respective item. In last frequent itemsets are extracted by intersecting two transaction lists of that item. The critical issue about the Big data is the privacy and security. Big data samples describe the review about the atmosphere, biological science and research. Life sciences etc. By this paper, we can conclude that any organization in any industry having big data can take the benefit from its careful analysis for the problem solving

purpose. Using Knowledge Discovery from the Big data easy to get the information from the complicated data sets

The overall Evaluation describe that the data is increasing and becoming complex. The challenge is not only to collect and manage the data also how to extract the useful information from that collected data.

According to the Intel IT Center, there are many challenges related to Big Data which are data growth, data infrastructure, data variety, data visualization, data velocity.

Garlasu, D.; Sandulescu, V. ; Halcu, I. ; Neculoiu, G. ;(17-19 Jan. 2013),"A Big Data implementation based on Grid Computing", Grid Computing offered the advantage about the storage capabilities and the processing power and the Hadoop technology is used for the implementation purpose. Grid Computing provides the concept of distributed computing. The benefit of Grid computing center is the high storage capability and the high processing power. Grid Computing makes the big contributions among the scientific research, help the scientists to analyze and store the large and complex data.

Mukherjee, A.; Datta, J.; Jorapur, R.; Singhvi, R.; Haloi, S.; Akram, W. (18-22 Dec. 2012) "Shared disk big data analytics with Apache Hadoop" Big data analytics define the analysis of large amount of data to get the useful information and uncover the hidden patterns. Big data analytics refers to the Mapreduce Framework which is developed by the Google. Apache Hadoop is the open source platform which is used for the purpose of implementation of Google's Mapreduce Model.

Apache Hadoop is an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license. It supports the running of applications on large clusters of commodity hardware. Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.

The Hadoop framework transparently provides both reliability and data motion to applications. Hadoop implements a computational paradigm named Map Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both map/reduce and the distributed file system are designed so that node failures are automatically handled by the framework.

In a larger cluster, the HDFS is managed through a dedicated Name Node server that hosts the file system index, and a secondary Name Node that can generate snapshots of the name node's memory structures, so preventing file system corruption and reducing loss of data. Similarly, job scheduling can be managed by a standalone Job Tracker server. In clusters where the Hadoop Map Reduce engine is deployed against an alternate file system, the Name Node, secondary Name Node and Data Node architecture of HDFS is replaced by the file system-specific equivalent

In this the performance of SF-CFS is compared with the HDFS using the SWIM by the facebook job traces .SWIM contains the workloads of thousands of jobs with complex data arrival and computation patterns.

Aditya B. Patel, Manashvi Birla, Ushma Nair (6-8 Dec. 2012) "Addressing Big Data Problem Using Hadoop and Map Reduce" reports the experimental work on the Big data problems. It describe the optimal solutions using Hadoop cluster, Hadoop Distributed File System (HDFS) for storage and Map Reduce programming framework for parallel processing to process large data sets.

Real Time Literature Review about the Big data According to 2013, facebook has 1.11 billion people active accounts from which 751 million using face book from a mobile. Another example is flicker having feature of Unlimited photo uploads (50MB per photo), Unlimited video uploads (90 seconds max, 500MB per video), the ability to show HD Video, Unlimited storage, Unlimited bandwidth. Flickr had a total of 87 million registered members and more than 3.5 million new images uploaded daily.

PROPOSED SYSTEM

After studying, we can propose such problem which will use try structure and grouping system but it will locally prune dataset at base level as well as at next level by grouping node into cluster. The cluster of the dataset is more pruned toward the size not easily get from the structural formation. This algorithm will have less applicable as well as message passing over all. From the compared matrix pattern one can say that a more work should be carried out for making the algorithm more efficient in the mode of reducing message passing overhead as well as problem of FIM. Our proposed algorithm will follow the steps starting from dividing the data nodes in clusters based on any namly clustering property. The nodes that are geographically near area bounded in one cluster. As well as the more cluster formation is necessary for performin the prunig through thr clusters. First the local itemset for each node is essential for the size of data and the frequently divided the information into different clusters. The same

method is continued rotation for all nodes in clusters. At the end we will find intermediate global itemset for specific cluster.

For dissertation of our propose algorithm hadoop distributed file system (HDFS) can be used. It is used to perform task operation on unstructured data, the data in form of logs and generation of analysis of massive pruning. Data mining performs operation on structured data that is stored in form of tables in that row and column. Apache big data is platform for the distributed parallel processing of the chunk of data all over the programming method.

Hadoop components:

1) HDFS- That is used for storage in distributed platform.

2) Map Reduce-This approach is used for working method of programming in distributed environment.

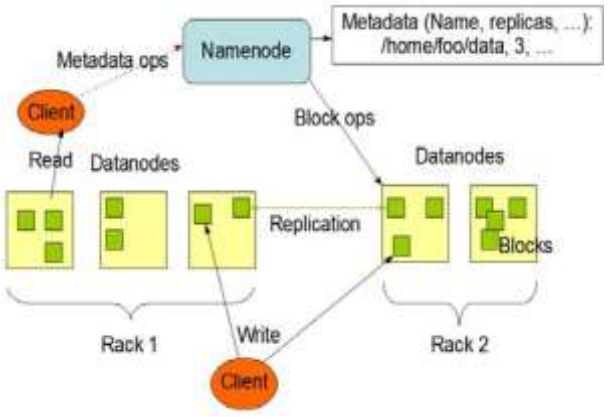


Figure 2: HDFS Architecture

1.1 Hadoop Components

HDFS: It is used for storage in distributed environment ˆ Map Reduce: It is used for processing in distributed environment

Name node: They are the master of system that maintains and manages the blocks which are present on data node and its implementation requires expensive hardware

Data node: They are slaves which are deployed on each machine and provide actual storage and are responsible for serving read and write request for client The proposed system can be deployed in distributed environment after installation of apache hadoop. The system can be

implemented in single node cluster environment in which one Name node and one Data node is there. Here single machine configuration is required but the problem with this approach is if name node fails then whole system fails same as if data node fails then also system crashes.

Thus, by implementing the proposed system in multi node clusters environment in which one name node and many data nodes are available. Here multiple machines can be configured together. Benefit of this approach is fault tolerance is achieved by having replica of name node as supporting name node and replication of data node can be achieved by name node as it stores the two or more copies of data node.

The proposed methodology can be divided into two phases namely:

1) Local Communication

2) Grouping Nodes

1.2 Local Communication

In this phase every node maintains a tree data structure. Nodes synchronize at end of each node so that all the nodes have same data at end of each round. The pruning operation is performed after the communication between the nodes is terminated. Hence as result the node maintains the similarity among them.

1.3 Grouping Nodes

After the first phase, the local candidates along with their corresponding supports are sent to the central nodes. The central node at each group aggregates the candidate item sets and their supports for their groups. This phase incorporates to the concept of parallel processing as the central nodes from each group enter into the gossip for determination of the global support. This global support is taken into account for pruning the infrequent item set. We can process these files parallel by placing the les on HDFS and running a Map Reduce job. The processing time theoretically improves by the number of nodes in the cluster. The advantage of the Map Reduce abstraction is that it permits a user to design his or her data-processing operations without concern for the inherent parallel or distributed nature of the cluster itself.

The HDFS stores the input data as it can store huge chunks of data. HDFS also helps in data localizations for the map/reduce task.

The input data is divided into parts and allotted to the mapper. The output from the mapper is key-value pair. The key is itemset and value is support count then output is passed the

combiner. It combines all the count value related to a particular item which is known as key. The result from this is taken in by the reducer which combines and sum up of the values corresponding to an item.

After getting the sum of values for an item the reducer is checked whether the value exceeds the given threshold value. If the value exceeds the threshold value for an item then the item with support count is written as the output. The item is discarded if it is less than the minimum support threshold value thus generating frequent-1 itemset containing a set of one item pattern. The same process is repeated for generation of frequent itemset in all chunks of HDFS and then the results are combined in intermediate map reduce function the algorithm is applied again and final results will be generated.

Apriori Algorithm

Apriori is the first frequent itemset mining algorithm which has been put forward by Agarwal. Transactional database has transaction identifier and set of items presenting transaction.

Apriori algorithm scans the horizontal database and finds frequents items of size 1-item using minimum support condition. From these frequent items discovered in iteration 1 candidate itemsets are formed and frequent itemsets of size two are extracted using minimum support condition. This process is repeated till either list of candidate itemset or frequent itemset is empty. It requires repetitive scan of database. Monotonicity property is used for removing frequent items.

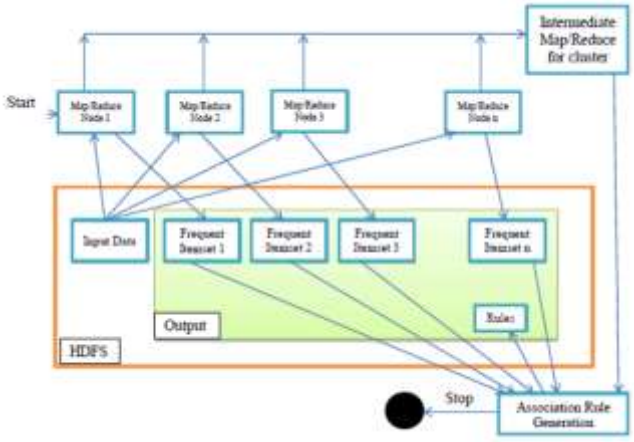
Eclat Algorithm

Eclat algorithm is proposed by Zaki et al. which works parallel on vertical database. In that we are using distributed version of Eclat all processing partition on their memory .we are using frequent pattern transaction on itemset using support and confidence. The list of each item is calculated and combination of original list of items is used for extracting frequent itemset of size k+1.

K-means Algorithm

The k-means algorithm is well known technique of clustering which takes number of clusters as input, in different points are chosen as center of gravity and distance measures to calculate distance of each point from center of gravity. In that we are received the part of document which support the count. The reducers combined all clusters n give report to global frequencies. This frequent itemset goes into mapper and search the document. Each point is

assigned to only one cluster based on high intra-cluster similarity and low inter-cluster similarity.



Flow diagram of the system

CONCLUSION

Cluster of Frequent itemset mining for big data with map reduced plat form is a new essential technology topic because it is widely applied in human real world to find frequent itemsets and to mine document patterns and behavior. In this paper comparative study of number of FIM technique we are going too presented. Clusters of FIM for big data on the platform of map reduced process are both memory and compute intensive. In the couple of year there is development of computer application to solver the memory problem. Map reduced solved the issue of database and its whole communication. Cluster generate huge dataset apriority processed to FIM through clusters and map reduced coding. In this paper, Dataset run with huge efficiency and higher speed clusters of big data comparatively. In current search we are trying too reduced the speed in huge amount of data.

REFERENCES

1. M. Bagheri, S. Mirian Hosseinabadi, H. Mashayekhi and J. Habibi, ƒMining Distributed Frequent Itemset using Gossip Based Protocol,, in Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC), 2012.
2. Sandy Moens, Emin Aksehirli and Bart Goethals,\ Frequent Itemset Mining for Big Data", 2013 IEEE International Conference on Big Data..
3. L. Brankovic and V. Estivill-Castro, ƒPrivacy issues in knowledge discovery and data mining ,, in Proc. Austral. Inst. Comput. Ethics Conf., 1999.

4. Wei Fan and Albert Bifet, ƒBig Data: Current Status and Forecast to the Future ,, in Special Interest Group on Knowledge Discovery and Data Mining Explorations, 2012.
5. R. Agrawal, T. Imielinski, and A. Swami, ƒMining association rules between sets of items in large databases in Proc. Special Interest Group on Management Of Data, 1993.
6. Sung-Hwan Kim, Jung-Ho Eom and Tai-Myoung Chung, ƒData Security Hardening Methodology Using Attributes Relationship ,, in Information Science and Applications, 2013.
7. M. Z. Ashra, D. Taniar and K. Smith, ƒDistributed Association Rule Mining ,, in IEEE distributed system online, 2004.
8. E Ansari, G.H. Dastghaibifard and M. keshatkaran, ƒDistributed Trie Frequent Itemset Mining ,, in International MultiConference of Engineers and Computer Scientists, 2008.
9. Tirumala Prasad and Dr. MHM Krishna Prasad, ƒDistributed Count Association Rule Mining Algorithm ,, in International Journal of Computer Trends and Technology, 2011.
10. Mohammed J. Zaki, ƒParallel and Distributed Association Mining: A Survey ,, in Association for Computing Machinery, 1999.
11. M. A. Mottalib, Kazi Shamsul Are n, Mohammad Majharul Is-lam, Md. Arif Rahman, and Sabbeer Ahmed Abeer, ƒPerformance Analysis of Distributed Association Rule Mining with Apriori Algorithm ,, in International Journal of Computer Theory and Engineering, 2011.
12. Lai Yang, Zhongzhi Shi, Xu L.D., Fan Liang and I. Kirsh, ƒDHTRIE Frequent Pattern Mining on Hadoop using JPA ,, in IEEE International Conference on Granular Computing, 2011.
13. Tao Xiao, Chunfeng Yuan and Yihua Huang, ƒPSON: A Parallelized SON Algorithm with MapReduce for Mining Frequent Sets ,, in Fourth International Symposium on Parallel Architectures, Algorithms and Programming, 2011.
14. Suhasini A. Itkar and Uday V. Kulkarni, ƒDistributed Algorithm for Frequent Pattern Mining using HadoopMap Reduce Framework ,, in Association of Computer Electronics and Electrical Engineers, 2013.
15. Assaf Schuster and Ran Wolff, ƒ Communication Efficient Distributed Mining of Association Rules ,, in Association for Computing Machinery Special Interest Group on Management of Data, 2001.
16. David W. Cheung, Jiawei Han, Vincent T. Ng, Ada W. Fu and Yongjian Fu, ƒA Fast Distribution Algorithm for Mining Association Rule,, in Association for Computing Machinery, 1996.
17. Ferenc bodon, ƒA Trie-based Apriori Implementation for Mining Frequent Item sequences ,, in Association for Computing Machinery, 2005.
18. Gang Wu, Huxing Zhang, Meikang Qui, Zhong Ming, Jiayin Liand Xiao kin, ƒA Decentralized Approach for Mining Event Correlations in Distributed System Monitoring ,, in Journal of Parallel and Distributed Computing, 2013.