INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 3 ISSUE 3 – MARCH 2015 – ISSN: 2349 – 9303

A Survey on Approaches for Frequent Item Set Mining on Apache Hadoop

Ahilandeeswari.G.[1]
[1]Research Scholar,
Department of Computer Science,
NGM College, Pollachi, India,
ahilaplatinum@gmail.com

Dr. R. Manicka Chezian[2]
[2] Associate Professor,
Department of Computer Science,
NGM College, Pollachi, India,
chezian_r@yahoo.co.in

Abstract— In data mining, association rule mining is one of the major techniques for discovering meaningful patterns from large collection of data. Discovering frequent item sets play an important role in mining association rules, sequence rules, web log mining and many other interesting patterns surrounded by complex data. Frequent Item set Mining is one of the classical data mining tribulations in most of the data mining applications. Apache Hadoop is a major innovation in the IT market place last decade. From modest beginnings Apache Hadoop has become a world-wide adoption in data centers. It brings parallel processing in hands of average programmer. This paper presents a literature analysis on different techniques for mining frequent item sets and frequent item sets on Hadoop.

Index Terms— Data mining, Association rules, Frequent item set, Apache Hadoop.

1. INTRODUCTION

GENERALLY, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of numeral analytical tools for analyzing data. It allows users to analyze data from several different extent or angles, categorize it, and review the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. One of the most important areas of data mining is association rule mining; it is a task is to find all items or subsets of items which repeatedly occur and the relationship between them. This is achieved in two main steps: (i) finding frequent item sets and (ii) generating association rules. Frequent Item set Mining (FIM) tries to discover information from database based on frequent occurrences of an event according to the minimum frequency threshold provided by user. Association Rule mining is one of the most important data mining tools used in many real life applications. It is use to expose unexpected relationships in the data. In this paper, we will discuss the problem of computing association rules within a projected partitioned database. We assume homogeneous databases. All sites have the same schema, but each site has information on different entities. The goal is to produce associate ion rules that hold globally, while limiting the information shared about each site to maintain the privacy of data in each site. Association rule mining finds interesting associations and/or correlation relationships among large sets of data items. Association rules show attributes value conditions that occur frequently together in a given dataset. Frequent item sets are appearing in a data set with frequency no less than a user-specified threshold. For example, a set of items, such as milk and cereal that appear frequently together in a transaction data set is a recurrent item set.

2. FREQUENT ITEM SET

There are several algorithms for finding repeated patterns. Association rule mining first mooted by Agarwal has now become one of the main pillars of data mining and knowledge discovery tasks [1]. Intuitively, a set of items that appears in many baskets is assumed to be “frequent.” To be formal, we assume present is a number *s*, called the *support threshold*. If *I* is a set of items, the *support* for *I* is the number of baskets for which *I* is a subset. We say *I* is *frequent* if its support is *s* or more [2].Let I = {I1, I2, …Im} be a set of items. Let D, the task related data, be a set of database transactions where each transaction T is a set of items such that T C J.

Support (A=>B) = P (AUB)
Confidence (A=>B) = P (B/A)

Frequent sets play a primary role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems of all these. The classification group of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in Data Mining. The original inspiration for searching frequent sets came from the need to analyze supermarket transaction data, that is, to inspect customer behavior in terms of the purchased products [3]. Frequent sets of products describe how often items are purchase together.

2.1. Apache Hadoop

The Apache™ Hadoop® develops open-source software for reliable, scalable, distributed computing [6]. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. Hadoop is a, Java-based programming framework that supports the processing of large data sets in a distributed computing environment and is part of the Apache project sponsored by the Apache Software Foundation. Hadoop was originally conceived on the basis of Google's Map Reduce, in which an application is broken down into numerous small parts [9]. It is proposed to scale up from single servers to thousands of machines, each present local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is planned to detect and handle failures at the application layer, so delivering a highly-available. Service on top of a group of computers, each of which possibly prone to failures. Hadoop skeleton is popular for HDFS as well as Map Reduce. The Hadoop Ecosystem also contains different projects which are discussed below [7] [8]: The Hadoop includes these modules:

- **Hadoop Common**: The common utilities that hold the other Hadoop modules. It contains libraries with utilities needed by other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data. HDFS is the Hadoop file system and comprises two major components: namespaces and blocks storage service. The namespace service manages operations on files and directories, such as creating and modifying files and directories. The block storage service implements data node cluster management, block operations and duplication.
- **Hadoop YARN**: A framework for job development and cluster resource management. YARN is a resource manager that was formed by separating the processing engine and resource management capabilities of Map Reduce as it was apply in Hadoop 1. YARN is often called the operating system of Hadoop because it is in charge for managing and monitoring workloads, maintaining a multi-tenant surroundings, implementing security controls, and managing high accessibility features of Hadoop.
- **Hadoop Map Reduce**: A YARN-based system for parallel processing of vast data sets. The Hadoop Map Reduce framework consists of one Master node termed as Job Tracker and many Worker nodes called as Task Trackers.

HDFS components:

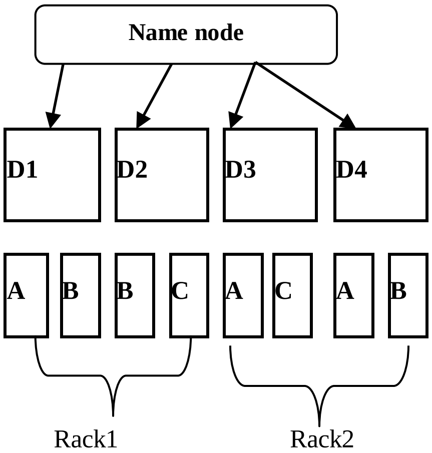

Fig 1: HDFS architecture of Apache Hadoop

Hadoop Distributed File System [3] [4] is an open-source file system that has been deliberate specifically to handle large files that traditional file system cannot handle. The large amount of data is split, replicated and scattered on multiple machines. The replication of data facilitates rapid computation and reliability. That is why HDFS can also be called as self-healing distributed file system meaning that, if a particular copy of the data gets corrupt or more specifically to say if the Data Node on which the data was residing fails then replicated copy can be used. This ensures that the on-going work continues without any disruption. HDFS is a distributed file system that provides high-performance access to data across Hadoop clusters. Like other Hadoop-related technologies, HDFS has become a key tool for managing pools of big data and supporting big data analytics application.

Name node:

They are master of system. That maintains and manages the blocks which are present on data node. Its execution requires expensive hardware. The Name node actively monitors the number of replicas of a block. When a copy of a block is lost due to a Data Node failure or disk failure, the Name Node creates another replica of the block. The Name Node maintains the namespace tree and the mapping of blocks to Data Nodes, holding the entire namespace image in RAM.

Data node:

They are slaves which are deployed on each machine and provide actual storage space. It is responsible for serving read and write request for client. The implementation strategy followed is mentioned as below: Future work can be deployed in dispersed environment after installation of apache hadoop. By implementing our propose system 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. Better way is to implementing our propose 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 duplication of data node can be achieved by name node as it stores the two or more copies of data node.[10] The Name Node does not unswervingly send requests to Data Nodes. It sends commands to the Data Nodes by replying to heartbeats sent by those Data Nodes. The instructions include commands to: replicate blocks to other nodes, remove local block replicas, re-register and send an immediate block report, or shut down the node.

3. FREQUENT ITEM SET MINING TECHNIQUES

Frequent patterns, such as frequent item sets, substructures, sequences term-sets, phrase sets, and sub graphs, generally exist in real-world databases. Identifying recurrent item sets is one of the most important issues faced by the knowledge discovery and data mining society. Recurrent item set mining plays an important role in several data mining fields as association rules warehousing, correlations, clustering of high-dimensional biological data, and classification. The frequent item set mining is motivated by problems such as market basket analysis. A row in a market basket database is a set of items purchased by customer in a transaction. An association rule mined from market basket database states that if some items are purchased in transaction, then it is likely that some other items are purchased as well. As frequent data item sets mining are very important in mining the association rules. Therefore there are various techniques proposed for generating frequent item sets so that association rules are mined efficiently. The mining process of frequent item sets (sets of items)can be started from transactional, relational data sets or other kinds of frequent patterns from other kinds of data sets.

There are number of algorithms used to mine repeated item sets. The most important algorithms are briefly explained here. The algorithms vary in the generation of candidate item sets and support count. The approaches of generating frequent item sets are divided into basic three techniques:

a) Horizontal outline based data mining techniques.
b) Vertical outline based data mining techniques.
c) Projected record based data mining techniques.

Dataset organizations can be processed straight or perpendicular. For several decades and especially with the pre-eminence of relational database systems, data is almost always formed into parallel record structure and then processed up and down. In a flat enumerated data organization, each transaction contains only items positively associated with a customer purchase. In a straight layout, the database is organized as a set of rows, with each row representing a customer's transaction in terms of the items that are purchased in the transaction. There is an alternative approach to this data layout such as perpendicular layout. It consists of each item associated with a column of values representing the transaction in which it is present. It has smaller effective database size, compact storage of the database and better support of dynamic database.

a) Horizontal outline based data mining techniques:

The algorithms used in the straight outline based data mining technique are Apriori Algorithm, Direct Hashing and Pruning (DHP) Algorithm, Partitioning Algorithm, Sampling Algorithm, Dynamic Itemset Counting (DIC) and Continuous Association Rule Mining Algorithm (CARMA).

b) Vertical outline based data mining techniques:

Eclat algorithm is used for perpendicular outline based data mining technique this algorithm is also used to execute item set mining. The basic idea for the eclat algorithm is use tidset intersections to compute the support of a candidate itemset avoiding the generation of subsets that does not exist in the prefix tree.

c) Projected record based data mining techniques:

Tree projected record based approaches use tree structure to store and mines the item sets. The projected based layout contains the record id separated by column then record. Tree Projection algorithms based upon two kinds of ordering breadth-first and depth-first. This type of database uses divide and conquer strategy to mine item sets therefore it counts the support more efficiently than Apriori based algorithms. FP-Growth Algorithm and H-mine Algorithm are used for projected layout based data mining technique.

III Related work: Projected database based techniques:

In this section we assess Apriori Algorithm, FP growth algorithm And H-mine algorithm

Apriori Algorithm: Apriori proposed by [1] is the fundamental algorithm. It searches for frequent itemset browsing the lattice of itemsets in breadth. The database is scanned at each level of lattice. Additionally, Apriori uses a pruning technique based on the properties of the item sets, which are: If an item set is frequent, all its sub-sets are repeated and not need to be considered.

The working of Apriori algorithm is fairly depends upon the Apriori property which states that" All nonempty subsets of a frequent itemsets must be frequent"[1] (table 1).

Table 1: Apriori Algorithm Parameters

Storage structures	**Array Based**
Technique	Use Apriori property and join and prune method
Memory utilization	Due to large amount of candidate are produced so required large memory space
Database	Suitable for space database as well as dense datasets
Time	Execution time is more as time wasted in producing candidates at every time

FP Growth Algorithm: FP growth or frequent pattern growth algorithm uses a data structure (FP tree or prefix tree) to pile up the entire database in a dense form. By storing the database in a tree like structure, the costly step of database scan is avoided. FP growth uses a divide and conquer approach which is a tree based frequent pattern mining method used to avoid costly process of candidate generation. (table 2).

Table 2: FP Growth Algorithm Parameters

Storage Structures	**Array Based**
Technique	It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support
Memory Utilization	Due to compact structure and no candidates generation require less memory
Database	Suitable for large and medium datasets
Time	Execution time is large due to complex compact data structure

H-mine Algorithm H-Mine is an algorithm for discovering frequent itemsets from a transaction database developed by Pei et al.[3] in 2007. They proposed a simple and novel data structure using hyper-links, H-struct, and a new mining algorithm, Hmine, which takes advantage of this data structure and dynamically adjusts links in the mining process. A distinct feature of the proposed method is that it has a very limited and precisely predictable main memory cost and runs very quickly in memory-based settings. (table 3).

Table 3: H-mine Algorithm Parameters

Storage structures	**Array Based**
Technique	Partition the database for finding the local frequent itemset first
Memory utilization	Each partition is easily occupy in main memory.
Database	Suitable for large databases
Time	Execution time is more because of Finding locally frequent than Globally frequent.

4. FREQUENT ITEMSET ON APACHE HADOOP

As association rule mining has attracted a major amount of research awareness, many data mining techniques have been proposed in the past decades. Frequent item set on Apache Hadoop helps to shrink time and cost, decrease the number of database scan, works on big data set, scattered framework and availability to overcome failure. Anjan Kumar [4] have implemented Apriori algorithm on Apache Hadoop platform. Contrary to the believe that parallel processing will take less time to get Frequent item sets, they experimental examination proved that multi node Hadoop with differential system configuration (FHDSC) was taking more time. The reason was in way the data has been portioned to the nodes. Dhamdhere Jyoti L [6] presented a new map-reduce based algorithm addressing problem of mining frequent itemsets using dynamic workload management during a block-based partitioning.

Due to limitations of main memory, Frequent Item set Mining becomes ineffective on large databases. This problem can be solved by using Apriori algorithm [7], where database is scanned multiple times for frequency count of each size of candidate item sets. Unluckily, single machines are unable to fulfill the memory requirements for handling the complete set of candidate item sets. Also existing algorithms care to organize the output and runtime by increasing the minimum frequency threshold, involuntarily reducing the number of candidate and frequent item sets [8].

5. CONLCUSION

The overall goal of the frequent item set mining process helps to form the association rules for further use. Association rules prove to be the most effective technique for frequent pattern matching over a decade. This paper gives a brief survey on different approaches for mining frequent itemsets and also frequent item set mining in Apache Hadoop.

REFERENCES

[1] R.Agarwal and R. Srikant,"Fast Algorithms for Mining Association Rules.", International Conference on very large Databases, proc.20th , pp 487-499,june 1994.

[2] Agrawal, R., Shafer, J.C., "Parallel mining of association rules.", IEEE Transactions on Knowledge and Data Engineering, Volume.8, no.6, pp.962-969, Dec 1996.

[3] Ramesh C. Agarwal, Charu C. Agarwal, V. V. V. Prasad, "A Tree Projection Algorithm for Generation of Frequent Item Sets.", Journal of Parallel and Distributed Computing, Volume 61, Issue 3, pp. 350-371, March 2001.

[4]Anjan K Koundinya, Srinath N K, ,K A K Sharma, Kiran Kumar, Madhu M ,and Kiran U Shanbag ,"Map Reduce Design and implementation of Apriori algorithm for handling colounius data Mining.",An International Journal Advanced Computing (ACIJ),Vol.3, No.6, November 2012.

[5]Brin.s Motwani,R,Ullman.J.D and S. Tsur,"Dynamic itemsets counting and implication rules for market basket analysis.", ACM-SIGMOD ,pages 255-264, May 1997.

[6]Ms. Dhamdhere Jyoti L., Prof. Deshpande Kiran B. "An Effective Algorithm for Frequent Itemset Mining on Hadoop.", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 8, August 2014.

[7]Ferenc Kovacs and Janos Illes "Frequent Itemset Mining on Hadoop.",IEEE 9th International conference on Computational Cybrnetics, Volume 2 Issue 4, june 2013.

[8]Ms. Dhamdhere Jyoti L., Prof. Deshpande Kiran B. "A Novel Methodology of Frequent Itemset Mining on Hadoop.", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 8, August 2014.

[9]Tao Gu, Chuang Zuo, Qun Liao, Yulu Yang and Tao Li, "Improving Map Reduce Performance by Data Prefetching in Heterogeneous or Shared Environments.", International Journal of Grid and Distributed Computing,Vol.6, No.5, pp.71-82, june2013.

[10]A. Swami, T. Imielienski, R. Agrawal," Mining association Rules between Sets of Items in Large databases.", ACM Press, pp 207–216, july 1993.

Authors Biography

Ms Ahilandeeswari.G received MCA degree From Anna University, Chennai in 2006 and 2009 respectively. Currently a Research Scholar at Department of Computer Science, NGM College, Pollachi. Her research interest lies in the area of Data Mining

DR. R.Manicka chezian received his M.Sc., degree in Applied Science from P.S.G College of Technology, Coimbatore, India in 1987. He completed his M.S. degree in Software Systems from Birla Institute of Technology and Science, Pilani, Rajasthan, India and Ph D degree in Computer Science from School of Computer Science and Engineering, Bharathiar University, Coimbatore, India. He served as a Faculty of Maths and Computer Applications at P.S.G College of Technology, Coimbatore from 1987 to 1989. Presently, he has been working as an Associate Professor in N G M College (Autonomous), Pollachi under Bharathiar University, Coimbatore, India since 1989. His research focuses on Network Databases, Data Mining, Distributed Computing, Mobile Computing, Real Time Systems and Bio-Informatics.