International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Review on Apriori Based Frequent Item Set Mining Using Various Techniques

Ravishankar Sahu[1], Abhishek Badholia[2]
[1](*M. Tech Computer Science & Engineering, Central College of Engineering and Management, Raipur, India*)
[2](*Assistant Professor, Computer Science & Engineering, Central College of Engineering and Management, Raipur, India*)

Abstract: ***Frequent Item set Mining is a standout amongst the most prominent systems to extract knowledge from data. Be that as it may, these mining strategies turn out to be more risky when they are connected to Big Data. Luckily, recent developments in the field of parallel programming give numerous devices to handle this issue. In any case, these instruments accompany their own technical difficulties, for example, balanced data distribution as well as inter-communication costs. In this paper, we are showing a point by point survey of Hadoop, which helps in putting away data and parallel processing in a distributed situation. Here we have surveyed different Frequent Item set Mining method on parallel and distributed condition. The point of this paper is to show a correlation of various frequent item set mining methods and help to create proficient and versatile frequent item set mining strategies.***

Keywords: ***Big Data, Data mining, Distributed data mining, Frequent Itemset Mining, Hadoop.***

I. INTRODUCTION

Data mining is the procedure of extraction of data from vast databases and it is an effective new innovation having an extraordinary potential to help researchers and additionally organizations on the most imperative data in their data warehouse [1]. Data mining apparatuses are utilized to anticipate the future patterns and practices in this way enabling organizations to settle on learning driven choices.

Frequent item set mining in distributed condition is an issue and should be performed utilizing a distributed algorithm that does not require trade of crude data between the taking an interest locales. Distributed data mining is the way toward mining data in distributed data sets. As indicated by Zaki in [2], two predominant structures exist in the distributed conditions i.e., distributed memory architecture (DMA) and shared memory architecture (SMA).

In DMA, every processor has its own particular database or memory and approaches it. DMA frameworks access to other nearby databases is conceivable just by means of message trade. DMA offers a straightforward programming technique, however restricted data transfer capacity may decrease the adaptability. Then again, in SMA every processor has immediate and measure up to access to the database in the framework. In this way, parallel projects on such frameworks can be actualized effectively.

An arrangement of items in a database is known as item set. On the off chance that the occurrence of items in a specific exchange is frequent, it is called as frequent item set and the help (or check) of frequent item set is more noteworthy than some client indicated least help. Frequent Pattern Growth (FP-Growth) algorithm is a standout amongst the most prevalently utilized data mining approach for finding frequent item sets from vast datasets [3]. Be that as it may, the fundamental test looked by different frequent item set mining algorithm is its execution time in distributed environment.

Distributed sources of voluminous data have made the requirement for distributed data mining. The regular data mining algorithms/strategies which work effectively on concentrated databases have it is very own few constraints when connected on distributed databases. In distributed data mining, data is situated at distributed areas and mining is performed on each nearby database to discover internationally mined data. Figure 1 delineates the design for distributed data mining.

In the next section, we discuss about Hadoop technology, its architecture and its working in a distributed environment.

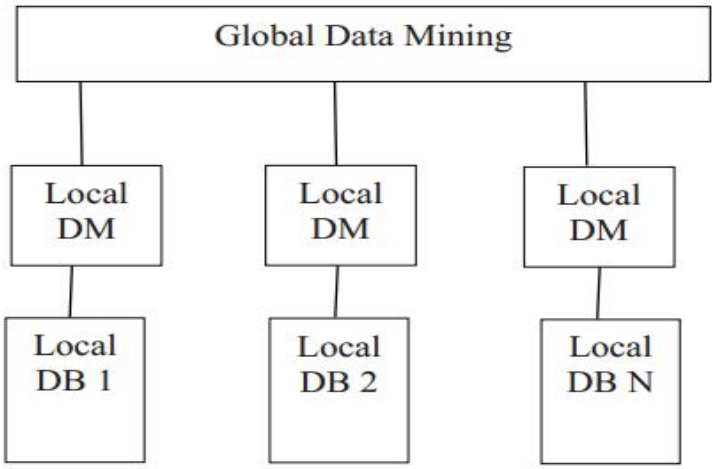

Fig.1 Architecture of Distributed Data mining

II. HADOOP

Hadoop is a free open source stage, which helps in putting away data and parallel handling in a distributed domain. Hadoop parts the expansive database into pieces of data and disperses over the clusters in the distributed condition. To process the data, Map Reduce is utilized for parallel processing on the clusters, in this way lessening the execution time.

The Hadoop Distributed File System (HDFS) is fundamentally a distributed document framework which is intended to keep running on item equipment. It is numerous like the current distributed record frameworks. Be that as it may, there are a few contrasts amongst HDFS and other distributed document frameworks which makes it noteworthy. HDFS is exceedingly fault tolerant and is planned such that it can be conveyed on minimal effort equipment. HDFS likewise gives high throughput access to application data and is exceptionally appropriate for applications that have substantial data sets.

Figure 2 demonstrates the HDFS master/slave architecture. A HDFS cluster comprises of two sections viz., a single NameNode and more than one DataNode. NameNode is a master server that directs access to records by clients and deals with the record framework namespace. There are various DataNodes in HDFS, generally one for every node in the cluster. The DataNode deals with the capacity which is connected to the nodes that they are running on. HDFS uncovered a record framework namespace and enables the client data to be stored in documents. Inside in a HDFS, a document is part into at least one pieces and these blocks are then put away in an arrangement of Data Nodes.

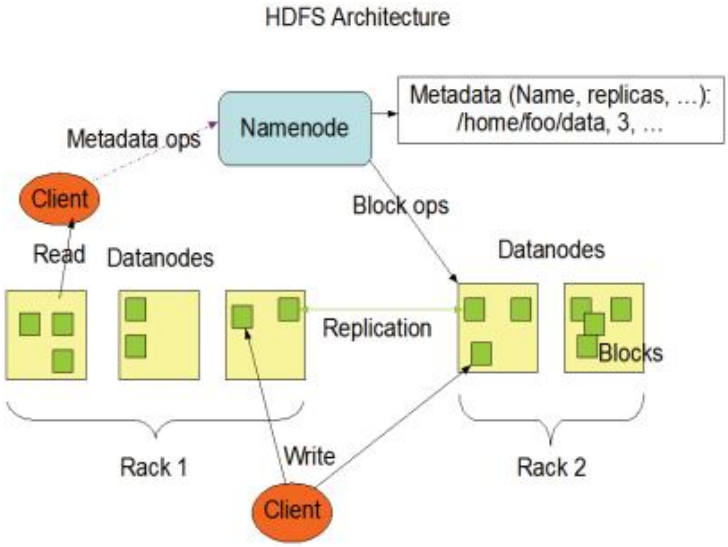

Fig.2 Architecture of HDFS

The Name Node is likewise used to execute document framework namespace activities which incorporate opening a document, closing a record and renaming records and catalogs in the HDFS. It additionally plays out the mapping of pieces of data to the Data Nodes. On the client's side, the Data Nodes are in charge of serving the read and compose demands from the HDFS. The Data Nodes likewise perform tasks, for example, block creation, deletion, and replication upon the direction gave from the Name Node.

III. LITERATURE SURVEY

Agrawal et al. [1], in this paper author propose algorithms for age of frequent item sets by progressive development of the nodes of a lexicographic tree of item sets. Author talks about various systems in age and traversal of the lexicographic tree, for example, breadth-first search, depth-first search, or a mix of the two. These procedures give diverse exchange offs as far as the I O, memory, and computational time necessities. author utilize the various leveled structure of the lexicographic tree to progressively extend exchanges at every node of the lexicographic tree and utilize matrix relying on this decreased arrangement of exchanges for finding frequent item sets. Author tried our algorithm on both genuine and engineered information.

Zaki et al. [2], various vertical mining algorithms have been proposed as of late for association mining, which have appeared to be extremely successful and typically outflank flat methodologies. The fundamental preferred standpoint of the vertical arrangement is bolstering for quick frequency tallying by means of crossing point activities on exchange ids (tids) and programmed pruning of unimportant information. The primary issue with these methodologies is when middle of the road aftereffects of vertical tid records turn out to be too vast for memory, hence influencing the algorithm versatility.

Han et al. [3], Association rule mining is an information mining procedure. It is utilized for finding the items from an exchange list which happen together frequently. A portion of the algorithms which are utilized most prevalently for association rule mining are i) Apriori algorithm ii) FP-tree algorithm. This paper researches on utilization of present day algorithm Apriori for book search for prescribing a book to a client who needs to purchase a book in light of the data that is kept up in the exchange database. The aftereffect of this contrasted and other algorithm accessible for association rule mining.

Lin et al. [4], numerous parallelization procedures have been proposed to improve the execution of the Apriori-like frequent item set mining algorithms. Portrayed by both guide and diminish capacities, Map Reduce has risen and exceeds expectations in the mining of datasets of terabyte scale or bigger in either homogeneous or heterogeneous clusters. Limiting the booking overhead of each guide decrease stage and amplifying the use of nodes in each stage are keys to fruitful Map Reduce usage. In this paper, Author propose three algorithms, named SPC, FPC, and DPC, to explore compelling usage of the Apriori algorithm in the Map Reduce structure.

Li et al. [5], Searching frequent examples in value-based databases is considered as a standout amongst the most vital information mining issues and Apriori is one of the commonplace algorithms for this errand. Growing quick and productive algorithms that can deal with vast volumes of information turns into a testing errand because of the huge databases.

Hammoud. et al. [6], over the most recent couple of years, various cooperative grouping algorithms have been proposed, i.e. CPAR, CMAR, MCAR, MMAC and others. This theory likewise presents another Map Reduce classifier that based Map Reduce affiliated rule mining. This algorithm utilizes diverse methodologies in rule disclosure, rule positioning, rule pruning, rule forecast and rule assessment techniques. The new classifier chips away at multi-class datasets and can deliver multi-mark predications with probabilities for each anticipated name.

Li et al. [7], frequent itemset mining (FIM) is a valuable device for finding frequently co-occurrent items. Since its beginning, various significant FIM algorithms have been produced to accelerate mining execution. Shockingly, when the dataset estimate is immense, both the memory utilizes and computational cost can at present be restrictively costly. In this work, we propose to parallelize the FP-Growth algorithm (we call our parallel algorithm PFP) on disseminated machines.

Zhou et al. [8], As a critical piece of finding association rules, frequent item sets mining assumes a key part in mining associations, relationships, causality and other imperative information mining undertakings. Since some conventional frequent item sets mining algorithms can't deal with gigantic little records datasets viably, for example, high memory cost, high I/O overhead, and low figuring execution, an enhanced Parallel FP-Growth (IPFP) algorithm and talk about its applications in this paper. Specifically, a little documents handling system for gigantic little records datasets to remunerate imperfections of low read/compose speed and low preparing effectiveness in Hadoop.

Riondato et al. [9], In this paper, author have portrayed PARMA, a parallel algorithm for mining semi ideal accumulations of frequent item sets and association rules in Map Reduce. Author appeared through hypothetical examination that PARMA offers provable assurances on the nature of the yield accumulations. Through experimentation on an extensive variety of datasets going in measure from 5 million to 50 million exchanges, we have exhibited a 30-55% runtime change over PFP.

Moens et al. [10], Frequent Item set Mining (FIM) is a standout amongst the most understood strategies to separate learning from information. The combinatorial blast of FIM techniques turn out to be much more risky when they are connected to Big Data.

Table1. Shows comparisons of existing methods and its limitation

Author's Name	Technique	Characteristics	Dataset	Tool/ Platform	Parameter	Benefits	Limitation
Agrawal et al.(2000)	Apriori	Level wise search, Monotonicity property and Easy to implement	Synthetic Transaction	Java	Number of transactions Number of items	Generates frequent itemsets and association rules	Scalability
Zaki et al. (2003)	dEclat	Uses vertical databases and diffsets over tidset	Mushroom	Hadoop	Minimum support Execution Time	Significant performance improvements	For sparse database diffsets loses its advantage over tidset
Han et al.(2001)	FP-Growth	Recursive approach, Employs FP-tree data structure	Connect Accident	RedHat, Linux C++	Runtime , Memory Consumption, Scalability	Focused search of smaller databases	Poor Performance
Lin et al.(2012)	SPC,FPC,DPC	SPC-Simple implementation of Apriori on Map Reduce framework, FPC-single Map Reduce phase with merging of fixed passes and DPC-Dynamically combine passes	Accident dataset, T1014D100K, Chess, Mushroom Retail Market	Hadoop with 7 map task and 1 reduce task	Confusion Matrix Size up Minimum Support	FPC and DPC provide efficient implementat ion of Apriori on Map Reduce framework and reduce scheduling, i nvocation, increasing node utilization, workload balancing.	SPC increasing scheduling and waiting overhead and FPC may get overloaded in case of large number of candidates.
Li et al. (2012)	PApriori	Sizeup, Speedup and Scaleup are used for performance evaluation	Retail Chess	Hadoop Map Reduce	Minimum Support	Efficient and Good performance for large database	User need to give number of reducers
Hammoud. et al. (2011)	MRApriori	Single scan of data in original format and Hybrid data structure, both horizontal and vertical	Retail	Hadoop Map Reduce	No of mapper No of Reducer	Efficient and Good performance for large database	No significant reduction in processing time
Li et al.	Parallel FP	Parallel version –	URLs,Tags	Hadoop	Scalability,	Linear scalability	Not efficient

(2008)	Growth	FP – Growth ,Independent mining of FP-tree and grouping of items	Transaction	Map Reduce	Run Time		in terms of memory and speed.
Zhou et al. (2010)	Balanced FPGrowth	Improvement in FP-Growth and uses frequencies of frequent items to balance the groups of PFP	Retail	Hadoop Map Reduce	No of Transaction	Faster execution using singletons with balanced distribution	Search Space partition using single item is not most efficient way
Riondato et al.(2012)	PARMA	Use random sampling method	Mushroom	Hadoop Map Reduce	No of Transaction Speedup Runtime Accuracy	Minimizes data replication, Scaling linearly, Runs faster	Finds approximate collection of frequent itemsets
Moens et al. (2013)	Dist-Eclat	Distributed version of Eclat	Mushroom Retail	Hadoop Map Reduce	Minimum Support	Minimum Support	Scalability When Data size increases it does not work.

IV. TOOLS USED

There are many tools available for processing data and extracting frequent patterns. Some of them are presented below.

A. Hadoop Mapper Tool
B. Hadoop Reducer Tool
C. Hadoop Distributed File System

V. CONCLUSION

As we have surveyed various methods of frequent item set mining in parallel and distributed situations, the vast majority of the systems/algorithms have inadequacies of their own. Despite the fact that, Hadoop innovation can gives a superior stage to defeat the inadequacies of the previously mentioned mining procedures.

REFERENCES

[1] R. Agrawal, C. Aggarwal, and V. Prasad, "A Tree Projection Algorithm for Generation of Frequent Item Sets," Parallel and Distributed Computing, pp. 350-371, 200

[2] Mohammed J. Zaki, Karam Gouda, "Fast Vertical Mining Using Diffsets", 2003 ACM.

[3] Han Jiawei, KamberMiceline. Fan Ming, MengXiaofeng translation, "Data mining concepts and technologies". Beijing: Machinery Industry Press. 2001.

[4] Lin, M. Y., Lee, P. Y., & Hsueh, S. C. (2012, February). Apriori-based frequent itemset mining algorithms on MapReduce. In proceedings of the 6th international conference on ubiquitous information management and communication (p. 76). ACM

[5] Li, N., Zeng, L., He, Q., & Shi, Z. (2012, August). Parallel implementation of apriori algorithm based on MapReduce. In Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on (pp. 236-241). IEEE

[6] S. Hammoud. MapReduce Network Enabled Algorithms for Classification Based on Association Rules. Thesis, 2011

[7] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang and Edward Chang," PFP: Parallel FP-Growth for Query Recommendation", ACM 2008

[8] Zhou, L., Zhong, Z., Chang, J., Li, J., Huang, J. Z., & Feng, S. (2010, November). Balanced parallel FP-growth with mapreduce. In Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on (pp. 243-246). IEEE

[9] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal. PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce. In Proc. CIKM, pages 85–94. ACM, 2012

[10] Moens, S., Aksehirli, E., & Goethals, B. (2013, October). Frequent itemset mining for big data. In Big Data, 2013 IEE International Conference on (pp. 111-118). IEEE.