Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Efficient Data Partitioning In Frequent Item Set Mining on Hadoop

M.Swathi
Assistant Professor,
Department of CSE,
Siddhartha Institute of Tech & Sciences,
swathimuppavaram@gmail.com.

R.Saritha
M. Tech Student,
Department of CSE,
Siddhartha Institute of Tech & Sciences,
saritharudhararaj@gmail.com.

Abstract:
A Parallel Frequent Item sets mining algorithm called FiDoop using Map Reduce programming model. FiDoop includes the frequent items ultra metric tree(FIU-tree), in that three Map Reduce jobs are applied to complete the mining task. The scalability problem has been addressed by the implementation of a handful of FP-growth-like parallelFIM algorithms.

InFiDoop, the mappers independently and concurrently decompose item sets; the reducers perform combination operations by constructing small ultra metric trees as well as mining these trees in parallel. Data Deduplication is one of important data compression method for erasing duplicate copies of repeating data and reduce the amount of storage space and save bandwidth. The technique is used to improve storage space utilization and can also be applied to reduce the number of bytes. The first Map Reduce job discovers all frequent items, the second Map Reduce job scans the database to generate k-item sets by removing infrequent items, and the third Map Reduce job complicated one to constructs k-FIU-tree and mines all frequent k-item sets. In this paper, we applying Deduplication technique in third Map Reduce job to avoid the replication of data in frequent item sets and improve the performance. It produces highly related mining results with less time and increase the storage capacity. Hadoop supports nine different tools, while Mahout is based on core algorithm and classifications. Having sequence algorithm to produce the output in better way. We aim to implement recommendation algorithm using Mahout, a machine learning device, on Hadoop platform to provide a scalable system for processing large data sets efficiently. This can be performed on such platforms for quicker performance.

Keywords: ***FiDoop, Parallel Mining, Frequent Item sets, Mahout.***

Introduction

FREQUENT item sets mining (FIM) is a core difficulty in association rule mining (ARM), sequence mining, and the similar to. Speeding up the procedure of FIM is critical and crucial, because FIM expenditure accounts for a significant section of mining instance due to its high computation and input/output (I/O) intensity. Frequent item sets mining algorithms can be divided into two categories namely, Apriori and FP-growth schemes. Apriori is a standard algorithm with the generate-and-test process that generates a huge number of aspirant item sets; Apriori has to frequently scan an whole database. Earlier developed parallel FIM algorithms were built leading the Apriori algorithm. Unfortunately, in Apriori-like parallel FIM algorithms, every processor have to check a database several times and to exchange an unnecessary number of candidate item sets with other processors. Data Deduplication is a focused data compression technique for eliminating photocopy copies of repeating data in storage. It brings a lot of benefits, security and privacy concerns happen as users, sensitive data are subject to both inside and outside attacks.

Fixed encryption, while providing data privacy, is incompatible with data Deduplication. Particularly, traditional encryption requires different users to encrypt their facts with their individual keys. To avoid unauthorized access, a secure proof of ownership procedure is also essential to provide the evidence that

Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

the user indeed owns the same file when a duplicate is establish. Hadoop has two sub-divisions namely HDFS (Hadoop Distributed File System) with Map Reduce programming model. Hadoop perfectly breaks the data into large chunks and distributes it to its product hardware cluster nodes for additional processing using Map Reduce programming model for distributed computing thus able to handle large datasets. Map Reduce was initially developed by Google for counting the no. of times a word occurs in particular document. It works well for applications where data is stored at distributed file system which allows local computing on each data node.

LITERATURE SURVEY

D. Chen et al.[1], proposed a "Tree partition based parallel frequent pattern mining on shared memory systems". In this paper, we show a tree-segment algorithm for parallel mining of frequent item set examples. Our work is taking into account FP-Growth algorithm, which consist of construction of tree and mining. The fundamental thought is to built single FP-Tree in the memory, segment it into a few free parts and appropriate them to diverse strings. A heuristic calculation is to equalization the workload over the cluster of computers. this algorithm cannot just reduce the effect of locks amid the tree-building stage, in any case, likewise dodge the overhead that do awesome damage to the mining stage. It show the trials on various sorts of datasets and contrast the outcomes and other parallel methodologies. The outcomes propose that the methodology has extraordinary point of preference in effectiveness, particularly on certain sorts of datasets. As the quantity of processors builds, our parallel algorithm indicates great adaptability.

Y.-J. Tsay et al.[2], proposed a "FIUT: A new method for mining frequent item sets,". This paper proposes a productive technique, the frequent item set ultra metric trees (FIUT), for mining incessant item sets in a database. FIUT utilizes an extraordinary continuous things ultra metric tree (FIU-tree) structure to improve its proficiency in getting continuous item sets. Looked at to related work, FIUT has four noteworthy advantages. To start with, it minimizes I/O overhead by examining the database just twice. Second, the FIU-tree is an enhanced approach to segment a database, which results from grouping exchanges, and fundamentally diminishes the inquiry space. Third, just incessant things in every exchange are embedded into the FIU-tree for fully packed storage. At long last, all successive item sets are produced by checking the leaves of each FIU-tree, without crossing the tree recursively, which altogether decreases processing time. FIUT was contrasted and FP-development, an understood and broadly utilized algorithm and the reenactment results demonstrated that the FIUT beats the FP-development. E.-H. Han, G. Karypis, and V. Kumar[3],depicts "Scalable parallel data mining for association rules,". One of the vital issues in information mining is discovering association rules from databases of exchanges where every exchange comprises of an arrangement of items. The most time devouring operation in this revelation procedure is the calculation of the recurrence of the events of intriguing subset of things (called applicants) in the database of exchanges.

To prune the exponentially extensive space of applicants, most existing calculations, consider just those applicants that have a client characterized least backing. Indeed, even with the pruning, the errand of discovering all affiliation rules requires a great deal of calculation power and time. Parallel PCs offer a potential answer for the calculation prerequisite of this undertaking, gave proficient and versatile parallel calculations can be composed. In this paper, it exhibit two new parallel calculations for mining affiliation rules. The Intelligent Information Distribution calculation effectively utilizes total memory of the parallel PC by utilizing wise applicant partitioning plan and uses proficient correspondence component to move information among the processors. The Half and half Distribution calculation further enhances the InteUigent Information Distribution calculation by powerfully dividing the hopeful set to keep up great burden equalization. The trial results on a Cray T3D parallel PC demonstrate that the Hybrid Distribution algorithm scales directly

Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

furthermore, misuses the total memory better and can create more association rule with a solitary output of database per pass.

K.-M. Yu et al.[4],presented "A load-balanced distributed parallel mining algorithm,". Because of the exponential development in overall data, organizations need to manage a perpetually developing measure of advanced data. A standout amongst the most essential difficulties for information mining is rapidly and effectively finding the relationship among information. The Apriori calculation has been the most well known strategy in finding continuous examples. Nonetheless, while applying this strategy, a database must be checked numerous times to figure the checks of countless item sets. Parallel and dispersed figuring is a viable technique for quickening the mining process. In this paper, the Distributed Parallel Apriori (DPA) calculation is proposed as an answer for this issue. In this reference, metadata are put away as Transaction Identifiers (TIDs), such that just a solitary sweep to the database is required. The approach likewise takes the element of item set tallies into thought, along these lines producing an adjusted workload among processors and diminishing processor unmoving time. Probes a PC bunch with 16 processing hubs is likewise made to demonstrate the execution of this approach and contrast it and some other parallel mining calculations. The test results demonstrate that the proposed approach beats the others, particularly while the base backings are low.

L. Zhou et all[5] proposed a "Balanced parallel FP-growth with Map Reduce,". Regular item set mining (FIM) assumes a key part in mining affiliations, connections and numerous other critical information mining errands. Lamentably, as the volume of dataset gets bigger step by step, most of the FIM calculations in writing get to be ineffectual because of either excessively tremendous asset prerequisite or as well much correspondence cost. In this reference, it propose an adjusted parallel FPGrowth calculation BPFP, in light of the PFP calculation [1], which parallelizes FPGrowth in the Map Reduce approach. BPFP includes into PFP load parity highlight, which enhances parallelization and consequently enhances execution. Through exact study, BPFP beat the PFP which utilizes some straightforward gathering procedure.

K. W. Lin, P.-L. Chen, and W.-L. Chang,[6] presented "A novel frequent pattern mining algorithm for very large databases in cloud computing environments," . FPgrowth is the most renowned calculation for finding incessant examples. As the database size developments on the other hand the base bolster diminishes, in any case, both of the memory prerequisite and execution time increment enormously. Numerous scientists attempted to take care of this issue by using conveyed registering procedures to enhance the adaptability and execution proficiency. In this paper, we propose a strategy for finding continuous examples from extensive database in distributed computing situations. To construct the entire FP Tree, we utilize the circle as the auxiliary memory. Since the plate access is much slower than fundamental memory, a proficient information structure for putting away and recovering FPTree from circle is moreover proposed. Through observational assessments on different reproduction conditions, the proposed strategy conveys magnificent execution as far as adaptability and execution time.

S. Hong, Z. Huaxuan, C. Shiping, and H. Chunyan,[7] proposed "The study of improved FP-growth algorithm in Map Reduce," . As FP-Growth calculation produces a lot of contingent example bases and restrictive example trees, prompting low productivity, propose an enhanced FP-Growth (IFP) calculation which firstly consolidates the same examples taking into account the circumstance whether the backing of the exchange is more prominent than the base support (min_sup) to mine the continuous examples. Accordingly the IFP eliminates the space and enhances the effectiveness. It likewise makes it simple to be paralleled. Advance more, consolidate the IFP calculation with the Map Reduce registering model, named MR-IFP(Map Reduce-Improved FP), to enhance the ability to manage the mass information.

Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Existing System

In Existing System Rather than considering Apriori and FP-growth, we incorporate the frequent items ultrametric tree (FIU-tree) in the design of our parallel FIM technique. We focus on FIU-tree because of its four salient advantages, which include reducing I/O overhead, offering a natural way of partitioning a dataset, compressed storage, and averting recursively traverse.

Disadvantage:-

- Parallel algorithms lack a mechanism that enables automatic parallelization, load balancing, data distribution, and fault tolerance on large computing clusters.

Proposed System

In Proposed System a new data partitioning method to well balance computing load among the cluster nodes; we develop FiDoop-HD, an extension of FiDoop, to meet the needs of high- dimensional data processing.

Advantage:-

- FiDoop is efficient and scalable on Hadoop clusters.

Association Rules

ARM provides a considered resource used for decision support by extracting the most significant regular patterns that concurrently happen in a large transaction database. A usual ARM application is market basket analysis. The final object of ARM is to notice all policy that satisfies a user-specified minimum sustain and minimum confidence.

The ARM method can be decomposed into two phases: 1) identifying all regular item sets whose support is better than the minimum support and 2) forming qualified implication system among the frequent item sets. The first stage is more demanding and difficult than the second one. As such, most previous studies are mainly focused on the topic of discovering frequent item sets. The design aim of FiDoop is to construct a mechanism that enables repeated parallelization, load balancing, and data sharing for parallel mining of frequent item sets on huge clusters. To assist the appearance of FiDoop. Aiming to recover data storage efficiency and to prevent structure provisional pattern bases, FiDoop incorporates the idea of FIUtree rather than traditional FP trees.

Map Reduce Framework

A Map Reduce program is collected of a Map()procedure (method) that executes filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The "Map Reduce System" (also named "infrastructure" or "framework") arranges the treating by marshalling the distributed servers, running the various jobs in parallel, handling all communications and data transfers among the various parts of the system, and providing for redundancy and fault tolerance. The model is motivated by the map and reduce functions usually used in functional programming, while their purpose in the Map Reduce framework is not the similar as in their unique forms.

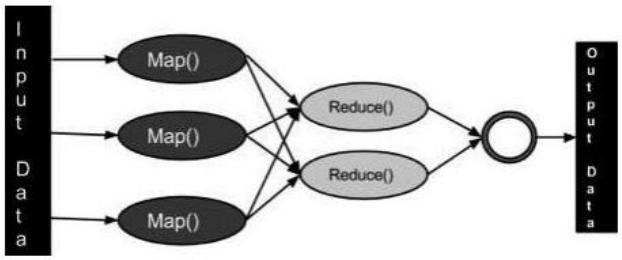

Fig 1: Hadoop Prospective

The key helps to the Map Reduce framework are not the real map and reduce purposes, but the scalability and fault-tolerance realized for a change of requests by optimizing the execution engine once. As such, a single-threaded implementation of Map Reduce will usually not be earlier than a traditional (non-Map Reduce) application; any gains are typically only seen with multithreaded applications.

Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

The usage of this typical beneficial only while the improved distributed shuffle process (which reduces network communication cost) and fault tolerance structures of the Map Reduce framework arise into tragedy. Raising the statement cost is vital to a good Map Reduce algorithm. The three Map Reduce jobs of our proposed FiDoop are described in detail. The first Map Reduce job discovers all frequent items or frequent one-item sets (see Algorithm 2). In this phase, the input of Map tasks is a database, and the output of Reduce tasks is all frequent one-item sets. The second Map Reduce job scans the database to generate k-item sets by removing infrequent items in each transaction The last Map Reduce job—the most complicated one of the three—constructs k-FIU-tree and mines all frequent k-item sets.

Effective four steps to Data Deduplication

Around adozen major vendors for Deduplication applications, Irrespective of retailer implementation Data Deduplication can be considered into four major steps: 1. Identifying the unit of comparison 2. Creating smaller unique identifier of these units to be compared. 3. Match for duplicates 4. Saving unique data blocks Implementation of each of these stages differs from vendor to vendor. But, the main objective of any implementation is to: Achieve maximum Deduplication ratio (Size of Real Data / Size of Data once Deduplication:1)Maximize Data Deduplication quantity (Megabytes of Data Deduplicated per sec)Minimize system resource utilization.

Mahout

Apache Mahout is Java carved library for machine learning algorithms that are scalable and can be applied on the top of Hadoop using Map Reduce framework for studying Big Data. It's an open source machine learning library from the Apache Software Substance. It implements many data mining algorithms similar Recommend engines (), clustering(), classification() and is accessible to very big data sets (up to terabytes and pet bytes) that are in the Big Data realm. These methods are also used in outlier discovery (also called anomaly detection), which means classifying events or explanations that do not conform to an estimated outcome, to support in classifying fraud in online transactions, etc. The Clustering algorithms applied in Apache Mahout are K-Means, Fuzzy K-Means, Streaming K-Means and Spectral Clustering. Clustering a cluster of objects includes three things: An algorithm, which is the technique used to collection things composed. Anidea of both similarity and dissimilarity — which item goes to an existing stack and which must start a new one. A ending situation, which capacity be the point past which objects can't be arranged any more, or while the stacks are previously quite different.

Conclusion

To solve the scalability and load paired tasks in the existing parallel mining algorithms for frequent item sets, we functional the Map Reduce encoding model to improve a parallel frequent item sets mining algorithm called FiDoop. FiDoop combines the frequent items ultra metric tree or FIU-tree rather than conventional FP trees, thus achieving compressed storing and avoiding the need to build qualified pattern bases. We also offered some new de-duplication creations supportive certified duplicate check in frequent item sets Time wanted to solve the difficult has reduced. Mahout is able to handle big data but it still want some algorithms. The reference for single user want to be developed for better results. .

New dividing platforms like Apache Spark are attainment prominent in the field of Big Data analysis. Approval algorithms can be completed on such stages for quicker performance.

Future Enhancement

This system in future can be enhanced with a diplomatic sentiment anslysis and redefine process of computation under big data environment. Apparently the proposed project can be used and appended in medical static data analysis and also the medical crisis and resource sharing analysis. This application is highly simulative and is active on all the medical conditions and diseases v/s resource mapping and decision analysis can be fetched.

Volume No:2, Issue No:12 (May-2017)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

References

[1] http://www.tcs.com/SiteCollectionDocuments/White%20Papers/HiTech_White paper_Effective_Data_Deduplication_Implementation_05_2011.pdf#page=5&zoom=auto,-107,644.

[2] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6802424&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6802424.

[3] https://www.irjet.net/archives/V2/i4/Irjet-v2i418.pdf.

[4] http://www.lemenizinfotech.com/2015/hadoop/FiDoop%20Parallel%20Mining%20of%20Frequent%20Itemsets%20Using%20MapReduce.pdf.

[5] http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf.

[6] X. Lin, "Mr-apriori: Association rules algorithm based on mapreduce," in Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on. IEEE, 2014, pp. 141–144.

[7] L. Zhou, Z. Zhong, J. Chang, J. Li, J. Huang, and S. Feng, "Balanced parallel fp-growth with mapreduce," in Information Computing and Telecommunications (YCICT), 2010 IEEE Youth Conference on. IEEE, 2010, pp. 243–246.

[8] S. Hong, Z. Huaxuan, C. Shiping, and H. Chunyan, "The study of improved fp-growth algorithm in mapreduce," in 1st International Workshop on Cloud Computing and Information Security. Atlantis Press, 2013.

[9] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, "Parma: a parallel randomized algorithm for approximate association rules mining in mapreduce," in Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012, pp. 85–94.

[10] C. Lam, Hadoop in action. Manning Publications Co., 2010.

[11] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang, "Pfp: parallel fp-growth for query recommendation," in Proceedings of the 2008 ACM conference on Recommender systems. ACM, 2008, pp. 107–114.