



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

A Survey on Ultrametric Tree for Frequent itemsets for Big Data

Sonal R.Londhe, Prof. Varsha R. Dange

Student, Dept. of Computer, Dhole Patil College of Engineering, Savitribai Phule Pune University, Pune, India

Guide, Dept. of Computer, Dhole Patil College of Engineering, Savitribai Phule Pune University, Pune, India

ABSTRACT: In parallel mining algorithms for frequent itemsets multiple mechanism are used (for eg.load balancing, data distribution, automatic parallelization, and fault tolerance on large clusters). For solution to this problem, a new parallel frequent itemsets mining algorithm called FiDooop is proposed using the MapReduce programming model. FiDooop incorporates the frequent items ultrametric tree for achieving reduces storage and avoids building conditional pattern bases, rather than conventional FP trees. In FiDooop, three MapReduce Jobs are implemented to complete the mining task. In third MapReduce job, mappers decompose itemsets independently and reducer constructing small ultrametric trees, mining of these trees separately. In this paper, FiDooop is implemented on an in-house Hadoop cluster. It shows that FiDooop on the cluster is sensitive to data distribution and dimensions, because itemsets with different lengths have different decomposition and construction costs. For improving FiDooop's performance and workload balance metric to measure load balance across the cluster's computing nodes, a FiDooop-HD is developed. FiDooop-HD helps to speed up the mining performance for high-dimensional data analysis. Extensive experiments using real-world celestial spectral data demonstrate that our proposed solution is efficient and scalable. In my proposed scheme, I will add various approaches to provide privacy to sensitive user datafor FiDooop running on Hadoop clusters. Using Load Balancing across different nodes, further energy efficiency can also be improved.

KEYWORDS: MapReduce, Energy efficiency, frequent itemsets, Frequent Items Ultrametric Tree (FIU-tree), Hadoop cluster, Load balance.

I. INTRODUCTION

An elementary necessity for mining for mining association rules is mining frequent itemsets. Numerous algorithms exist for frequent itemset mining. Apriori and FP-Growth are the traditional method. Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by recognizing the frequent individual items in the database and widening them to larger item sets providing those item sets appear adequately often in the database. It works with Candidate Generation and Test Approach.FP-Growth is used to overcome the problem of candidate generation. FP-growth is a program to find frequent item sets with the FP-growth algorithm, which corresponds to the transaction database as a prefix tree which is enhanced with links that organize the nodes into lists referring to the same item. The search is carried out by prognostic the prefix tree, working recursively on the result, and trimming the original tree. The implementation also supports sifting for closed and maximal item sets with conditional item set repositories, although the approach used in the program differs in as far as it used top-down prefix trees rather than FP-trees. FP-growth condense a large database into a compact, Frequent-Pattern tree (FP-tree) structure with highly reduced, but complete for frequent pattern mining and avoid costly database scans. It develops an efficient, FP-tree-based frequent pattern mining method with a divide-and-conquer methodology which decomposes mining tasks into smaller ones and avoids candidate generation. The disadvantage of this algorithm consists in the TID_set being too long, taking considerable memory space as well as computation time for intersecting the long sets. Incremental data mining is not held by this algorithm.FREQUENT itemsets mining (FIM) is a center issue in association rule mining (ARM), grouping mining, and so forth. Accelerating the procedure of FIM is basic and essential, on the grounds that FIM utilization represents a huge part of mining time because of its high calculation and input/output (I/O) force. At the point when datasets in present day information mining applications turn out to be too much substantial, consecutive FIM calculations running on a solitary machine experience the ill effects of execution



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

decay. To address this issue, I research on how to perform FIM utilizing MapReduce—a generally embraced programming model for preparing enormous datasets by misusing the parallelism among registering hubs of a bunch. This demonstrates to appropriate an extensive dataset over the group to adjust load over all bunch hubs, accordingly improving the execution of parallel FIM. Big information for the most part incorporates information set with sizes past the capacity of generally utilized programming devices to catch, oversee and handle information inside a fair passed time. Its size is continually moving focus starting 2012 going from a couple of Dozen of terabyte to numerous petabytes of information "greatly parallel programming running on tens, hundreds, or even a large number of servers".

II. OBJECTIVE

1. Subset selection to achieve fast retrieval.
2. Compressed storage and load balancing is achieved by calculating T-relevance.
3. Compute F-correlation and construct a Minimum Spanning Tree to improve speed and accuracy.
4. Efficiency and effectiveness of fast clustering algorithm are evaluated.
5. The similar data are clustered using clustering algorithm. The final data set is stored in HDFS (Hadoop Distributed File System) using map reduce technique. The performance after redundancy removal is evaluated.
6. Improving energy efficiency of Fidoop running on Hadoop clusters.
7. Providing privacy to sensitive user data.

III. LITERATURE SURVEY

1. **Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, NingfangMi, "Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters" 2168-7161 (c) 2015 IEEE.**

The paper proposes simple yet effective schemes which use slot ratio between map and reduce tasks as a tunable knob for reducing the makespan of a given set. By leveraging the workload information of recently completed jobs, our schemes dynamically allocates resources (or slots) to map and reduce tasks. It implemented the presented schemes in Hadoop V0.20.2 and evaluated them with representative MapReduce benchmarks at Amazon EC2. The experimental results demonstrate the effectiveness and robustness of the schemes under both simple workloads and more complex mixed workloads.

2. **Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters" COMMUNICATIONS OF THE ACM January 2008/Vol. 51, No. 1**

MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a *map* and a *reduce* function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

3. **R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22, no. 2, pp. 207–216, 1993.**

The system proposes, (1) efficient generation for large itemsets by hash method (2) effective reduction on itemsets scan required by the division approach and (3) the option of reducing the number of database scans required by the proposed hash and division-based techniques.

4. **Assaf Schuster, Ran Wolff, "Communication Efficient Distributed Mining of Association Rules"**

This paper presents a set of new algorithms that solved the Distributed Association Rule mining Problem using far less communication.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

5. **Jong Soo Park; Ming-Syan Chen and Philip S. Yu, “Efficient Parallel Data Mining for Association Rules”**
Through this paper, an algorithm, called PDM, is developed to conduct parallel data mining for association rules. Consider a transaction as a collection of items, and a large itemset is a set of items such that the number of transactions containing it exceeds a pre-specified threshold. PDM is so designed that the global set of large itemsets can be identified efficiently and the amount of inter-node data exchange required is minimized.
6. **Sandy Moens, Emin Aksehirli and Bart Goethals, “Frequent Itemset Mining for Big Data.**
This paper investigates the applicability of FIM techniques on the MapReduce platform. It introduces two new methods for mining large datasets: Dist-Eclat focuses on speed while BigFIM is optimized to run on really large datasets. The experiments show the scalability of the methods.
7. **Trupti Kenekar, A. R. Dani, “ A Study of Differentially Private Frequent Itemset Mining ” International Journal of Science and Research (IJSR), Volume 4 Issue 10, October 2015.**
Differential privacy aims to get means to increase the accuracy of queries from statistical databases while minimizing the chances of identifying its records and itemset. The studied algorithm consists of a preprocessing phase as well as a mining phase. It under seeks the applicability of FIM techniques on the MapReduce platform, transaction splitting. It analyzed how differentially private frequent itemset mining of existing system as well.
8. **Wei Lu, Yanyan Shen, Su Chen, Beng Chin Ooi, “Efficient Processing of k Nearest Neighbor Joins using MapReduce”**
The paper investigates how to perform kNN join using MapReduce which is a well-accepted framework for data-intensive applications over clusters of computers. The mappers cluster objects into groups; the reducers perform the kNN join on each group of objects separately.
9. **JIAWEI HAN, JIAN PEI, YIWEN YIN, RUNYING MAO “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach “**
This paper proposes a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, *FP-growth*, for mining *the complete set of frequent patterns* by pattern fragment growth.
10. **Shekhar Gupta, Christian Fritz, Johan de Kleer, and Cees Witteveen, “Diagnosing Heterogeneous Hadoop Clusters ” 23rd International Workshop on Principles of Diagnosis**
The paper proposes efficient use of Hadoop on virtual/cloud infrastructure, which violate the peer-similarity assumption. To this end, it has implemented and presented preliminary results of an approach for automatically diagnosing the health of nodes in the cluster, as well as the resource requirements of incoming MapReduce jobs. It shows that the approach can be used to identify abnormally performing cluster nodes and to diagnose the kind of fault occurring on the node in terms of the system resource affected by the fault (e.g., CPU contention, disk I/O contention). It also describes their future plans for using this approach to increase the efficiency of Hadoop on heterogeneous and virtual clusters, with or without faults.

IV. EXISTING SYSTEM APPROACH

Existing system also works with frequent itemset mining. The base paper approach system also used HDFS framework but in FIUT having some data leakage issue. The system was able to find all possible itemset, and that was drawback of existing base approach so there is a need to eliminate such proposal.

V. PROPOSED SYSTEM APPROACH

The base system having dimension reduction issue, the proposed solution focuses on eliminating such problems. The system also focuses on SQL injection and prevention as well as data collusion attacks. Develop the system in HDFS 2.0 with MongoDB with 16 cluster node environment. Proposed system use HDFS framework with R package called R-hadoop. The proposed system extends up to node cluster. It also uses transaction management system base on ACID



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

properties which will help to avoid data inconsistency. The privacy of sensitive user data on the end user systems will be protected various approaches.

VI. CONCLUSION

To solve the scalability and load balancing challenges in the existing parallel mining algorithms for frequent itemsets, the MapReduce programming model to develop a parallel frequent itemsets mining algorithm called FiDooop is applied. FiDooop incorporates the frequent items ultrametric tree or FIU-tree rather than conventional FP trees, thereby achieving compressed storage and avoiding the necessity to build conditional pattern bases. FiDooop seamlessly integrates three MapReduce jobs to accomplish parallel mining of frequent itemsets. The third MapReduce job plays an important role in parallel mining; its mappers independently decompose itemsets whereas its reducers construct small ultrametric trees to be separately mined. The performance of FiDooop is improved by balancing I/O load across data nodes of a cluster.

REFERENCES

- [1] Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, Ningfang Mi, "Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters" 2168-7161 (c) 2015 IEEE.
- [2] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters" COMMUNICATIONS OF THE ACM January 2008/Vol. 51, No. 1.
- [3] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22, no. 2, pp. 207–216, 1993.
- [4] A. Schuster and R. Wolff, "Communication-efficient distributed mining of association rules," Data Min. Knowl. Disc., vol. 8, no. 2, pp. 171–196, 2004.
- [5] Jong Soo Park; Ming-Syan Chen and Philip S. Yu, "Efficient Parallel Data Mining for Association Rules"
- [6] Sandy Moens, Emin Aksehirli and Bart Goethals, "Frequent Itemset Mining for Big Data", 2013 IEEE International Conference on Big Data, 978-1-4799-1293-3/13/\$31.00 ©2013 IEEE
- [7] Trupti Kenekar, A. R. Dani, "A Study of Differentially Private Frequent Itemset Mining" International Journal of Science and Research (IJSR), Volume 4 Issue 10, October 2015
- [8] Wei. Lu, Y. Shen, S. Chen, and B. C. Ooi, "Efficient processing of k nearest neighbor joins using MapReduce," Proc. VLDB Endow., vol. 5, no. 10, pp. 1016–1027, 2012.
- [9] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Min. Knowl. Disc., vol. 8, no. 1, pp. 53–87, 2004.
- [10] Shekhar Gupta, Christian Fritz, Johan de Kleer, and Cees Witteveen, "Diagnosing Heterogeneous Hadoop Clusters" 23rd International Workshop on Principles of Diagnosis.