www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 11 Nov. 2016, Page No. 18793-18794

Data Deduplication in Parallel Mining of Frequent Item sets using MapReduce

Pavithra.K[1]
[1]Assistant Professor, Department of Computer Science,
KG College of Arts and Science, Coimbatore-641035, Tamilnadu, India
kpavithramsc@gmail.com

Abstract: ***A Parallel Frequent Item sets mining algorithm called FiDoop using MapReduce programming model. FiDoop includes the frequent items ultrametric tree(FIU-tree), in that three MapReduce jobs are applied to complete the mining task. The scalability problem has been addressed bythe implementation of a handful of FP-growth-like parallelFIM algorithms. InFiDoop, the mappers independently and concurrently decompose item sets; the reducers perform combination operationsby constructing small ultrametric trees as well as miningthese 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 MapReduce job discovers all frequent items, the second MapReduce job scans the database to generate k-item sets by removing infrequent items, and the third MapReduce job complicated one to constructs k-FIU-tree and mines all frequent k-item sets.***

In this paper, we applying Deduplication technique in third MapReduce 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.

1. Introduction

FREQUENT itemsets 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 significantsection of mining instance due to its high computation and input/output (I/O) intensity.Frequent itemsets 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 itemsets; Apriori has to frequently scan anwhole 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 itemsets 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 insider and outsider 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 the user indeed owns the same file when a duplicate is establish. Hadoop has two sub-divisions namely HDFS (Hadoop Distributed File Syetem) withMapReduce programming model. Hadoopperfectly breaks the data into large chunks and distributes it to its product hardware cluster nodes for additional processing using MapReduce programming model for distributed computing thus able to handle large datasets.

MapReduce 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.

2. 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 itemsets. 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 itemsets.

The design aim of FiDoop is to construct a mechanism that enables repeated parallelization, load balancing, and data sharing for parallel mining of frequent itemsets 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 FIU-tree rather than traditional FP trees.

3. MapReduce Framework

A MapReduce 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 "MapReduce 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 MapReduce framework is not the similar as in their unique forms. The key helps to the MapReduce 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 MapReduce will usually not be earlier than a traditional (non-MapReduce) application; any gains are typically only seen with multi-threadedapplications. The usage of this typical beneficial only while the improved distributed shuffle process (which reduces network communication cost) and fault tolerance structures of the MapReduce framework arise into tragedy. Raising the statement cost is vital to a good MapReduce algorithm.

The three MapReduce jobs of our proposed FiDoop are described in detail.

The first MapReduce job discovers all frequent items or frequent one-itemsets (see Algorithm 2). In this phase, the input of Map tasks is a database, and the output of Reduce tasks is all frequent one-itemsets. The second MapReduce job scans the database to generate k-itemsets by removing infrequent items in each transaction The last MapReduce job—the most complicated one of the three—constructs k-FIU-tree and mines all frequent k-itemsets.

4. 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 stagesdiffers 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.

5. Mahout

Apache Mahout is Java carved library for machine learning algorithms that are scalable and can be applied on the top of Hadoop using MapReduce framework for studying Big Data.Its an open source machine learning library from the Apache Software Substance. It implements many data mining algorithms similar Recommendengines (), clustering(), classification() and is accessible to very big data sets (up to terabytes and petabytes) 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.Aendingsituation, which capacity be the point past which objects can't be arranged any more, or while the stacks are previously quite different.

6. Conclusion

To solve the scalability and load pairedtasks in the existing parallel mining algorithms for frequent itemsets, we functional the MapReduceencoding model to improve a parallel frequent itemsets mining algorithm called FiDoop. FiDoopcombines the frequent items ultrametric tree or FIU-tree rather than conventional FP trees, thusachiev-ing compressed storing and avoiding the need to build qualified pattern bases. We also offeredsome new deduplication creationssupportivecertified 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.

References

[1] http://www.tcs.com/SiteCollectionDocuments/White%20Papers/HiTech_Whitepaper_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.

Author Profile

Pavithra.K, received the B.Sc Computer Science degree in Sri Ramakrishna College of Arts and Science for Women and M.Sc Computer Science degree in Dr. G.R.Damodaran College of Science, in 2010 and 2012 respectively. Pursuing part time Ph.D Computer Science in Dr.G.R.Damodaran College of Science. Now, Working as a Assistant Professor in KG College of Arts and Science, Saravanampatti, Coimbatore- 641035, Tamilnadu, India.