ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Frequent Itemset Mining for Distributed Systems using Hadoop

Tushar M. Chaure, Kavita R. Singh

Dept. of Computer Technology, YCCE, Nagpur, India

ABSTRACT: Frequent Itemset Mining is one of the most popular techniques to extract knowledge from data. But, these mining methods become more problematic when they are applied to Big Data. Fortunately, recent improvements in the field of parallel programming provide many tools to tackle this problem. However, these tools come with their own technical challenges such as balanced data distribution and inter-communication costs. In this paper, we perform the applicability of FIM techniques on Hadoop platform. Here we introduced two new methods for mining large datasets, one where FP growth focuses on speed and the other where Hadoop Map/Reduce platform will be used to reduce fault tolerance. The algorithm will store the data in a tree structure at every node, which will facilitate the process and reduce the execution time required for mining.

KEYWORDS: frequent itemsets, map/reduce, distributed mining.

I. INTRODUCTION

Data mining is the process of extraction of information from large databases and it is a powerful new technology having a great potential to help researchers as well as companies on the most important information in their data warehouses [1]. Data mining tools are used to predict the future trends and behaviors thus allowing businesses to make knowledge-driven decisions.

Frequent itemset mining in distributed environment is a problem and must be performed using a distributed algorithm that does not require exchange of raw data between the participating sites. Distributed data mining is the process of mining data in distributed data sets. According to Zaki in [2], two dominant architectures exist in the distributed environments i.e., distributed memory architecture (DMA) and shared memory architecture (SMA).

In DMA, each processor has its own database or memory and has access to it. DMA systems access to other local databases is possible only via message exchange. DMA offers a simple programming method, but limited bandwidth may reduce the scalability. On the other hand, in SMA each processor has direct and equal access to the database in the system. Thus, parallel programs on such systems can be implemented easily.

A set of items in a database is known as itemset. If the occurence of items in a particular transaction is frequent, it is called as frequent itemset and the support (or count) of frequent itemset is greater than some user-specified minimum support. Frequent Pattern Growth (FP-Growth) algorithm is one of the most popularly used data mining approach for finding frequent itemsets from large datasets [3]. But the main challenge faced by various frequent itemset mining algorithm is its execution time in distributed environment.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

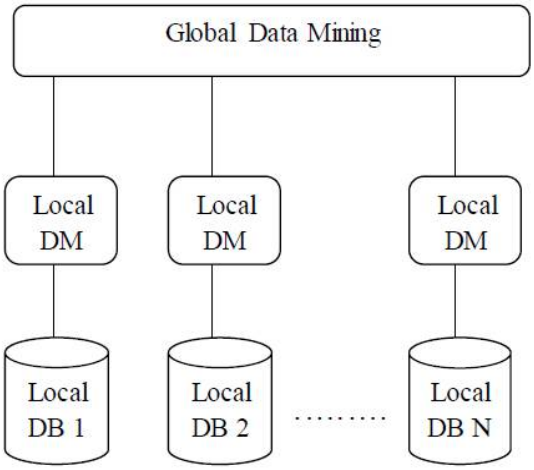

Figure 1: Architecture for distributed data mining

Distributed sources of voluminous data have created the need for distributed data mining. The conventional data mining algorithms/techniques which work efficiently on centralized databases have some limitations of its own when applied on distributed databases. In distributed data mining, data is located at distributed locations and mining is performed on every local database to find globally mined data. Figure 1 depicts the architecture for distributed data mining.

II. RELATED WORK

In the proposed system, at first the input database (i.e., shopping dataset [22]) will be splitted into a number of parts which will be equal to the number of data nodes in the Hadoop Distributed File System (HDFS). Figure 2 depicts the flow of the system in which partition of the data is given to different data nodes. For instance, if the database consists of 2000 entries and there are 4 data nodes, the data will be splitted into 4 parts each consisting of 500 entries. After allotting the data, the data nodes will store the data in a key-value pair and generates the local frequent itemsets. The generation of local frequent itemsets can be done in parallel by running a MapReduce job which will reduce the time required to find frequent itemsets.

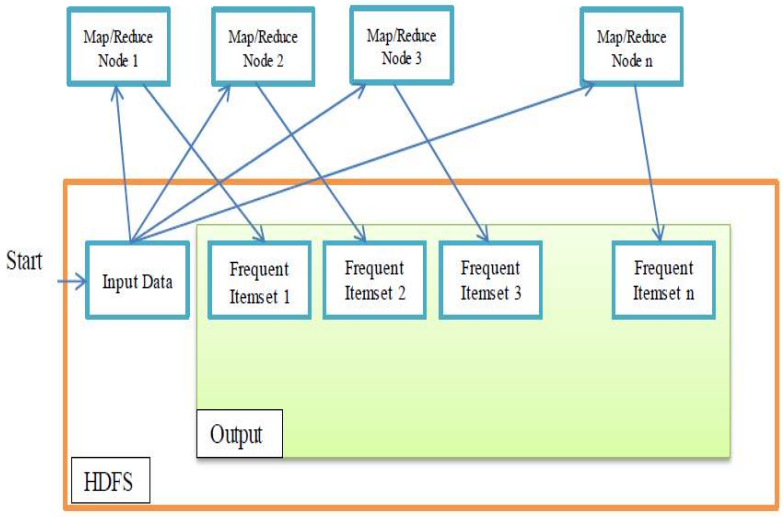

Figure 2: Flow diagram of the system

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

In the next phase, all nodes will send their local frequent itemsets to the central node which will generate the global frequent itemsets. The global frequent itemsets will be used for generation of association rules which will be used for decision making about marketing activities.

III. EXPERIMENTAL RESULTS

The main parameters to consider are performance time and communication cost. Since hadoop provides data replication and resolves the issue of node failure, the communication cost is reduced to a great extent as compared to the existing systems. The time taken by the existing system is reduced as Map/Reduce operations simplify the process of mining. The Map/Reduce operation will store the database in several key-value pairs, which will be further used for mining frequent itemsets.

The experiments were performed on a Mobile Shopping Dataset with 60366 records. The analysis was initially performed on 1-Node (as shown in Figure 4) and dataset of 10000, 20000, 30000, 40000, 50000 and 60000 records was applied to the system. The similar experimentation was performed on 2-Nodes (as depicted in Figure 6) and 3-Nodes (as shown in Figure 7).

The table depicted in figure 3 shows the time required for mining using multiple nodes with varying records of the dataset.

No. of Records	1 Node		2 Nodes		3 Nodes	
	Without Map/Reduce (ms)	With Map/Reduce (ms)	Without Map/Reduce (ms)	With Map/Reduce (ms)	Without Map/Reduce (ms)	With Map/Reduce (ms)
10000	716	90	528	88	94	31
20000	1012	130	951	102	125	32
30000	1664	219	1184	155	297	63
40000	2126	328	1178	300	328	94
50000	2702	355	1669	345	382	103
60000	3106	408	1906	386	425	125

Figure 3: Time required in milliseconds along with the nodes and dataset records

The time required for generating frequent itemsets is calculated on a dataset of 10000 records initially, then 20000 and so onupto 60000 records. Figure 4 shows the time required for mining on a single client node, and the graph clearly shows that the time required without using Map/Reduce operations is more as compared to the time required for mining using Map/Reduce.

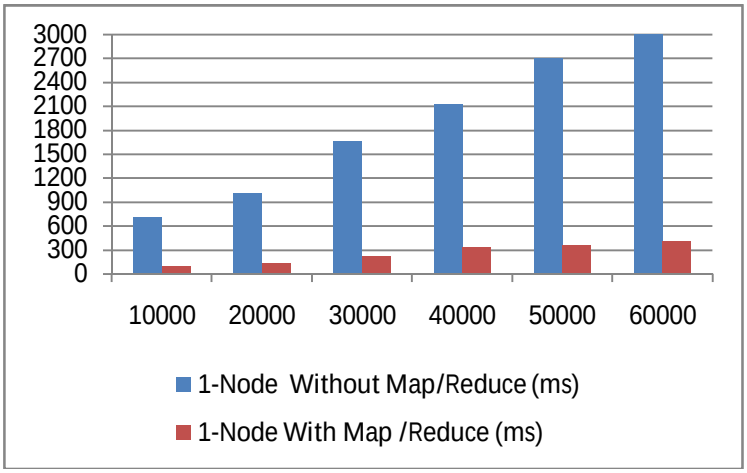

Figure 4: Time required for mining on 1-Node

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Similarly, from Figure 5 we can conclude that the time required for mining reduces if we add another client node to the system.

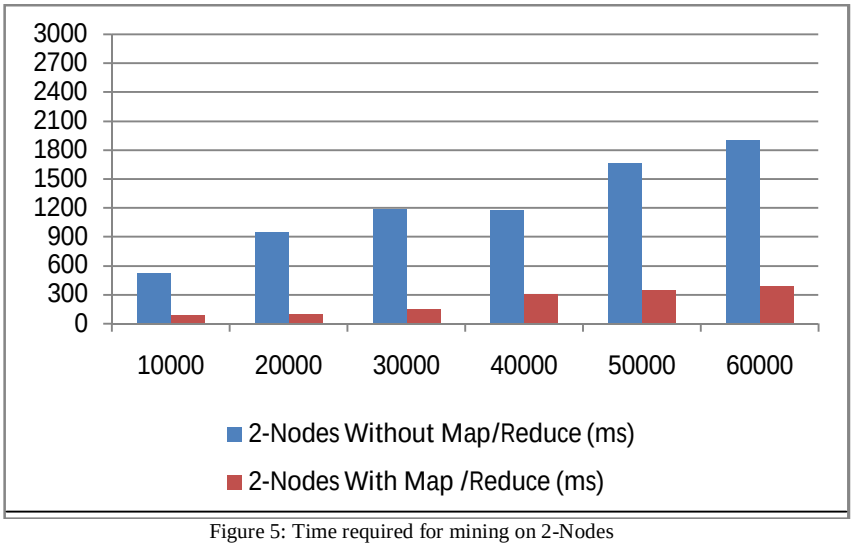

Figure 5: Time required for mining on 2-Nodes

If we further add one more client node to the system, from Figure 6 we can conclude that the time required goes on decreasing as the number of client nodes are increased.

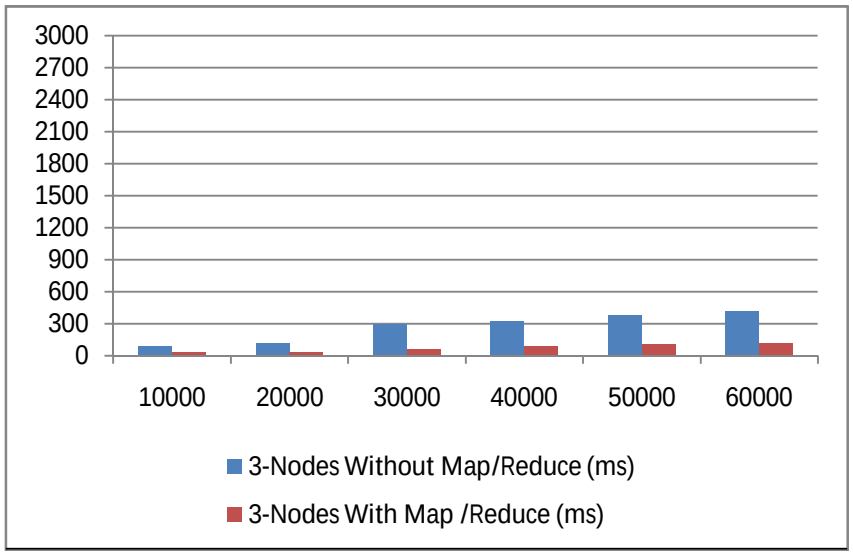

Figure 6: Time required for mining on 3-Nodes

From the experimentation analysis, it is found out that the time required to mine frequent itemsets between various ranges of records is reduced linearly as the number of nodes are increased. Also, the time required for mining frequent itemsets with Map/Reduce operations is much less as compared to the time required for mining without Map/Reduce operations.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a system to mine frequent itemsets in distributed systems using Hadoop. This system will reduce the time required for execution as it uses hadoop technology. So far, we have created Single-Node cluster and Multi-Node cluster using which the Namenode or central node is connected to multiple datanodes or client nodes. Thus, a distributed environment has been created using Hadoop. Also, the central node is able to distribute the data to the multiple client nodes for further processing. And the data received by the client nodes will be used to generate frequent itemsets at the central node by using Map/Reduce operations.

In Future, the system can be developed to generate association rules from the frequent itemsets and can be made data independent, *i.e.*, the system will work on variety of databases without making any changes to the system.

REFERENCES

1. Folorunso. O and Ogunde. A. O, "Data Mining as a Technique for Knowledge Management in Business Process Redesign," In Electronic Journal of Knowledge Management, Volume 2, Issue 1, pp. 43-54, 2004.
2. Mohammed. J. Z, "Parallel and Distributed Association Mining: A Survey," In Proceedings of Concurrency, IEEE, Volume 7, Issue 4, pp. 14-25, 1999.
3. Borgelt. C, "An Implementation of the FP-Growth Algorithm," In Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 1-5, 2005.
4. Dunham. M. H, "Data Mining: Introductory and Advanced Topics," Prentice Hall, 2002.
5. Tirumala. P and Prasad. M. K, "Distributed Count Association Rule Mining Algorithm," In International Journal of Computer Trends and Technology, Volume 1, Issue 3, pp. 370-374, 2011.
6. Cheung. D. W, Han. J, Vincent T. N, Ada W. F and Yongjian. F, "A Fast Distribution Algorithm for Mining Association Rule," In Proceedings of Parallel and Distributed Information Systems, pp. 31-42, 1996.
7. Ashrafi. M. Z, Taniar. D and Smith. K, "Optimized Distributed Association Rule Mining," In Proceedings of IEEE distributed system online, Volume 5, No. 3, pp. 1-18, 2004.
8. Ansari. E, Dastghaibifard. G. H and keshatkaran. M, "Distributed Trie Frequent Itemset Mining," In Proceedings of International Multi Conference of Engineers and Computer Scientists, Volume 1, pp. 978-988, 2008.
9. Bagheri. M, Hosseinabadi. S. M, Mashayekhi. H and Habibi. J, "Mining Distributed Frequent Itemset using Gossip Based Protocol," In Proceedings of Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC), pp. 780-785, 4-7 Sep 2012.
10. Yadav. C, Wang. S and Kumar. M, "Algorithm and approaches to handle large Data-A Survey," In International Journal of Computer Science and Network, Volume 2, Issue 3, 2013.
11. Brankovic. L and Estivill. C. V, "Privacy issues in knowledge discovery and data mining," In Proceedings of Australian Institute of Computer Ethics Conference, pp. 89-99, 1999.
12. Wei. F and Bifet. A, "Big Data: Current Status and Forecast to the Future," In Proceedings of Special Interest Group on Knowledge Discovery and Data Mining Explorations, Volume 14, Issue 2, pp. 1-5, 2012.
13. Agrawal. R, Imielinski. T, and Swami. A, "Mining association rules between sets of items in large databases," In Proceedings of Special Interest Group on Management Of Data, Volume 22, Issue 2, pp. 207-216, 1993.
14. Kim. S. H, Eom. J. H and Chung. T. M, "Data Security Hardening Methodology Using Attributes Relationship," In Proceedings of International Conference on Information Science and Applications, pp. 1-2, 2013.
15. Bodon. F, "A Trie-based Apriori Implementation for Mining Frequent Item sequences," In Journal of Association for Computing Machinery, pp. 56-65, 2005.
16. Gang. W, Zhang. H, Qui. M, Ming. Z, Jiayin. L and Xiao. K, "A Decentralized Approach for Mining Event Correlations in Distributed System Monitoring," In Journal of Parallel and Distributed Computing, Volume 73, Issue 3, pp. 330-340, 2013.
17. Mottalib. M. A, Arefin. K. S, Mohammad. M. I, Rahman. A, and Abeer. S. A, "Performance Analysis of Distributed Association Rule Mining with Apriori Algorithm," In International Journal of Computer Theory and Engineering, Volume 3, No. 4, pp. 484-488, 2011.
18. Schuster. A and Wolff. R, "Communication Efficient Distributed Mining of Association Rules," In Journal of Association for Computing Machinery Special Interest Group on Management of Data, Volume 8, Issue 2, pp. 171-196, 2001.
19. Yang. L, Zhongzhi. S, Xu L.D., Fan. L and Kirsh. I, "DH-TRIE Frequent Pattern Mining on Hadoop using JPA," In Proceedings of IEEE International Conference on Granular Computing, pp. 875-878, 2011.
20. Xiao. T, Yuan. C and Huang. Y, "PSON: A Parallelized SON Algorithm with MapReduce for Mining Frequent Sets," In Proceedings of Fourth International Symposium on Parallel Architectures, Algorithms and Programming, pp. 252-257, 2011.
21. Suhasini. A.I and Kulkarni. U. V, "Distributed Algorithm for Frequent Pattern Mining using Hadoop Map Reduce Framework," In Journal of Association of Computer Electronics and Electrical Engineers, pp. 15-24, 2013.
22. "Frequent Itemset Mining Dataset Repository", http://fimi.ua.ac.be/data [accessed August 8, 2015].