PARALLELIZING APRIORI ALGORITHM USING OPENMP AND ITS APPLICATION IN FOREST FIRE PREDICTION

Rahul Pabba
(Rahul.pabba01@student.csulb.edu)
Prudhvi Raju Madam
(Prudhviraju.madam01@student.csulb.edu)
Geethika Penukula
(Geethika.Penukula01@student.csulb.edu)
Charitha Reddy Mandli
(Charithareddy.mandli01@student.csulb.edu)

Distributed Computing Course
Computer Engineering and Computer Science Department
CSULB
Spring 2024

Instructor:
Dr. Pooria Yaghini
April 20, 2024

Contents

Abstract:

We all know that forests are home to a large variety of species of flora and fauna and the need of preserving it. A catastrophic calamity like Forest Fire burns down hundreds of acres of land and perishes all the life which are caught in it. If there is presence of human communities in the vicinity of the fire then there is loss of infrastructure as well. The traditional ways of detecting a fire through meteorological department is not accessible by remote locations and the timely updates are not always available readily. To tackle this, if we are able to find the strong factors which will contribute to the fire based on probabilistic measures then we can improve the survival rate. A popular Association Rule Mining algorithm called Apriori algorithm helps in finding various frequent itemsets in the database. The constraints for finding these itemsets are given by the user in terms of support – measured by the proportion of transactions in which an itemset appears, and confidence - measured by the proportion of transactions with an itemset, in which another item set also appears. Our project increases its efficiency with the help of OpenMP threads. We use data decomposition to split the transaction database into various parts, each taken by a thread to find the support count of all the candidate itemsets for all the transactions assigned to that particular thread. To give an example of the application, this project is used to determine the probability of the occurrence of a forest fire. Here, the transaction database can consist of various occurrences of natural phenomena, in which a few transactions also have the forest fire phenomenon, which means that it has occurred in the presence of the other item sets in the transaction. Hence, if a new transaction is taken from the user, then the probability (or confidence) that a forest fire occurs, given this transaction, is calculated

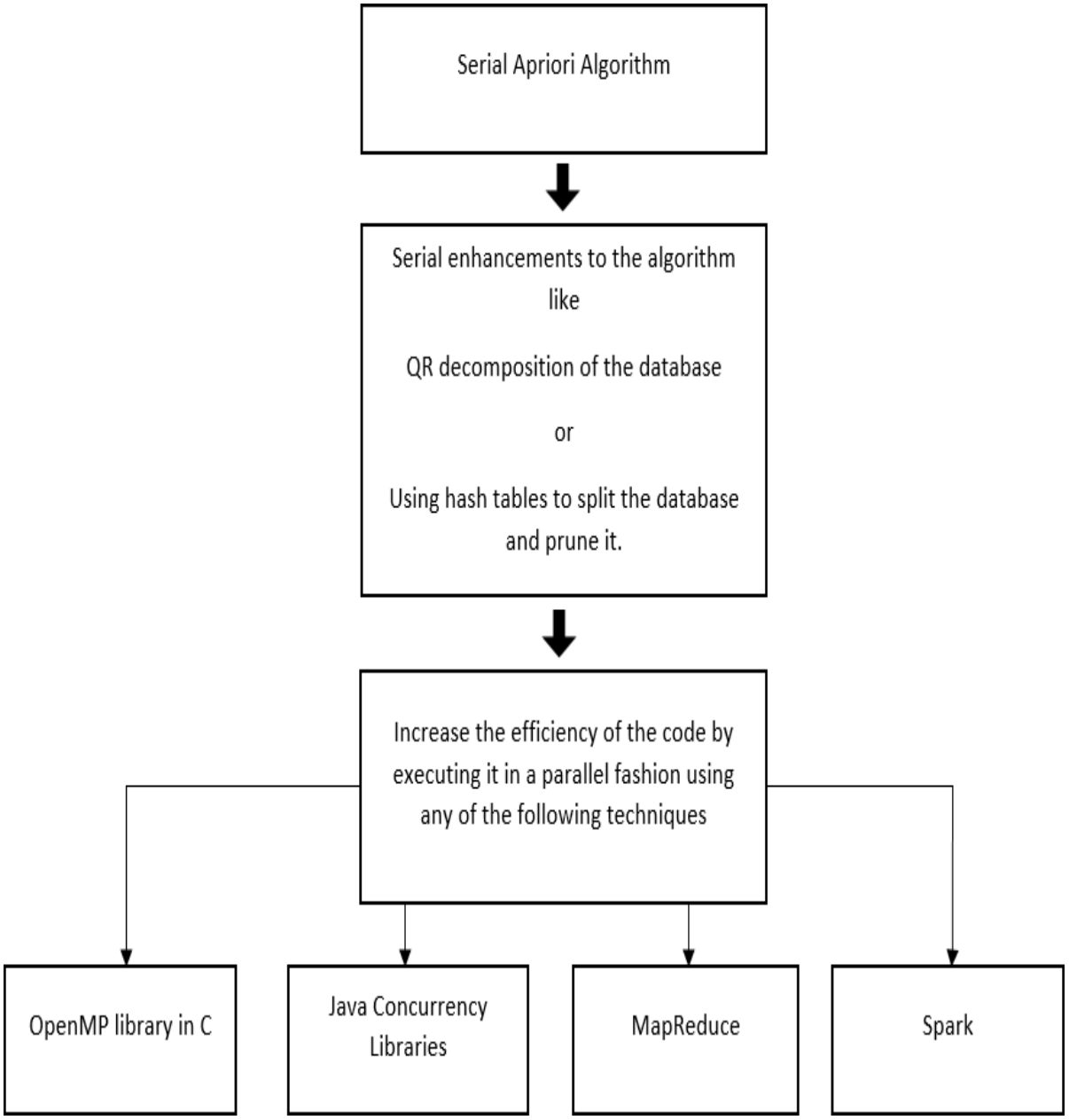
Introduction:

This is the era of data. There are huge amounts of data being generated every day. Data mining deals with the mining or the understanding of these data and deriving useful information from it. Large databases/datasets are scanned and the raw data present in them is run through various algorithms to produce something with a context, which is useful for industrial as well as research purposes. A common way of inferring something from the data is to understand the relationship between the different instances in the data or tuples in the table. This particular form of data mining is called Association Rules Mining (ARM), where associations between the different itemsets in the database are found with the help of various algorithms.

A popular ARM algorithm is called Apriori algorithm. This algorithm helps in finding various frequent itemsets in the database. The constraints for finding these itemsets are given by the user in terms of support - measured by the proportion of transactions in which an itemset appears, and confidence - measured by the proportion of transactions with an itemset, in which another itemset also appears. The problem with this algorithm is that it is highly iterative and thus its efficiency rapidly decreases with increase in size or dimension of the dataset. Our project increases its efficiency with the help of OpenMP threads. We use data decomposition to split the transaction database into various parts, each taken by a thread to find the support count of all the candidate itemsets for all the transactions assigned to that particular thread.

Suppose we start with a small data containing ‘n’ transactions then during the 1st iteration we generate n C 2 combinations and chuck out the intermediate transactions which are less than the minimal support. That means for n value of 200 we have 19,900 first generated item sets which have to be scanned individually which consumes a lot of time. Giving a few transactions to each thread and running them parallel decreases the execution time by a fraction of the number of threads taken into the code. The scope of our project is not restricted to forest fires alone. Any application which requires the analysis of large amount of data where an event is dependant of a set of known factors contributing to it in varied amounts in varied times, our project can be used to tailor the requirement of the particular application. The most famous being the Market Basket Analysis.

Schematic Representation:



Conclusion:

Implementing the Apriori algorithm in parallel decreases the scanning time of the database, enhancing the algorithm's efficiency and applicability for real-time uses. The Forest Fire prediction system is an example of such real-time application where the Apriori algorithm is employed to forecast forest fire incidents.

Time taken for executing the data with 90 transactions, 15 states with 5 as minimum support and 65% as minimum confidence is 0.0189.

References:

1. Hu, L., Zhuo, G., & Qiu, Y. (2009, August). Application of Apriori algorithm to the data mining of the wildfire. In Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on (Vol. 2, pp. 426-429). IEEE.
2. Korde, N. S., & Shende, S. W. (2014). Parallel Implementation of Apriori Algorithm. IOSR Journal of Computer Science, 01-04.
3. Chai, S., Yang, J., & Cheng, Y. (2007, June). The research of improved apriori algorithm for mining association rules. In Service Systems and Service Management, 2007 International Conference on (pp. 1-4). IEEE.
4. Changsheng, Z., Zhongyue, L., & Dongsong, Z. (2009, March). An improved algorithm for apriori. In Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on (Vol. 1, pp. 995-998). IEEE.
5. Spandana, K., Sirisha, D., & Shahida, S. (2016). Parallelizing Apriori Algorithm on GPU. International Journal of Computer Applications, 155(10).
6. Rathee, S., Kaul, M., & Kashyap, A. (2015, October). R-Apriori: an efficient apriori based algorithm on spark. In Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management (pp. 27-34). ACM.
7. Parsania, V., Kamani, G., & Ghodasara, Y. R. (2014). Mining Frequent Itemset Using Parallel Computing Apriori Algorithm.
8. Wang, G., Yu, X., Peng, D., Cui, Y., & Li, Q. (2010, June). Research of data mining based on Apriori algorithm in cutting database. In Mechanic Automation and Control Engineering (MACE), 2010 International Conference on (pp. 3765-3768). IEEE.
9. 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.
10. Shah, A. (2016, July). Association rule mining with modified apriori algorithm using top down approach. In Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on (pp. 747-752). IEEE