Analysis of Big data using MapReduce Framewok with comparison between Apriori Algorithm and FP Growth Algorithm

Vishal Juneja[1]
SIT, Lonavala
S.P.Pune University

Lahu Gavade[2]
SIT Lonavala
S.P.Pune University

Niranjan Yadav[3]
SIT Lonavala
S.P.Pune University

Abstract—Now days data is generated at high rate from different sources such as Business , Education, Research Internet Archive ,Social Sites etc .In Short data that is generated from different sources at high rate having massive volumes is known as Big Data . Earlier Centralized systems were not designed keeping Big Data needs in mind which made difficult to process Big Data on these systems. Hadoop is Parallel Distributed Infrastructure developed to store and handle Big Data.We are implementing MapReduce Framework to harness its parallel and simultaneously processing capabilities to analyze Big Data. MapReduce is Parallel Distributed Programming Paradigm that runs on HDFS and processes Big Data. In this Project we are using two algorithms Apriori and FP-Growth on MapReduce Framework for Analysing Big Data. Apriori and FP-Growth Algorithms are used to for finding association rules and frequent patterns which is nothing but knowledge which and enterprise or an individual can utilize to make profit to make better decision or to bulid a strategy which yield best result and much more We will be using online shopping data as our dataset for the particular operations.

Index Terms—Big Data,Apriori Algorithm,FP Growth,Map Reduce,Candidate Generation,Itemsets,Frequent Itemsets .

I. INTRODUCTION(BIGDATA)

Nowadays tremendous amount of data is generated by various sources such as social sites, enterprises, ecommerce, telecommunication industries,sensors etc. Big Data is a term popularly used to describe data collected from heterogeneous sources having massive volumes and in different format i.e. data having four characteristics via Big Volume ,Velocity ,Variety and Veracity is Big Data . Big Data comprises of data in structured ,semi structured and unstructured form. Earlier centralized systems were not designed keeping Big data processing needs in mind and are inefficient in handling such huge amount of data .Organization are starting to realize the importance of Big data and want to harness this data in order to make optimum use of dataset.

Mapreduce:

MapReduce is a parallel distributed framework that runs on commodity hardware and is used to process large datasets

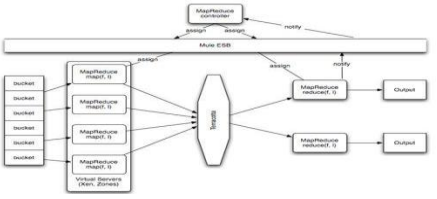

Fig. 1. Map Reduce Framework

.The computation is specified by user using two primary functions map and reduce and if needed with another optional function i.e. combine. The map function takes a set of key/value pair and generate a set of intermediate result after which a reduce function takes the intermediate values and merges them to obtain a resultant set of key value pairs

In this paper, we are compairing the basic algorithms that are used for the analysis of big data using map reduce frame-work.As per the survey and results shown apriori algorithm performs slower in comparison with fp growth algorithm .As apriori algorithm takes many more steps with respect to fp growth. Fp tree formation itself differentiates fp growth from apriori algorithm. So the user can easily take the best algorithm from the comparison to analyse big data in day to day life.

II. DATA MINING

Enterprises are maintaining huge amount of customers everyday transaction details . This data collected in data warehouses has a limited practical use unless it is properly processed to mine useful knowledge from it .Data mining is process of finding useful information from such large datasets .The knowledge extracted from these sources can be used to make strategic decisions ,target customers of interest and improve overall business performance .

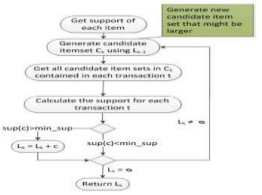

Fig. 3. Activity diagram - Apriori frequent item set generation - Notation: [27]

Fig. 2. Apriori Algorithm

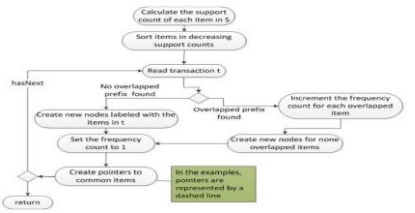

Fig. 8. Activity diagram - Construction of the FP-Tree - Notation: [27]

Fig. 3. FP Growth Algorithm

III. APRIORI ALGO RITHM

This algorithm elaborates to determine subsets which are common to atleast a minimum number of the item sets. We demonstrate frequent pattern mining based on support and confidence measures produced desired output in various fields.

LI largel-itemsets //count item frequency
for (K=2; Lkl0; k + +)
dobegin
Ck = Apriori − gen(Lk − I); //newconditions
f oralltransactionstD
dobegin
Ct = subset(Ck, t); //candidatesintransaction
f orallC andidates cC, do
c.count + +; //determinesupport
end
Lk = CCklc.count2 : minsup//createnewset
end
Result = U kLk;

Apriori Algorithm's Function:

In general, Apriori Algorithm can be viewed as a two step process[S]:
1. All item sets are generated which have support factor greater than or equal to, the user specified minimum support.
2. All rules which have the confidence factor greater than or equal to the user specified minimum confidence are generated.

IV. DISADVANTAGES OF APRIORI ALGOR ITHM

-The Candidate generation is extremely slow in some cases(pairs, triplets etc.)
-The Candidate generation could generate duplicates depending on the different implementations.
-The Counting method iterates many time with each transaction.
-Constant items and iterations make this algorithm quite heavier.
-It consumes huge memory.

V. ADVANTAGES OF APRIORI ALGORITHM

-The Apriori Algorithm calculates more sets of frequent items.

VI. FP GROW TH ALGORITHM

The FP-Growth Algorithm is one of the alternative way to find frequent itemsets without using candidate generations, thus it helps in improving performance. For so much it uses a divide-and conquer strategy. The core of this method is using a special data structure named frequent-pattern tree (FP-tree), which keeps the itemset association information. In simple words, this algorithm working as follows: first it will compress the input database creating an FP-tree instance which represents frequent items. After this step it divides the compressed database into a set of conditional databases, each one is associated with one of the frequent pattern. Finally, the FP-Growth will reduce the search costs which is looking for short patterns recursively and it then concatenating them in the long frequent patterns, offering good selectivity. In large databases, its not possible to keep the FP-tree resides in the main memory. A strategy to take with this problem is to first partition the database into a set of smaller set of databases (called projected databases), and then construct an FP-tree from each of these smaller databases.

VII. ASSOC IATION RULE

Association rule of data mining involves preference out the nameless inter-relation of the data and fmding out the rules between individual items[lO]. We inference expression of the form
P –+Q where P and Q are item-set .
For example 10, 20 O

Support: I = I), 12, 13, 1m is a collection of items. T is a collection of transactions linked with the items. Every operation has an identifier Tid [11]. We define parameter is Support (A=¿B) = Support (AUB) = P (AUB).
Confidence: The confidence defmed as a conditional probability
Confidence (A=¿B) = Support (AUB) f Support(A) = P (BfA).

VIII. FP GROWTH BOTTLENECKS

There is the biggest problem of interdependency of data. The interdependency problem is that for the parallelization of the algorithm some has to be shared, which creates a bottleneck in the shared memory

IX. COMPARISON BETWEEN APRIORI ALGORITHM AND FP GROWTH ALGORITHM

Apriori algorithm based on Hash-based items counting, when a k-item set whose corresponding hashing count is below the threshold cannot be frequent .It uses bottom-up search approaches that in step generates a frequent sequences of length n, all subsequence's have to be produces. Apriori algorithm implies that in any item-set that is potentially frequent in database must be frequent in at least one of the partition of database (DB). FP-Growth firstly creates the root of the tree, labeled with "null". FP Growth scans the database D a second time (First time when scanned, it crate l-itemset and then Ll), whenever the same node is encountered in another transaction, we only increment the support count of the common node. This transforms the problem of mining frequent patterns in database to that of mining the FP-tree.

X. PERFORMANCE BASED COMPARISON OF APRIORI AND FP-GROWTH ALGORITHM

APRIORI ALGORITHM:

Algorithm-Candidate generation with Different pruning strategies
Speedup time - Fairly High
Memory Size - All candidates holds in dataset
Scalability - High when supporting in very well
Databases(Transactions) - Transactional item-set
Efficiency - Slower because concatenate candidate generation
Scan Entire Dataset m-lterative for pattern matching
Input data-size - Exponential number of candidates
Based on increasing time - Increases
Frequency -Improve because lower support threshold

FP GROWTH ALGORITHM:

Algorithm-Demonstrate FP finds long frequent patterns short-term searching and concatenate suffix
Speedup time - Lower
Memory Size - Holds in FP tree pattern in memory
Scalability - No when support is very bad otherwise Yes
Databases(Transactions) - Tree based data-structure
Efficiency - Magnitude faster because divide and conquer methodology
Scan Entire Dataset -Twice construct for Frequent Pattern tree
Input data-size - Bushy FP tree may not fit the main memory
Based on increasing time - Reduces search time because substantially method
Frequency -Less because Descending order arrange of dataset

XI. FUTURE SCOPE

The main purpose of this system is to differentiate between diferrent analysis algorithm on parameters and apply only those algorithms which will give the fast and precise results for the big data analysis. As data is increasing day by day in this fast increasing world in every aspects so to conquer this quest this comparison makes an edge for the analysis of Big data.which will help us analyze any type of data such as weather forcasting,online shopping transactions,railways and airlines databases.

XII. CONCLUSION

This is paper is giving a discussion on brief comparison betweeen apriori algorithm and fp growth algorithm for analyizing big data and having a fast and efficient process. As per the conclusion the FP Growth is faster than apriori algorithm as it takes less steps to find the frequent itemsets

. FP Growth Algorithm produces fp-tree on the basis of minimum support count which makes it more useful than apriori algorithm.FPGrowth beats Apriori by far. It has less memory usage and less runtime. The differences are big related to others. FP-Growth is more scalable because of its linear running time. Dont think twice if you want to make a decision between these algorithms. Use FP-Growth.

REFERENCES

[1] J .Nandimath, A.Patil,E.Banerjee,P.Kakade, S.Vaidya,Big Data analysis using Apache Hadoop,in proceedings of the 2013 IEEE 14th International Conference on Information reuse Integration ,IEEE IRI,SanFrancisco,CA,United States,no.6642536,pp.700-703,August 2013.

[2] Z.Rong,D.,Z.Zhang,Complex stastical analysis of Big data :Implmentation Application of Apriori FP-Growth Algorithm Based on Map Reduce.In proceedings 4th IEEE International Conference on software engineering and service science ,ICSESS,Beijing,China,2013.

[3] D Markoins,R.Schaer,I.Eggel,H.Muller,A.Depursinge,Using Map Reduce For large scale Medical Image Analysis,in proceedings of the 2nd conference on Healthcare Informatics,Imaging Systems Biology,HISB,SanDiego,CA, United States,September,2012.

[4] D. Taniar, W. Rahayu, V. Lee, and O. Daly, Exception rules in association rules mining, Applied Mathematics and Computation, vol. 205, no. 2, 2008, pp. 735-750.

[5] T. Herawan, and M. M. Deris, A soft set approach for association rules mining, Knowledge-based systems, vol. 24, no. 1,2011, pp. 186-195.
[6] K. C. Lin, I. E. Liao, and Z. S. Chen, An improved frequent pattern growth method for mining association rules, Expert Systems with Application, vol. 38, no. 5, 2011, pp. 5154-5161.
[7] http://en.wikipedia.org. Google web page.
[8] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation,San Mateo, CA: Morgan Kaunann;2005:
[9] H. Peng, Discovery of Interesting Association Rules Based onWeb Usage Mining, International Conference, 2010.
[10] R. Mishra and A. Choubey, Discovery of Frequent Patterns from Web Log Data by using FP-growth algorithm for Web Usage Mining, International Journal of Advance Research in Computer Science and Software Engineering, vol. 2, pp. 311-318,2012.
[11] X. Bai, R. Guerraoui, A.-M. Kermarrec, and V. Leroy, Collaborative personalized top-k processing, ACM Trans. Database Syst., vol. 36, no. 4, pp. 26:126:38, Dec.2011.

