ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Performance Improvement Technique for Frequent Itemset Mining Using MapReduce

Prajakta G. Kulkarni[1], Prof. S. R. Khonde[2]

PG Student, Modern Education Society's College of Engineering College, Pune, Maharashtra, India[1]

Assistant Professor, Modern Education Society's College of Engineering College, Pune, Maharashtra, India[2]

ABSTRACT: Frequent itemset mining is a fundamental and basic work in numerous information mining applications. Extraction of frequent itemstes with standard rules helps the applications like Association rule mining, co-relations likewise in item deal and advertising. For the mining procedure of frequent itemsets there are number of techniques utilized like FP-growth-clat and so on. Regardless, deplorably the algorihm used in these techniques are not capable to balance and distribute the load, when it come across the huge information. Automatic parallelization is additionally impractical with these calculations. To remove these issues of existing techniques, there is need to build the algorithm which will support the missing elements, for example, utomatic parallelization, adjusting and great dissemination of information. This paper is focusing on a capable method to expel frequent itemsets with the popular MapReduce approach. This new approach comprises an algorithm called as Modified Apriori algorithm. This system works with three mappers, and reducers by utilizing the decompose procedure. These mappers and reducers use hash tables to improve the result. Using Hash tables in Modified Apriori exection time of jobs will be reduced.

KEYWORDS: Association Rules, Frequent item sets, Load balancing, MapReduce, Modified Apriori, Hash tables.

I.INTRODUCTION

Frequent itemset mining is a significant research subject in characterization, groupings and other fundamental data mining works. To discover frequent item sets is one of the essential computational assignments in association rule mining. As per the Association rule mining approach the frequent itemset is the itemset whose minimum support is greater than the given threshold value. These standard rules are useful for finding similar connections in the datasets and offers knowledge to the technique that produced the data [12]. Presently, there are numorous information creates from different sources like IT ventures, administrations, advances what's more, information. This vast information is available with different structures. To deal with such extreme information is exceptionally troublesome in light of the fact that it has billions of exchanges of clients, items and so on. There are number of techniques to get frequent itemsets from database. These strategies works on regular datasets, however not appropriate for massive information. To utilize frequent itemset mining strategy on extreme database is very hard process. To speed up the procedure of FIM is unpredictable and complex, because FIM expends huge time for calculatation and high input / output intencity. One of the answers for this issue is to utilize another parallel frequent itemsets mining approach with MapReduce called Modified Apriori [12]. Now a day's datasets are becoming very large and hence only sequential algorithms are not able to find the frequent itemsets correctly. Their performance gets degraded because of vast data. Therfore here used a Modified Apriori technique with the hash tables and MapReduce paradigm. MapReduce can tackle these issues with extensive number of databases crosswise over number of clusters. MapReduce will solve the parallel mining issue of large database and Modified Apriori algorithm will focus on the speed. This method improves the capacity of storage and computation of problem.

Modified Apriori algorithm utilizes the procedure of hash tables in which it stores past values to comapre with new itemsets. It utilizes buckets in hash tables for storing results. Each buckets stores frequent items and will give a unique name, hence no need to scan repetadly. FIUT is replaced with Modified Apriori. In this strategy, issues will be illuminated that are available in the current framework like data partitioning, load balancing of datasets. The working of mappers and reducers is done simultaneously to streamline the speed, well adjusting the load over different clusters [12].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

II. LITERATURE SURVAY

Association Rule Mining (ARM) for separating regular itemsets from the gigantic database the creators have exhibited an issue of finding the incessant things from expansive number of database. The creators discovered principles that have least value-based support and least certainty with new calculation. This calculation that painstakingly assesses the itemsets for one go and for every one of the information. This pruning system used to keep away from certain itemsets. Subsequently it gives right Affiliation itemsets from unnecessary databases [1][10]. The creators have displayed three calculations; those are DPC, FPC, and SPC [2], to look at fruitful executions of the Apriori calculation in the MapReduce worldview. SPC has straight-forward capacities and the FPC has static passes merged checking capacities. DPC calculation assembles the dataset of different lengths by using dynamic technique and it gives better execution. Subsequently, three calculations will scale up regarding the expanded dataset[2].

In the paper of parallel FP-growth algorithm used with MapReduce".frequent itemset mining is an exceptionally fundamental part [12]in the various information mining applications. Yet, when the dataset gets noticeably bigger, the mining algorithms can not manage such huge databases. Along these lines BPFP is used to adjust the load in PFP(an augmentation of PFP algorithm [1]), which supports parallelization and consequently this component upgrades execution. BPFP gives more noteworthy execution by utilizing PFP's gathering framework [3].

FIUT is another approach for mining frequent itemsets. It is productive framework for frequent itemset mining(FIM) called as Frequent Itemset Ultrametric Tree (FIUT)[4][11]. It has two primary stages. In the essential stage it processes with the minimum support for all itemsets a huge database. In next stage it uses pruning methodology and gives frequent itemsets. When frequent one itemsets are computed, phase two will build the ultrametric trees. These outcomes will be shown in small ultrametric trees.

Dist-Eclat, BigFIM are two kind of FIM estimation used for MapReduce System in the paper "frequent Itemset Mining for Big Data". Dist-Eclat concentrates on speed by load balancing technique using k-FIS. BigFIM transcendently focuses on hybrid approach for mining lrge databases[5]. Apriori estimation is moreover used to create k th FIS. The K th FIS is required to discover frequent itemsets in perspective of the E-clat methodology. Dist-Eclat, BigFIM and k-FIS are utilized with round robin approach.

In the technique of PARMA the authors have presented a parallel approach with Map Reduce to find the most similar itemsets. PARMA has two approaches, first it accumulates the randomized information sets and another it utilizes parallel computation, to quicken the mining system. This approach removes the replication that is especially costly. In this it divides the dataset into small random parts. Mining algorithm will be applied on the small random parts [6].

In the paper of k -nearest Neighbor, the MapReduce approach is used to show the k-nearest Neighbor i. e. KNN join, for the largest datasets. This huge data is spread on number of datanodes. In this the mappers flows the data into various data datanodes and the reducers consolidates the aftereffect of the KNN join[7]. It uses Voronoi diagram to show the similarity between the datasets to avoid the replication of items. This gives better performance while calculating the frequent itemsets.

III. PROBLEM STATEMENT

Problem statement focuses on investigation of Frequent ItemsetUltra-metricTree (FIUT) to find the issues of existing system and the efficient way for its execution in HDFS framework Implementation on FiDoop. To show that FiDoop on the cluster is sensitive to data distribution and dimensions, because itemsets with different lengths have different decomposition and construction costs. Improving energy efficiency of Fidoop running on Hadoop heterogenous clusters and to integrate FiDoop with the data-placement mechanism on clusters.To improve FiDoops performance,a workload balance metric to measure load balance across the clusters computing nodes is developed.

IV.EXISTING SYSTEM

Existing system is divided in three mapreduce levels .The first mapreduce level gets frequent one itemset. Database is divided into number of input splits and given to each mapper. Each mapper scans the databse and calculates the frequent one itemsets. Second MapReduce level applies and rescan the frequent one itemsets. Here it applies pruning

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

techniqueto to remove infrequent itemsets. This level generates k- frequent itemsets . In third mapreduce level it uses FIUT algorithm.This level decomposes the itemsets which are obtained from second phase and then it will create ultrametric tree[12].FIUT raises an issue with an item regarding dimensionality reduction from available sets. No any well balancing techniques has used, hence it shows high complexity.

V. PPROPOSED SYSTEM ARCHITECTURE AND METHODOLOOGY

Proposed system is using Modified Apriori algorithm to remove the problems of FIUT algorithm. This reduces the time required to scan the transactions. It works faster even,the support count is less. This is faster than the traditional Apriori algorithm and shows 60 percent reduced time.

Objectives of Proposed System:

- To develop a system using hybrid algorithm which is based on Modified Apriori and hash tables for frequent itemset mining.
- This system works with heterogeneous cluster nodes.
- Increased performance and accuracy with automatic parallel processing. It takes less time to scan the database.

In this paper three MapReduce phases are used. The input database is given to the mapper of the first phase of Map Reduce. Its output is obtained from reducer, in the form of frequent one itemset. This candidate 1 itemset shows the completion of the first phase of MapReduce [12]. Second phase accepts the input from first phase in the form of frequent 1- itemsets. Second phase of MapReduce scans all the data to give frequent k-itemsets. It applies pruning technique, to discover frequent and infrequent itemsets. The result is obtained from reducer, in the form of k- frequent itemsets.

Proposed System implements a new data partitioning method with balancing the load among the cluster nodes. Following figure shows the architecture of the proposed system.

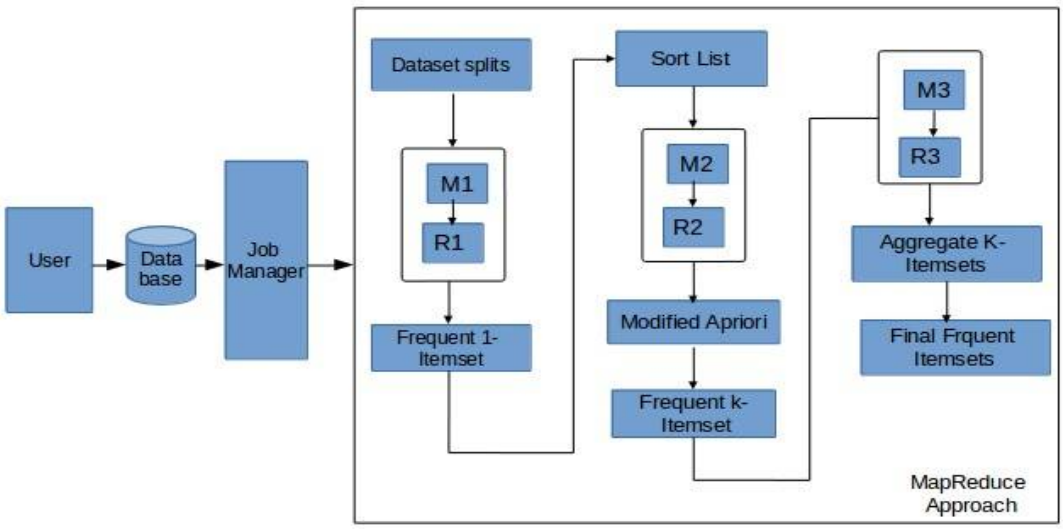

Fig.1System architecture

System architecture is based on HDFS framework and Mapreduce Approch. It uses HDFS which is a hadoop distributed file system and it stores the log files required for HDFS. HDFS framework accepts the data from user. This is done by Jobmanager. Jobmanager destributes the jobs to the datanodes. Each datanode accepts the job from the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Namenode, computes it using MapReduce parallel mining paradigm. Final results will again sent to the jobmanager in the reduced form. As shown in fig. 1 the architecture uses the MapReduce approach. When the jobs distributed among the datanodes ,the working of mappers and reducers will be as follows: When database upload is done the first MapReduce accepts the dataset. All dataset will be split into available mappers and the output is combined in the form of frequent 1 itemsets from the reducer. This output will be in the form of key and values. This frequent-1 itemsets is is loaded to the next mapreduce . This MapReduce scans the data again and remove the items which are not frequent. Here in each phase of MapReduce the Modified Apriori will be used to calculate the candidate itemsets.

Modified Apriori used with hash tables. When the new itemsets will come from the user it is store in the hash table. Different values are given to the each items and store in the hash table. Hash value is calculated as per the order of items and will be stored in hash table as per their accurances. Depending upon the hash value the items will be stored in the table and the count of place will be increased and given to the items . This value is assumed as its count and will be compared against the minimum-support. If the place value is greater than the minimum-support then it is termed as frequent itemset.This procedure is used in thre MapReduce jobs so that the candidate itemsets will be minimized. Automatically there is reduced frequent itemsets extraction time will be reduced if there is minimized candidate itemsets. Hence the speed is achieved through the reduced candidate itemsets. Frquent item sets extraction time is reduced by using hash based technique.

ALGORITHMS:

1) Frequent- 1 itemset Algorithm:
Input: Dataset D , Min support
Output: 1 candidate item set

Mapper (items, DB)
Step 1: input dataset with k
Step 2: Generate mappers with m maps
Step 3: get all items from transaction T
Step 4: For item in items
Process item as output
End for
Reducer (Key items, Value 1)
Step 1: take all ietms from list
Each list form each mapper
Step 2: extract relevant set from items
Create countsum for al ietms
Step 3: if (countsum¿minsupport)
Add items into itemset list with transaction
End for.

2) Modified Apriori Algorithm:
Step 1: Input dataset D and set min support,hash table,hash_place
Step 2: Generate items, for all transactions
Step 3: use a function f1 to store frequent itemsets from dataset
Step 4: for each iteam I to n, i=2
Each kth iteration items which having min support
end for
Step 5: create all items which are equals with k itemset list n hash table

Step 6: Store the result in a variable fr_output of hash table
Step 7: compare the result with min support,increased by 1 and stored in new input

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Step 8:Modify the list l
Step 9:use if to check the hash tables raised 1 or 0
Step 10: if 1 then add into fr_output else discard
Step 11: fr_output= f1(new itemsets)
Step 12: modify the list k-1 items reach
Step 13: Final top most frequent itemsets.

V. IMPLMENTATION

The implementation of the system is as shown in fig, 2 , fig.3 and fig. 4. Systems start with lofin id of users for the system authentication purpose. In this paper three MapReduce phases are used. The input database is given to the mapper of the first phase of MapReduce.

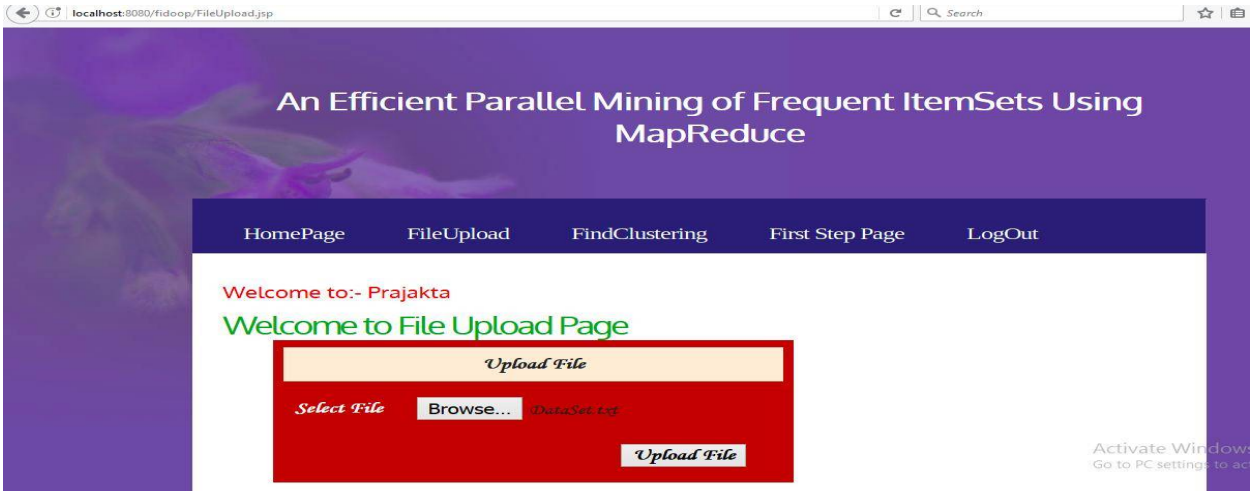

Fig.2 File Upload

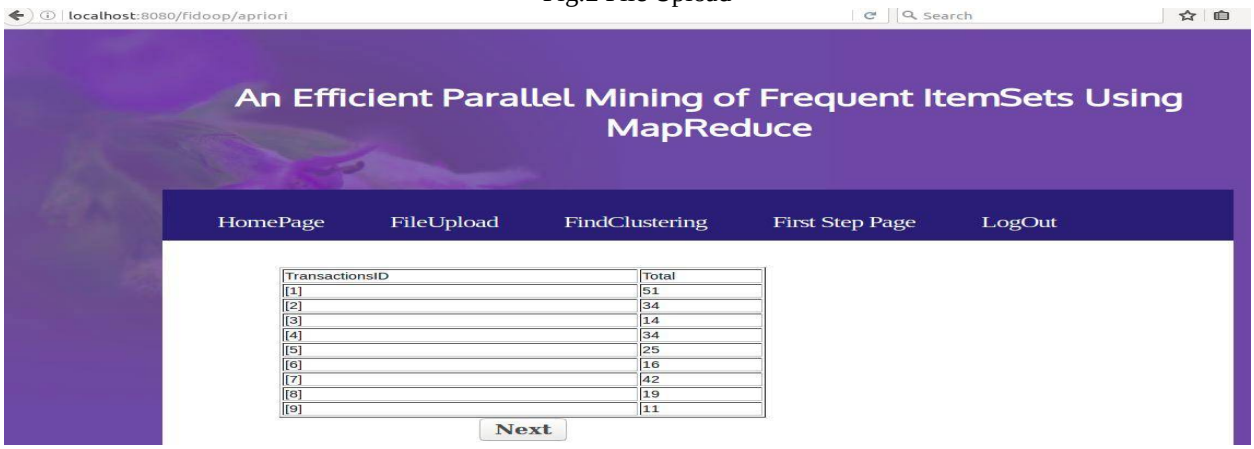

Fig. 3 Frequent-1 Itemsets

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

This database will be accepted by using the option"upload file" as shown in fig2. Its output is obtained from reducer, in the form of frequent one itemset. This candidate 1 itemset shows the completion of the first phase of MapRe duce [12]. The implementation of Frequent 1- itemset is in Fig.3. Second phase accepts the input from first phase in the form of frequent 1- itemsets. Second phase of MapReduce scans all the data to give frequent k-itemsets. It applies pruning technique, to discover frequent and infrequent itemsets. The result is obtained from reducer, in the form of k-frequent itemsets. The implementation of k-frequent itemsets is in Fig.4.

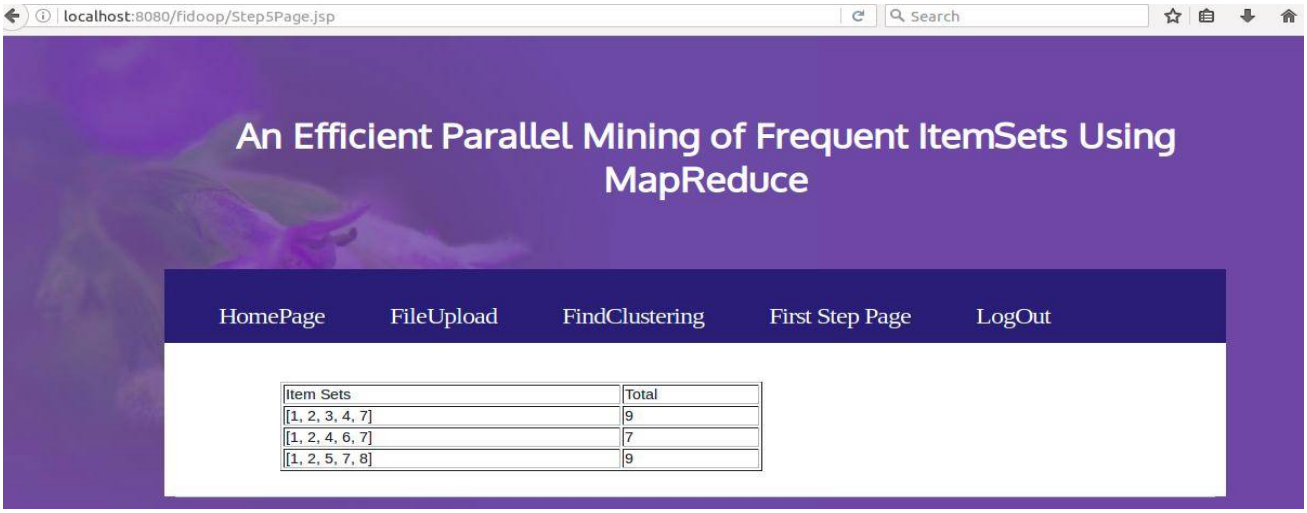

Fig. 4 Frequent-k Itemsets

VI. RESULTS AND DISCUSSION

Following figure 5.shows the comparative results of the existing methodology FIUT and proposed technique and are Minimum support is set as threshold than having even yet been considered. Here time is in miliseconds and this shows that FIUT takes more time than the Modified Apriori. These overhauls came at no execution cost, as demonstrate by the way that our usage accomplishes the execution contrast with different techniques less time.Prior to this work, it has been acknowledged that the execution of Apriori is moderate. Through this examination we have exibited that at increased support our execution creates similar outcomes. When we decrease threshold value ,other methods takes larger memory as well as more time.

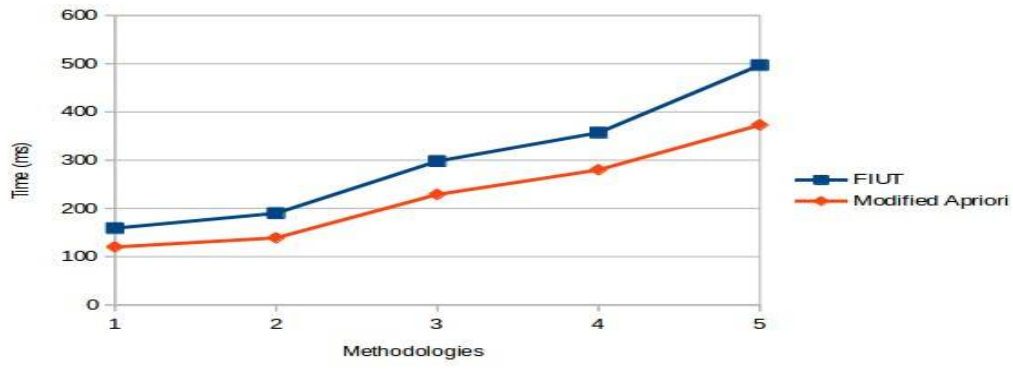

Fig.5 Cmparision Graph of two methodologies

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

TABLE I
TIME REQUIRED FOR FINDING THE FI FROM DATASET

Size of Dataset (Number of Records)	FIUT	Modified Apriori
1000	159	120
2000	190	139
3000	298	229
5000	357	280
10000	497	373

Table I shows the time required to execute the jobs for MapReduce approach. Time is miliseconds for FIUT and Modified Apriori.

VII.CONCLUSION

To defeat the timing issue for execution of lage datasets on Hadoop systems and load balancing challenges in the existing parallel mining algorithms for frequent itemsets, the system executed the MapReduce programming approach to develop a parallel frequent itemsets mining algorithm called Modified Apriori. Modfied Apriori used with hash tables. So hash tables stores the record of previous frequent itemsets. For new input has techniq compared with previous items. Hence it removes the issues which are presen in the existing algorithm and reduces time, achieves speed with MapReduce approach. Modified Apriori is used throughout the stages of MapReduce to calculate frequent-1 itemset upto the frequent-k itemsets. It increasees the CPU enerey efficiency upto 60 % and gives better performance.

REFERENCES

[1] R. Agrawal, T. Imieli´nski, and A. Swami, “Mining association rules between sets of items in large databases,” ACM SIGMOD Rec., vol.22,no. 2, pp. 207–216, 1993.
[2] M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, “Apriori-based frequent itemset mining algorithms on MapReduce,” in Proc. 6th Int. Conf. Ubiquit. Inf. Manage. Common. (ICUIMC), Danang, Vietnam, 2012, pp. 76:1–76:8. [Online].
[3] L. Zhou et al., “Balanced parallel FP-growth with MapReduce,” in Proc. IEEE Youth Conf. Inf. Compute. Telecommun. (YC-ICT), Beijing, China, 2010, pp.243–246.6.
[4] Y.-J. Tsay, T.-J. Hsu, and J.-R. Yu, “FIUT: A new method for mining frequent itemsets,” Inf. Sci., vol. 179, no. 11, pp. 1724–1737, 2009..
[5] Kiran Chavan, Priyanka Kulkarni, Pooja Ghodekar, S. N. Patil,” Frequent itemset mining for Big data ”, IEEE,Green Computing and Internet of Things (ICGCIoT), pp. 1365–1368, 2015
[6] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, “PARMA:A parallel randomized algorithm for approximate association rules mining in MapReduce,” in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.,Maui, HI, USA, pp. 85–94 , 2012.
[7] Wei Lu,Yanyan Shen,Su Chen,Beng Chin Ooi,“Efficient Processing of k Nearest Neighbor Joins using MapReduce”2012 VLDB Endowment 2150-8097/12/06,Vol. 5, No. 10,pp.1016-1027.
[8] Shekhar Gupta, Christian Fritz, Johan de Kleer, and Cees Witteveen,Diagnosing Heterogeneous Hadoop Clusters,2012, 23 rd International Workshop on Principles of Diagnosis
[9] Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, Ningfang Mi, Self Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters DOI 10.1109/TCC.2015.2415802, IEEE Transactions on Cloud Computing
[10] He Lijun. ”Comparison and Analysis of Algorithms for Association Rules”, 2009 First International Workshop on Database Technology and Applications, 04/2009
[11] Yaling Xun, Jifu Zhang, Xiao Qin,FiDoop-Dp:Data Partitioning in Frequent Itemset Mining on Hadoop clusters,2016
[12] Yaling Xun, Jifu Zhang, and Xiao Qin, FiDoop: Parallel Mining of Frequent Itemsets Using MapReduce IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 46, NO. 3, MARCH 2016
[13] Ferenc Kov acs, Janos Illes, Frequent itemset mining on Hadoop, ICCC 2013,IEEE 9th International Conference on Computational Cybernetics, July 8-10, 2013, pp. 241-245.
[14] S Deshpande, H Pawar, A Chandras, A Langhe ,Data Parti- tioning in Frequent Itemset Mining on Hadoop Clusters- 2016 irjet.net,https://irjet.net/archives/V3/i11/IRJET-V3I11229.pd