International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, November 2015
www.ijates.com
ijates
ISSN 2348 - 7550

PARALLEL ROUGH SET APPROXIMATION USING MAPREDUCE TECHNIQUE IN HADOOP

Ms.Suruchi V. Nandgaonkar[1], Prof.A.B.Raut[2]

[1]Student ME CSE, [2]Associate Prof. HVPM COET Amravati Maharashtra, (India)

ABSTRACT

Rough set theory can be considered as a tool to study the uncertain, indescendent data by classifying the set into ternary classification as lower, upper and boundary region. This paper helps explaining the rough set theory using map-reduce in Hadoop. Some features can reduce the number of input dimension, so that the process can be executed in comparative less time. Using this method this can help many organizations that are facing problem related to the data handling and data analysis. Existing method calculates rough set approximation in serial way. Therefore a parallel method used Map-Reduce have developed to manage many large-scale computations. Recently introduced Map-Reduce technique has received much consideration from both scientific community and industry for its applicability in big data analysis. The effective computation of approximation is essential step in improving the performance of rough set. For mining the massive data, parallel computing modes, algorithms and different methods get used in research fields. In this paper, we have explained a parallel method for computing rough set. Using map-reduce we can achieve the same. Because of map-reduce we can generate rules and abstract attributes of massive data. To mine knowledge from big data, we present consequently algorithm corresponding to the MapReduce based on roughest theory are put forward to deal with the massive data.

Keywords*: *Data Mining, Mapreduce, Hadoop, Rough Set

I. INTRODUCTION

Handling the huge amount of data has been the serious problem now days and with this massive data comes the application associated with it such as association rule mining, sequential pattern mining, text mining and temporal mining. But, with the emergence of hadoop implementation into the map-reduce technique we can solve the problem of the many organization that they are facing with it. From last few decenniums, the size of the data stored in the databases has been increasing each day and therefore we face lots of difficulties about obtaining the worthful data. It has become difficult to reach accurate and useful information as the data stored in the databases is growing each day. To find out the rules or interesting and useful patterns within stored data in the databases, data mining techniques are used. Storing huge amount of increasing data in the databases, which is called information explosion, it is necessary to transform these data into necessary and useful information. Using conventional statistics techniques fail to satisfy the requirements for analyzing the data. Data mining is a nontrivial process of determination of valid, unknown and potential useful and easily understandable dependencies in data. With the development of information technology, data volumes processed by many applications will routinely cross the peta-scale threshold, which will in turn increase the computational

requirements. For massive data is always a hot topic in data mining data mining is the deficiency and indeterminateness. This problem is solved by using new theories and procedures, for example fuzzy sets, genetic algorithms or rough sets. Data processing and knowledge discovery for massive data has always been an active research area in data

Dean and Ghemawat from Google firstly presented a parallel programming model in MapReduce, which was a framework for processing huge data sets on certain kinds of distributable problems using a large number of computers (nodes), collectively referred to as a cluster. MapReduce is a popular computing model. MapReduce framework offers clean abstraction between data analysis task and the underlying systems challenges involved in ensuring reliable large-scale computation. Finally, the MapReduce runtime system can be transparently explore the parallelism and schedule these components to distribute resource for execution.

II. ROUGH SET THEORY

Rough set theory was proposed by Zdzislaw pawlak in1982.This method is considered as one of the non-static approach in data analysis and deals with the uncertain, indescendent and incomplete data.

Finding rough approximations comes in following steps.

- Finding decision set
- Equivalence class
- Association class

Lower, Upper and Boundary approximations

Consider a set X is a subset of U. We want to characterize the set with respect to Y using rough set theory then,

Lower approximations: It is a set of items which can be definitely classified as X.

Upper approximation: It is defined as a set of item which can be possibly classified as item of X**.**

Boundary Approximation: It is a set of items which can be classifies either as items of X or may not be.

Consider the following example for the better understanding of data. In the following example, first column refers to the objects while the next columns such as 'Income', 'Student 'and 'Credit Rating' are the attributes while the last column refers to the decision of the attribute.

In the below table we can see person X3 and X3 have same attributes still the results are different. we call this type of data as indescendent or inconsistent data. In such situations rough set proves very useful.

Decision class Yes=[X3,X4,X5,X7,X9,X10,X11,X12,X13]

Decision class No=[X1, X2, X6, X8, X14]

Equivalence class: Eq1=[X1, X3] Eq2=[X6, X7] Eq3=[X4, X8] Eq4=[X5, X9] Eq5=[X6, X7] Eq6=[X10] Eq7=[X11] Eq8=[X12, X14]

Association class: We compare each of the equivalence class with the decision class.

Approximations:

Lower approximation = [{X5, X9},{X10},{X11}]

Upper Approximation=[{X2}]

Boundary region= [{X2,{X6,X7},{X12,X14},{X1,X3},

{X4, X8}]

The regularity hidden in the data are generally expressed in terms of the rules. Which are the basic tool of data mining .Rule are expressed of the form If (attribute1, value1) and (attribute2, value2) and Then (decision value) The number of consistent rule in the table is known as factor of consistence which is denoted by $_{\gamma}$ (C,D)=1. Here, C is the condition and D is the decision, if $_{\gamma}$(C, D)≠1 then the table is not consistent.

Table 1: An Example of Data Set (Decision Table S)

Person	Income	Student	Credit rating	Buys computer
X1	High	No	Fair	No
X2	High	No	Excellent	No
X3	High	No	Fair	Yes
X4	Medium	No	Fair	Yes
X5	Low	Yes	Fair	Yes
X6	Low	Yes	Excellent	No
X7	Low	Yes	Excellent	Yes
X8	Medium	No	Fair	No
X9	Low	Yes	Fair	Yes
X10	Medium	Yes	Fair	Yes
X11	Medium	Yes	Excellent	Yes
X12	Medium	No	Excellent	Yes
X13	High	Yes	Fair	Yes
X14	Medium	No	Excellent	NO

2.1 Advantages of Implementing Rough Set in Hadoop

Design for huge data: Hadoop is mainly design to handle and process a large amount of data systematically and using parallel method.

1. Reliability: It achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAID storage on hosts .Replication is by default set to 3.

2. Streaming pattern: It can handle lots of streaming reads and infrequent writes.

3. Not fully POSIX: HDFS is not completely POSIX (Portable Operating System Interface)-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application.

4. Increased Bock size: The increase in the block size (64MB) results in handling reduced numbers of blocks and fast processing.

5. Parallel processing: The rough set that we are using is parallel in nature and again parallel processing of it will reduce time processing at much extend.

MAPREDUCE

Map-Reduce allows for distributed processing of the Map and Reduce functions. The Map-Reduce divides the input file into no of blocks by method “input split”. It is used for processing data on commodity hardware.

Step 1: Input reader:

It divides the input into appropriate size splits i.e. blocks and assigns one split to each map function. The input reader reads data from stable storage and generates key/value pairs.

Step 2: Mapper

It processes a key/value pair and generates another key/value pair. A number of such map functions running in parallel on the data that is partitioned across the cluster, produce a set of intermediate key/value Pairs. Mapk, v → <k′, v′ >

Step 3: Compare function

The input for each reduces is pulled from the machine where the map ran and stored using the applications *comparison* function. Computek′, v′ → <k′, v′ >

Step 4: Partition Function

The *partition* function is given the key and the number of reducers and returns the indexes of the desired reduce. It is important to pick a partition function that gives an approximately uniform distribution of data reducers assigned more than their share of data for load-balancing operation to finish.

Step 5: Reducer

The *reduce* function then merge all intermediate values that are associated with the same intermediate key. Reducek′, v′ → <k′, v′ >

Step 6: Output

Map-Reduce allows developers to write and deploy code that runs directly on each data-node server in the cluster. That code understands the format of the data stored in each block in the file and can implement simple algorithms and much more complex ones. It can be used to process vast amounts of data in-parallel on large clusters in a reliable and fault tolerant fashion. Consequently, it renders the advantages of the Map/Reduce available to the users

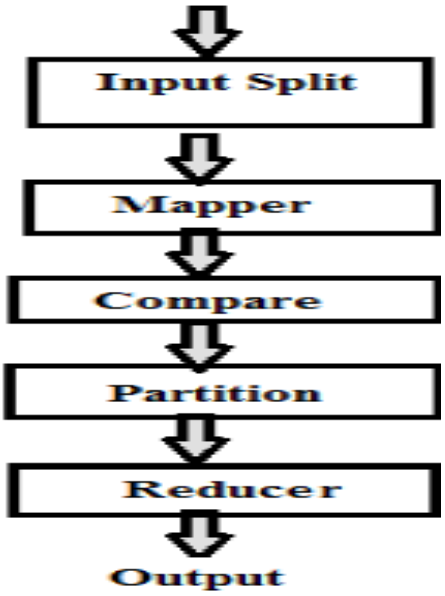

Fig 1: Programming Model for Map Reduce

The rough set approximations obtained by the parallel method are the same as those obtained by the serial method. But using map reduce we can run independent phases in parallel as computing equivalence class, computing decision class, constructing associations based on map-reduce. Therefore time required is very less as compared to traditional method of rough set calculation. In addition to that we can also generate the rules for massive data and able to abstract attributes in more efficient way using map-reduce with rough set.

III. PARALLEL ROUGH SET SYSTEM

Existing method for calculating rough set performs in-memory processing. Firstly, it calculates the equivalence class and then decision class. And at the last approximation of decision class calculated. Existing method calculates the rough set serially. It cannot deal with large data sets. Therefore we have used parallel method performs on the Hadoop platform which may remove the data size limit due to transparent spilling. Data partitioning, fault tolerance, execution scheduling is provided by MapReduce framework itself. MapReduce was designed to handle large data volumes and huge clusters (thousands of servers). MapReduce is a programming framework that allows executes user code in a large cluster. All the user has to write two functions: Map and Reduce .During the Map phase, the input data are distributed across the mapper machines, where each machine then processes a subset of the data in parallel and produces one or more <key; value> pairs for each data record. Next during the Shuffle phase, those <key, value> pairs are repartitioned (and sorted within each partition) so that values corresponding to the same key are grouped together into values {v1; v2; :::}. Finally, during the Reduce phase, each reducer machine processes a subset of the <key, {v1; v2; :::}> pairs in parallel and writes the final results to the distributed file system. The map and reduce tasks are defined by the user while the shuffle is accomplished by the system .Even though the former pseudo code is written in terms of string inputs and outputs, conceptually the map and reduce functions supplied by the user have associated types.

Map (k1, v1) → list (k2, v2)

Reduce (k2, list (v2)) → list (v2)

Two programmer specified f• Map Input: key/value pairs (k1, v1)

Output: intermediate key/value pairs list (k2, v2)

• Reduce Input: intermediate key/value pairs (k2, list (v2))

Output: List of values list (v2)

That is, the input keys and values are drawn from a different domain than the output keys and values. The k1,k2 are the two different keys used in MapReduce phase and same as v1, v2 are the different values. The intermediate keys and values are from the same domain as the output keys and values

In this system we can compute both rough set equivalence classes and decision classes in parallel using the Map-Reduce technique. The associations between equivalence classes and decision classes of the decision table can also be executed in parallel. Lower and upper approximations are computed by associations between equivalence classes and decision classes. However, if we compute the approximations directly, memory may overflow since equivalence classes and decision classes both contain too many objects while dealing the large data set. Hence we can compute the Indexes of Rough Set Approximations i.e. the set of information for each decision class.

After computing the indexes of approximations, we output the approximations directly. The diagram of the parallel method for computing rough set approximations is shown below.

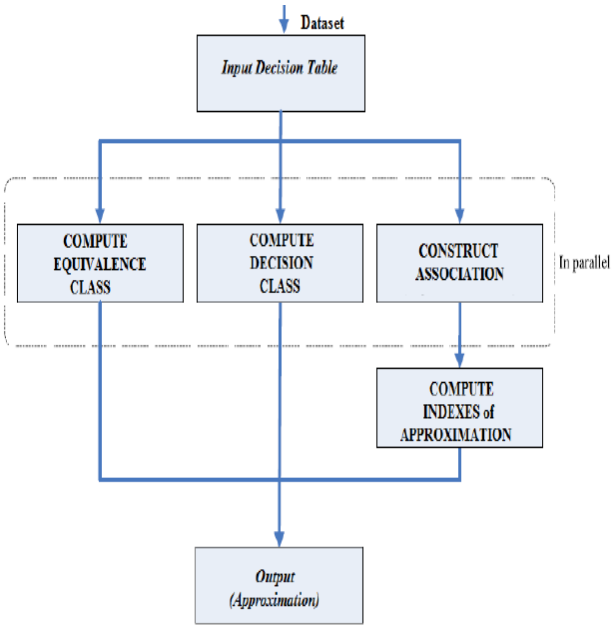

Fig 2: Method to Calculate Approximation Based on Map Reduce

The rough set approximations obtained by the parallel method are the same as those obtained by the serial method. But using map reduce we can run independent phases in parallel as computing equivalence class, computing decision class, constructing associations based on map-reduce. Therefore time required is very less as compared to traditional method of rough set calculation. In addition to that we can also generate the rules for massive data and able to abstract attributes in more efficient way using map-reduce with rough set. Map-Reduce framework offers clean abstraction between data analysis task and the underlying systems challenges involved in ensuring reliable large-scale computation. Map-Reduce runtime system can be transparently explore the parallelism and schedule components to distribute resource for execution.

3.1 Characeristics of Parallel Rough Set System

We can measure the performance of this system using three characteristics as described below.

• Speed up:

To measure the speedup, we keep the data set constant and increase the number of nodes (computers) in the system. Speedup given by the larger system is defined by the following formula :

Speedup (p) = T1 / Tp;

where p is the number of nodes (computers), T1 is the execution time on one node, Tp is the execution time on p nodes. We can perform the speedup evaluation on data sets with quite different sizes and structures. The number of nodes (computers) varied from one to many. In Serial existing system with p times the number of computers yields a speedup of p. However, linear speedup is difficult to achieve because the communication cost increases with the number of clusters becomes large.

• Scale up:

Scale up is defined as the ability of a p-times larger system to perform a p-times larger job in the same execution time Scale up (D, p) = TD1/TDp where D is the data set, TD1 is the execution time for D on one node, TDp is the execution time for p × D on p nodes .To check whether the proposed system handles larger data sets when more nodes are available. Therefore we can perform scale up experiments, where we can increase the size of the data sets in direct proportion to the number of nodes in the system

• Size up:

Size up is defined as the following formula :

Size up (D, p) = TSp/ TS1

where TSp is the execution time for p × D, TS1 is the execution time for D.Size up analysis holds the number of computers in the system constant, and grows the size of the data sets by the factor p. Size up measures how much longer it takes on a given system, when the size of data set is p-times larger than that of the original data set.

IV. LIMITATIONS

Hadoop is not meant for storing large number of small files.

1. HDFS is not suitable to efficiently access small files: it is primarily designed for streaming access of large files. Reading through small files normally causes lots of seeks and lots of hopping from data node to data node to retrieve each small file, which is an inefficient data access pattern?

2. Every file, directory and block in HDFS is represented as an object in the name node's memory, each of which occupies 150 bytes. The block size is 64 Mb. So even if the file is of 20 kb, it would be allocated an entire block of 64 Mb. That's a waste disk space

V. CONCLUSION

Up till now, many rough sets based algorithms have been developed for data mining. But enlarged data in applications made these algorithms based on rough sets a challenging task. Computation of rough set approximation is very important step. We can improve the quality and speed of calculating approximation. This is one way where we have lots of opportunities to achieve speed and accuracy.

In this paper, we proposed a parallel method for rough set. Using map-reduce we can achieve the same. Because of map-reduce we can generate rules and abstract attributes of massive data. Future work will involve the parallel frequent pattern mining exploration of an alternative method that calculate the attribute space, so that information systems with a large number of attributes, such as those used in mathematical, may be analyzed effectively.

VI. ACKNOWLEDGMENT

I am thankful to all the teachers and the principal for their valuable guidance. I would like to sincerely thank Prof. Anjali Raut Madam whose knowledgeable guidance helped me to make more descriptive. I would also like to thank my parents and friends for their constant support.

REFERENCE

[1] Ing.Pavel Jurka, Using rough set in data mining.

[2] Z. Pawlak, W.Ziarko CommunicationofACM,38,1995,p.88

[3] Zdzislaw.pawlak,Rough set theory and its applications ,Journals of telecommunication and information technology.

[4] DevarajDas,Owen O'Melly,Sanjay Radia,KanZang,Adding security to apache hadoop, HortonworksTechnical Re-port,www.Hortonworks.com

[5] Junbo Zhang a, Tianrui Li , Da Ruan, ZizheGao, Chengbing Zhaoa,' A parallel method for computing rough set approximations', J. Zhang et al./ Information Sciences 194 (2012) 209–223

[6] A.Pradeepa1,Dr.AntonySelvadossThanamaniLee2,'hadoop file system and Fundamental concept of MapReduce interior and closure Rough set approximations', International Journal of Advanced Research in Computer and Communication EngineeringVol. 2, Issue 10, October 2013

[7] 'Rough set based decision support' Roman Slowinski Institute of Computing Science Poznan University of Technology and Institute for Systems Research Polish Academy of Sciences Warsaw, Poland

[8] MertBaMathematical Engineering Department, Yildiz Technical University, Esenler, İstanbul, TURKEY 'Rough Sets Theory as Symbolic Data Mining Method: An Application on Complete Decision Table

[9] J. Han, M. Kamber, Data Mining: Concepts and Techniques, second ed., Morgan Kaufman, San Francisco, 2006.

[10] P.Y. Hsu, Y.L. Chen, C.C. Ling, Algorithms for mining association rules in bag databases, Information Sciences 166 (2004) 31–