iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** X **Month of publication:** October 2017

DOI: http://doi.org/10.22214/ijraset.2017.10124

Evolution Mining: A Novel Accelerated Framework for Large Data processing in the big Data Paradigm

Mrs.M.Rajalakshmi[1], Ms. A. Kalaiselvi[2]
[1]Assistant Professor, Dept. of Master of Computer Applications, Sankara College of Science and commerce, Coimbatore
[2]Research Scholar, Sankara College of Science and commerce, Coimbatore

Abstract: The big data focus on the mining of the large volume, complex and exploiting datain the large scale application and Social network communities. The map reduce paradigm is been used to handle large scale data in the big data paradigm through classification and clustering algorithm as it is treated as computational intensive task. The Data Classification and Clustering algorithm using supervised and unsupervised learning models failed in processing dynamic updates as concept and feature evolution to map reduce paradigm in the large streaming application. Therefore it becomes mandatory to build a new accelerated framework which is capable of handling of the data evolution in the large data. In this Paper, we propose a novel technique named as "EvolutionMining" which develop a deep learning model in the map reduce paradigm using key value pair processing &Semantic Mining techniques. Sophisticated iterative models for feature extraction, classification and feature or concept evolution determination been devised. The proposed Technique is capable of handling the diverse data by producing the increased value to the computation characteristics such as accuracy and complexity. The Experimental results prove that proposed System outperforms other state of approaches using evaluation metrics such Runtime &Mean Error and reduces I/O overhead to much extent.

Index Terms – Big Data, Map reduce Paradigm, Key Value Pair, Incremental Processing

I. INTRODUCTION

Recently a lot of data updates and data evolution is exploiting in the cloud based server which consider as big data problem. Big Data is manifested in three different issues such as velocity, Variety and Volume of the data handling and analysing. The data handling and analysis leads to integration problem, computation problem, data placement problem, and finally Memory related problems. The Map Reduce paradigm is employed to handle large volume of data with high velocity. The map reduce functions used in production because of its simplicity, generality, and maturity.

data is constantly evolving and exploiting in different kind of applications. As new data and updates are being collected, the input data of a big data mining algorithm will gradually change, and the computed results will become more complex and less reliable. The traditional data mining approaches such data classification and clustering which are based on the full batch-mode learning may run short in meeting the demand of analytic efficiency[1]. The Learning model may be supervised or unsupervised. During new data updates arrival to the distributed resource of cluster, data management into the resulting cluster is typical in the data collection process that makes the big data inflate to bigger data, the traditional induction method needs to re-run and the model that was built needs to be built again with the inclusion of new data with evolution in terms of concept and feature. Example of Data Evolving is web pages which grow with new hyperlink about new information about the recent updates. Usually the web page is structured in the graph model to enable efficient searching in the web.

However, the web graph structure is constantly evolving; traditional data mining algorithm will become obsolete. Therefore, it is desirable to propose a new model capable of identifying the data updates with concept evolution and feature evolutions regularly. Hence it becomes mandatory to model a system which is capable of handling of the data evolution in the large data.

Incoop[2] and I2map reduce [3] have used for evolution data processing by extending MapReduce tosupport incremental processing. However, it has two mainlimitations. First, Incoop supports only task-level incrementalprocessing. That is, it saves and reuses states at the granularityof individual Map and Reduce tasks. Each tasktypically processes a large number of key-value pairs (kvpairs).In this Paper, we propose a novel technique named as Evolutionmining which develop a learning model in the map reduce paradigm using key value pair processing & Concept based Mining techniques for map and reduce functions. The proposed

Technique is capable of handling the diverse computation characteristics such as accuracy and complexity. It will be handled using sophisticated iterative models for feature extraction, classification and feature or concept evolution determination.
The rest of paper is organized as follows Section 2 describes the related work, section 3 describes the proposed model in detail, section 4 deals with experimental analysis and finally section 5 is concluded.

II. RELATED WORK

A. I2Mapreduce- Incremental Processing Approach

I2MapReduce supports only task-level incrementalprocessing. That is, it saves and reuses states at the granularityof individual Map and Reduce tasks. Each tasktypically processes a large number of key-value pairs (kvpairs).If Incoop detects any data changes in the input of atask, it will rerun the entire task. While this approach easilyleverages existing MapReduce features for state savings, itmay incur a large amount of redundant computation if onlya small fraction of kv-pairs have changed in a task [4]. Second,Incoop supports only one-step computation. Incoop would treat each iteration as a separateMapReduce job. However, a small number of input data changes may gradually propagate to affect a large portion ofintermediate states after a number of iterations, resulting inexpensive global re-computation afterwards.

B. DATMin- Map Reduce Computation

It is an extension to MapReduce that supports fine-grain incremental processing for both one-step and iterative computation. Supports kv-pair level fine-grain incremental processing in order to minimize the amount of re-computation as much as possible. It is modelled as kv-pair level data flow and data dependence in a MapReduce computation as a bipartite graph, called MRBGraph. A MRBG-Store is designed to preserve the fine-grain states in the MRBGraph [5] and support efficient queries to retrieve fine-grain states for incremental processing. It uses reuse the converged state from the previous computation and employ a change propagation control (CPC) mechanism.

III. PROPOSED SYSTEM

A. Data Partitioning

The data is initially partitioned into different clusters and placed in distributed file system using random partitioning algorithm or equal sized blocks. The equal sized partitioning is obtained from the dataset in order to establish the computation using map reduce framework. The Clustered undergoes analysis in the map reduce framework for merging mechanism in the data management.

B. Modelling aMapReduceArchitecture

The Map function takes a key value-pair as Input and it computes more Key value pair. The Reduce function is used to produce the final key value pair. The data is placed in cluster in the distributed file system. The Map reduce undergoes following process

1) *Map Function :*For a MapReduce program, the MapReducesystem runs a Job Tracker process on a master node tomonitor the job progress, and a set of Task Tracker processeson worker nodes to perform the actual Map andReduce tasks. The detailed architecture of the proposed framework is depicted in the figure 1

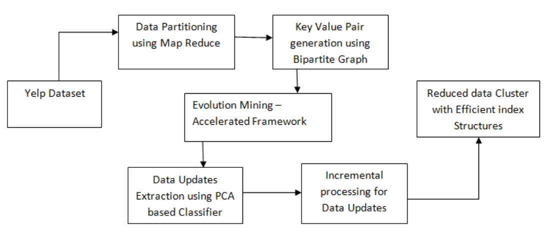


Figure 1 - Architecture Diagram of the proposed System

The Job Tracker starts a Map task per data block, and typically assigns it to the Task Tracker on the machine that holds the corresponding data block in order to minimize communication overhead. Each Map task calls the Map function for every input of each input key value pair, and stores the intermediate key value pair for the redundant and similar contents.

2) *Reduce Function :*Intermediate results are shuffled to Reduce tasks according to a partition function.Reduce task obtain and merge intermediate results from all Map Tasks.

C. Data Merging using Map Reduce Bipartite Graph

Each vertex in the Map task represents an individualMap function call instance on a pair. Each vertexin the Reduce task represents an individual Reduce functioncall instance on a group. An edge from a Mapinstance to a Reduce instance means that the Map instance generates key value pair that is shuffled to become part of theinput to the Reduce instance. Edges are the fine-grain states M that wewould like to preserve for incremental processing [6]. An edgecontains three pieces of information: (i) the source Mapinstance, (ii) the destination Reduce instance, (iii) the edge value.

he Map input is the adjacency matrix of the graph. Every record corresponds to a vertex in the graph. For incremental processing, we preserve the fine-grain MRBGraph edge states.

D. EvolutionMining: Cluster processing for incremental data

It employs the Concept based mining to identify the concept evolution and feature evolution using adjacency matrix. The key value pair is estimated for the merging with already available cluster using map reduce framework. Data indexing [7] is carried out for the key value pair based similarity computation using vertices and edges of the graph. Word similarity is computed using wordnet tool. Indexing is kept for easy retrieval of the data search in the graph structure model [8].

E. Algorithm – Evolution Mining

Input – Key value pair and Delta input for Increment with Data Evolution
Output – Data merging for Similarity
Process
Map Phase
Select the random point k as Centroid
Map instance of key value for random point
Edge -→weight between the values
Vertices --- values of the each key
Merge ()
For key=1&& key ==2
If (values 1== value 2)
Agg key 1& key 2 as Agg 1
Else
Create a new id agg 2
Reduce Phase
For new evolution
Find (concept similarity)
If (value 1= = value 2)
Merge ( )
Else
Create new data for key

IV. EXPERIMENTAL RESULTS

We implement the prototype named as Evolution Mining in order to support accelerated updates of the data in the cluster. The framework implements the map reduce and bipartite graph for data merging and data organization into different clusters. Initially one step algorithm is followed in the i2map reduce and EvolutionMining. The size of the dataset is taken in large scale and it is named as yelp dataset.

Accumulator Reduce optimization in incremental processing is implemented in the reduce Function. For incremental processing, we model a dataset as a delta input from each incremental data for dimensionality reduction using Principle component analysis [9].

A. Performance Analysis

EvolutionMining Produces the low processing power with less complexity. The graph structure is constructed for dataset. The i2map reduce and EvolutionMining will return the exact updated result but at the same time prolong the runtime for i2mapreduce. This is because i2MapReduce pays additional cost for accessing and updating the MRBGraph file.

Data Structure of the graph often reflects the problem structure and is read -only during computation in the data merging in the map and reduce function. In contrast, state data is the target results of the new data of concept evolution being updated in each iteration by the algorithm. Structure (state)data can be represented by a set of structure (state) kv-pairs.
The Evolutionmining technique speed up convergence, and reduce the time and space overhead for saving states.

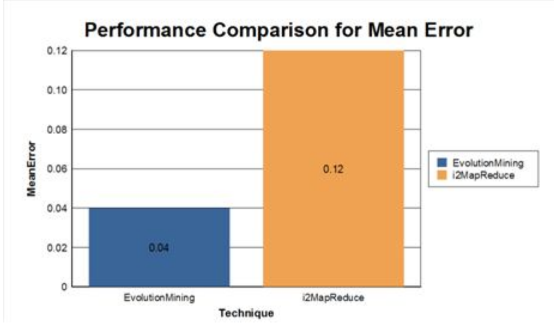


Figure 2: Performance Comparison of Mean error for incremental processing

he Figure 2 describes the Mean error computation of the incremental processing
Further improves the performance with incremental processing of EvolutionMining, reducing the run time as described in the figure 3.

Figure 3- Performance Comparison of the Execution Time against the Incremental processing Techniques

Moreover, we find that the change propagation control mechanism plays a significant role for merging the data. It filters the kv-pairs with tiny changes at the prime. Reduce, greatly decreasing the number of Map instances in the next iteration.

Table 1-Performance Table for the Incremental Processing Techniques

Technique	Run time for Map Function	Run Time for Reduce Function	Mean Error
I2Map Reduce	40s	78s	0.04%
EvolutionMining	20s	43s	0.02%

Consequently, the MRBGraph file will consist of multiple batches of sorted chunks, corresponding to a series of iterations. If a chunk exists in multiple batches, a retrieval request returns the latest version of the chunk.The Performance values is depicted in the table 1.
Only the Map and Reduce instances that are affected by the delta input are re-computed. The overhead of the backward transfer can be fully removed if the number of state kv-pairs in the application is greater than or equal to map function.
The index is stored in an index file and is preloaded into memory.

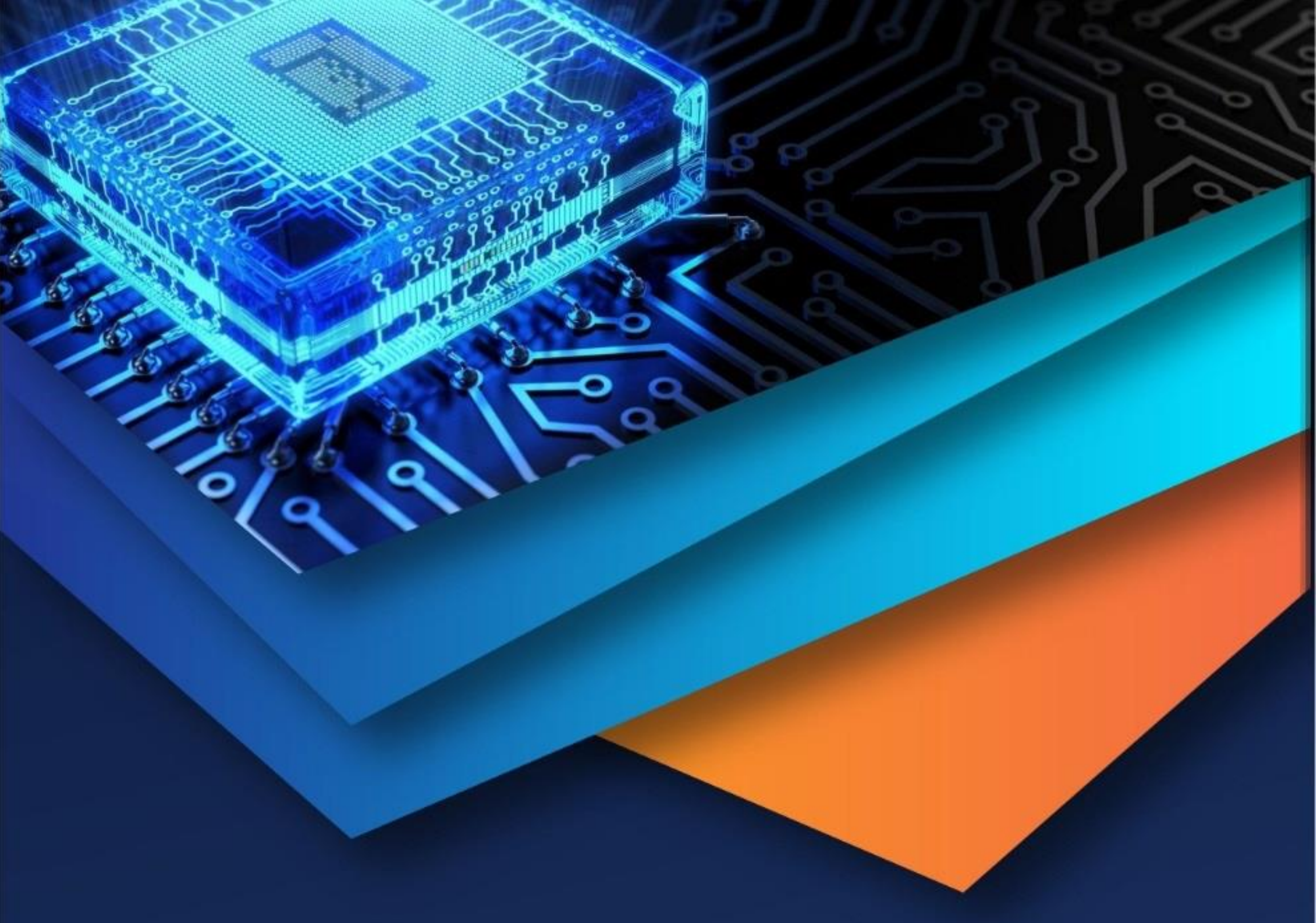
V. CONCLUSION

We designed and implemented the Accelerated clustering framework for data evolution using map Reduce model for in the large scale data streaming. Classification model combines unsupervised learning model, concept based mining techniques and Map reduce computation functionalities is used generate the reduced clusters for new data updates without re-computation process.

Experimental results prove that proposed process significantly reduces the run time and mean error for clustering the data with feature evolution and concept evolutions against re-computation based processes.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Opear. Syst. Des. Implementation, 2004, p. 10.
[2] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce for incremental computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 7:1–7:14
[3] Yanfeng Zhang, Shimin Chen, Qiang Wang, and Ge Yu "i2MapReduce: Incremental MapReducefor Mining Evolving Big Data" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 7,JULY 2015
[4] . Y. Wu, J. S. Pan, and C. F. Lin, "Improving accessing efficiency of cloud storage using de-duplication and feedback schemes," IEEE Syst. J., vol. 8, no. 1, pp. 208–218, Mar. 2014, doi:10.1109/JSYST.2013.2256715
[5] P. Wang, H. Wang, X. Wu, W. Wang, and B. Shi, "A Low- Granularity Classifier for Data Streams with Concept Drifts and Biased Class Distribution," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 9, pp. 1202-1213, Sept. 2007
[6] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Integrating Novel Class Detection with Classification for Concept- Drifting Data Streams," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 79-94, 2009.
[7] . Yan, X. Yang, Z. Yu, M. Li, and X. Li, "IncMR: Incremental data processing based on mapreduce," in Proc. IEEE 5th Int. Conf. Cloud Comput., 2012, pp.pp. 534–541.
[8] . Zhang, Q. Gao, L. Gao, and C. Wang, "Priter: A distributed framework for prioritized iterative computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 13:1–13:14
[9] N.Meenakshi, Dr.Vijaykumar "Location based Hierrachical classifier for multidimensional Spatial Data mining "International Journal of Scientific Engineering and Innovative Research, volume 10, issue 1, paper id: I1V1020,



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY