ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

AN IMPROVED MAP REDUCE COMPUTATION WITH PARALLELIZED INCREMENTAL CLUSTERING

Dr.V.Venkatesakumar[1], A.Subashini[2]

Department of CSE, Anna University Regional Centre, Tamil Nadu,India
email id: mail2venkatesa@gmail.com
Department of CSE, Anna University Regional Centre, Tamil Nadu, India
email id: subashini.4be@gmail.com

Abstract: **Map reduce is programming model which enables the users to evaluate the large volume of data at run time. Handling large volume of data with less computational cost is the important features of map reduce frame work which needs to be handled with more concern. In the existing work, incremental map reduce is introduced which focus to eliminate the computation cost of users by make use of already processed input data for the new computation instead of doing it from start. The existing work does an incremental processing in the KV pair level in which only the input data which might affects the output would be processed. However this works lacks to reduce the computational cost in the data set which are collected from various sources with different characteristics. This is overcome in the proposed methodology by introducing the Parallelized incremental clustering which can process the data that are gathered from multiple sources in run time. The experimental tests conducted proves that the proposed methodology is improved in its performance in terms of less computational cost.**

I. INTRODUCTION

Big data is data that exceeds the processing capacity of conventional database systems because of its large volume of data. It describes a massive volume of both structured and unstructured data it is difficult to process using a traditional database and software techniques. It used in companies to make operations faster which is used to take decisions.

The advantage gained from the ability to process large amounts of data is the main attraction of big data analytics. Hadoop is a platform for distributing computing problems across a number of servers. Hadoop has an implementation of distributed file systems called Hadoop Distributed File System which is a base to process the large amount of data. A typical Hadoop usage involves three steps:

- loading data into HDFS,
- MapReduce operations, and
- retrieving results from HDFS.

NameNode contains details regarding the block's location as well as the data of the entire directory structure and files. DataNode runs on all the worker machines and stores all the data of the cluster. DataNode frequently reports to NameNode(master) with the list of blocks stored in it for processing .

Job Tracker runs on the master node and Task Tracker runs on worker nodes where the data actually resides. Each Task Tracker is responsible for running multiple task-instances, and every

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Task Tracker reports to Job Tracker at regular a interval, which also contains details of the current job it is processing. Job Tracker schedules jobs and takes care of failed ones by re-executing them on some other worker nodes.

II. EXISTING WORK

In the existing work, i2MapReduce, onestep and iterative computations an extension to MapReduce that supports fine-grain incremental processing is used for processing the input data. Fine-grain incremental processing is done by using Map Reduce Bipartite Graph store and it supports kv-pair level processing in order to minimize the amount of re-computation as much as possible for the incoming data. Iterative computation in this work targets types of iterative computation where there is a one-to-one or all-to-one correspondence from Reduce output to Map input. It cannot efficiently handle the spatial type of data which are collected in the distributed manner. More computational cost in case of distributed environment because if the same type of data or same data comes for processing once again as input the intermediate result will be computed once again for the entire data. The computation time will also be more for such processing of data.

III. PROPOSED SYSTEM

Parallelized incremental and distributed processing is the major tool to reduce the computation time, especially when very large data sets are involved. Online(Dynamic) Aggregation is an alternative approach where the incoming data from multiple sources are gathered, combined and stored in the buffer for processing at run time. The goal is targeted by the extension of the open-source Hadoop framework which allows dynamically distributed computations under the Map-Reduce model. First, the incoming data are gathered from multiple sources and then the similar data are combined to make clusters using clustering algorithm. Then Map function is performed for the clusters in as usual manner, one copy of the intermediate result will be stored in MRBG Store and other copy will be forwarded to the Reduce function for further processing.

Instead of computing the result for the entire data once again, the result if the data will be taken from the store for processing. The intermediate result for the data will be stored along with the Map Key, which is used to store the data in MRBG store and helps for fast retrieval of the intermediate results. PageRank Algorithm is also used for the fast retrieval of the intermediate results where the mostly or recently used results will come first in the list of the store.

IV. MODULES

The proposed system has six modules. They are,

- Network Setup
- MRB Graph Abstraction
- Fine Grain Incremental Processing
- Storing Processing States
- Parallel Computation of Tasks
- Performance Evaluation

The modules above specified are used to gather the data from multiple sources , process the data by computing the work assigned to it and generate the results of processing efficiently in terms of less computational cost and time.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

V. BLOCK DIAGRAM

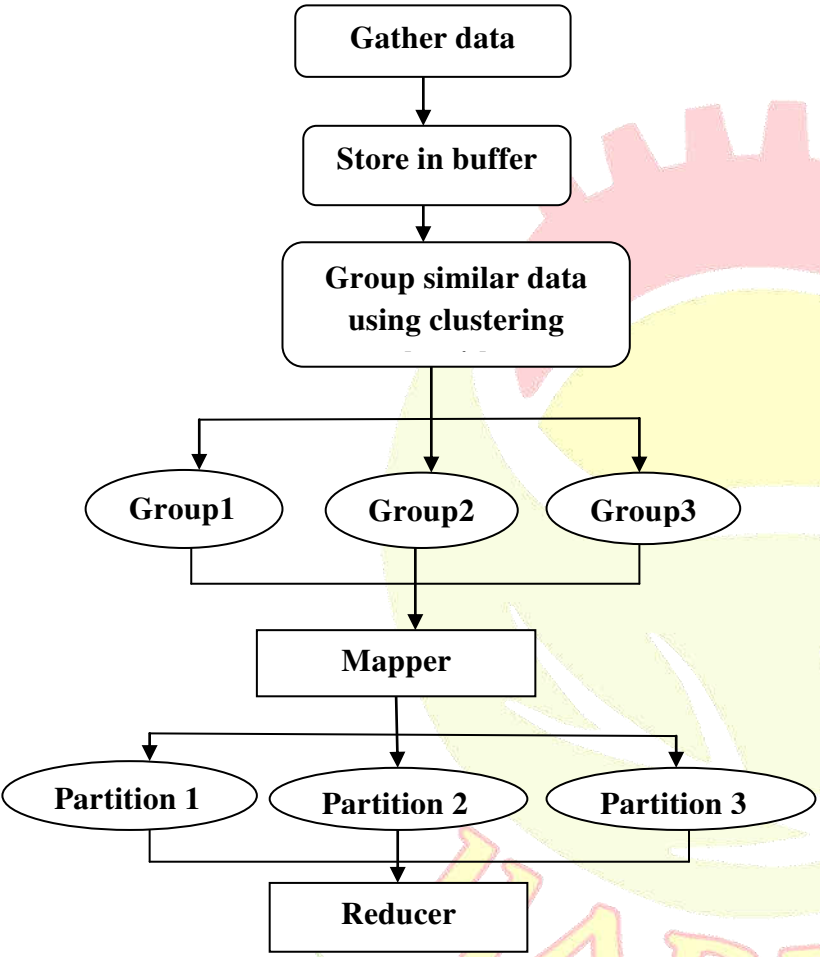

Fig.1 Work flow

A. Network Setup

In this module, the Hadoop setup is done(i.e) Hadoop tool kit environment to process those big data will be setup. The location data will be gathered from the various sources for processing. Tasks will be partitioned and distributed to the different data nodes using the map reduce framework by name node. Name node is a master, where the data node is responsible for the processing and resulting an output.

B. MRB GRAPH ABSTRACTION

Each vertex in the Map task indicates an individual Map function call instance on a pair of <K1, V1>. Each vertex in the Reduce task indicates an individual Reduce function call instance on a group of <K2, {V2}>. An edge contains (a) the source Map instance, (ii) the destination Reduce instance and (iii) the edge value .Therefore, for each MRBGraphedge,i2Map Reduce will preserve (K2, MK, V2).

C. Fine Grain Incremental Processing

For fine grain incremental processing, it preserve the MRBGraph edge states to reduce the processing time and cost . The engine transfers the globally unique MK(Map key) along with <K2, V2> during the shuffle phase. Then it saves the states (K2, MK, V2) in a MRBGraph store at every Reduce task. If for example task A arrives it is proceesed and intermediate result is produced next task A1 arrives where A1=A+d, then only the value of data part d will be computed. The result of A, which is already computed is be used to reduce the computation time and cost of data processing.

D. Storing Processing States

The MRBG Store provides support for preservation and retrieval of fine-grain MRBGraph states for incremental processing because the intermediate results will be stored in the store for reuse. The unique Map key will be stored for each edge along with the key and value and this Map key is used to retrieve the intermediate results that are stored in MRBG store if same input is evolved for processing once again. Instead of computing the results for entire data for inputs , only the changed data is processed and the early intermediate result is used which reduces the computation cost and time.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

E. Parallel Computation Of Tasks

Incremental algorithm is performed in the Distributed environment using Hadoop. Data are gathered from various sources and then similar data are clustered using the clustering algorithm. Algorithm will be applied on different site individually and one local cluster (LC) is created for each partitions.

LC will be send to the Master Node of the Hadoop system from all the sites involved in processing. At the Master Node one global cluster (GC) is generated using all Local clusters gathered from data nodes . Master Node is also called as client. If any client sends any request to the master node, it send back the data node list to the client and then client will communicate with that particular data node involved in processing. At the data node that query or request will be executed with the help of the map reduce function and that result will be send back to the client.

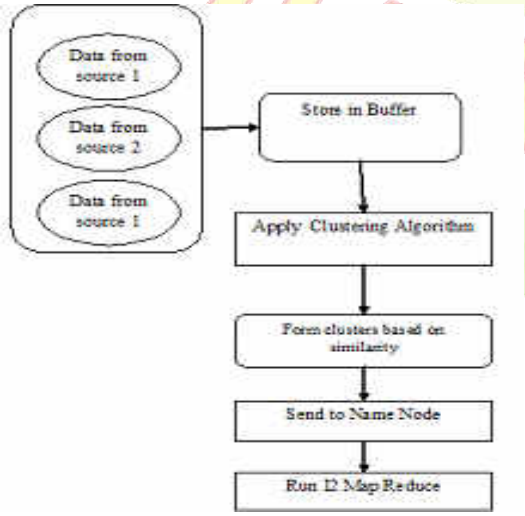

Fig.2 Computation process

VI. CONCLUSION

Map reduce is most featured component in the Hadoop simulation toolkit which is used to handle the large volume of data in the efficient manner. The performance of the map reduce toolkit might get reduced in case of processing the same kinds tasks again and again. This is due to because immediate deletion of cache stored data once it forwarded to the reduced phase. This is resolved in this work by introducing the incremental algorithm which will buffer the intermediate generated results for the particular period. The buffer would be refreshed in the periodic manner to support the large volume of data. The data gathered from the multiple nodes is processed in the efficient manner by clustering them based on most similar fields through which system performance can be improved considerably there by reducing the computation cost and time

REFERENCES

[1] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Communications of the ACM - 50th anniversary issue: 1958 - 2008.

[2] Russell Power, Jinyang Li, "Piccolo: Building Fast, Distributed Programs with Partitioned Tables", Proceedings of the 9th USENIX conference on Operating systems design and implementation, Article No. 1-14, 2010.

[3] Grzegorz Malewicz, Matthew H. Austern, Aart J.C. Bik, James C. Dehnert, Ilan Horn, NatyLeiser, and GrzegorzCzajkowski, "Pregel: A System for Large-Scale Graph Processing", Proceedings of the 2010 ACMSIGMOD International Conference on Management of data, Pages 135-146, 2010.

[4] MateiZaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael Franklin, Scott Shenker, Ion Stoica, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing", Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation,Pages 2-2, 2012.

[5] Svilen R. Mihaylov Zachary G. Ives SudiptoGuha, "REX: Recursive, DeltaBasedDataCentric Computation", Proceedings of the VLDB Endowment, Vol. 5, No. 11, 2012.

[6] Yucheng Low, Joseph Gonzalez, AapoKyrola, Danny Bickson, Carlos Guestrin, Joseph M. Hellerstein, "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud", Proceedings of the VLDB Endowment, Vol. 5, No. 8, 2012.

[7] Yingyi Bu, Bill Howe Magdalena, Balazinska Michael D. Ernst, "HaLoop: Efficient Iterative Data Processing on Large Clusters", Proceedings of the VLDB Endowment, Vol. 3, No. 1, 2010.

[8] JaliyaEkanayake, Hui Li, Bingjing Zhang, ThilinaGunarathne, Seung-HeeBae, Judy Qiu, Geoffrey Fox, " Twister: A Runtime for Iterative MapReduce", Conference'10, Month 1–2, 2010.

[9] Yanfeng Zhang, Qixin Gao, LixinGaoy, Cuirong Wan, "iMapReduce: A Distributed Computing Framework for Iterative Computation", Journal of Grid Computing, March 2012, Volume 10, Issue 1, pp 47-68.

[10] Daniel Peng and Frank Dabek, "Large-scale Incremental Processing Using Distributed Transactions and Notifications", Proceedings of the 9th USENIX conference on Operating systemsdesign and implementation, 2010.

[11] Murray D J ,McSherry F, Isaacs, Isard M, Barham P, and Abadi M, 'Naiad: A timely dataflow system,' in Proc. 24th ACMSymp. Oper. Syst. Principles, 2013,

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

pp. 439–455..Olston C and Najork M, 'Web crawling,' Found. Trends Inform. Retrieval, vol. 4, no. 3, pp. 175–246, 2010.

[12] Peng D and Dabek F, 'Large-scale incremental processing using distributed transactions and notifications' in Proc. 9th USENIX Conf. Oper. Syst. Des. Implementation, 2010, pp. 1–15.

[13] Svilen R. Mihaylov Zachary G. Ives SudiptoGuha,'REX: Recursive, DeltaBasedDataCentric Computation, Proceedings of the VLDB Endowment, Vol. 5, No. 11, 2012.