© 2016 IJSRSET | Volume 2 | Issue 5 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Implementation of Aggregation of Map and Reduce Function for Performance Improvisation

Varsha B. Bobade

Department of Computer Engineering, JSPMs Imperial College of Engineering &Research, Wagholi, Pune, Maharastra, India

ABSTRACT

Big Data is term that refers to data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to capture, manage, process or analyzed. To analyze this enormous amount of data Hadoop can be used. Hadoop is an open source software project that enables the distributed processing of large data sets across clusters of commodity servers.

I proposed a modified MapReduce architecture that allows data to be pipelined between operators. This reduces completion times and improve system utilization for batch jobs as well. I present a modified version of the Hadoop MapReduce framework that supports online aggregation, which allows users to see early returns from a job as it is being computed. The objective of the proposed technique is to signicantly improve the performance of Hadoop MapReduce for efficient Big Data processing.

Keywords: Big Data, Hadoop Framework, Online Aggregation, Combiners.

I. INTRODUCTION

Every three years ,World data can be increases so much as doubling its size. So that very large data are generated from digital media such as images, audio, video and Effectively storing, querying, analyzing, understanding and utilizing these very large datasets present makes challenges to the industry. and research communit Big data is a collection of large data sets that cannot be processed using traditional computing techniques.Currently for solving this problem Hadoop is used. Hadoop is framework that allows processing of large data sets.Hadoop has two parts: Hadoop Distributed File System and MapReduce. Hadoop cluster uses Hadoop Distributed File System (HDFS) to manage its data.

HDFS is a file system designed for storing very large files with straming data access patterns,running clusters on commodity hardware. It is designed as a highly fault-tolerant, high throughput, and high capacity distributed file system. It is suitable for storing terabytes or petabytes of data on clusters and has flexible hardware requirements, which are typically comprised of commodity hardware like personal computers. MapReduce is software framework for easily writing applications which process large amount of data in parallel on large cluster of commodity node. MapReduce has two functions Map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. I have proposed a simple online aggregation technique to improve the performance of Hadoop framework.

The main objectives of the proposed system are implementation of online aggregation taking to provide the user to see early returns from a job. The paper of article is organized in part II as previous work and part III necessary background respectively. In section IV,I have discussed about motivation of the work. In section V, discussed with proposed system architecture and In section VI discuss with experimental setup and results are given. Finally Section VII paper final conclusion is given.

II. METHODS AND MATERIAL

1. Related Work

- ✓ S. Vikram Phaneendra & E. Madhusudhan Reddy[2] In RDBMS minimum data easily handled but Now it is so much difficult to handle huge data through RDBMS tools, that is called as "big data". Hadoop architecture handle large data sets, scalable algorithm does log management application
- ✓ Chris Jermaine proposes an Online Aggregation for Large-Scale Computing. They are interested on very large scale and data-oriented computing.
- ✓ The EARL library [3] is an extension to the Hadoop framework and focuses on correct estimation of final results, while providing reliable error estimations. It uses the bootstrap technique, which is applied to a single pre-computed sample
- ✓ Vasiliki Kalavri,Vaidas Brundza,Vladimir Vlassov [6] Here the block-level sampling technique used to provides a random sample of the provided dataset. And provides Integrated block-level sampling with the MapReduce Online framework is done. System provides accurate estimations of results, without requiring a-prior knowledge of the query.
- ✓ The BlinkDB [7] approximate creates query engine and uses pre-computed samples of various sizes to provide fast answers. It contain two types of samples: large uniform random and smaller multi-dimensional stratified. Queries are evaluated with of selected samples and an initial estimate is produced and The system can give estimate results of standard aggregate queries easily.

2. Background

Here a briefly review MapReduce programming model, the MapReduce Online framework and the Online Aggregation technique is given Hadoop is an Apache open source framework written in Java that allows distributed processing of large dataset across cluster of computers using simple programming model. Hadoop is a framework that can run applications on systems with thousands of nodes and terabytes. Allows to distributes the file among the nodes and allows to system continue work in case of a node failure. Fig1.shows MapReduce Word Count example

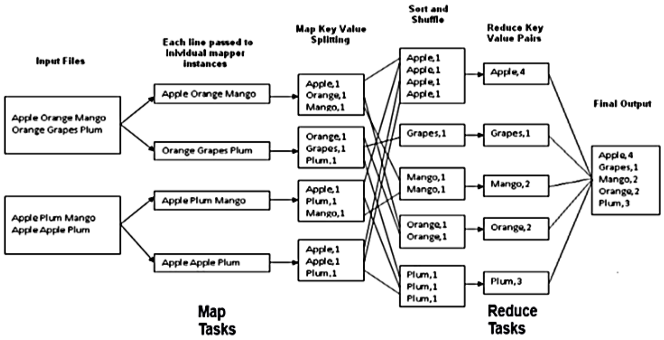

Figure 1: MapReduce

- ✓ MapReduce programming model : MapReduce is powerful large-scale datasets processing in a distributed and parallel fashion MapReduce framework has two types of tasks: map and reduce. The data are fed into map function as key value pairs to produce intermediate key value pairs. Once the mapping is done all the intermediate results from various nodes are reduced to create the final output.
- ✓ MapReduce Online framework : The Map Reduce Online is a framework for Hadoop.and MapReduce are used to supports Online Aggregation and stream processing and also improving utilization and reducing response time. Traditional Map Reduce technique used to materialize the intermediate results of mappers and also not allow pipelining between the map and the reduce phases .In MapReduce Online technic having advantage of simple recovery in the case of any failures occurs. and In Traditional technique first all mapper finishing all tasks then reducers can start executing tasks .To avoid these problem Map Reduce Online technic should be used to allowing pipelining between operators with preserving fault-tolerance guarantees.
- ✓ Online Aggregation : Online Aggregation [4] is a framework that used idea of batch mode to obtained approximate result .Map Reduce can be implemented using Online Aggregation. Online Aggergation technic explain partial query processing and without knowing previous output of the query specifications, such as types of operators and data structures and also, users can observe the progress of running queries and control their execution

3. Motivation

In Big Data processing having problems that good performance requires both good data locality and good resource utilization.

A characteristic of Big Data processing is that the amount of data being processed is typically large with compared to the amount of computation done on it. The processing can be benefit from data locality, that can be achieved by moving the computation close to the data using Hadoop. Still there are few challenges with Hadoop too which are discussed in introduction. The data from a MapReduce of Job can be fed to a system for online processing.

The current Hadoop MapReduce implementation does not provide such capabilities. For that result each MapReduce job, a customer can assign a dedicated cluster to run that particular application, one at a time. Each cluster is dedicated to a single MapReduce application so if a user has multiple applications, the he has to run them in serial on that same resource or buy another cluster for the additional application.

4. Proposed System

We are considering how online aggregation can be built into a Map Reduce system.

The Map Reduce Online can supports Online Aggregation and also reduces response time. In Traditional Map Reduce technique materalize the intermediate results of mapper and not allowing pipelining between the map and the reduce phases. In Map reduce approach has advantage of simple recovery in the case of failures and reducers cannot start executing tasks before all mapper have finished.

In traditional method has limitation of lowers resource utilization and leads to inefficient execution for many applications. The main motivation of Map Reduce Online is to overcome these problems, by allowing pipelining between operators, while preserving.

A. System Architecture

When map task execution is done reduce task collect outputs by giving request to node. The reduce task cannot issue this request until it has received notification of execution of map task has been completed and its final output has been committed to disk. This means that map task execution is completely decoupled from reduce task execution . In modified map task paradigm map will pushing data to reducer. Consider a job in that reducer has some different values and then the reducer applies aggregate function.

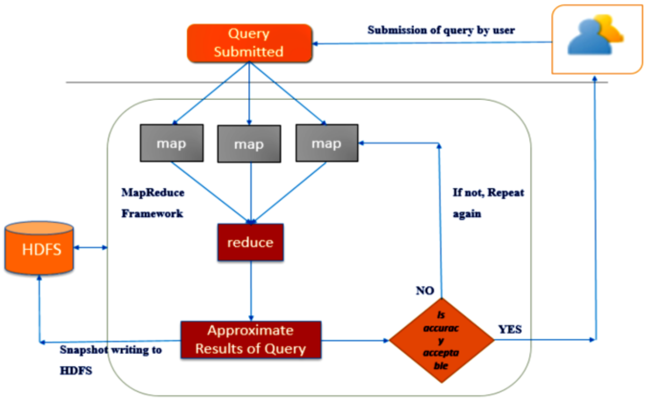

Figure 2 : Architecture for Online Aggregation of MapReduce

1) Partitioner and Combiners: Combiner is used for aggregation.Combiner is a used to reducing intermediate data. It is thought as minireducer. Minireducer that processes the output of mappers. The MapReduce library handles the output of the function. The output of a reduce function is written to the final output file.

The output of a combiner function is written to an intermediate that will be sent to a reduce task. In word count example , the combiners aggregate term counts on data processed by each map task.This results in a reduction in the number of intermediate key-value pairs that need to be shuffled across the network. An associative array (i.e., Map in Java) is introduced inside the mapper. This array tally up term counts within a single document. This version gives a key value pair for each unique term in the data instead of a key value pair for each term in the data being processed. For the same example we can consider some words appear frequently within a document.

This method can save number of intermediate key-value pairs emitted, especially for huge data.

2) Taking Snapshots: Snapshots means the output of intermediate reduce operation The reducer will process the data given by mapper as input and generates the results. This result is considered as a snapshot and will be shown to user.If result is acceptable then user can stop further processing of map reduce .If result is not acceptable then mapper and reducer will continue with the processing.

B) Mathematical Model

This section describes mathematical modelling of proposed work. Turing machine are used to
Describe the processing of the system.

- ✓ Multiplexer Logic : In addition to mathematical model, the multiplexer logic also discussed here. As the fig.3. indicates ,the data from the HDFS is given to mapper on the input. The mapper processes the data and the result is then sent to the reducers.The number of mappers and the reducers will be depending on the programming logic used and the number blocks to be processed. As figure indicates, the result of reducers is written to the HDFS.and same is then sent to the mapper for further processing . The whole processis executed continuously until the desired output is not achieved .The difference between the traditional Hadoop and The proposed system is that ,the intermediate results of mapper are forcefully sent to the reducers to obtain the early results. Where as in traditional system the reducer cannot start until the mapper finishes it processing completely.

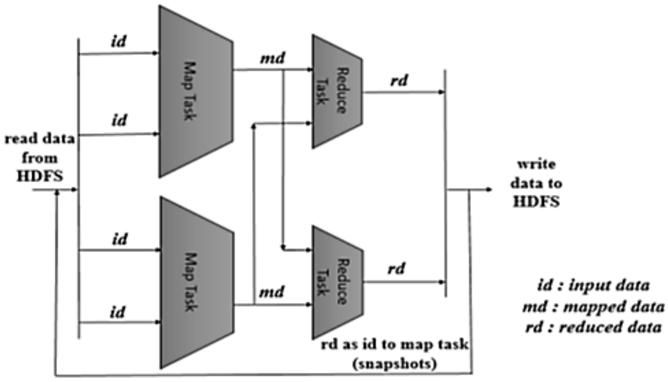

Figure 3 : Multiplexer Logic for Aggregated MapReduce Framework

- ✓ Performance Model: The time taken for a Hadoop to execute a job on given data set can be described as $T = ov+(ps+pa)_h$ (1)
- ✓ where, ov = the fixed overhead in running a job ps = processing time for Hadoop system

pa = processing time in map and reduce
h = time over which data is collected

Here, the processing time can be divided into two parts, one part related to Hadoop system and the other is with the application logic.

There is a fixed overhead cost in running any Hadoop job, which is represented by ov. The overhead is mostly associated with the work involved in job. It is independent of the data size. Time has two components. The first component represented by ps, contains all time taken by Hadoop system for processing a job. This part is mostly disk and network IO bound.

Bradly speaking it constitutes the following: a) Input and output for map b) Shuffle c) Input and output for reduce The second component represented by pa. It is the time taken by the application logic . It having the following parts: a) Execution of map b) Execution of reduce The overhead is fixed and not much can be done about it.

III. RESULTS AND DISCUSSION

Experimental Results

The results obtained from the traditional Hadoop framework and implemented proposed system are taken on different sets of data. Both results are then compared to find out the conclusion. For both traditional and proposed system the same environment is used. The setup used for Hadoop and the results obtained are discussed below.

A. Environment

The fig.4. describes the experimental setup of proposed system. The hadoop-1.2.1 is used for setup. Ubuntu 12.04 operating system is installed on all the nodes. Total 3 nodes are configured, one as master node and remaining two as slave nodes. The processors and memory allocated to the nodes is
given in table 1

	Description	Total Virtual Machines
Machine	Processor: Intel Corei5-4210U CPU @ 1.70 GHz 2.40 GHz RAM: 4.00 GB	1- Master 2- Slaves
Hadoop Version	Hadoop-1.2.1	
Virtual Machine	VMware Workstation Version 11.0.0	

Figure 4: Experimental Environment

TABLE I
SPECFICATION FOR EACH NODE

Sr.No.	Machine	CPU	Memory	Disk
1	Master	4	2GB	500GB
2	Slave1	2	1GB	500GB
3	Slave2	2	1GB	500GB
4	Slave3	2	1GB	500GB
5	Slave4	2	1GB	500GB
6	Slave5	2	1GB	500GB

B. Results

The comparison in between execution of traditional approach of word count job and new approach is shown in fig.5. It shows that time required to complete a job by new approach is less than the traditional system.

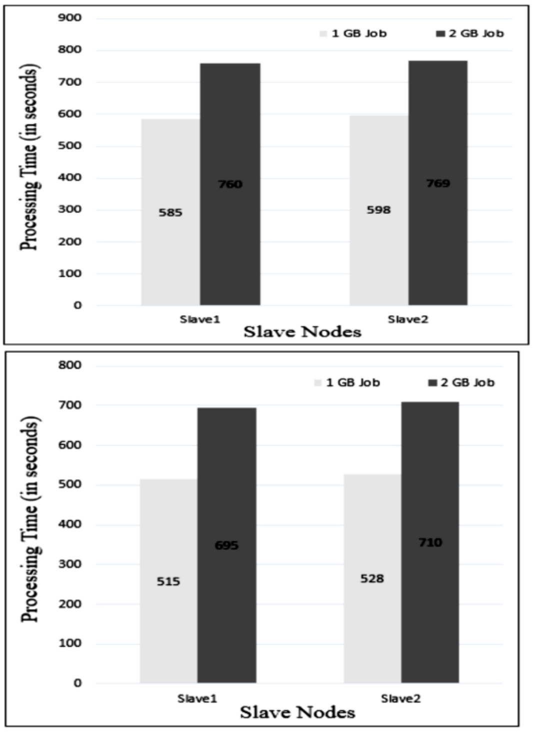

Figure 5: Execution Time of WordCount Job

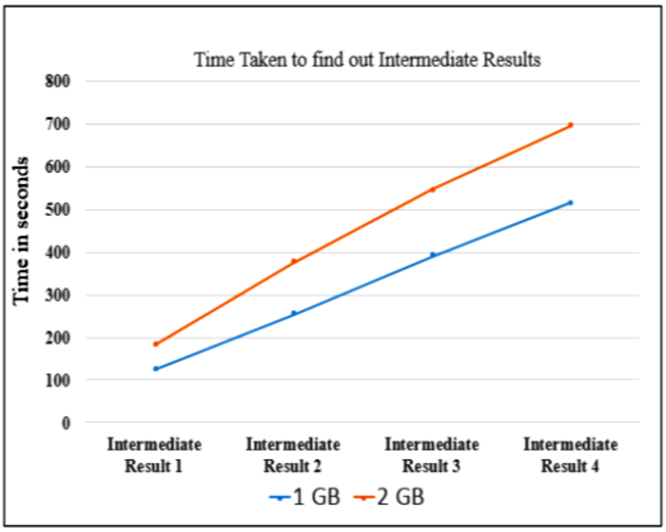

Figure 6: Time taken to find out Intermediate Results

Fig.6. shows the time taken by the system to find out the intermediate results. For any particular job performed on big data, the results are obtained for few initial records from the input data. And the same process is followed until user accepts the results. According to the proposed system, whole input data will not be processed at one glance. For example for the input data of size 1 GB, system will process first 250 MBs of data and will produce some result. Then next 250 MB of input data will be processed and both the results are combined. Likewise the system processes the input data and intermediate results are given to user.

IV. CONCLUSION

In this paper I have proposed an Online Aggregation approach for efficient big data processing. The aggregated results will be shown to the user and if user is agree with the obtained results then the further processing of MapReduce can be stopped. The early result will be shown to user by taking snapshots of the intermediate results obtained. As a local aggregate the combiners are implemented alongside with mapper and reducers for the aggregation. The combiners processes intermediate data produced by the mappers. And the result is then forwarded to reducers for further processing. The processing time of Hadoop is reduced at some instance.

V. REFERENCES

[1] S. Vikram Phaneendra and E. Madhusudhan Reddy, Big Data- solutions for RDBMS problems- A survey, In 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19, 2013)

[2] Kiran kumara Reddi & Dnvsl Indira, Different Technique to Transfer Big Data : survey, IEEE Transactions on 52(8) (Aug.2013) 2348 2355

[3] Jimmy Lin MapReduce Is Good Enough?, The control project. IEEE Computer 32 (2013).

[4] Jiawei Han and Micheline Kamber, Classification and Prediction in Data Mining: Concepts and Techniques, 2nd ed., San Francisco, CA The Morgan Kaufmann, 2006.

[5] N. Laptev, K. Zeng, and C. Zaniolo, Early accurate results for advanced analytics on mapreduce, vol. 5, no. 10. VLDB Endowment, 2012, pp.10281039.

[6] Report from Pike research, http://www.pikeresearch.com/research/smartgrid-data-analytics.
[7] National Climate Data Center Online]. Available:http://www.ncdc.noaa.gov/oa/ncdc.html
[8] D. Borthakur, The Hadoop Distributed File System: Architecture and Design, 2007.
[9] J. Hellerstein, P. Haas, and H. Wang,Online aggregation, In SIGMOD Conference, pages 171182, 1997.
[10] C. Jermaine, S. Arumugam, A. Pol, and A. Dobra, Scalable approximate query processing with the dbo engine, In SIGMOD Conference, pages 725736, 2007.
[11] The apache hadoop project page, http://hadoop.apache.org/, 2013, last visited on 1 May, 2013.
[12] J. Dean and S. Ghemawat, Mapreduce: simplied data processing on large clusters, Communications of the ACM, vol. 51, no. 1, pp. 107 113,2008.
[13] S. Agarwal, A. Panda, B. Mozafari, S. Madden, and I. Stoica, Blinkdb: Queries with bounded errors and bounded response times on very large data, in ACM EuroSys 2013, 2013.
[14] N. Pansare, V. R. Borkar, C. Jermaine, and T. Condie, Online aggregation for large mapreduce jobs, vol. 4, no. 11, 2011, pp. 11351145.