 ISSN: 2321-8169
58 – 62

Tools and Techniques of Big Data Processing with Incremetnal Map Reduce

Ravikiran G. Deshmukh
Department of CSE
MGI-COET, Shegaon, India
Email Id: - ravideshmukh611@gmail.com

Dhanshri S. Khade
Department of CSE
MGI-COET, Shegaon, India
Email Id: -dhanshriskhade@gmail.com

Pooja G.Mankar
Department of CSE
MGI-COET, Shegaon, India
Email Id:- mankar96pooja@gmail.com

Mahadeo J. Pathak
Department of CSE
MGI-COET, Shegaon, India
Email Id: - mahadeopathak1@gmail.com

Vaishali A. Kshirasagar
Department of CSE
MGI-COET, Shegaon, India
Email Id: - vaishalikshirasagar96@gmail.com

Abstract—A collection of a huge and complex data is known as big data which is used to capture, store, process, retrieve and analyze data with the help of different data processing techniques. Big Data composed of 3V's viz as Volume, Variety, and Velocity. A popular data processing engine for big data is Hadoop. Hadoop is an open source framework which allows to store and process big data in distributed environment across clusters of computers using simple programming models. In this paper, we explores incremental Map Reduce, the most widely used framework for processing Big Data. Incremental MapReduce performs key-value pair level incremental processing and also supports complicated iterative computation, which is widely used in data mining applications which makes Incremental MapReduce to process big data in a less time and stores it in a more optimized form.

Keywords-*Big data, Hadoop, MapReduce, incremental processing.*

*****

I .INTRODUCTION

Big data is a collection of large data sets that cannot be process using traditional data processing applications.It is not a single technique or a tool, rather it involve in many business and technologies. Big data have three characteristics Volume of data,Variety of data (structured, semi structured or unstructured) and Velocity of data (how fast data processes).Hadoop is to process such data, it is an open source framework written in java that allows distributed processing of large data files across clusters of computers using simple programming
Models.

The key to agreement with such composite data is, in our estimation, to combine streaming with distributed computing and open source. Extremely large quantity of digital data is being collected in numerous significant areas, including e-commerce, social network, finance, health care, education, environment etc. The large amount of multi-structured data with organizations find it difficult to use traditional relational DBMS technology for processing and analyzing such data. Therefore hadoop technology is used to process this big data.

MapReduce is a framework using which we can write applications toprocess huge amount of data, in parallel, on large clusters of commodity hardware in a reliable manner. It shows good performance for batch parallel data processing. MapReduce frameworks is used in many fields like life science computing, text processing, web searching, graph processing, relational data processing, data mining, machine learning and video analysis. Due to the simplicity and high efficiency MapReduce programming model is popular.
The data explosion is creating a surge in demand of storage and computing, e.g., web log analysis and web site statistics. Incremental data processing approaches can help them lighten the computation burden and save computation resources. This section first introduces several observations of Big Data including MapReduce implementation, then proposes an incremental computation framework.

II.BIG DATA

Bigdata is a term that refers to data sets or combinations of data sets whose size, complexity, and rate of growth make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases and desktop statistics or visualization packages, within the time necessary to make them useful. While the size used to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, most analysts and practitioners currently refer to data sets from 30-50 terabytes(10 12 or 1000 gigabytes per

 ISSN: 2321-8169
58 – 62

terabyte) to multiple petabytes (1015 or 1000 terabytes per petabyte) as big data.In order to gain better performance and quality services mining of such big data are become popular. Big Data as the name describes a large data sets that is growing beyond the ability to manage and analysis using with the traditional data processing tools. Big data represents large and incremental volume of information that is mostly untapped by existing data warehousing systems and other analytical applications.

The data in it will be of three types.

- Structured data: Relational data.
- Semi Structured data: XML data.
- Unstructured data: Word, PDF, Text, Media Logs.

Big data can be described in three characteristics:

- Volume of data: Volume refers to amount of data. Volume of data stored in enterprise repositories have grown from megabytes and gigabytes to petabytes.
- Variety of data: Different types of data and sources of data. Data variety exploded from structured and legacy data stored in enterprise repositories to unstructured, semi structured, audio, video, XML etc.
- Velocity of data: Velocity refers to the speed of data processing. For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value.

2.1Benefits of Big Data

- Using the information kept in the social network like Facebook, the marketing agencies are learning about the response for their campaigns, promotions, and other advertising mediums.
- Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.
- Using the data regarding the previous medical history of patients, hospitals are providing better and quick service.

2.2Big Data Challenges

The major challenges associated with big data are as follows:

- Capturing data
- Curation
- Storage
- Searching
- Sharing
- Transfer
- Analysis
- Presentation

To fulfill the above challenges, organizations normally take the help of enterprise servers.

2.3Big Data Solutions

Using the solution provided by Google, **Doug Cutting** and his team developed an Open Source Project called **HADOOP**. Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.

III.HADOOP

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. Hadoop is open-source software that enables:

- Reliable: The software is fault tolerant; it expects and handles hardware and software failures.
- Scalable: Premeditated for huge size of processors, memory, and neighboring attached storage space.
- Distributed: Handles duplication. Offers particularly parallel programming model, Map Reduce.

Hadoop have two basic components :

i) HDFS for storage and
ii) MapReduce for large computations

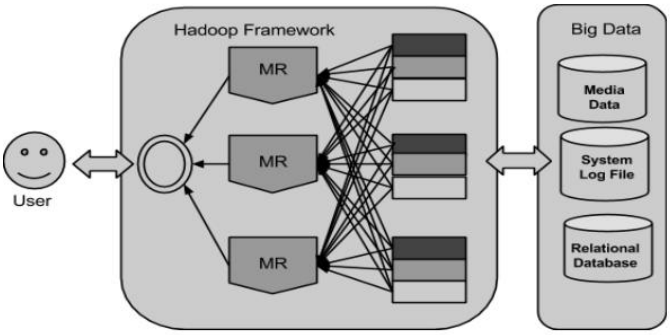

Fig1:Hadoop Framework

IV.HDFS

The processing support in the Hadoop environment is the MapReduce framework. The framework allows the requirement of a procedure to be useful to a massive data set, divide the problem and data, and run it in parallel. In the MapReduce framework, a distributed file system (DFS) firstly partitions data in several machines and data is represented as (key, value) pairs. The MapReduce framework executes the main function on a single master machine where we may preprocess the input data before map functions are called or post process the output of reduce functions. A two of a kind of map and reduce functions may be executed just the once or frequent times as it depends on the distinctiveness of an application.

4.1MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. MapReduce program poised of a Map function and Reduce function. Map() procedure that performs filtering and sorting, reduce (

 ISSN: 2321-8169
58 – 62

) method that performs a summary operation. MapReduce execution sequence is shown in fig.2.
map(K1,V1)□ [<K2,V2>]
reduce(K2,{V2})□ [<K3,V3>]

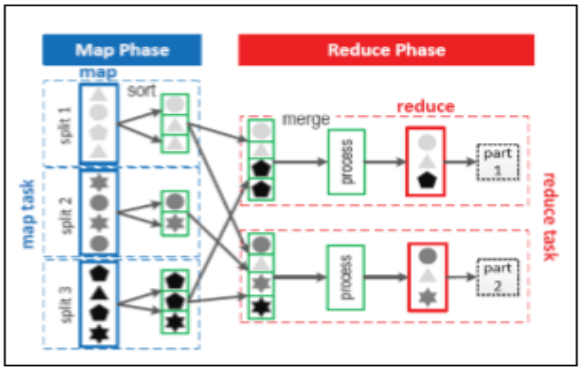

Fig2: MapReduce execution sequence

The MapReduce system orchestrates the processing by marshalling the distributed servers, running the different tasks in parallel, organization all communications and data relocate between the different parts of the system and provided that for redundancy and fault tolerance. An addition to MapReduce is Incremental MapReduce used for processing big data efficiently and timely with the optimization in data content of big data and also required optimized time to process this big data.

V.INCMR DATA PROCESSING FRAMEWORK

Because of the limitations of mapreduce , MapReduce model has to be extended for incremental computation. Following table shows some typical solutions for incremental or continuous computation related to MapReduce programming model and other similar models. They feature different input patterns, separate or coupled control approaches between state and dataflow, compatible or new added interfaces. Batch parallel processing refers to those that provide high efficient largescale parallel computation with one batch input and produce one output such as Google MapReduce, Hadoop and Dyrad/DryadLINQ. Incremental algorithms, more complicated than the original algorithms, can improve the runtime by modifying algorithms. Its input includes newly added data and the latest running result. Continuous bulk processing is another kind of solution to support incremental processing by providing new primitives for developers to design delicate dataflow oriented applications. It takes the intermediate results of the prior executions as a part of explicit input. CBP of Yahoo and Percolator of Google both provide such incremental computation frameworks. Incremental computation based on MapReduce, making full use of MapReduce programming model, supports incremental processing by modifying the kernel implementation of map and reduce stages. Because HDFS does not support appends currently, some approaches also modify the distributed file system to support incremental data discovery and intermediate result storage. IncMR an improved framework for large-scale incremental data processing. The framework, as shown in Figure 1, inherits the simplicity of the original MapReduce model. It does not modify HDFS and still uses the same APIs of MapReduce. All algorithms or programs can complete incremental data processing without any modification.

Approaches	**Input**	**State& dataflow**	**Advantages**	**Disadvantages**
batch parallel processing	One batch	Stateless	simple implementation; one input, one output	waste of time for re-running the entire data set
incremental algorithm	Incremental	Stateful &uncoupled	unchanged parallel framework	complicated algorithm design; manual user interaction
continuous bulk processing (CBP)	Incremental	Coupled	new model and primitives	delicate data flows built for different applications
incremental computation based on MR	Incremental	Coupled	same algorithm; same MapReduce APIs;	modified HDFS; modified programming mode
IncMR	incremental	coupled	same algorithm; same MapReduce APIs; same HDFS without modification	repartition of state data

Table 1
Typical Incremental Data Processing Approaches

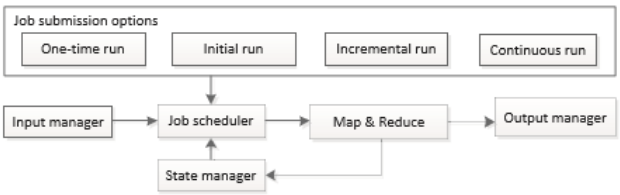

Fig. 3. Framework for incremental data processing based on MapReduce

VI. LITERATURE SURVEY

1. Iterative processing

A number of distributed frameworks have newly emerged for big data processing. HaLoop improves the efficiency of iterative computation by making the task scheduler loop aware and by employing caching mechanisms. Twister employs a lightweight iterative MapReduce runtime system by sensibly constructing a Reduce-to-Map loop. IMapReduce supports iterative processing by directly passing

 ISSN: 2321-8169
58 – 62

the Reduce outputs to Map and by distinguishing variant state data from the static data.

2. Incremental processing for one-step application.

Besides Incoop, several recent studies aim at supporting incremental processing for one-step applications. Incoop detects changes to the inputs and enables the automatic update of the outputs by employing an efficient, fine-grained result reuse mechanism. This incremental nature of data suggests that performing large-scale computations incrementally can improve efficiency dramatically. But Incoop supports only task-level incremental processing. So, Incoop do not allow for reusing the large existing base of MapReduce programs. Incoop supports only one step computation.

3. Incremental processing for iterative application.

Naiad proposes a timely dataflow paradigm that allows stateful computation and arbitrary nested iterations. To support incremental iterative computation, programmers have to completely rewrite their MapReduce programs for Naiad. In comparison, we extend the widely used MapReduce model for incremental iterative computation. Existing Map-Reduce programs can be slightly changed to run on i2MapReduce for incremental processing.

3.1 Continuous MapReduce

Ad-hoc data processing is a critical paradigm for wide-scale data processing especially for unstructured data. Ad-hoc data processing abstraction is a distributed stream processor based on MapReduce programming model to support continuous inputs. MapReduce Online adopts pipelining technique within a job and between jobs, supports single-job and multi-job online aggregation, and also provides database continuous queries over data streams. MapReduce programming model with the continuous query model characterized by CutRewind to process dynamic stream data chunk. CMR, continuous MapReduce, is architecturefor continuous and large-scale data analysis. Continuous processing is a special case of incremental processing.

3.2 Incremental parallel data processing

It is a generalized architecture for continuous incremental bulk processing. It takes the prior state as an explicit input combined with the new input. A set of dataflow primitives is also provided to perform web analytics and mine large scale and evolving graphs. Percolator is a system for incrementally data processing by using distributed transactions and notifications. It is mainly used to create Google Web search index. PIESVM is a special parallel incremental extreme SVM classifier. It is designed based on MapReduce model and can save training time for SVM algorithm.

4. MapReduce

Implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers and the system easy to use: hundreds of MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google's clusters every day. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The runtime system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine Communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

5. Dataflow

Systems such as CIEL, Spark, Spark Streaming and Optimus extend acyclic batch dataflow to allow dynamic modification of the dataflow graph, and thus support iteration and incremental computation without adding cycles to the dataflow. By adopting a batchcomputation model, these systems inherit powerful existing techniques including fault tolerance with parallel recovery; in exchange each requires centralized modifications to the dataflow graph, which introduce substantial overhead that Naiad avoids.

6. Asynchronous computation

Several systems have abandoned synchronous execution in favor of a model that asynchronously updates a distributed shared data structure, in order to achieve low-latency incremental updates and fine-grained computational dependencies. Percolator structures a web indexing computation as triggers that run when new values are written into a distributed key-value store.

7. HaLoop

Bu etal used a new technique called as HaLoop which is modified version of Hadoop MapReduce Framework, as Map Reduce lacks built-in-support for iterative programs HaLoop allows iterative applications to be assembled from existing Hadoop programs without modification, and significantly improves their efficiency by providing inter- iteration caching mechanisms and a loop-aware scheduler to exploit these caches. HaLoop is built on top of Hadoop and extends it with a new programming model and several important optimizations that include (1) a loop-aware task scheduler, (2) loop-invariant data caching, and (3) caching for efficient fix point verification modification, and significantly improves their efficiency by providing inter-iteration .

VII.IMPLEMENTATION RESULTS

In this section we present comparison results of MapReduce and the extension of MapReduce i.e. incremental MapReduce. In this result, comparison is done on the basis of data content optimization and time optimization. We have given input as a one base file and four iterative files. Then each iterative file is compared with the base file and gives the output in the reduced format. Output of MapReduce is a file of large size while the output of incremental MapReduce is a file of small size than the result of the MapReduce. Comparison results are shown in fig. 4.

 ISSN: 2321-8169
58 – 62

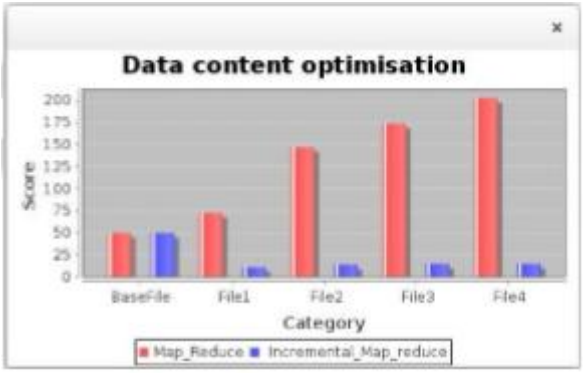

Fig. 4: Comparison graph of data content optimization in incremental MapReduce vs. plain MapReduce

Output time required for processing big data file in MapReduce is more while the output time required for processing big data file in incremental MapReduce is less than the result of the MapReduce. Comparison results are shown in fig. 5.

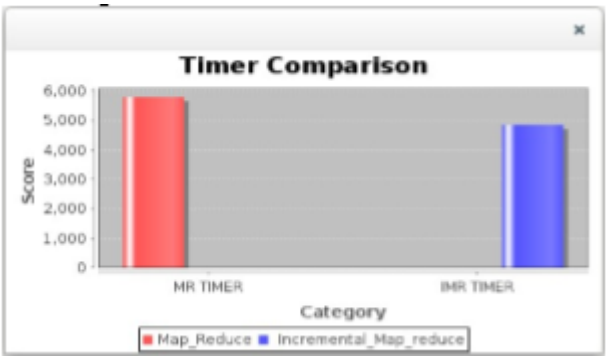

Fig. 5: Comparison graph of time optimization in incremental MapReduce vs. plain MapReduce

As shown in above fig. for the same input data sets there are different outputs for MapReduce and incremental MapReduce. Base file is of the same size in both MapReduce and incremental MapReduce. After the processing of iterative files with its base file, optimization of data content is done and processing time required in incremental MapReduce is also optimized. From the above graph, we see that the more optimization of data content is occurred in the incremental MapReduce than the plain MapReduce. As more the iterative files taken, more optimized data in the incremental MapReduce. For this output, the processing of the algorithms is iterative processing. Which means the complete processing of algorithms should be iteratively in the order of their execution. Execution of PageRank algorithm, k-means clustering algorithm, MapReduce and Incremental MapReduce is done iteratively. For the execution of these algorithms there is no. of dependencies. In this method the major merit is the optimization in the data content and processing time required. We are going to evaluate the performance of our projected scheme in terms of the computation overhead introduce by each operation. Base file and iterative files are taken as the computing parameter. When the number of iterative files increases, then there is more optimization on data content and time required for processing in incremental MapReduce.

VIII.CONCLUSION

For Big Data Processing, Hadoop is one of the most useful tool, while MapReduce is used as programming model for processing big data efficiently. Along with this, Incremental MapReduce Optimizes the Data Content and minimizes the time required for Processing Big Data.

REFRENCES

[1] Yanfeng Zhang, Shimin Chen, Qiang Wang, and Ge Yu,Member, IEEE, “i2MapReduce: Incremental MapReduce for Mining Evolving Big Data” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 7, JULY 2015.

[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] Y. Zhang and S. Chen, “i2mapreduce: Incremental Iterative MapReduce,” CoRR, vol. abs/1501.04854, 2013.

[4] C. 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.

[5] Jeffrey Dean et al. MapReduce: Simplified data processing on large clusters. In Proceedings of the 6th USENIX OSDI, pages 137–150, 2004.

[6] D. G. Murray, F. McSherry, R. Isaacs, M. Isard, P. Barham, and M. Abadi, “Naiad: A timely dataflow system,” in Proc. 24th ACM Symp. Oper. Syst. Principles, 2013, pp. 439– 455.

[7] Yingyi Bu_ Bill Howe _ Magdalena Balazinska _ Michael D. Ernst “The HaLoop Approach to Large-Scale Iterative Data Analysis” VLDB 2010 paper “HaLoop: Efficient Iterative Data Processing on Large Clusters.

[8] Solanke Poonam G. and B. M. Patil “Incremental MapReduce for Big Data Processing” International Journal of Computer Engineering and Applications, Volume X, Issue II, Feb. 16 .