Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review on Data Merging and Data Movement to Perform Accelerate Hadoop System

Prof Ajit R. Patil*
Department of Computer Engineering, University of Pune,
Bharati Vidyapeeth's College of Engineering Lavale, Pune, Maharashtra, India

Abstract— Hadoop is popular framework for storing and processing big data in cloud computing. Map Reduce and HDFS are the two major components of Hadoop. Map reduce is programming model for Hadoop and HDFS is a Hadoop distributed file system which is mainly used as storage component for Hadoop framework. Existing Hadoop system has many performance issues like serialization barrier, repetitive merges, portability issues for different interconnects. An effective and efficient I/O capability is also required for Hadoop. As data size is increasing day by day the performance of Hadoop is becoming a critical issue. To handle large dataset needs to improve the performance by modifying existing Hadoop system. The network levitated merge algorithm is used to avoid repetitive merges. A full pipeline is designed to overlap the Hadoop shuffle merge and reduce phase. Hadoop-A i.e. Hadoop Acceleration framework overcomes the portability issue for different interconnects. It also speeds up data movement and also reduces the disk access.

Keywords— Hadoop, HDFS, Cloud Computing, Big data.

I. INTRODUCTION

In the current age of information the requirement of data is high. The data generated from different sources is in terabytes a day which is also known as big-data. Big-data isnot just big in size, but big data have data of different types and of different sizes. This big-data is used for many application and business related services like business intelligence. To store and process this large amount of data we need an efficient and fault tolerant system. Hadoop is open source software framework to store and process this big data efficiently. It is designed in java language. Map reduce is implementation of Hadoop system for cloud, map reduce is a programming model to write applications for processing. big data. Hadoop is used by many organizations like Yahoo,Google, Facebook and it is maintained by Apache Foundation. Map-reduce is implemented with the help of two components: a job tracker and multiple task trackers. The job tracker is responsible to command the task trackers through two main functions i.e. map tasks and reduce tasks, the task trackers used to process data as per the commanded by job tracker. Job tracker is also in-charge of scheduling map task and reduces task to task trackers, it assigns job to the task trackers and also collects the intermediate results. Hadoop has name node and data node to manage and process data. Name node is node which stores file system metadata, and data node is actually store the data. In the overall operation, reduce task fetches intermediate data from map task which leads to shuffling and merging of data and it leads to serialization barrier which delays reduce operation. Hadoop merges these intermediate data segments of map task when number of data segments goes over a threshold. The current merging algorithm repetitively merges. data segments which causes multiple rounds of disk access for same data. This degrades the performance of Hadoop.

Many studies have been carried out to improve the performance of Hadoop. Yu [1] proposed new merging algorithm and a new framework of Hadoop. Jiang [2] identified the four factors that have effect on map reduce performance. Condie [3] proposed a direct channel between map task and reduce task.

II. LITERATURE SURVEY

As data size is growing day by day the management of data is becoming a critical issue. The data becomes a Big Data. Big data is not only big in size but it also contains data of different size, format, and from different sources. As an example New York stock exchange generate data in Terabytes a day, popular social networking site Facebook uploads millions of images and videos per day whose size in Terabytes. This data may require for future references or some business intelligence purpose for many organizations. Existing database management system is not capable of managing this large amount of data. It has many reasons like data security, manageability, portability, recovery from failure etc. which leads to the need of effective and fault tolerant system. Google invented a new file system known as Google file system for his own purpose of big data management. On basis of Google's file system the Hadoop HDFS and Map Reduce is designed. Hadoop is maintained by Apache Foundation and it is supported by many organizations like Yahoo, Facebook.

Existing Hadoop system has many performance and security issues. Hadoop has two main components map and reduce. The data which is supplied by client to Hadoop system is divided into multiple splits. Each split is assigned to each map task. Map task generates the key value pair of input data. The mappers job is map input key value pair to intermediate key value. Mapper transforms the input record to intermediate record. The number of maps are depends upon number of input blocks. This intermediate data is supplied to reduce task. Reduce task merges these key value pair into single value. During the map task the data is sorted according to user query and shuffled.

We can improve performance with various factors like merging, data I/O etc. Many studies have been carried out to improve the performance over existing system. Numbers of different improvements are made.

Yu[1] suggested many changes to existing system. He invented a new algorithm network levitated merge. Reduce task fetches intermediate data i.e. output of map task and store it locally onto memory. This leads to multiple disk access and many I/O operations which require more execution time. New network levitated merge overcomes this problem. In this algorithm instead of fetching whole segment only small header is fetched.

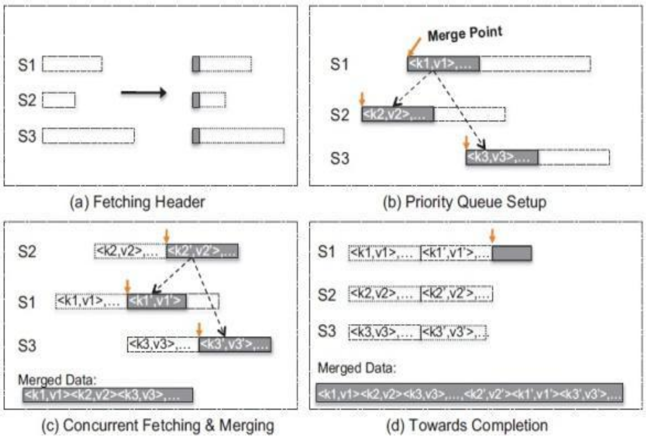

Fig.1 Network Levitated Merge [1]

For example suppose three segments S1, S2 and S3 are to be fetched merged instead of fetching the segment to local disk only the header file of every segment is fetched. Each header file consists of offset, partition length and key/valuepair. By using these key/value pair a priority queue is constructed to organize these segments. The fetching process of header is continues till a certain limit is reached. When the process of fetching is completed the priority queue is building is also completes. And then the merging operation is started. Because the header file is small so memory I/O is also less, which leads improvement in performance. When all the remaining segments are completely fetched then the complete priority queue is established. Then the root of priority queue is extracted i.e. first key/value pair from queue and the merging operation is started. The next root is the next key value pair from the queue. When merging of record from segment is completed next record from next segment is used to resume merge operation. As data comes the fetching and building of priority queue is concurrently in process. This algorithm uses only one queue but when data sizeincreases performance also decreases. So there is need to change the currents algorithm. There is need to analyze the performance of this on cloud computing.

In this paper they also proposed a pipelined structure of Hadoop shuffle, merge and reduce phase which overcomes the serialization barrier between reduce and merge phase as there is pipeline structure so without waiting reduce task directly fetches the header. Due to this intermediate data is fetched as map out file is generated, this decreases the execution time.

A portable Hadoop acceleration framework i.e. Hadoop- A[1] is proposed. Existing Hadoop system doesn't have support for different network interconnects. It only supports TCP/IP transport protocol it has no support for RDMA i.e. Reduced Direct Memory Access which good in high performance communication. In Hadoop-A two new plugin components are added MOF-Supplier and NET-Merger to support RDMA capable interconnects. These two components have C implementation which allows choice of connections like RDMA. This framework is optional i.e. user can enable or disable this framework by setting a parameter. Implementation results shows that Hadoop-A decreases the execution time by 50% and improves the throughput.

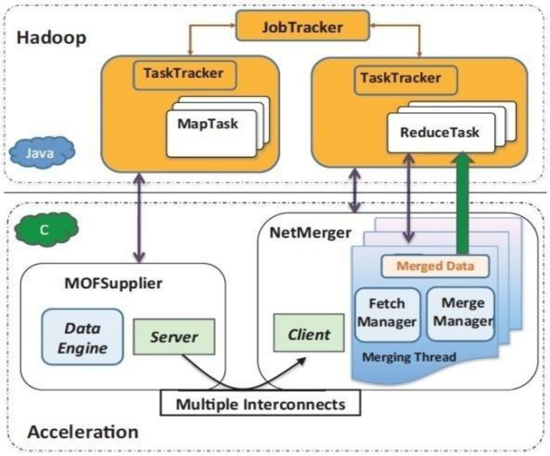

Fig.2 Hadoop-A Framework Architecture [1]

The problem in this system is as number of segments increases the buffer memory requirement is also increases. This may create a scalability problem in future. Jiang[2] identified four important elements that notably effect on performance of Hadoop which are I/O mode, Indexing, Parsing, and Sorting. Map function doesn't take input directly from storage, a mediator called as reader is used. A reader takes data from storage and put it into map buffer. There are two methods for reading data i.e. direct I/O an streaming I/O. Streaming is mainly used for nodes which are placed locally or remotely. And direct I/O used for local nodes. Existing Hadoop uses only streaming I/O method no matter the node locally placed or remotely placed. And experiment shows that streaming I/O has 15% more performance than direct I/O method. Map-reduce uses range indexing scheme that uses same size data chunk to create data index. Next factor is parsing i.e. means when reader reads data from storage to buffer then there is need to convert this raw data into set of record having key/value pair. In this the fields from value part needs decoding into appropriate type. This decoding has two types immutable and mutable. Immutable decoder is much slower than mutable decoder. Next important factor they considered is sorting and merging. Sorting and merging of data is important task in map reduce. A finger print mechanism is proposed to improve the sorting of keys, which reduce key comparisons to improve performance.

Condie [3] designed a architecture which directly connects map and reduce tasks. The intermediate data is pipelined within different operators. They designed a new prototype known as Hadoop online prototype[3]. This modification in architecture has several advantages like reducers fetches data quickly by mapper, it also helps to solve continues queries like different logs. But this raises several design challenges in pipeline. This also indirectly causes wolfish communication. This approach requires coordination between map and reduce.

III. CONCLUSIONS

In this paper study of various papers are performed. To improve Hadoop performance different methods and algorithms are suggested. Many of them are also shows effective results in performance. But there is still need to improve the different factors related to performance such as merging. Network levitated merge has improved the merging performance but it uses only single priority queue. There is lack cloud implementation. Suggested algorithms are not tested on cloud computing. The performance may change on cloud environment. So there is need to improve and test the performance on cloud environment.

REFERENCES

[1] Weikuan Yu, *Member, IEEE*, Yandong Wang, and Xinyu Que Design and Evaluation of Network- Levitated Merge for Hadoop Acceleration IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS

[2] Dawei Jiang Beng Chin Ooi Lei Shi Sai Wu The Performance of MapReduce: An Indepth Study Proceedings of the VLDB Endowment, Vol. 3, No. 1

[3] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein MapReduce Online Yahoo! Research

[4] Apache Hadoop Project, http://hadoop.apache.org

[5] Hadoop- The Definitive Guide, 3rd Edition by Tom White

