

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.459 – 462

RESEARCH ARTICLE



ADDRESSING BIG DATA WITH HADOOP

TWINKLE ANTONY

Department of Computer Science & Engineering
Jyothi Engg. College, Cheruthuruthy

SHAIJU PAUL

Assistant Professor
Department of Computer Science & Engineering
Jyothi Engg. College, Cheruthuruthy

Abstract— Nowadays, a large volume of data from various resources such as social media networks, sensory devices and other information serving devices are produced. This large collection of unstructured, semi structured data is called big data. The conventional databases and data ware houses can't process this data. So we need new data processing tools. Hadoop addresses this need. Hadoop is an open source platform that provides distributed computing of big data. Hadoop composed of two components. A storage model called hadoop distributed file system and computing model called MapReduce. Map reducer, is a programming model for handling large complex task by doing two steps called map and reduce. In map stage the master node partition the problem into sub problems and distribute the task into worker nodes. The worker nodes pass the result to master node after solving the problem. In the reduce phase the master node reduce the answers of the sub problem to a final solution.

I. INTRODUCTION

The size of the databases used in today's enterprises has been growing at exponential rates day by day. There are a number of applications that produces this data which include social media networks, sensory devices, mobile, tracking devices etc. This large collection of complex unstructured, semi structured data is called Big data. The big data is mainly characterized by three v's. That is velocity, variety, volume. The data volume is increased in high rate. Data Volume growing by a factor of 44. That is the total volume of digital universal in 2009 was 0.8 zettabytes in 2020 the volume will be approximately 35zb. Variety implies Various formats, types, and structures. The data can be Text, numerical, images, audio, video, sequences, social media data, multi-dim arrays, etc. A single application can be generating/collecting many types of data. Data is begin generated fast and need to be processed fast. This data contain useful information and can be mined for getting the result. The traditional relational databases can't address process these data. So we need new data processing technique .Hadoop addresses this need. Hadoop is an open platform to process large amount of big data.

When a company needed to store huge amount of data they will adopt two options. That is either buy a big machine with more CPU, RAM, disk space etc or contacting some database vendors for a bigger solution. The two options have their own problem. There is a limit to how big of machines that you can buy. Also the scaling is very difficult. The second option implies scale horizontally. This solution is not cheap and requires significant investment. Today, the source of data being generated not only by the users and applications but also "machine-generated and such data is exponentially leading the change in the Big Data

space. Dealing with big datasets in the order of terabytes or even petabytes is a challenging. In Cloud computing environment a popular data processing engine for big data is Hadoop-MapReduce due to ease-of-use, scalability, and failover properties.

Hadoop provide distributional storage and distributed computing capabilities. Hadoop distributed system file system (HDFS) provides distributional storage and Map Reduce which is a programming model that provide distributed computing capability. MapReduce handle complex combinations of several task and it was published by Google.

II. BIG DATA: A REVIEW

Big data is a collection of data sets so large and complex which is also exceeds the processing capacity of conventional database systems. Since the data is too big, moves too fast, or doesn't fit the structures of our current database architectures. Big Data is large volume of un-structured (or semi structured) and structured data that gets created from various organized and unorganized applications, activities and channels such as emails, twitter, web logs, Facebook, etc. The process of research into massive amounts of data helps to reveal hidden patterns and secret correlations. This process is known as big data analytics. This useful information's helps the companies or organizations to gain richer and deeper insights and getting an advantage over the competition like reviewing their products and making the business strategies etc

The characteristics of big data include volume, variety and velocity. IBM research find that every day we add about 2.5 quintillion bytes of data. Facebook alone add 500TB of data on daily basis. 90% of worlds data is generated in last 2 years. Google process about 1 petabyte of data every hour. The data getting added is also of various types ranging from unstructured feeds, social media, and multi-media data, sensor data etc. There can be video or audio data, multidimensional data, streaming and static data. The data growth is also increasing. Gartner research finds that data is growing at 80% rate out of which 80% is unstructured. We need to process the data very fast. EMC research indicates that the data increase is following Moore' law by doubling every two year.

III. HADOOP

Hadoop is the one of the big data processing platform. It is a project from the Apache Software Foundation written in Java to support data intensive distributed applications. Hadoop provide distributed computing and distributed storage. Hadoop have mainly two component hadoop distributed file system and MapReduce. Hadoop enables applications to work with thousands of nodes and petabytes of data. The inspiration comes from Google's MapReduce and Google File System papers. Hadoop's biggest contributor is the search Yahoo, where Hadoop is used extensively at many business platforms. A small Hadoop cluster includes a single master node and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications.

An HDFS cluster has two types of node operating in a master-worker pattern: a NameNode (the master) and a number of DataNodes (workers). The NameNode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. DataNodes are the workhorses of the filesystem. They store and retrieve blocks when they are told to (by clients or the NameNode), and they report back to the NameNode periodically with lists of blocks that they are storing.

Name Node decides about replication of data blocks. In a typical HDFS, the default block size is 64MB and replication factor is 3 (second copy on the local rack and third on the remote rack). The replication factor can be changed according to the user wish. HDFS have default replication policy such that No DataNode contains more than one replica of any block and no rack contains more than two replicas of the same block, provided there are sufficient racks on the cluster. The NameNode ensure that each block has sufficient number of replicas. It will continuously check whether the block is over replicated or under replicated. If it is over replicated NameNode will select a replica to remove. If it is under replicated NameNode will create another replica in any node or in the same node. The replica removal or replica creation is done based on the replication policy.

To read a file from the HDFS the client first contact the NameNode and get information such as where are the blocks of the needed file is residing and replicas of each node. Then the client read the block from a node which is more close to the client. In order to write a file again the client need to contact the NameNode and get permission for a write. Then the client write the data into any one of the node and pass the block content into another node or same node in order to provide replication.

During each operation the DataNode send a heartbeat message to the NameNode indicating that the DataNode is still alive and the block replicas are available .If the NameNode doesn't get the Heartbeat within a particular time that DataNode is considered to be out of service and the replicas residing that DataNode is copied into another node.

MapReduce is a data processing or parallel programming model introduced by Google. In this model, a user specifies the computation by two functions, Map and Reduce. The input file is splitted into number of files. In the mapping phase, MapReduce takes the input file and pass each data element to the mapper. In the reducing phase, the reducer processes all the outputs from the

mapper and arrives at a final result. The Map function in each node will not wait for the entire Map process. That is as soon as each Map task is completed the result is passed to the Reduce function. There is an intermediate optional phase called shuffle/sort perform sorting or shuffling of Map output. The underlying MapReduce library automatically done the computation, and handles complicated issues like data distribution, load balancing and fault tolerance. Massive input, spread across many machines, need to parallelize. Map Reduce has gained a great popularity as it gracefully and automatically achieves fault tolerance. It automatically handles the gathering of results across the multiple nodes and returns a single result or set. MapReduce model advantage is the easy scaling of data processing over multiple computing nodes. Map, written by the user, which takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key is and passes them to the reduce function. The reduce function, also written by the user, accepts an intermediate key and a set of values for that key. It merges these values together to form a possibly smaller set of values.

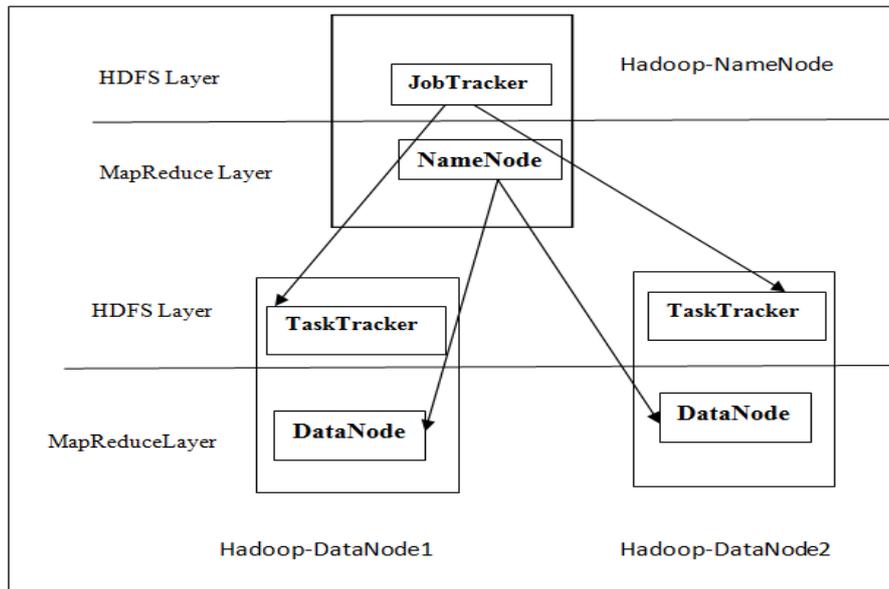


Figure 3.1: Architecture of Hadoop

Figure 1.1 illustrates the Hadoop architecture which contains HDFS and MapReduce. The components inside both HDFS and MapReduce are shown.

IV EXECUTION OVERVIEW OF HADOOP

The MapReduce library in the user program first splits the input files into M pieces of typically 16-64MB per piece. It then starts up many copies of the program on a cluster of machines. The master controls all the function. The rest nodes are workers that are assigned work by the master. There are M map tasks and R reduces tasks to assign. The master picks idle workers and assigns each one a map task or a reduce task. A worker who is assigned a map task reads the contents of the corresponding input file. It parses key/value pairs out of the input data and passes each pair to the user-defined map function. The intermediate key/value pairs produced by the map function are buffered in memory. Periodically, the buffered pairs are written to local disk, partitioned into R regions by the partitioning function. The Map output is stored in the local disk temporarily. As soon as the Reduce function is over it will be removed. This temporary storage of Map output will help in fault tolerance and recovery. That is when an Reduce function is failed the system need not start the Map process. The Map output on the local disk are then passed back to the master who is responsible for forwarding these locations to the reduce workers.

When a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of the map workers. When a reduce worker has read all intermediate data for its partition, it sorts it by the intermediate keys so that all occurrences of the same key are grouped together. The sorting is needed because typically many different keys map to the same reduce task. If the amount of intermediate data is too large to fit in memory, an external sort is used. The reduce worker iterates over the sorted intermediate data and it passes the key and the corresponding set of intermediate values to the user's reduce function. In case of a system failure Completed reduce tasks do not need to be re-executed since their output is stored in a global system. The output of the reduce function is appended to a final output file for this reduce partition. When all map tasks and reduce tasks have been completed, the master wakes up the user program.

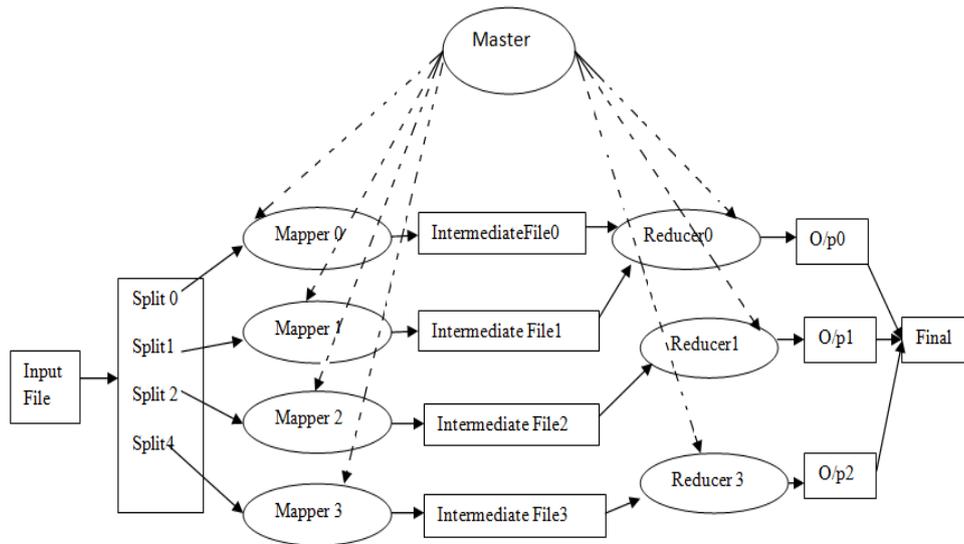


Figure 4.1: Execution overview of MapReduce

Figure 2.1. Map and Reduce function is performed by the worker. The output of both process individually carried out by the worker is passed to the master.

CONCLUSION

Hadoop is an open-source framework for developing and executing distributed applications that process very large amounts of data. It provides the infrastructure that distributes data across a multitude of machines in a cluster and that pushes analysis code to nodes closest to the data being analyzed. Hadoop provides distributed storage called Hadoop distributed file system (HDFS) and distributed computing through a programming model called MapReduce. Map Reduce perform the task by dividing the entire program into smaller program and executed independently. The effectiveness and ease-of-use of the MapReduce method in parallelization involves many data analysis algorithms. An important characteristic of Hadoop is the partitioning of data and computation across thousands of nodes, and the execution of application computations in parallel close to their data. A Hadoop cluster scales computation capacity, storage capacity and I/O bandwidth by simply adding commodity servers. Hence it uses cheap commodity of hardware's the cost of Hadoop is relatively low. Hadoop is not an alternative for databases and data ware houses. It is a framework that provides processing of big data. The processing of structured data is much easier in relational databases.

REFERENCES

- [1] Sanjay P. Ahuja & Bryan Moore " State of Big Data Analysis in the Cloud" Published by *Canadian Center of Science and Education* Dean A. P, *Neural Network Vision for Robot Driving*, (Carnegie Mellon University, 1989)
- [2] Konstantin Shvachko, Hairong Kuang, Sanjay Radia and Robert Chansler," The Hadoop Distributed File System" *Yahoo*
- [3] Fay chang, jeffrey dean, sanjay ghemawat, wilson c. Hsieh, Deborah a. Wallach, mike burrows, tushar chandra, Andrew fikes, and robert e. Gruber "Bigtable: A Distributed Storage System for Structured Data" *Google*
- [4] Ghemawat, S., Gobioff, H., and Leung, S.-T. 2003. "The Google File System". In *19th Symposium on Operating Systems*
- [5] Rabi Prasad Padhy " Big Data Processing with Hadoop-MapReduce in Cloud Systems". *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*
- [6] Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S.R., and Stonebraker, M. "A comparison of approaches to large-scale data analysis". In *Proceedings of the 35th SIGMOD International Conference on Management of Data*. ACM Press, New York, 2009, 165-178.
- [7] bouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J., Silberschatz, A., and Rasin, A. "HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads". In *Proceedings of the Conference on Very Large Databases, 2009 Implementation* (Berkeley, CA, 2004).
- [8] D eWitt, D.J. and Gray, J." Parallel database systems: The future of high-performance database systems". *Commun. ACM* 35,6(June1992),8