Volume: 9 Issue: XI Month of publication: November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38877

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Using HDFS to Load, Search, and Retrieve Data from Local Data Nodes

Shubh Goyal[1], Rahul Agrawal[2], Malav Seth[3]
[1, 2, 3]*Student Final year, Information Technology, SVKM's NMIMS MPSTME, Shirpur, India*

Abstract: By utilizing the Hadoop environment, data may be loaded and searched from local data nodes. Because the dataset's capacity may be vast, loading and finding data using a query is often more difficult. We suggest a method for dealing with data in local nodes that does not overlap with data acquired by script. The query's major purpose is to store information in a distributed environment and look for it quickly. In this section, we define the script to eliminate duplicate data redundancy when searching and loading data in a dynamic manner. In addition, the Hadoop file system is available in a distributed environment.
Keywords: HDFS; Hadoop distributed file system; replica; local; distributed; capacity; SQL; redundancy.

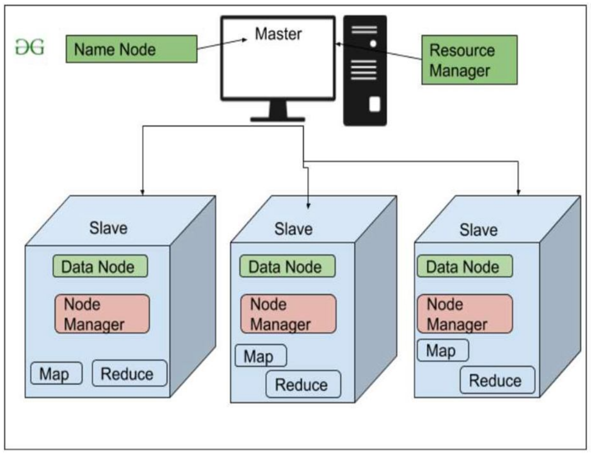
I. INTRODUCTION

The Hadoop file system is the Hadoop application environment's primary storage. Name Node is the core of the Hadoop system. The name node's job is to keep track of the file's path and where it's stored in the data node. Data nodes are storage discs that store information in a specific location on the local disc. Only one Name Node and numerous data nodes function locally in a distributed system. This is the notion of replica analysis, and it is the biggest benefit of the Hadoop file system to retain the file in three separate data nodes in three different places. In a distributed context, the Hadoop file system allows for quick data movement between nodes and local data nodes. The Map-Reduce system for data transmission and storage in Hadoop is introduced. The map reduction framework reduces the time it takes to send data and search for it in a distributed operation. Replica techniques are one of the greatest ways here. Data may be simply obtained using this way. Because there are three copies of the file in three separate locations. HDFS employs the Master/Slave skeleton, which consists of a Master and a Slave. Hadoop local is made up of many data nodes and a Name Node (NN). Every action that isn't done in the Name Node is performed on the Data Node. The Hadoop framework is used to analyze huge datasets. Hadoop scripts written in Java are used in this process and allow distributed applications to be computed.

A. Architecture Of Hadoop

The following architecture explains how to construct an application that can run on numerous local nodes (Figure 1).

1) Map-Reduce Technique
2) HDFS Map-Reduce is a Java-based programming framework for big distributed applications. Google is a great example of a search engine. It is cost- effective to track a big volume of data with a single operation. Although a huge number of nodes are participating in a single operation, each node provides accurate data to the client.



II. MAP-REDUCE STAGES

The java program is executed by Hadoop in three stages: mapper, shuffle, and reducer.

1) *Mapper Stage:* The input data from the data node is taken by the mapper step. On the Hadoop file system, the input file is accessible. The input file is sent to the mapper, which examines each line and runs the input file. Create many data pieces as well. The input file might be in any structure, semi- structure, or unstructured format.
2) *Reducer Stage:* The file reached the reducer stage after passingthrough the mapper stage. The reducer at this level is a mix of Shuffle and Reduce. Here, the file is run, and a new output is generated. The Hadoop file system will be used to store the results. The actions that follow are used to process the reducerstage.

The Hadoop framework organizes all information about data transferring, such as issuing jobs, verifying job execution, and storing results on distributed local nodes

The tasks are executed on local disc, which reduces network traffic

When the job is complete, the local gather reduced data from the data node

III. MAP-REDUCE INPUT/OUTPUT

The mapper's input file is made from of (key value). Thereducer receives the input file after the mapper has divided it into (key value) pairs. The reducer takes the data from the mapper and shuffles it before producing a set of outputs in the form of keys and values. input and output The following are the output kinds of a Map- Reduce job: k1, v1> input k2, v2>map v3>reduce, k3>reduce, v3>reduce, v3>reduce, v3>re The final product is data that has been reduced.

Google's file system is used. The Hadoop file system isa distributed storage system is designed to run on low- cost hardware.

Hadoop framework is made up of two parts:

1) Hadoop Common: Java libraries and common utilities are necessary to conduct Map-Reduce jobs in the Hadoop environment.
2) Hadoop YARN: This stands for Yet Another Resource Negotiator s foundation for a task schedule and local resource.

IV. HADOOP DISTRIBUTED FILE SYSTEMSTORAGE PROCEDURE

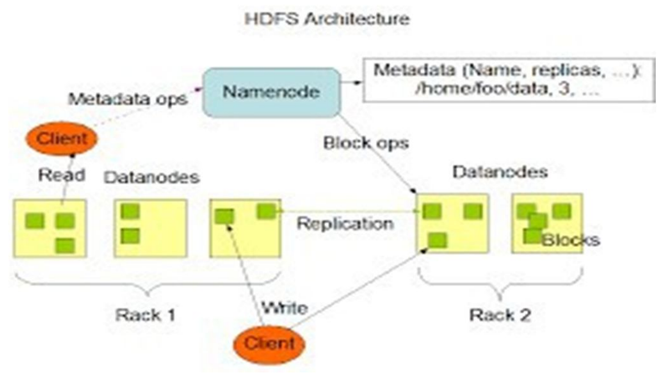
Hadoop file system effectively stores files on local data nodes. Each file is divided into 64MB blocks. The idea of replica is used to keep three copies of each filein distributer local contexts.

The Hadoop file system is used to effectively load data into a local Data Node. The input data is stored in Hadoop and is separated into chunks with a default block size of 64 MB.

The Hadoop file system is divided into five sections (Figure 2):

1) Job Tracker
2) Task Tracker
3) Name Node
4) Data Node
5) Secondary Name Node

Name Node, Secondary Name Node, and Job Trackerare the three Hadoop local components that are all assessed on the same server. The input file will be divided into blocks and saved in three separate datanodes and files named X, Y, and Z while doing the operation in Hadoop local. Name Node keeps track of the location and path of these files. One of the most significant concepts of the Name Node is that it merely saves the file paths and does not perform any of the tasksspecified by the client. If the Name Node fails, the Secondary Name Node will take over as the primary server until the Name Node is restored. JT-> JobTrackeris in charge of distributing jobs to Task Tracker, and Task Tracker is in charge of performing the tasks accessible from three data nodes. The rationale for assigning the job to three data nodes is to determine which of the three data nodes will perform the assignment first. The goal of this method is to cut down on time and improve programme efficiency. If one of the data nodes fails or is destroyed, the other two will continue to run the application. The essential criterion for executing jobs locally are transmitting Heart-Beat messages from Job Tracker to Task Tracker at regular intervals. Job Tracker sends out heartbeat notifications to let me know I'm alive. The execution of tasks in data nodes The Task Tracker keeps an eye on the Map Reduce-Program and verifies the status of any jobs that have been reported to the Job Tracker. On the intra communication system paradigm, the following suggested algorithm defines the full process of communication between Name Node and other local nodes.



V. RELATED WORK

1) Only the appropriate subset of total entities was selected by the 'Simon Miles' to explain the origin of the query results. Relevant data may not always provide the correct information to real-time processes. The query sources are assessed using SQL relational queries. The query provenance is not suited forSQL relational queries. It returns information depending on an entity or attribute.
2) Glavic and Alonso (2009) investigated the effectiveness of query rewriting for locating the query's origin in databases. They keep logsand transit time, i.e. query execution time, while rewriting a query. We utilised one method for query execution with origin, whichis combining the databases.
3) Widom (2004) describes a strategy for managing entity data joins based on uncertainty and lineage. That is to say, it is the system's storage object. As a result, the data base system stores the appropriate information.The query starts with the data in each row and column.
4) With no modifications to the SPARQL, Halpinand Cheney (2014) show how the SPARQL works to search the query origin within replica storage. We presented a single execution method in many data warehouses based on attribute data selection based on this approach.This isn't entirely true, but it follows the query's dynamic execution procedure.
5) The query re-optimisation for repeated query execution was determined by 'Grafe and Ward.'They suggested a novel query reduction model optimization approach. The query was dynamically compiled without delay by this model.
6) The researchers 'Kabra and DeWitt' devised a strategy for acquiring statics at the time of query execution. In order to optimise queries,they integrated both static and dynamic runtime execution.

VI. METHODS

A. *Apache Pig*

This is high level procedural language for applying the quires on big datasets in environment of Hadoop and Map Reduce. It contain Java packages and writing scripts are executed any language, by running on Java Virtual Machine. This script is like SQL queries and execute in local environment datasets. It canevaluate large size of datasets within the fraction of seconds fast and efficiently. The advantage of Apache PID script uses relational operation to data retrieval process.

B. *Apache Pig Latin*

For Hadoop applications, APL is fairly similar to SQLin terms of script creation. We used to construct the Apache pig Latin instead of MR programmes. ATL is a series of statements that are run and generate different outcomes for the user while loading, searching, and retrieving data from local data nodes on HDFS 47. Bags,maps, tuple, and scalar are used to compress data for a certain input and output file.

C. *Executing Modes*

There are two execution modes in Apache Pig:

1) *Local Mode:* The input file is obtained from the local data node in this mode. To execute the Apache pig scriptin local mode, use the following command. Thecommand 'pig –x local' can be used to specify the MR mode.
2) *Map-Reduce mode:* The Hadoop local environment will be used in this Map-Reduce mode. Use the Map- Reduce mode using the following command.

VII. CONCLUSION

We begin the first step in the process of storing and retrieving data and results from Hadoop's local data nodes. The query is run on static data on local nodes ona small number of rows in the data file. We're looking atthe ability to run a single query across several data nodeswithout interruption. Finally, instead of utilising SQL, the user uses the Apache Pig Script to obtain information from any local data store. We may also use other patterns to search the data. The data can be stored in any format, but it must be filtered before it can be shown to the consumers.

REFERENCES

[1] Glavic, B. and Alonso, G. (2009) 'The perm provenance management system in action', Proc. ACM SIGMOD Int. Conf. Manage. Data, pp.1055–1058.
[2] Halpin, H. and Cheney, J. (2014) 'Dynamic provenance for SPARQL updates', in Mika, P. et al. (Eds.): The Semantic Web, Springer, Berlin, Germany, pp.425–440 [Online]
[3] Widom, J. (2004) Trio: A System for Integrated Management of Data, Accuracy, and Lineage, Tech. Rep., Stanford Info Lab, August
[4] Arab, B., Gawlick, D., Radhakrishnan, V., Guo, H. and Glavic, B. (2014) 'A generic provenance middleware for queries, updates, and transactions', Proc. 6th USENIX Workshop Theory Practice Provenance, June [Online]
[5] Biton, O., Cohen-Boulakia, S. and Davidson, S.B.(2007) 'Zoom*User views: Querying relevant provenance in work flow systems', Proc. 33rd Int. Conf. Very Large Databases, pp.1366–1369.
[6] Chebotko, A., Lu, S., Fei, X. and Fotouhi, F. (2010) 'RDFProv: a relational RDF store for querying and managing scientific workflowprovenance', Data Knowl. Eng., Vol. 69, No. 8, August, pp.836–865.
[7] Gadelha Jr., L.M., Wilde, M., Mattoso, M. and Foster, I. (2012) 'MTCProv: a practical provenance query framework for many-task scientific computing', Distrib. Parallel Databases, Vol. 30, Nos. 5–6, October, pp.351– 370.
[8] Geerts, F., Karvounarakis, G., Christophides, V. and Fundulaki, I. (2013) 'Algebraic structuresfor capturing the provenance of SPARQL queries', Proc. 16th Int. Conf. Database Theory,pp.153–16 fvvvffvv