Volume 5, Issue 2, February 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Ad-hoc Reporting Using Hadoop

[1]Sayalee Rajesh Pote, [2]Minal Prakash Ugale, [3]Rohan Mahesh Choudhary, [4]Smita Dasharath Dange
[1, 2, 3] Computer, Mumbai University, Maharashtra, India
[4]Asst. Professor Computer Department (F.C.R.I.T), India

Abstract— Hadoop is being used widely for production and research in companies and organisations across the globe to handle their big data problems. This open source framework by Apache allows for the distributed processing of large data sets across clusters of computers, each offering local computation and storage. Recognizing the contribution of Hadoop to big data problems and the number of advantages it offers, this paper proposes a platform on top of Apache Hadoop which can be used for creating ad-hoc reports on the fly employing pig script templates for processing the queries.

Key words— Big data, Hadoop, HDFS, MapReduce, Pig

I. INTRODUCTION

The world is awash in a flood of data today. Not just human generated data but machine generated data is also entering our huge data stores in great volume and variety at unprecedented rates. Extracting relevant and useful information from this ocean of data in an economical and efficient way is a task at hand for the world. Thus, Big Data analysis now drives nearly every aspect of our modern society, including mobile services, social networking, retail, manufacturing, financial services, life sciences, and physical sciences. This Big Data is a very large and loosely structured data set that defies traditional storage thus defying the traditional methods and technologies of processing data. The technologies associated with big data analytics include NOSQL databases, Hadoop and MapReduce. Big Data analysis and the Apache Hadoop open source Technology are rapidly emerging as the preferred duo to address the increasing processing demands of Big Data. Some of the many advantages of Hadoop that help it have an edge over other Big Data Analysis Technologies is its ability to cost-effectively integrate data from multiple data sources, its scalability, flexibility and its speed of processing data. Owing to this, it is said that Hadoop is not just a fad but it is here to stay and rule the gigantic world of Big Data. This nature of Hadoop makes it the best pick for a platform to create a tool for adhoc reporting over a huge amount of data.

II. BIG DATA AND HADOOP

Big Data refers to massive volume of both structured and unstructured data that is so large, varied and exponentially growing that it is difficult to process using traditional databases and software techniques. Many companies are forced to discard valuable data because the cost of storing it is simply too high and new data sources make this problem much worse. The EMC Digital Universe data growth study2 predicts nearly 45-fold annual data growth by 2020[1]. Hadoop is a high-performance distributed data storage and processing system which provides two important services to tackle this situation- It can store any kind of huge amount of data from any source, very economically and it can do very sophisticated analysis of that data easily and quickly. Hadoop handles hardware and system failures automatically, without losing data or interrupting data analyses and thus is very reliable and robust. Hadoop's architecture is such that it can store and process terabytes, and even petabytes, of unstructured data inexpensively using clusters of low-cost commodity servers. One study by Cloudera suggested that Enterprises usually spend around $25,000 to $50,000 dollars per tera byte per year. With Hadoop this cost drops to few thousands of dollars per tera byte per year [2]. Each of the servers in the cluster of Hadoop has local CPU and storage.

The core components of Hadoop are:

- The Hadoop Distributed File System (HDFS)
- MapReduce.

III. HDFS

HDFS is the storage system for a Hadoop cluster which is a distributed file system that provides high-throughput access to data[3]. When data arrives at the cluster, the HDFS software breaks it into pieces and distributes these pieces among different servers participating in the cluster. Each server stores just a small fragment of the complete data set, and each piece of data is replicated on more than one server[1].

The main components of HDFS are as described below:

- **NameNode** is the master node of the entire cluster system. It maintains the name system i.e. the directories and files and manages the blocks which are present on the DataNodes.
- **DataNodes** are the slave nodes which are deployed on each machine to provide the actual storage. They are responsible for serving read and write requests for the clients.

- **Secondary NameNode** is responsible for performing periodic checkpoints. In the event of NameNode failure, the NameNode can be restarted using the checkpoint. It is basically a backup node in case of failure of the master node.[3]

IV. MAPREDUCE IN HADOOP

MapReduce is a distributed data processing framework which uses the MapReduce programming paradigm. In the MapReduce paradigm, each job has a user-defined map phase which is a parallel, share-nothing processing of input; followed by a user-defined reduce phase where the output of the map phase is aggregated. As Hadoop stores the entire dataset in small pieces across a collection of servers, analytical jobs are distributed, in parallel, to each of the servers storing part of the data. Each server evaluates the query against its local fragment simultaneously and reports its results back for collation into a comprehensive answer. Typically, HDFS is the storage system for both input and output of the MapReduce jobs. Thus, MapReduce is basically a plumbing that distributes the work and collects the results[1][3].
The main components of MapReduce are as described below:

- **JobTracker** is the master of the system which manages the jobs and resources in the clusters or in the TaskTrackers. The JobTracker tries to schedule each map on the TaskTracker which is running on the DataNode. JobTracker is deployed on the master node along with the NameNode.
- **TaskTrackers** are the slaves which are deployed on each machine. They are responsible for running the map and reducing tasks as instructed by the JobTracker.
- **JobHistoryServer** is a daemon that serves historical information about completed tasks or jobs. Typically, JobHistory server can be co-deployed with JobTracker, but it can also run as a separate daemon.[3]

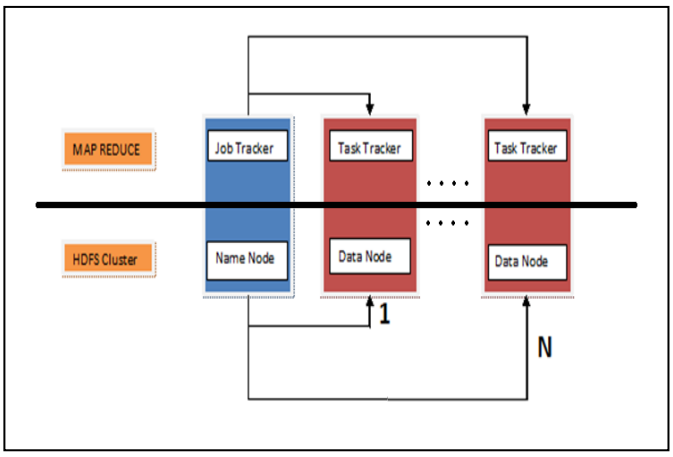

Fig.1. Components of a Hadoop System

V. PIG

Pig is a high level scripting language that is used with Apache Hadoop that enables data workers to write complex data transformations without knowing Java. Pig's simple SQL-like scripting language is called Pig Latin and the code written using Pig Latin is called Pig Latin Script. Pig was initially developed at Yahoo! to allow people using Hadoop to focus more on analyzing large data sets and spend less time having to write mapper and reducer programs [4]. Pig Latin defines a set of transformations on a data set such as aggregate, join and sort but it is sometimes extended using user defined functions which the user can write in Java or a scripting language and then call directly from the Pig Latin. It translates the Pig Latin script into MapReduce so that it can be executed within Hadoop.

Pig Latin statements follow this general flow:

- **Load** : Read data to be manipulated from the file system.
- **Transform** : Manipulate the data.
- **Dump/Store** : Output data to the screen or store for processing

Using Pig reduces the time needed to write mapper and reducer programs. There is the flexibility to combine Java code with Pig. Many complex algorithms can be written in less than five lines of human-readable Pig code [5]. Thus, Pig is an extensible, easy to program and a self-optimizing language which perfectly complements the Hadoop Framework.

VI. EXISTING WORK USING HADOOP

Scientists are employing Hadoop to acquire scalable, reliable and faster analysis and storage services for Big Data. One of the examples being that the University of Nebraska-Lincoln constructed a 1.6PB Hadoop Distributed File System to store Compact Muon Solenoid data from the Large Hadron Collider [6], as well as data for the Open Science Grid (OSG) [7]. In the University of Maryland, researchers developed blastreduce based on Hadoop MapReduce to analyze DNA sequences [8]. A novel Hadoop MapReduce framework executed on the Open Science Grid which spans multiple institutions across the United States – Hadoop On the Grid (HOG) has been proposed by Chen He, Derek Weitzel, David Swanson and Ying Lu from University of Nebraska – Lincoln [9]. A system for storage and retrieval of large Resource Description framework i.e. RDF graph using Hadoop and MapReduce and an algorithm to answer SPARQL query using

Hadoop's MapReduce framework to actually answer the queries has also been proposed [10]. Another model has been proposed for performance evaluation of stream log collection using HDFS based on analytics performed by Google for web pages [11]. Intel has implemented a low-cost, fully realized big data platform based on the Intel® Distribution for Apache Hadoop* software (Intel® Distribution) in just five weeks. This platform currently supports three use cases, with more in development, delivering BI results worth millions of dollars to Intel [12].

VII. PROPOSED DESIGN

We propose a system that will allow user to do on-demand reporting by choosing input parameters and deriving adhoc reports based on the parameters. This proposed design is for a tool that can work as a generic tool for adhoc reporting on a data set stored on the HDFS by the administrator. For this, the system has been broken down into some key components. The following Block Diagram shows the overall design of the system:

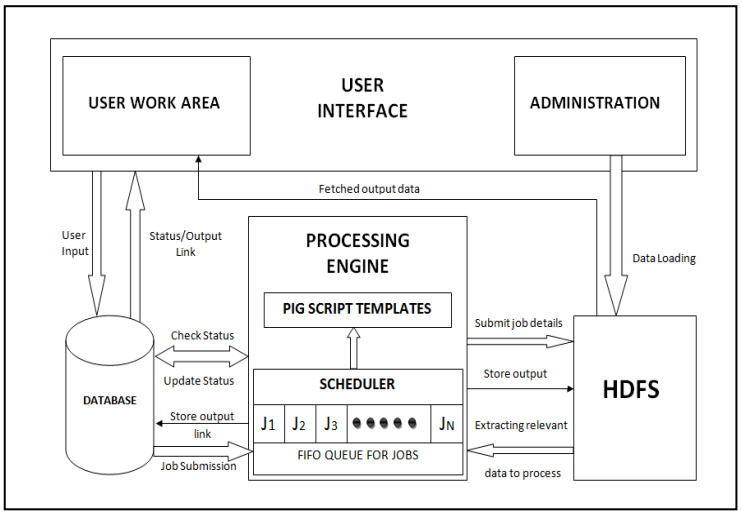

Fig.2. Block Diagram of the proposed system

System is divided into following parts:

A. Administration

This module will be controlled by the system administrator. It is meant for management of metadata, which includes:

- Generating the backend data.
- Loading data to HDFS.
- Purging old data after retention period.

B. User Interface

This module is the interface between the user and the tool. All the adhoc user input will be taken through this module and submitted to the database. This module will have the following features:

- It will provide the users with a list of attributes from which the users can choose the attributes required for their output report.
- It will aslo have aggregation functions or methods as an input in addition to attributes.
- It will redirect the input to a database (DB) where it will be stored in respective tables.
- It will display the Status/Progress of each submitted job to the user.
- Once the submitted job is completed or executed, it will display a link against the submitted job to download the output data.
- After clicking on the Download link, the data will be picked from HDFS will be made available to the user on the UI.

C. Database (DB)

A normal traditional Database will be used to maintain all the data pertaining to a job submitted by the user. DB will have the following records for a JOB:

- Job ID
- Job Entries
- Input parameter for each job
- Job Status
- Output link

When job is submitted from UI by user, its entry will be added in DB table giving each job an unique job ID. DB will be updated based on status of job execution on HDFS thus showing the current status of the submitted job which will

also be shown on the UI. On completion of the job, output link will be created and stored in Database table against job ID so that it can be displayed on the UI for the user.

D. Processing Engine

This module is meant for the actual processing of the job submitted using Pig script templates. It will interact with the DB and the HDFS in the following manner:

- It will have a scheduler that will pick up job details like job id and input parameters from DB on FIFO basis.
- It will then submit it to the HDFS.
- It will maintain PIG script templates to process or run the jobs fetched from DB.
- It will pick the appropriate PIG script template based on the inputs provided by user.
- This script will run in backend to generate output data and store it on HDFS.
- This module will also perform task of checking and updating job status in DB.

E. Output Report Maintenance

Report output will be maintained on HDFS or could also be uploaded on external storage, for instance on the cloud.

The proposed design will be deployed on every node in the Hadoop cluster employed for adhoc query processing.

The main objective of this design is to introduce a tool which will address the adhoc reporting problems of Big Data using Hadoop.

VIII. CONCLUSIONS

Thus we have proposed the design of a generic tool that can be used for adhoc reporting. This tool is a scalable generic tool which can be used to address the adhoc reporting problems associated with Big data using Hadoop framework, which is the best suited framework for big data problems. The fact that this tool uses Hadoop as its platform for data storage and processing, gives it all the advantages that Hadoop offers for big data analysis

REFERENCES

[1] Cloudera white paper,"Ten common Hadoopable problems, real-world Hadoop use cases"
[2] http://blog.cloudera.com/wp-content/uploads/2011/03/ten_common_hadoopable_problems_final.pdf
[3] Hadoop Illuminated by Mark Kerzner and Sujee Maniyam
[4] http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-Win-1.1.0/bk_getting-started-guide/content/ch_hdp1_getting_started_chp2_1.html
[5] http://www-01.ibm.com/software/data/infosphere/hadoop/pig/
[6] http://hortonworks.com/wp-content/uploads/2012/03/Hortonwworks_Tutorial_Pig-5-22.pdf
[7] CERN, "Large hadron collider," http://lhc.web.cern.ch/lhc/.
[8] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. W¨urthwein et al., "The open science grid," in Journal of Physics: Conference Series, vol. 78. IOP, Publishing, 2007, p. 012057.
[9] M. Schatz, "Blastreduce: high performance short read mapping with mapreduce," University of Maryland, http://cgis.cs.umd.edu/Grad/scholarlypapers/papers/MichaelSchatz.pdf.
[10] Chen He, Derek J. Weitzel, David Swanson, and Ying Lu, "HOG: Distributed Hadoop MapReduce on the Grid" (2012). CSE, Conference and Workshop Papers. Paper 231.
[11] Mohammad Farhan Hussain,Pankil Doshi, Latifur Khan and Bhavani Thuraisingham, "Storage and Retrieval of large RDF graph using Hadoop and MapReduce", http://link.springer.com/chapter/10.1007/978-3-642-10665-1_72#page-2
[12] N. Ramasubramanian , Srinivas V.V. , Praveen Kumar Yadav, "Performance Evaluation of Stream Log Collection Using HADOOP Distributed File System", Volume 3, Issue 6, June 2013 ISSN: 2277 128X , International Journal of Advanced Research in Computer Science and Software Engineering
[13] IT@Intel White Paper, Intel IT, Big Data and Business Intelligence, October 2013
[14] http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/best-practices-for-implementing-apache-hadoop-paper.pdf..

