Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

Handling Different Hadoop Platforms with Single Generic File Explorer: A New Approach

Ms. R. S. Sajjan
Department of Computer Science and Engineering
Vidya Vikas Pratishthan Institute of Engineering & Technology
Solapur,India

Sabir Naikwadi
Department of Computer Science and Engineering
Vidya Vikas Pratishthan Institute of Engineering & Technology
Solapur,India

Mudassar Mulla
Department of Computer Science and Engineering
Vidya Vikas Pratishthan Institute of Engineering & Technology
Solapur,India

Shivaling Achalare
Department of Computer science and Engineering
Vidya Vikas Pratishthan Institute of Engineering & Technology
Solapur,India

***Abstract*- Managing and viewing data in HDFS is an important part of Big Data analytics. Generic File Explorer is a GUI-based interface that makes Apache Hadoop easier to use, helps you do that through a GUI very easily instead of logging into a Hadoop gateway host with a terminal program and using the commands. By using Generic File Explorer, file operations in HDFS are only a few clicks away. The Generic File Explorer as well offer direct links to the outputs of your MapReduce jobs, perform all file related operation and admin operation in no time.The Generic File Explorer will be GUI based application which work with many Hadoop platforms simultaneously. In Generic File Explorer user can perform Hadoop operations without typing a single command.**

Keywords: Hadoop, GFE (Generic File Explorer), HDFS, MapReduce, Hadoop Platforms.

1. INTRODUCTION

Due to the rapid growth of Internet the large amount of data is generating. In today's life use of internet is increased because most of the people using social sites, e-commerce sites etc. . For every need people are using internet For example- From education to entertainment, food to shopping, banking to real estate, everything is going with internet. Due tothese reasons every day the volume of data is generating. It is too much difficult and coaster to store and manipulate these large data. To overcome this problem Hadoop is introduced by Apache.

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. The core of Apache Hadoop consists of a storage part Hadoop Distributed File System (HDFS) and a processing part (MapReduce). A Hadoop cluster is composed of two parts: Hadoop Distributed File System and MapReduce. A Hadoop cluster uses Hadoop Distributed File System (HDFS) [1] to manage its data. HDFS provides storage for the MapReduce job's input and output data. It is designed as a highly fault-tolerant, high throughput, and high capacity distributed file system. It is suitable for storing terabytes or petabytes of data on clusters and has flexible hardware requirements, which are typically comprised of commodity hardware like personal computers. The significant differences between HDFS and other distributed file systems are: HDFS's write once- read-many and streaming access models that make HDFS efficient in distributing and processing data, reliably storing large amounts of data, and robustly incorporating heterogeneous hardware and operating system environments. It divides each file into small fixed-size blocks (e.g., 64 MB) and stores multiple (default is three) copies of each block on cluster node disks. The distribution of a blocks increases throughput and fault tolerance. HDFS follows the master/slave architecture.

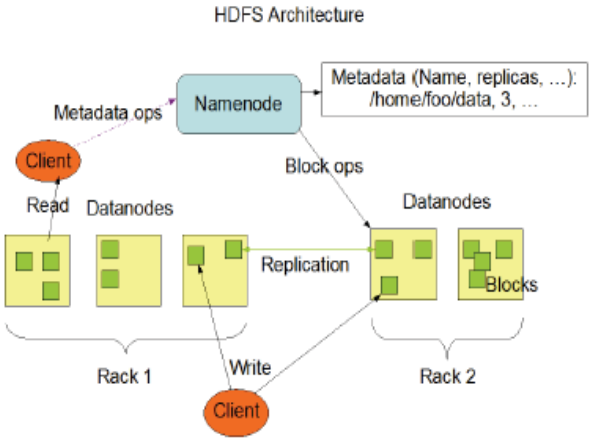

Fig. 1. HDFS Structure. Source: http://hadoop.apache.org

The master node is called the Namenode which manages the File system namespace and regulates client accesses to the data. There are a number of worker nodes, called Datanodes, which store actual data in units of blocks. The Namenode maintains a mapping table which maps data blocks to Datanodes in order to process write and read requests from HDFS clients. It is also in charge of file system namespace operations such as closing, renaming, and opening files and directories. HDFS allows a secondary Namenode to periodically save a copy of the metadata stored on the Namenode in case of Namenode failure. The Datanodes stores

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

the data blocks in its local disk and executes instructions like data replacement, creation, deletion, and replication from the Namenode. Figure1 (adopted from Apache Hadoop Project [2]) illustrates the HDFS architecture.

A Datanode periodically reports its status through a heartbeat message and asks the Namenode for instructions. Every Datanode listens to the network so that other Datanodes and users can request read and write operations. The heartbeat can also help the Namenode to detect connectivity with its Datanode. If the Namenode does not receive a heartbeat from a Datanode in the configured period of time, it marks the node down. Data blocks stored on this node will be considered lost and the Namenode will automatically replicate those blocks of this lost node onto some other Datanode. Hadoop MapReduce is the computation framework built upon HDFS. There are two versions of Hadoop MapReduce: MapReduce 1.0 and MapReduce 2.0 (Yarn [3]).

The Generic File Explorer will be GUI based application which work with many Hadoop platforms simultaneously. In Generic File Explorer user can perform Hadoop operations without typing a single command.

2. LITERATURE SURVEY

While surveying about different Hadoop File Manager we found that every Hadoop platform developed their own file manager or file explorer to overcome the command typing problem for their own platform. But it is also a disadvantage for the user to use different file manager for different platforms. To use any of these platforms we have to install and configure their own file manager and other supporting tools with it. It means if user using one platform and performing some operation on that due to some reason if user want to move from one to other it is too much complicated to move from one to another. To make this task easy means moving from one to other Hadoop platform and perform all the operation and manipulation smoothly we are creating this Generic file Explorer for different Hadoop platform.

Platforms and their file systems:

I. Hortonworks:
HDP 2.3 on Hortonworks Sandbox
- Release date: July 2015

To quickly browse, upload and download files to and from the Hortonworks sandbox from within Windows using HDFS Explorer. HDFS Explorer is a Free Windows Explorer based GUI file manager for the Hadoop Distributed File System (HDFS).

Once HDFS Explorer is connected to the Hortonworks Sandbox it can be used to rapidly upload files and data and download query results using the familiar Windows Explorer-like GUI.

HDFS Explorer supports common Windows Explorer functionality including:
- Copying, moving, renaming and deleting files
- Drag and drop
- Fast navigation of folder structures
- Support for multiple windows
- Bookmarked locations

II. Cloudera:
Cloudera Manager 5.1.6
- Release date: September 2015
Hue

The File Browser application lets you browse and manipulate files and directories in the Hadoop Distributed File System (HDFS) while using Hue. With File Browser, you can:

Create files and directories, upload and download files, upload zip archives, and rename, move, and delete files and directories. You can also change files or directory's owner, group, and permissions. See Working with Files and Directories.
Search for files, directories, owners, and groups. See Searching for Files and Directories.
View and edit files as text or binary. See Viewing and Editing Files.

III. MapR:
MapR Sandbox
MapR Control System

File Browser is an application that you can use to access files and directories in the MapR File System (MapR-FS). Use File Browser in HUE to perform the following directory tasks:
- Create directories
- Upload, rename, transfer, and delete files and directories
- Change the owner, group, and permissions of a file or directory
- View and edit files as text or binary or download the files to your local system
- View MapReduce job input and output files

Table 1: Platforms and their file systems

Hadoop Platforms	File Manager	Released date
Hortonworks	HDP	July 2015
Cloudera	Hue	September 2015
MapR	MapR Control System	March 2015
Wrox	Web based	——

3. PROPOSED SYSTEM

Hadoop is totally command based system. It means to perform any operation on the HDFS we have to go for command based approach in that we have to fire the command on HDFS all the time through terminal. In Hadoop every task is performed by using commands only. Like starting the Hadoop, performing file transfer operations, configurations operations etc.
For example:

I. START-ALL.SH:

This command will start the Hadoop System. It will start the NameNode, DataNode, SecondryNameNode, Job Tracker, and TaskTracker.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

II. COPYFROMLOCAL<SOURCE><DESTINATION>:

This command will copy the file from the local system to HDFS.

III.COPYTOLOCAL<SOURCE><DESTINATION>:

This command will copy the file from HDFS to Local System.

Every time to perform all the tasks on terminal by using commands seems very tedious for Hadoop users and remembering so many commands will be very harder for the Novice users. To avoid these above things many Hadoop Vendors has developed their file browser for Hadoop. Many of these are the GUI based applications which are very helpful for Hadoop users the operations smoothly without using any command. Few of these Hadoop platform and their file systems are as follows:
Hortonworks:
HDP 2.3 on Hortonworks Sandbox
Release date: July 2015
Cloudera:
Cloudera Manager 5.1.6 with Hue
Release date: September 2015
MapR:
MapR Sandbox
MapR Control System

The above mentioned Hadoop vendors had developed these file systems for their own platforms only. This means these file systems will work with their own platform only. If user wants to use these file manager he has to download the file system from the respective vendor and also need to install all related software. In case if user wants to switch from one to other platform he needs to download and install the all the platform software to which he want switch. This is very difficult task for users to work on many file system simultaneously. Hence we have proposed a Generic File Explorer for Different Hadoop platforms. This will be a Generic File Explorer which will work with different Hadoop platforms. This will be a GUI based Application in that user can work on Hadoop without typing a single command.
If user want to perform operation on Hadoop it doesn't required complicated commands instead of this complicated commands he just click a single button to perform operation on Hadoop which is given on the GUI.

In Generic File Explorer if user wants to work than first he needs to select the Hadoop Platform which he would like to work. After selecting the platform the user start the Hadoop platform in pervious Hadoop system user start the Hadoop using command and the command is "START-ALL.SH" but in our system user just click the Start button which is present on our GUI. After starting the Hadoop user can perform all the operation and manipulation on Hadoop using simply button which is present on our GUI.

Our application will work with the below mentioned platforms.

- Hortonworks
- Cloudera
- Wrox
- MapR

3.1 System Design

Fig 2 illustrates the system architecture of the GFE. This system design contents 3 layers.

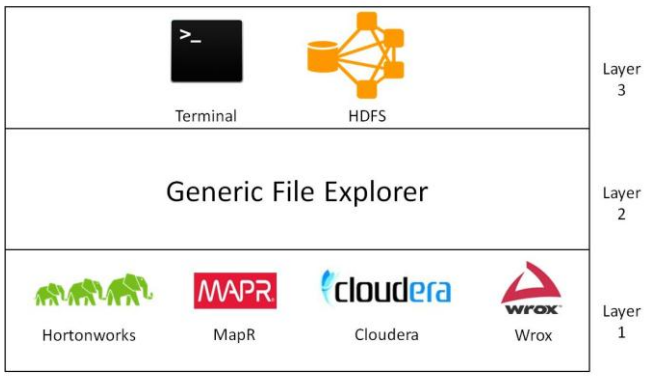

Fig 2: System Design

Each layer is defined as follows:

Layer1:
This is the bottom layer of Generic File Explorer in this layer all the Hadoop platforms are present.

Layer2:
This is the second layer of Generic File Explorer which contains the main application in that user can perform all the operation.

Layer3:
This is the third layer of Generic File Explorer in this layer terminal and HDFS is present.

3.2 Detailed Architecture of GFE

Figure [3] shows the architecture of our Generic File Explorer consist of three main modules:
A. Hadoop Platforms and Their XML files
B. Generic File Explorer (GFE)
C. Hadoop Distributed File System (HDFS)

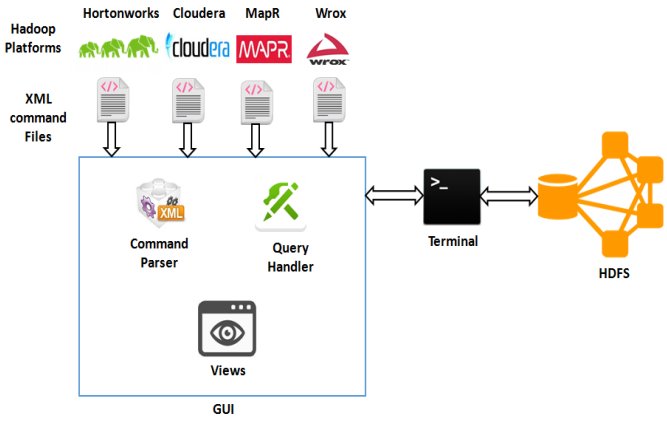

Fig 3: Detailed Architecture for GFE

A. Hadoop Platforms and Their XML files:
In this module we are taking the four Hadoop platforms which we are going to use in this project in this module we are store all the Hadoop command of the respected platforms into XML file. We create a XML file for each platform. These Commands will be parsed by using the XML_PARSER algorithm whenever user selects any command.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

B. Generic File Explorer (GFE):
In this module we design the GUI for Generic file system for different Hadoop platform in this we are going to parse the xml file which we are created in first module.
The GUI contains following tasks:

I. Command Parser
II. Query Handler
III. Views

I. Command Parser:
Whenever user wand to perform any operation then he will select the respective command on the main screen. After this The XML_PARSER program will parse the commands from the XML files and handover them to the Query Handler.

II. Query Handler:
It receives the command from the Command Parser and checks the query and its descriptions and the operand list. After checking the commands as per the command it prompts the user to select the further data which is required by the command.

III. Views:
This method is used to represent the results on the GUI. After the completion the operations on the HDFS the results are generated on the Terminal. Then it Fetches the results from Terminal and shows the results on the GUI.
C. Hadoop Distributed File System (HDFS):
This is the third module of our project. It contains the Terminal and HDFS. When the user selects any command it will go to HDFS through the Terminal. Then the operation will perform on the HDFS and again result of the operation is Display on GUI which fetched through Terminal.

3.3 Methodologies
The Fig4 shows the exact working of our Generic File Explorer. The Diagram shows the step by step execution of Application.

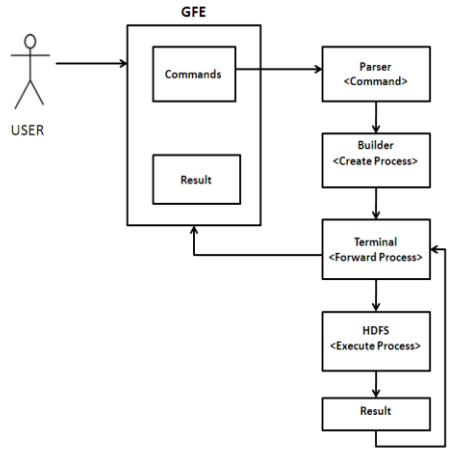

Fig 4: Methodology of GFE

In this user first select the command for performing the operations on the HDFS. After selecting, the selected command is sent to the XML_PARSER which parses the actual command from XML files respected to the platforms. After this the parsed command is forwarded to the Process Builder. The Process Builder will create the process which sends the command to the HDFS through the Terminal. Now as per the command the operation is performed on the HDFS. After performing the operation the HDFS will send he result to the terminal. After this the output of the terminal is displayed on the GUI.

4. RESULTS

For developing the Generic File Explorer we used 16GB RAM, I3 processor and 500GB HDD for performing operation efficiently with all the platforms.

Whereas following options are available for developing our proposed system GFE. Out of all there any one can be selected as per availability or respected to which platform user wants to work with

I. Platform Hortonworks
RAM 16GB
Processor i3
HDD 500G

II. Platform Cloudera
RAM 8GB
Processor i3
HDD 250GB

III. Platform Wrox
RAM 4GB
Processor C2D
HDD 250GB

IV. Platform MapR
RAM 8GB
Processor C2D
HDD 250GB

Till now we have worked on the designing part of our application below is the screenshots of our application.

Fig 5: Startup Frame

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

The Fig 5 shows the Startup Frame of our application in which user can start the Hadoop. To start the Hadoop user need to select the Bin folder that contains all the Hadoop shell files, this files starts the Hadoop. Here user needs to give the Password after that user can start the Hadoop. Below check boxes shows the running Daemon processes by which we can understand is the Hadoop is started or not. After the Successful Initialization User can proceed to the Main Frame By clicking the proceed button.

The Fig 6 shows the Main Frame of our application in this user select the platform on which user want to work. After selecting platform user can select the operation like File System Operation or Admin Operation. After this user can select the Command related to selected Operation. After the selection of Command user prompted to select the required files and operands related to Command. After clicking the execute button the command is executed and operation is performed on HDFS and result of this operation is displayed on the Main Frame.

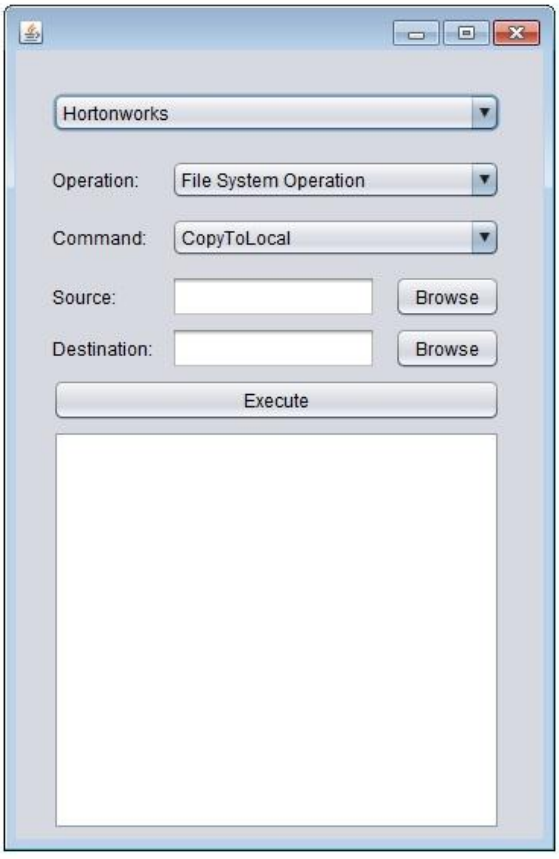

Fig 6: Main Frame

5. CONCLUSION

Our proposed GFE provides solution which is very useful to Hadoop users to perform operations with the options of multiple platforms without any need of downloading many supporting platform services. Still development phase of GFE is going on and we will need2 more months to complete the work.GFE is going to be one of the best solution for users of Hadoop in various ways. GFE is going to overcome the painstaking job of users while working on multiple Hadoop platforms.

6. REFERENCES

[1] Apache, “Hdfs,” http://apache.Hadoop.org/hdfs/.
[2] Hadoop, “HDFS Architecture,” September 2012, https://Hadoop.apache.org/docs/r0.20.2/hdfs design.html.
[3] A.Foundation,“Yarn,”https://Hadoop.apache.org/docs/r0.23.0/Hadoopyarn/Hadoop-yarn-site/YARN.html.
[4] http://hortonworks.com/Hadoop-tutorial/use-hdfs- explorer-manage-files-hortonworks-sandbox/
[5] http://hortonworks.com/hdp/
[6] http://www.cloudera.com/content/www/en-us/documentation/archive/cdh/4-x/4-2-1/Hue-2-User-Guide/hue24.html
[7] https://www.mapr.com/products/mapr-sandbox-Hadoop/tutorials/hue-tutorial-file-browser-metastore-manager-beeswax
[8] https://Hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[9] https://Hadoop.apache.org/docs/r0.23.0/Hadoopyarn/Hadop-yarn-site/YARN.html.
[10] https://Hadoop.apache.org/docs/r2.6.0/Hadoop-project-dist/Hadoop-common/CommandsManual.html
[11] https://Hadoop.apache.org/docs/current/Hadoop-project-dist/Hadoop-common/FileSystemShell.html

(This work is licensed under a Creative Commons Attribution 4.0 International License.)