Data Analysis Using Spark

An Industry Oriented Mini Project report submitted
In partial fulfillment of requirements
For the award of degree of

Bachelor of Technology
In
Information Technology

By

K. Vishnu Priya **(Reg No: 14131A1249)**

Under the esteemed guidance of

Dr. K. B. Madhuri

(Head of the Department)

Department of Information Technology

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING (AUTONOMOUS)

(Affiliated to JNTU-K, Kakinada)

VISAKHAPATNAM

2017- 2018

Gayatri Vidya Parishad College of Engineering(Autonomous)

Visakhapatnam

CERTIFICATE

This report on **"*Data Analysis Using Spark*"** is a bonafide record
of the mini project work submitted

By

K.Vishnu Priya **(Reg No:14131A1249)**

in their VII semester in partial fulfillment of the requirements for the Award of Degree of

Bachelor of Technology

In

Information Technology

During the academic year 2017-2018

Dr.K.B.Madhuri
Project Guide

Dr.K.B.Madhuri
Head of the Department
Department of Information Technology

External Examiner

DECLARATION

We hereby declare that this industry oriented mini project entitled **"DATA ANALYSIS USING SPARK"** is a bonafide work done by us and submitted to **Department of Information Technology G.V.P college of engineering (autonomous) Visakhapatnam,** in partial fulfilment for the award of the degree of B.Tech is of our own and it is not submitted to any other university or has been published any time before.

By

K.Vishnu Priya(Reg No:14131A1249)

ACKNOWLEDGEMENT

We would like to take this opportunity to extent our hearty gratitude to our esteemed institute "**Gayatri Vidya Parishad College of Engineering (Autonomous)**"where we got the platform to fulfill our cherished desire.

We express our sincere thanks to **Prof. Dr. A.B.KOTESWARA RAO**, Principal of Gayatri Vidya Parishad College of Engineering (Autonomous), for his support and encouragement during the course of this project.

We express our deep sense of gratitude to **Prof. Dr. K. B. MADHURI**, Head of Department, Department of Information Technology, for her constant encouragement.

We also thank **Asst. Prof. D.NAGA TEJ**, project coordinator, Department of Information Technology, for guiding us throughout the project and helping us in completing the project efficiently.

We are obliged to **Prof. Dr. K. B. MADHURI**, Department of Information Technology, who has been our guide, whose valuable suggestions, guidance and comprehensive assistance helped us a lot in realizing the project.

We would like to thank all the members of teaching and non-teaching staff of Department of Information Technology, for all their support.

Lastly, we are grateful to all my friends, for their relentless support in augmenting the value of work, our family, for being considerate and appreciative throughout.

Project Members

K.Vishnu Priya(Reg No:14131A1249)

ABSTRACT

Apache Spark is an open source big data processing framework built around speed, ease of use, and sophisticated analytics. Spark has several advantages compared to other big data and MapReduce technologies like Hadoop and Storm.

Spark enables applications in Hadoop clusters to run up to 100 times faster in memory and 10 times faster even when running on disk. Spark takes MapReduce to the next level with less expensive shuffles in the data processing. With capabilities like in-memory data storage and near real-time processing, the performance can be several times faster than other big data technologies. We collect log data from Server using Flume and store them dynamically into HDFS and process this data by using SPARK. The processed data will be visualized using Tableau. Spark streaming also supports machine learning and graph processing algorithms, and languages like scala, java and python. For high availability in production, spark streaming uses Zookeeper and HDFS.

CONTENTS

1. Introduction

1.1 Project Scope 1

1.2 Existing System 2

1.3 Proposed System 2

1.4 Functionality of the project 3

2. SRS Document 4

2.1 Modules Description 4

2.1.1 About Cloudera 5

2.1.2 About Tableau 7

2.1.3 About Hive 9

2.2 System Requirement Specification 10

2.2.1 Introduction 10

2.2.2 Purpose 10

2.2.3 Functional Requirements 11

2.2.4 Non Functional Requirements 11

2.3 Hardware Requirements 12

2.4 Software Requirements 12

3. Design 13

3.1 System Design 13

3.2 System model 13

3.2.1 Introduction to UML 13

3.2.2 Class Diagram 14

3.2.3 Sequence Diagram 15

3.2.4 Activity Diagram 16

3.2.5 Use case Diagram 17

3.2.6 Data Flow Diagram 18

4. Implementation 19
4.1 Sample Code 19
4.2 Output Screen 23
5. Testing 35
5.1 Testing methodologies 43
5.2 Test Cases
6. Conclusion 44
7. Bibliography 45
7.1 Web References 45

1. Introduction

1.1 Project Study

Spark Streaming is developed as part of Apache Spark. Spark Streaming is used to analyze streaming data and batch data. Spark Streaming can read data from HDFS, Flume, Kafka, Twitter , process the data using Scala, Java or python and analyze the data based on the scenario.

Software should implement the Streaming Data Analysis for:

a. Continuous Log Data.

b. Error extraction.

c. Analyze the type of errors.

Steps in making the process:

By consider different types of logs and store in one host.

Create large amount of logfiles and process the useful information from these logs which is required for monitoring purposes.

Using Flume send these logs into another host where it needs to be processed.

The solution providing for streaming real time log data is to extract the error logs.

Provide a file which contains the keywords of error types for error Identification in the spark processing logic.

Processing logic should be written in spark-scala or spark-java and store in HDFS/Hbase for tracking purposes

Use Flume for sending the streaming data into another port.

Spark-streaming to receive the data from the port and check the logs which contains error information, extract those logs and store into HDFS or HBase.

On the Stored error data categorize the errors using Tableau Visualization.

1.2 Existing System:

The existing system that we are using currently requires a person who is well-known about the complete domain

In the existing system we have to process the entire data after collection of total data

Monitoring of data is big task

Problems in existing system:

The present existing system requires continuous monitoring of data

Also requires an assistant who know complete working of the software

Lot of time wasting for analyzing the result

1.3 Proposed Systems:

In-Memory processing is done that that reduces Time-consuming

Spark is better implementation for Map Reduce paradigm

Proposed system covers all tools in big data stream

Data can be easily visualized

1.4 Functionality of the project:

This project is used to analyze the large amount of log data from servers. Create large amount of log files and process the useful information from these logs which is required for monitoring purposes. Using Flume send these logs into another host where it needs to be processed. The solution providing for streaming real time log data is to extract the error logs. Provide a file which contains the keywords of error types for error Identification in the spark processing logic. Processing logic should be written in spark-scala or spark-java and store in HDFS/Hbase for tracking purposes. Use Flume for sending the streaming data into another port. Spark-streaming to receive the data from the port and check the log which contains error information, extract those logs and store into HDFS or HBase.On the Stored error data categorize the errors using Tableau Visualization.

2. SRS Document

2.1 Module Description:

We implement totally in scala

Tableau: Analyzing the results.

Cloudera: Software for streaming.

The system after careful analysis has been identified to present with the following modules.

1. **Input Module:**

Input module contains different log data to process and analyze them.

2. . **Processing Module:**

We perform processing by using scala. we collect log data from various servers and get the data using flume .we store the streaming data in the hdfs. we process using spark in scala.

2.1.1 ABOUT CLOUDERA:

Cloudera is revolutionizing enterprise data management by offering the first unified Platform for Big Data: The Enterprise Data Hub. Cloudera offers enterprises one place to store, process, and analyze all their data, empowering them to extend the value of existing investments while enabling fundamental new ways to derive value from their data.

Founded in 2008, Cloudera was the first, and is currently, the leading provider and supporter of Apache Hadoop for the enterprise. Cloudera also offers software for business critical data challenges including storage, access, management, analysis, security, and search.

Customer success is Cloudera's highest priority. We've enabled long-term, successful deployments for hundreds of customers, with petabytes of data collectively under management, across diverse industries.

CDH-: CDH (Cloudera Distribution Hadoop) is open-source Apache Hadoop distribution provided by Cloudera.Inc which is a Palo Alto-based American enterprise software company. CDH (Cloudera's Distribution Including Apache Hadoop) is the most complete, tested, and widely deployed distribution of Apache Hadoop. CDH is 100% open source and is the only Hadoop solution to offer batch processing, interactive SQL and interactive search as well as enterprise-grade continuous availability. More enterprises have downloaded CDH than all other distributions combined.

CLOUDERA IMPALA: Impala is an open source massively parallel processing query engine on top of clustered systems like Apache **Hadoop**. It was created based on Google's Dremel paper. It is an interactive SQL like query engine that runs on top of **Hadoop** Distributed File System (HDFS).

CLOUDERA SEARCH: Cloudera Search provides near real-time (NRT) access to data stored in or ingested into Hadoop and HBase. Search provides near real-time indexing, batch indexing, Full-text exploration and navigated drill-down, as well as a simple, full-text interface that requires no SQL or programming skills. Search is fully integrated in the data-processing platform and uses the flexible, scalable, and robust storage system included with CDH. This eliminates the need to move large data sets across infrastructures to perform business tasks. +Cloudera Search incorporates Apache Solr, which includes Apache Lucene, SolrCloud, Apache Tika, and Solr Cell. Cloudera Search is included with CDH 5.

CLOUDERA MANAGER:-Cloudera Manager is the industry's first end-to-end management application for Apache Hadoop. By delivering granular visibility into and control over the every part of the Hadoop cluster, Cloudera Manager empowers enterprise operators to improve cluster performance, enhance quality of service, increase compliance and reduce administrative costs. Cloudera Manager provides many useful features for monitoring the health and performance of the components of your cluster (hosts, service daemons) as well as the performance and resource demands of the user jobs running on your cluster.

2.1.2 ABOUT TABLEAU:-

Data Visualization is an art of presenting the data in a manner that even a non-analyst can understand it. A perfect blend of aesthetic elements like colors, dimensions, labels can create visual masterpieces, hence revealing surprising business insights which in turn helps businesses to make informed decisions.

Data Visualization is an inevitable aspect of business analytics. As more and more sources of data are getting discovered, business managers at all levels embrace data visualization softwares, that allow them to analyze trends visually and take quick decisions. Currently, the most popular tools for visualizations / data discovery are Qlikview and Tableau.

Tableau is one of the fastest evolving Business Intelligence (BI) and data visualization tool. It is very fast to deploy, easy to learn and very intuitive to use for a customer. Here is a learning path to all those people who are new to Tableau. This path will help you to learn Tableau in a structured approach. Beginners are recommended to follow this path religiously. If you already have some background, or don't need all the components.

Speed of Analysis − As it does not require high level of programming expertise, any user with access to data can start using it to derive value from the data.

Self-Reliant − Tableau does not need a complex software setup. The desktop version which is used by most users is easily installed and contains all the features needed to start and complete data analysis.

Visual Discovery − The user explores and analyzes the data by using visual tools like colors, trend lines, charts, and graphs. There is very little script to be written as nearly everything is done by drag and drop.

Blend Diverse Data Sets − Tableau allows you to blend different relational, semi structured and raw data sources in real time, without expensive up-front integration costs. The users don't need to know the details of how data is stored.

Architecture Agnostic − Tableau works in all kinds of devices where data flows. Hence, the user need not worry about specific hardware or software requirements to use Tableau.

Real-Time Collaboration − Tableau can filter, sort, and discuss data on the fly and embed a live dashboard in portals like SharePoint site or Salesforce. You can save your view of data and allow colleagues to subscribe to your interactive dashboards so they see the very latest data just by refreshing their web browser.

Centralized Data − Tableau server provides a centralized location to manage all of the organization's published data sources. You can delete, change permissions, add tags, and manage schedules in one convenient location. It's easy to schedule extract refreshes and manage them in the data server. Administrators can centrally define a schedule for extracts on the server for both incremental and full refreshes.

2.1.3 ABOUT HIVE:

What is Hive:

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Hive is not :

A relational database

A design for Online Transaction Processing (OLTP)

A language for real-time queries and row-level updates

Features of Hive:

It stores schema in a database and processed data into HDFS.

It is designed for OLAP.

It provides SQL type language for querying called HiveQL or HQL.

It is familiar, fast, scalable, and extensible.

HIVEQL:

While based on SQL, HiveQL does not strictly follow the full SQL-92 standard. HiveQL offers extensions not in SQL, including *multitable inserts* and *create table as select*, but only offers basic support for indexes. HiveQL lacked support for transactions and materialized views, and only limited sub query support.[2] Support for insert, update, and delete with full ACID functionality was made available with release 0.14

Internally, a compiler translates HiveQL statements into a directed acyclic graph of MapReduce Tez, or Spark jobs, which are submitted to Hadoop for execution

2.2 SRS DOCUMENT:

2.2.1 Introduction

Intended Audience and Reading Suggestions

The document is prepared keeping in view of the academic constructs of my Bachelor's Degree/Master's Degree from university as partial fulfillment of my academic purpose the document specifies the general procedure that has been followed by me, while the system was studied and developed. The general document was provided by industry as reference guide to understand my responsibilities in developing the system, with respect to the requirements that have been pin pointed to get the exact structure of the system as stated by the actual client.

The system as stated by my project leader the actual standards of the specifications were desired by conducting a serious of interviews and questionnaires, the collected information was organized to form the specification document and then was modulated to suite the standards of the system as intended.

2.2.2 Purpose:

The overall documents for this project use the recognized modeling standards at the software industries level.

The physical dispense, which state the overall data search for the relational key whereas transaction is implemented on the wear entities.

Unified language modeling concepts to give a generalized blue print for the overall system

The standards of flowcharts at the required state that are the functionality of the operations need more concentration

2.2.3FUNCTIONAL REQUIREMENTS:

A functional requirements defines a function of a system or its components. A function described as a set of inputs, behavior and outputs.

Fully constructed network with multiple channels

Packet length

Weights to the every communication link

Probability

Number of channels

In this algorithm we are providing the active probability for every channel in the network, packet length and number of channels to calculate the smallest transmission delay to the every communication link in every channel and assign that delay as modified weights to that link.

2.2.4 NON-FUNCTIONAL REQUIREMENTS:

A non-functional requirements is a requirement that specify criteria that can be used to judge the operation of a system.

Route to Destination:

The routing architecture will provide for the routing of datagram from a single source to one or more destinations in a timely manner. The larger goal is to provide datagram delivery to an identifiable destination, one which is not necessarily immediately reachable by the source. In particular, routing is to address the needs of a single source requiring datagram delivery to one or more destinations.

2.3 HARDWARE REQUIREMENTS:

System **:** Pentium-III(or)Higher

Hard Disk **:** 10GB

Ram **:** 8GB (or)Higher

Cache **:** 512MB

2.4 SOFTWARE REQUIREMENTS:

Operating System **:** Windows 7/8/8.1/10

Coding Language **:**scala

Software **:** Cloudera

3.DESIGN

3.1 System Design:

This system helps us to analyze log data from different systems. Collecting the Streaming log data from servers and send them into Hadoop Distributed FileSystem.Data will be collected and analyzed using spark-scala. This data will be stored into HDFS and can be visualized into pictorial representation. To increase the performance of the server and capability of the server we analyze server log data.

3.2 System Model

3.2.1 Introduction to UML:

The Unified Modelling Language (UML) provides a standard format via construction of a model and using object oriented paradigm for describing software systems as well as non- software systems, business processes for the enterprise's problem areas and corporate infrastructure.

The model abstracts the essentials details of the underlying problems and provides a simplified view of the problem so as to make easy for the solution architect to work towards building the solution.

In context of the software development, the importance of UML can be comprehended using analogy of a construction process. Normally builders use the designs and maps to construct buildings. The services of a civil architect are needed to create designs, maps which act as reference point for the builder. The communication between architect and builder becomes critical according to the degree of complexity in the design of the building. Blueprints or Architectural designs are the standard graphical language that both architects and builders must understand for an effective communication.

Software development is a similar process in many ways. UML has emerged as the software blueprint methodology for the business and system analysts, designers, programmers and everyone involved in creating and deploying the software systems in an enterprise. The UML provides for everyone involved in software development process a common vocabulary to communicate about software design.

UML DIAGRAMS:-

We prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system

UML offers a way to visualize a system's architectural blueprints in a diagram (see image), including elements such as:

any activities

individual components of the system and how they can interact with other software components

how the system will run;

how entities interact with others (components and interfaces);

external user interface

Although originally intended for object-oriented design documentation, UML has been extended to a larger set of design documentation (as listed above), and been found useful in many contexts.

3.2.2 CLASS DIAGRAM

Class Diagram: The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

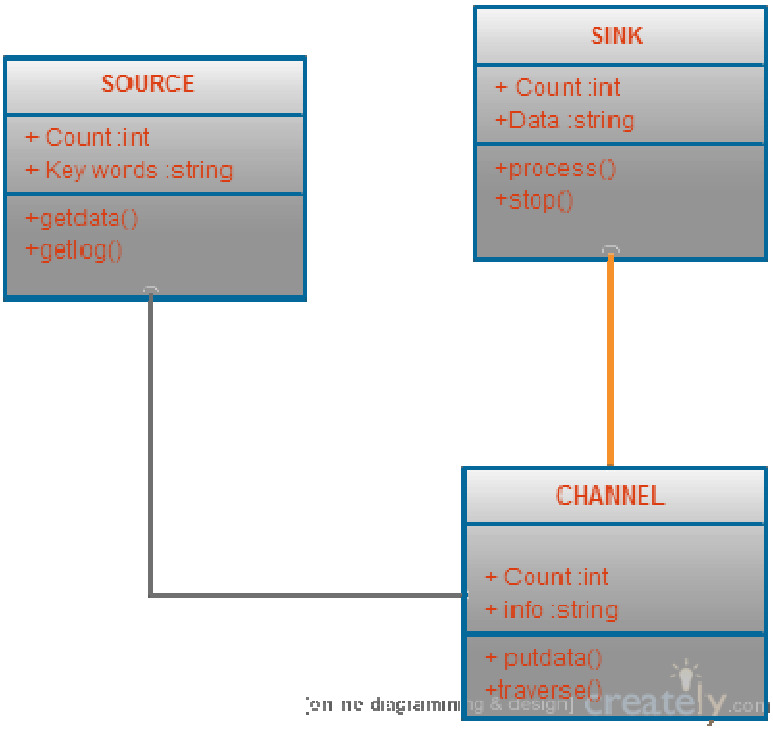


Figure: 3.1

3.2.3 SEQUENCE DIAGRAM

Sequence diagram: An interaction diagram, a subset of behavior diagrams, emphasizes the flow of control and data among the things in the system being modelled.

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

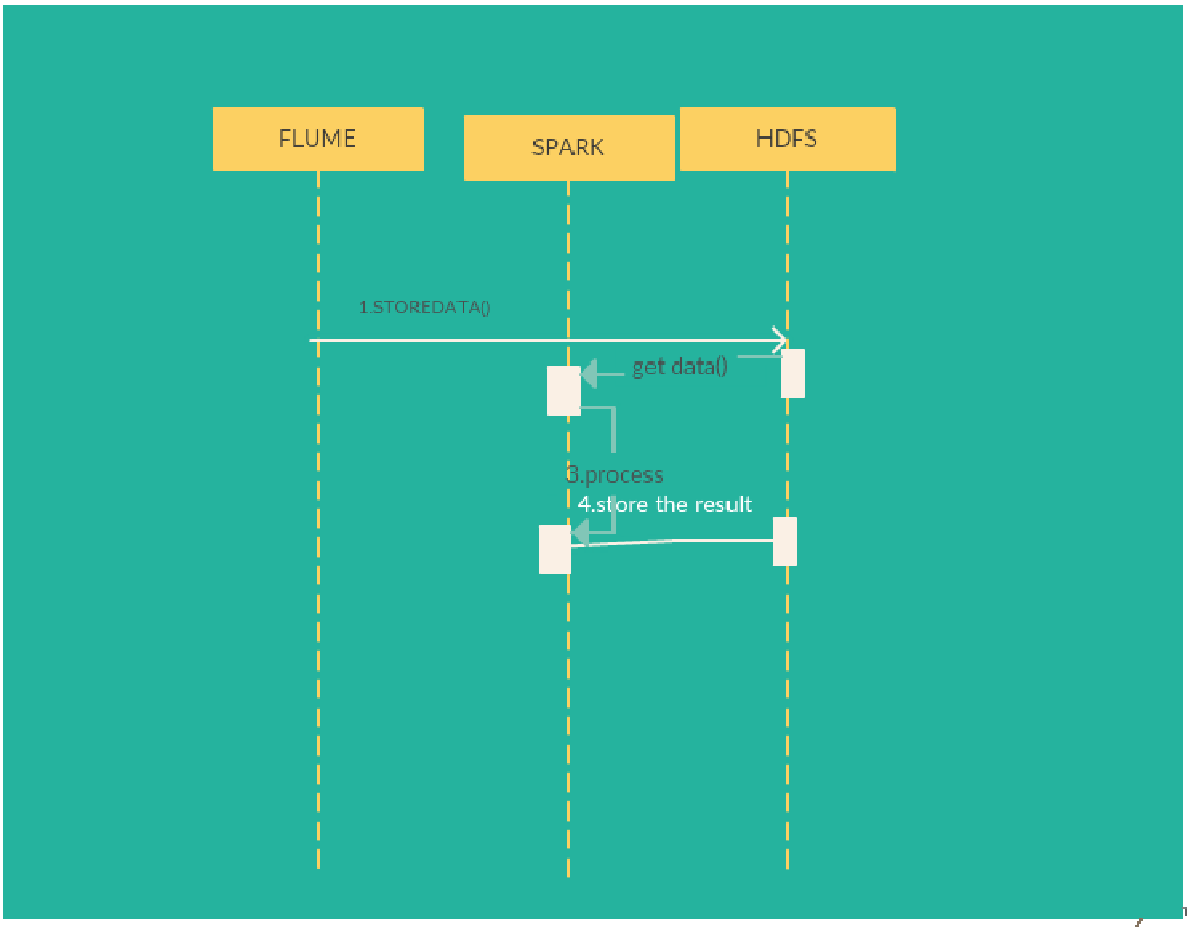


Figure: 3.2

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

3.2.4 USECASE DIAGRAM

UML Use Case Diagrams. Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

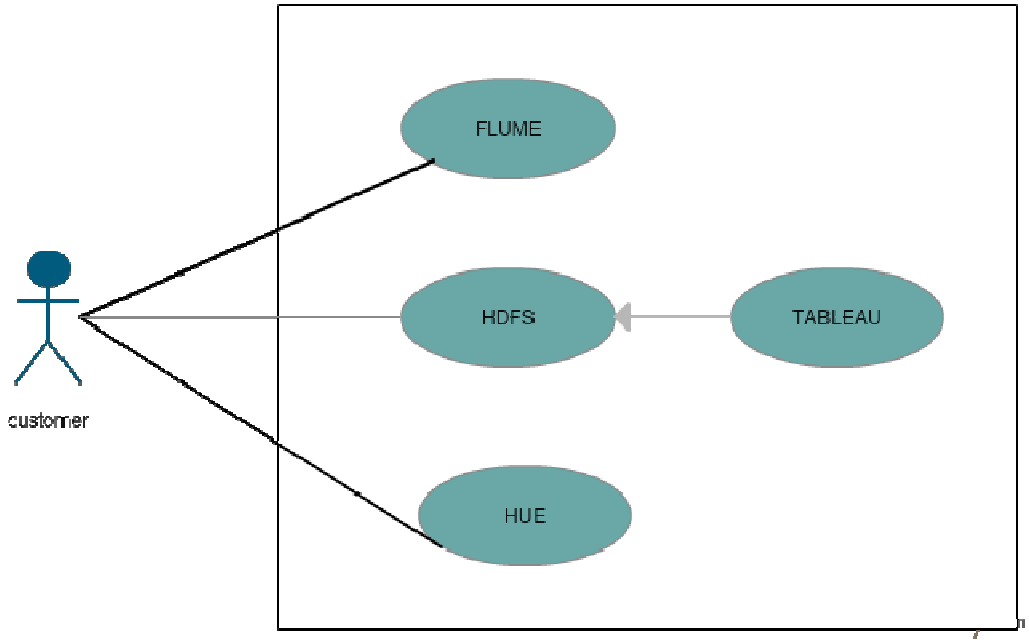


Figure: 3.3

3.2.5 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

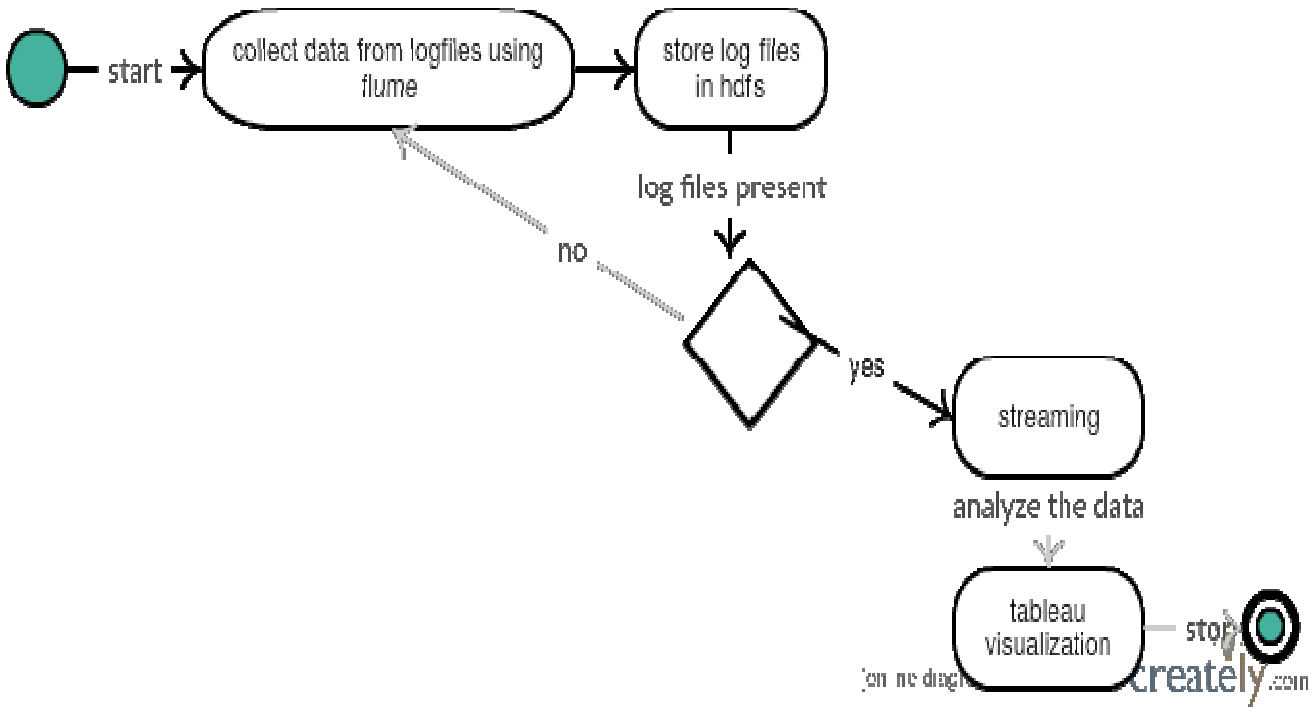


Figure: 3.4

3.2.6 FLOW DIAGRAM

Architecture &flow:

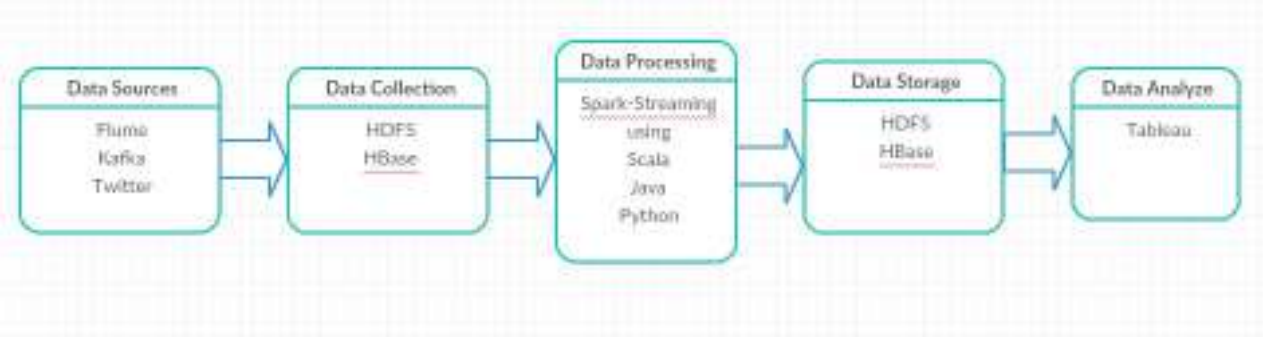


Figure: 3.5

Expected Results: The line which contains errors should store either in Hbase or HDFS itself ,later this data is used to generate error reports using Tableau .

4.IMPLEMENTATION

4.1 Samplecode:

Flume.conf:

```
agent.sources = src

agent.channels = memoryChannel

agent.sinks = mycluster



##Sources ################################################################

agent.sources.src.type = exec

agent.sources.src.command = cat /var/log/hue/hue_install.log

##Sinks ##################################################################

agent.sinks.mycluster.type =hdfs

agent.sinks.mycluster.hdfs.path=hdfs://quickstart.cloudera:8020/user/cloudera/project

agent.sinks.mycluster.hdfs.filePrefix=log

agent.sinks.mycluster.hdfs.rollInterval=0

agent.sinks.mycluster.hdfs.rollCount=0

agent.sinks.mycluster.hdfs.rollSize=10000000
```

agent.sinks.mycluster.hdfs.batchSize=10000

agent.sinks.mycluster.hdfs.fileType=DataStream

agent.sinks.mycluster.hdfs.UsePerfix=.

##Channels ###

agent.channels.memoryChannel.type = memory

agent.channels.memoryChannel.capacity=1000 agent.channels.memoryChannel.transactionCapacity = 100

#bind the sink and source to channel#################

agent.sinks.mycluster.channel =memoryChannel

agent.sources.src.channels =memoryChannel

Spark.scala

```
//execute this program in spark-shell

importorg.apache.spark._

importorg.apache.spark.SparkContext._

object WordCount {

defmain(args: Array[String]) {

        var s=sc.textFile("sparkStreaming/access.log")

        valcounts=s.flatMap(line=>line.split(" ")).map(word=>(word,1)).reduceByKey(_+_)

        counts.saveAsTextFile("sparkStreaming/results11")

        }

}

/*val pairs=lines.map(lines=>{

        val tokens=lines.split("\\s+") if(tokens.length>4)(tokens(4),line)

        else("unknown",line)}).filter(tup=?!((tup._1)equals("unknown")))

        valKeyCount=pairs.map(pair=>._1,1)).reduceByKey(_+_)

        val random=new random()

        pairs.print()

        pairs.foreachRDD(rdd=>
```

```
        if(!rdd.isEmpty)rdd.saveAsTextFile("hdfs:quickstart.cloudera:8020/user/cloudera/project/"+ran
dom.nextInt())

                ) keyCount.print()

        keyCount.foreachRDD(rdd=>

        if(!rdd.isEmpty)rdd.saveAsTextFile(""hdfs:quickstart.cloudera:8020/user/cloudera/project/"+ra
ndom.nextInt())

                )

        */
```

HIVE CODE:

Table creation :

```
CREATE TABLE IF NOT EXISTS tableau ( info string, no String)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ‘,’
STORED AS TEXTFILE;
```

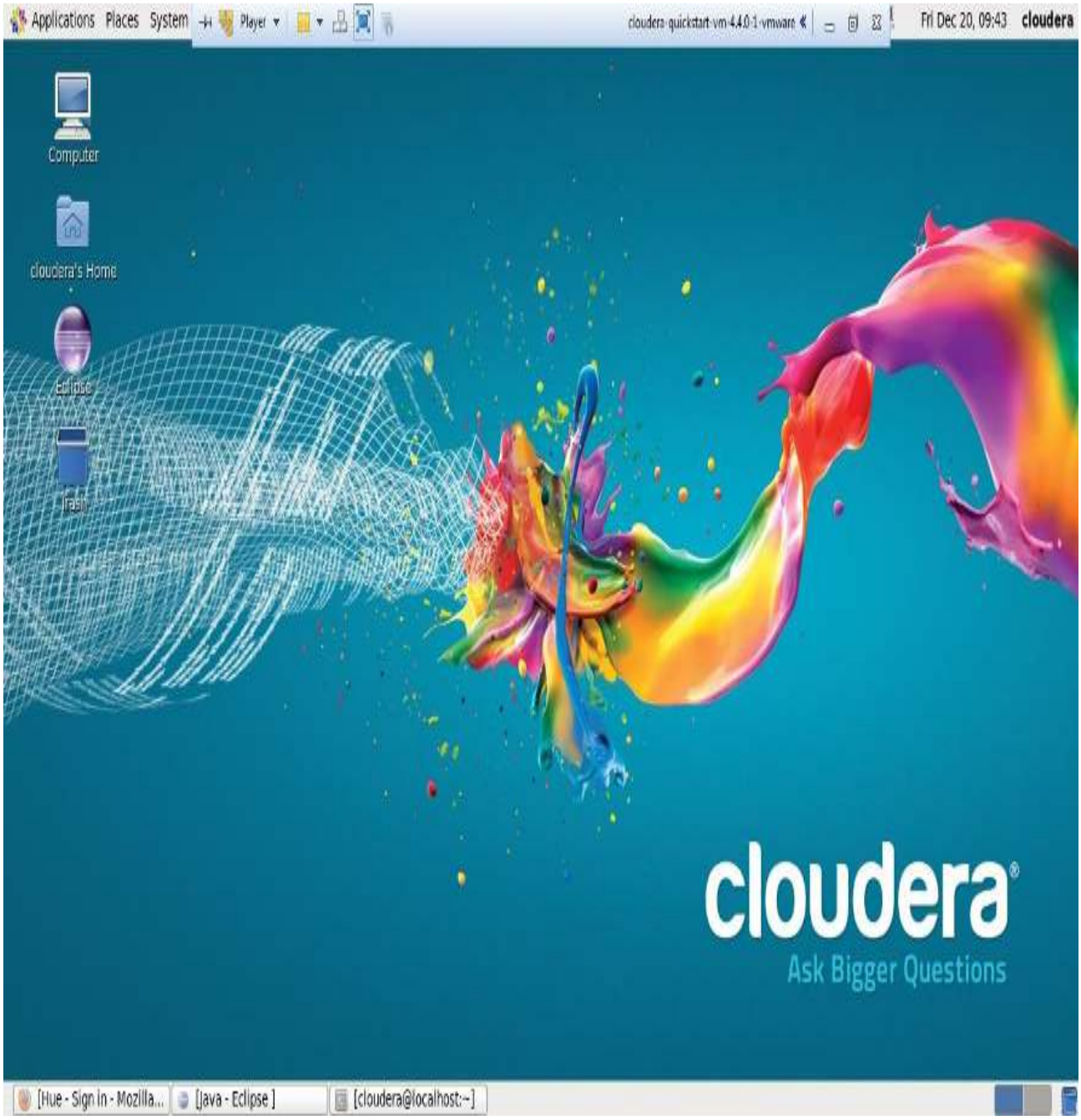
Load data into Hive

```
LOAD data inpath “Spark streaming/part-00000” from tableau ;
```

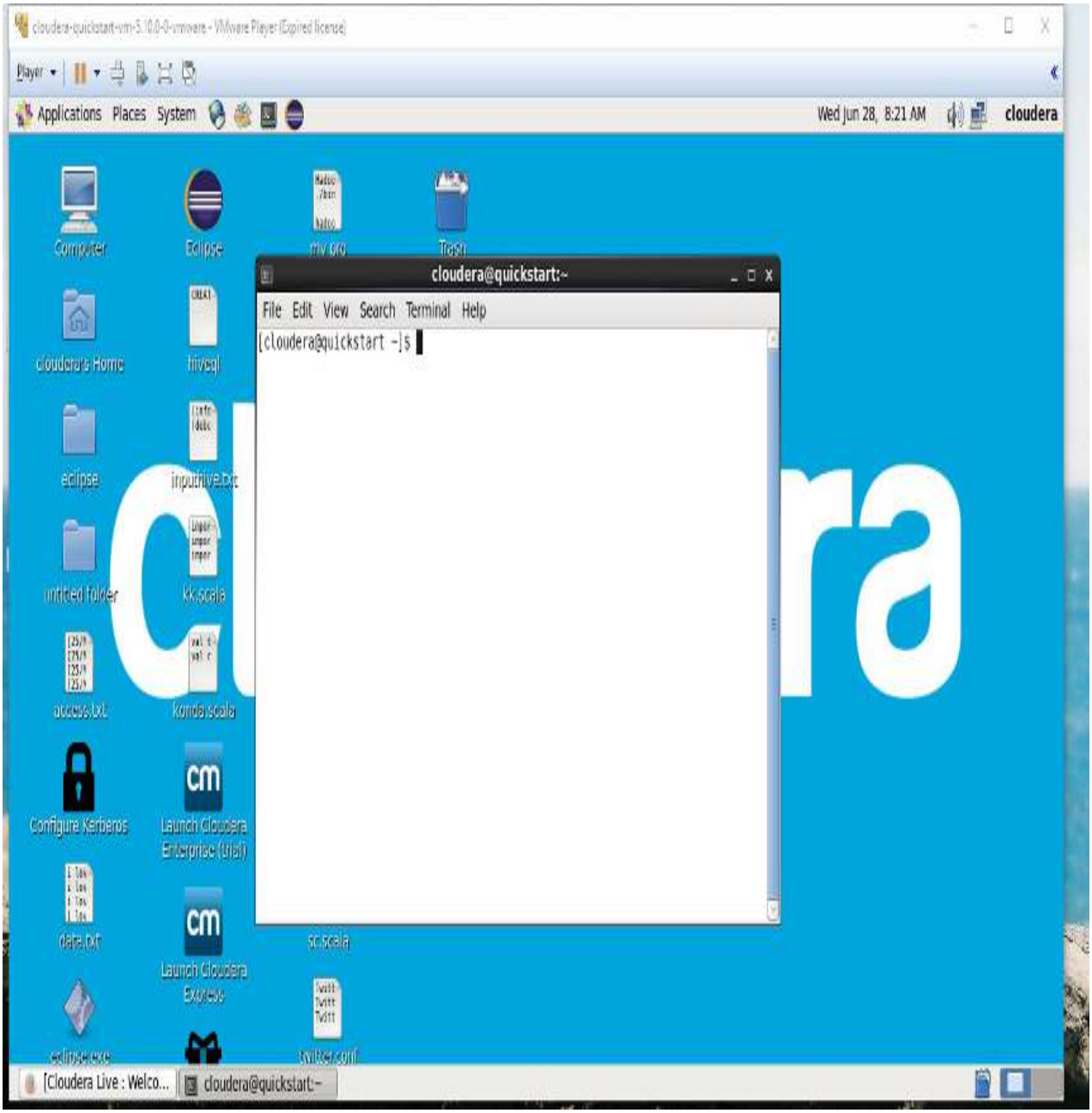
Retrieve data from hive

```
SELECT * from tableau;
```

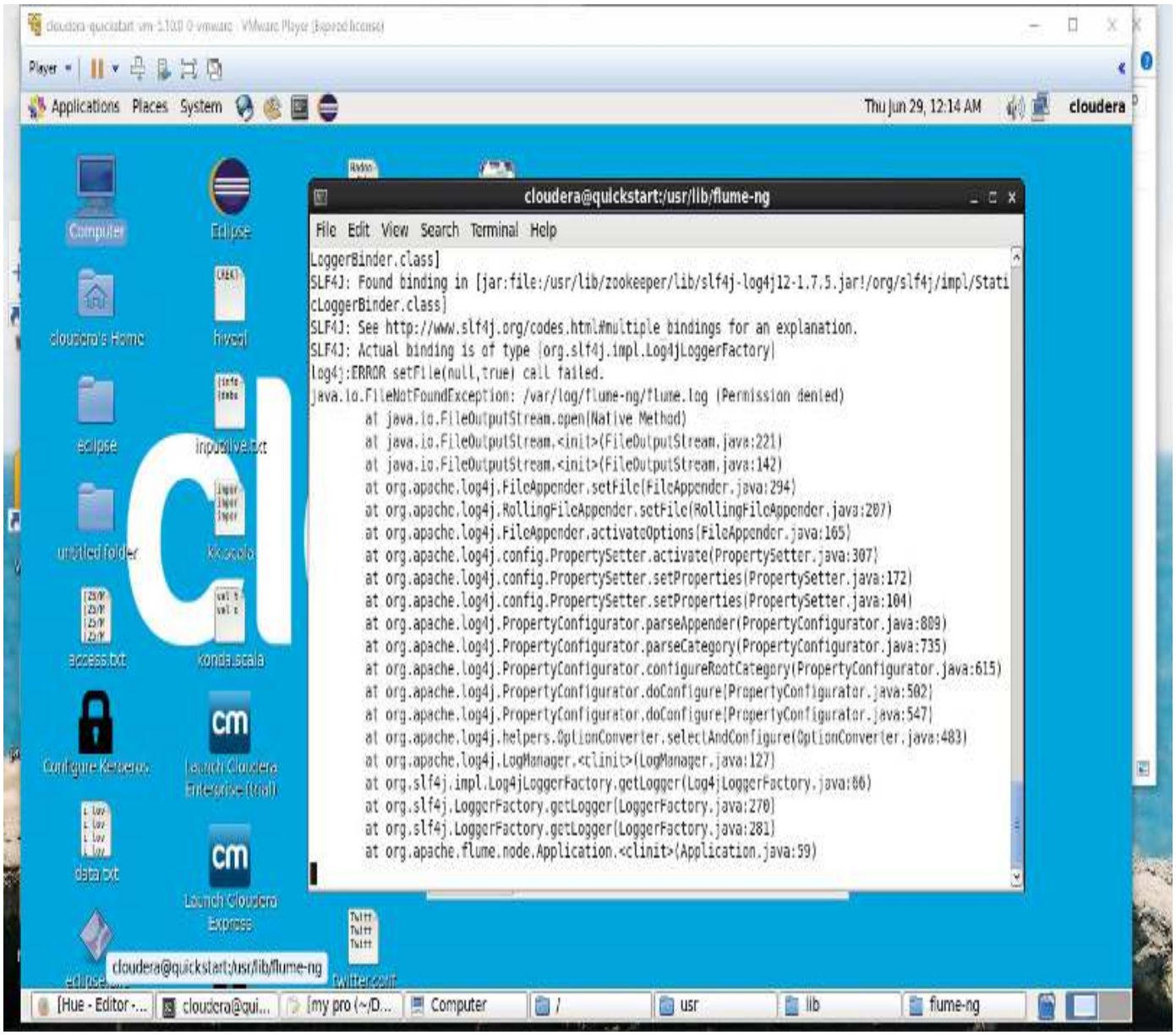
4.2 OUTPUT:



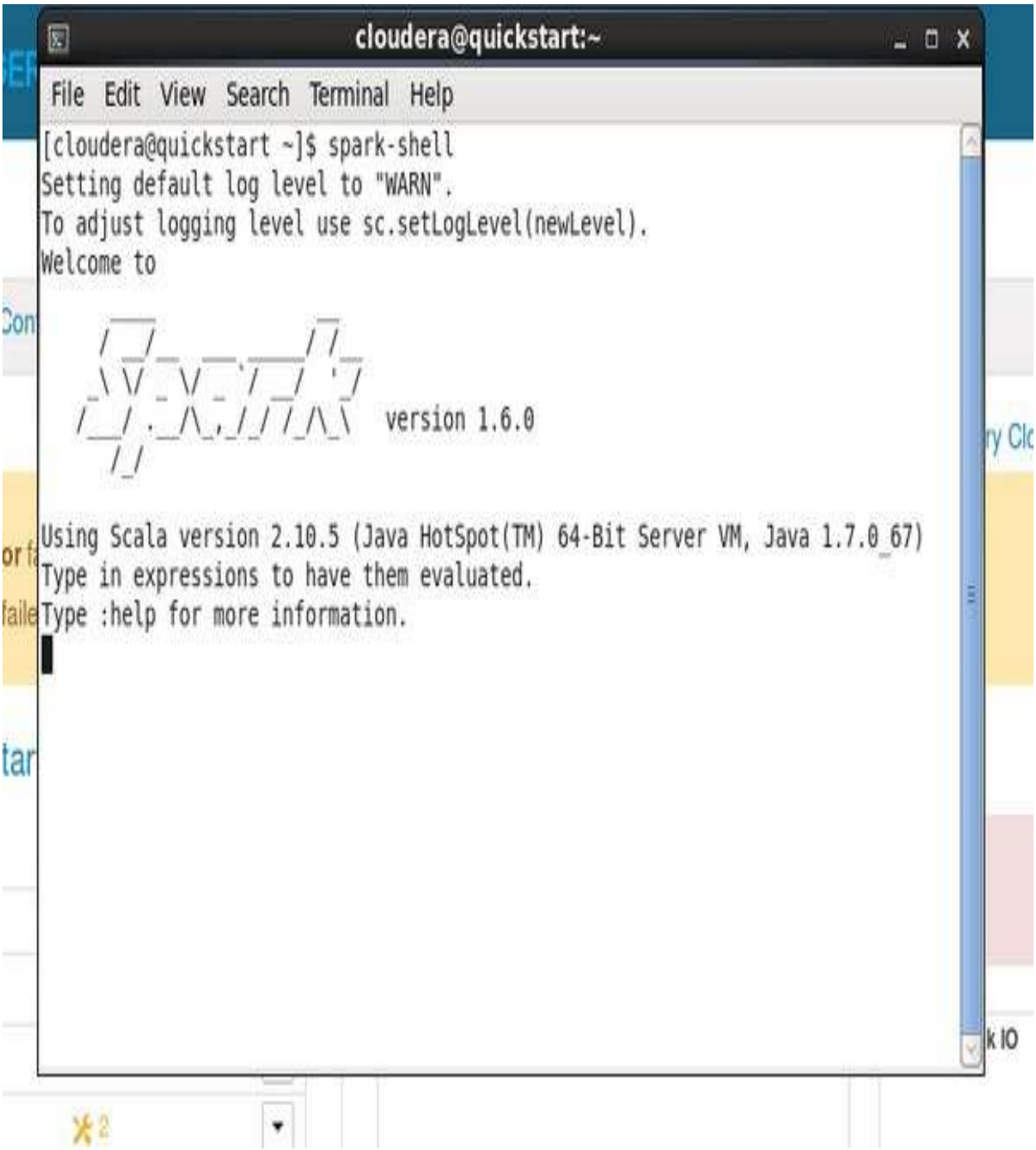
4.1Cloudera Desktop



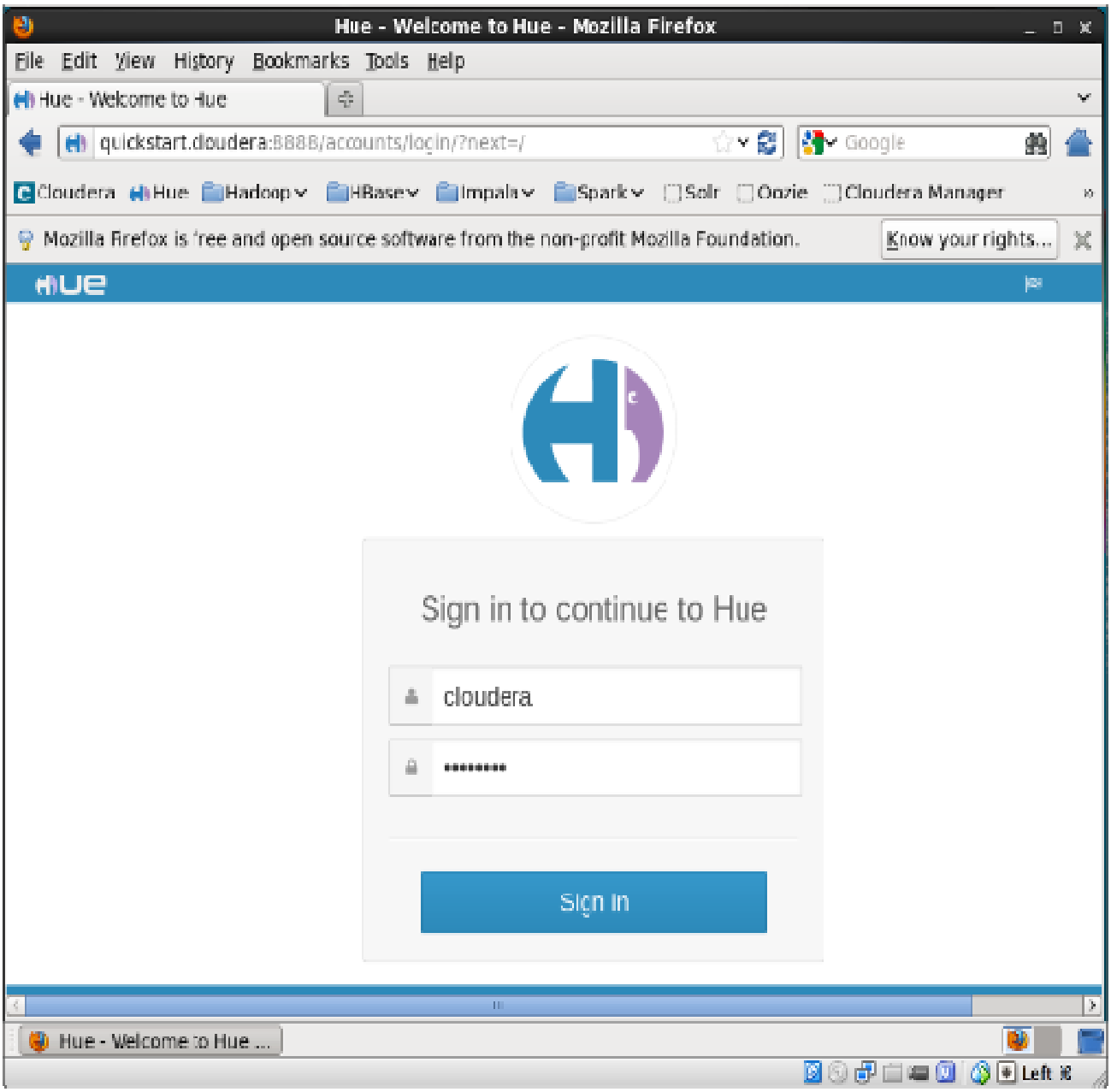
4.2Cloudera Terminal



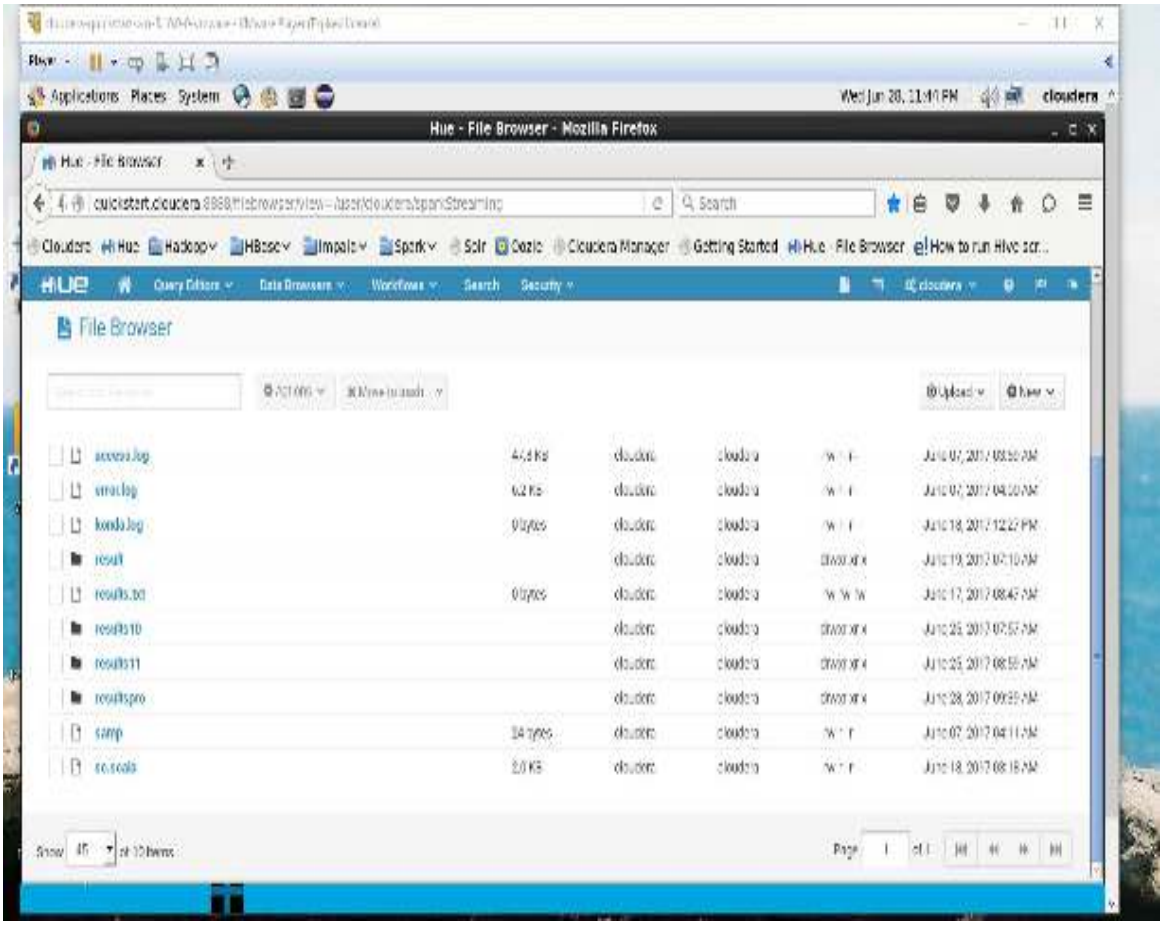
4.3Flume-ng Execution



4.4Spark Shell

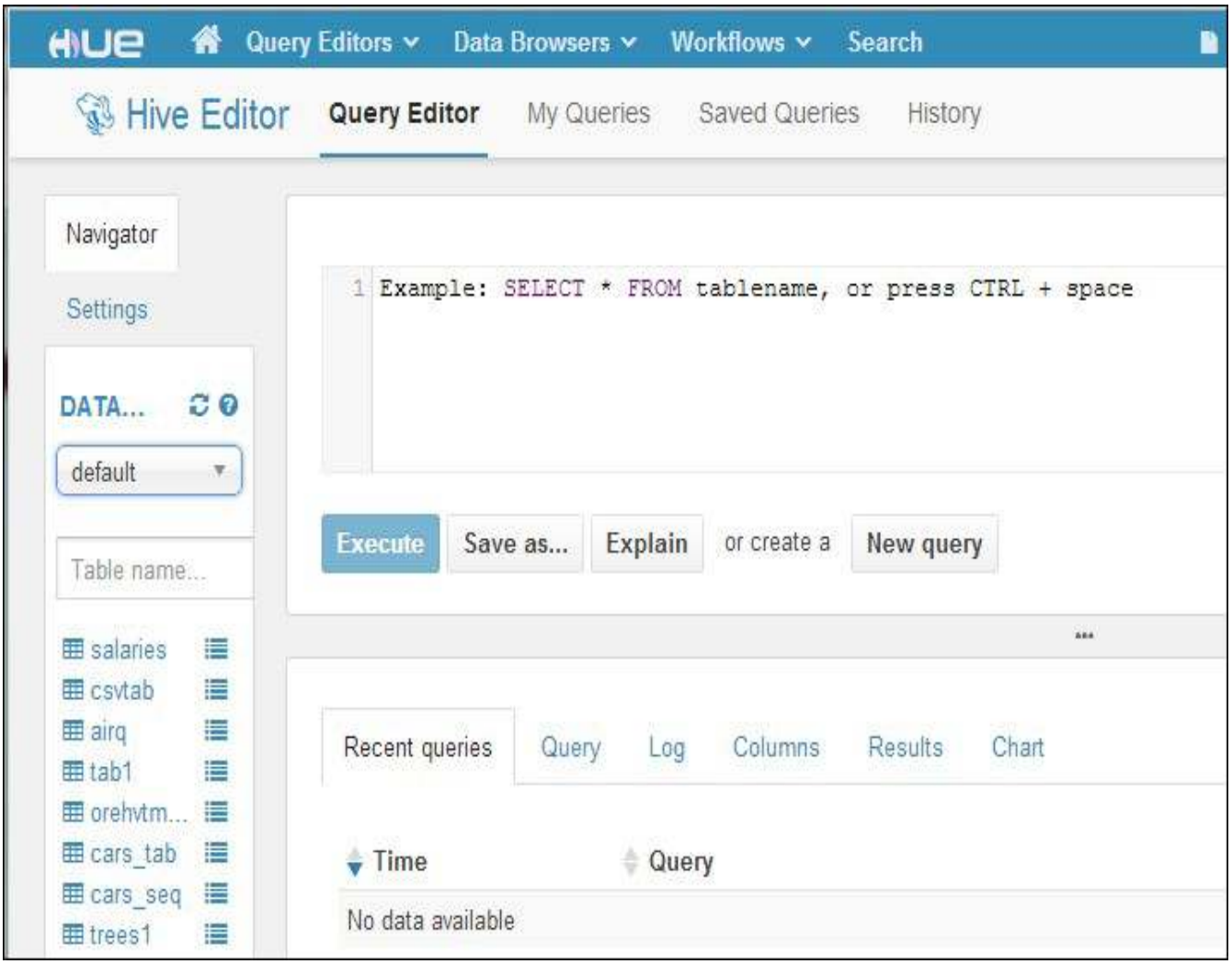


4.5Hue login page

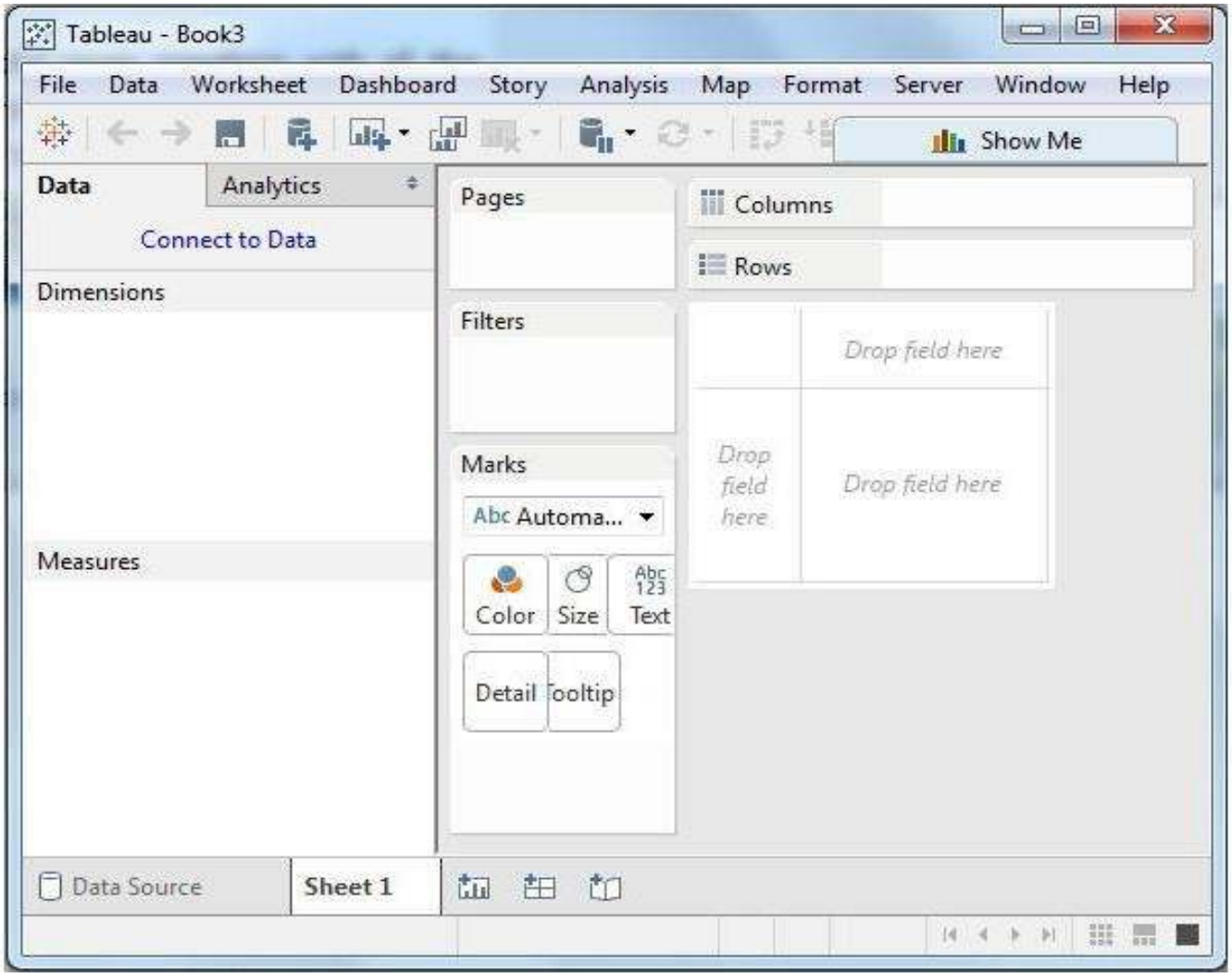


4.6 HDFS

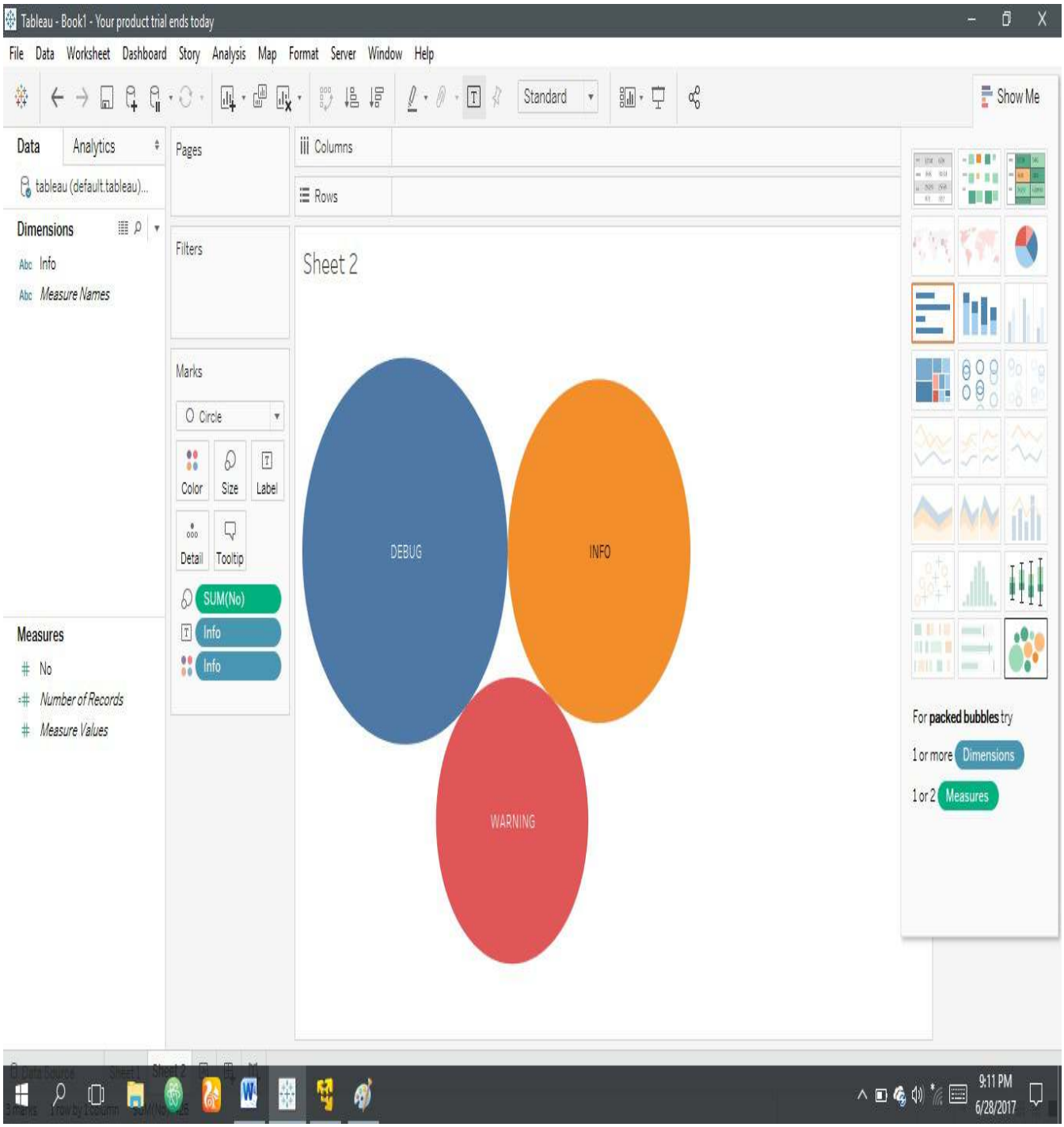
(Hadoop File System)



4.7 HIVE EDITOR



4.8 TABLEAU DESKTOP



4.9 Visualization

5. TESTING

5.1 Testing Methodologies

Testing is the process of finding differences between the expected behavior specified by system models and the observed behavior implemented system. From modeling point of view , testing is the attempt of falsification of the system with respect to the system models. The goal of testing is to design tests that exercise defects in the system and to reveal problems.

The process of executing a program with intent of finding errors is called testing. During testing , the program to be tested is executed with a set of test cases , and the output of the program for the test cases is evaluated to determine if the program is performing as expected . Testing forms the first step in determining the errors in the program. The success of testing in revealing errors in program depends critically on test cases.

Strategic Approach to Software Testing:

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirements analysis where the information domain , functions , behavior , performance , constraints and validation criteria for software are established. moving inward along the spiral , we come to design and finally to coding . To develop computer software we spiral in along streamlines that decreases the level of abstraction on each item.

A Strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing will progress by moving outward along the spiral to integration testing , where the focus on the

design and the concentration of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole are tested as a whole.

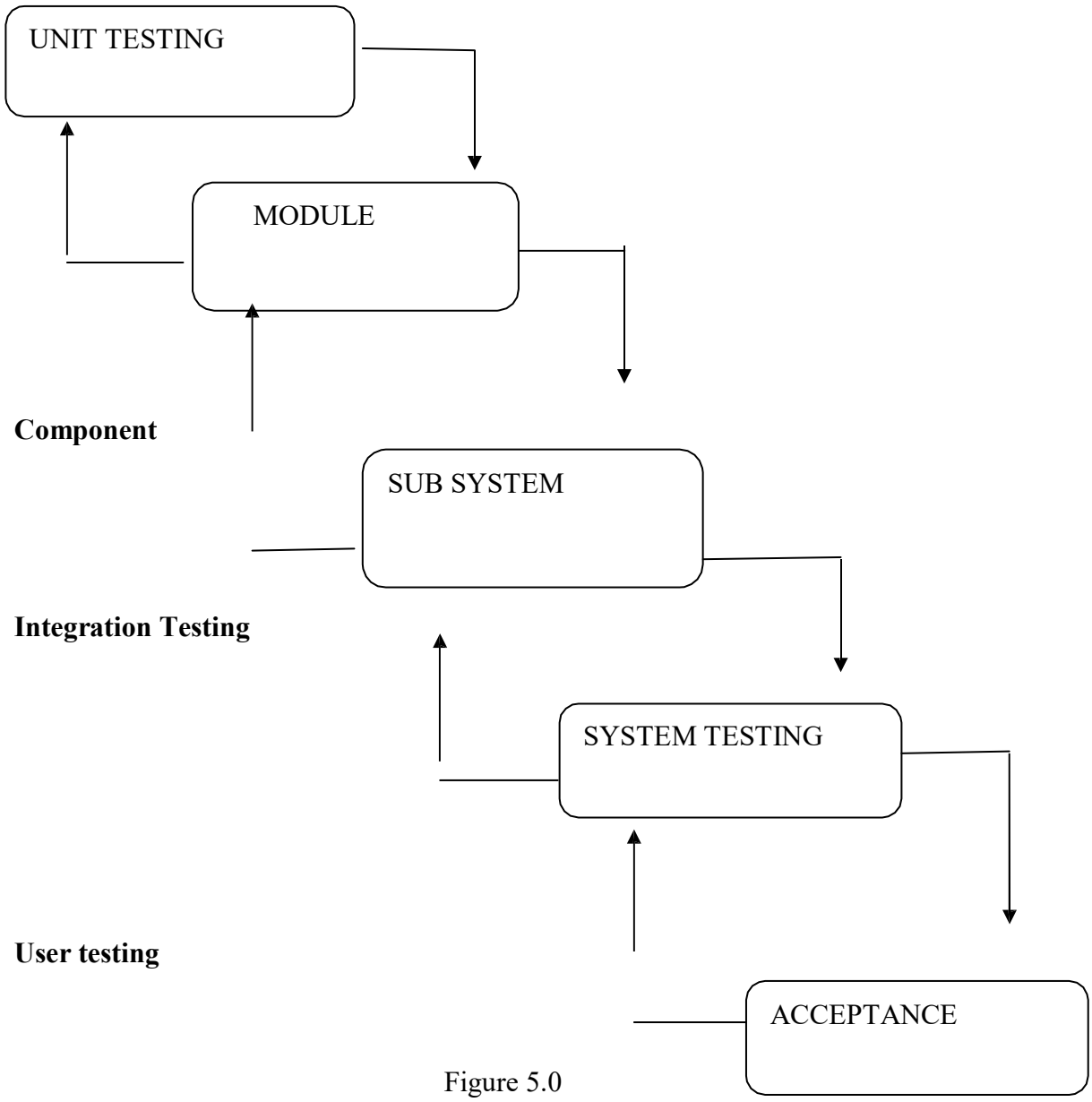


Figure 5.0

Different Levels of Testing

Client Needs	Acceptance Testing
Requirements	System Testing
Design	Integration Testing
Code	Unit Testing

Testing is the process of finding difference between the expected behavior specified by system models and the observed behavior of the implemented system.

Testing Activities:

Different levels of testing are used in the testing process , each level of testing aims to test different aspects of the system the basic levels are:

Unit testing

Integration testing

System testing

Acceptance testing

Unit Testing:

Unit testing focuses on the building blocks of the software system, that is, objects and sub system. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall tests activities, allowing us to focus on smaller units of the system. Second, unit testing makes it easier to pinpoint and correct faults given that few components are involved in this test. Third, Unit testing allows parallelism in the testing activities, that is each component can be tested independently of one another . Hence the goal is to test the internal logic of the module.

Integration Testing:

In the integration testing, many test modules are combined into sub systems , which are then tested . The goal here is to see if the modules can be integrated properly, the emphasis being on testing module interaction.

After structural testing and functional testing we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration, the test cases are generated and the results are tested.

System Testing:

In system testing the entire software is tested . The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

Acceptance Testing:

Acceptance testing is sometimes performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here focus on the external behavior of the system , the internal logic of the program is not emphasized . In acceptance testing the system is tested for various inputs.

Types of Testing:

1. Black box or functional testing
2. White box testing or structural testing

Black box testing:

This method is used when knowledge of the specified function that a product has been designed to perform is known . The concept of black box is used to represent a system whose inside workings are not available to inspection . In a black box the test item is a "Black" , since its logic is unknown , all that is known is what goes in and what comes out , or the input and output.

Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure
- Performance errors
- Initialization and termination errors

As shown in the following figure of Black box testing , we are not thinking of the internal workings , just we think about

What is the output to our system?

What is the output for given input to our system?

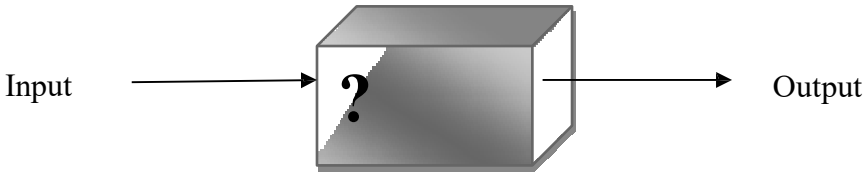


Figure 5.1

The Black box is an imaginary box that hides its internal workings

White box testing:

White box testing is concerned with testing the implementation of the program. the intent of structural is not to exercise all the inputs or outputs but to exercise the different programming and data structure used in the program. Thus structural testing aims to achieve test cases that will force the desire coverage of different structures . Two types of path testing are statement testing coverage and branch testing coverage.

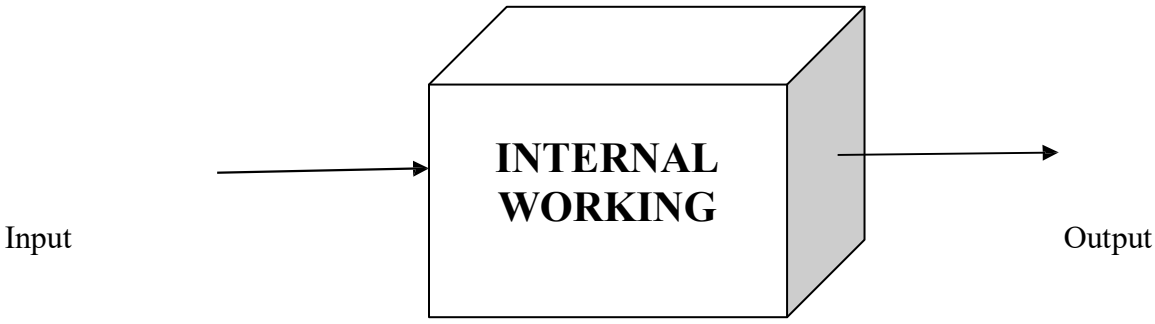


Figure 5.2

The White Box testing strategy, the internal workings

Test Plan:

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules , allocates the resources , and specified guidelines for testing . During the testing of the unit the specified test cases are executed and the actual result compared with expected output. The final output of the testing phase is the test report and the error report.

Test Data:

Here all test cases that are used for the system testing are specified. The goal is to test the different functional requirements specified in Software Requirements Specifications (SRS) document.

Unit Testing:

Each individual module has been tested against the requirement with some test data.

Test Report:

The module is working properly provided the user has to enter information. All data entry forms have tested with specified test cases and all data entry forms are working properly.

Error Report:

If the user does not enter data in specified order then the user will be prompted with error messages. Error handling was done to handle the expected and unexpected errors.

5.2 Test Cases:

TEST NO	TESTCASE DESCRIPTION/ACTION	INPUT	EXCEPTED OUTPUT	OBSERVED OUTPUT
1.	Validate HDFS Path	Without including port number to the path of our file in HDFS	Should redirect to the given path in HDFS	Cannot identify the path of file to where it should go.
2.	Validate PDF File	Give the pdf image formatted file to scala program	Should split the text into tokens	Cannot recognize the file of image format

6. CONCLUSIONS

Reducing the hardware and software cost

It is portable

The type of error also more than one computer can be interconnected and communicated so that streaming is possible from one system to another system

Data is visualized using tableau

We can easily analyze large amount of data

7. BIBLIOGRAPHY

7.1 List of Web References:

http://spark.apache.org/docs/latest/streaming-flume-integration.html/

http://datametica.com/integration-of-spark-streaming-with-flume/

https://www.sigmoid.com/getting-data-into-spark-streaming/