Hadoop HBase Weather Project

Hbase 1.1.2

Sujan Maddumage Don
Index: 985333
Ms in Computer Science
Maharishi University of Management
10/23/2016

Table of content

Solution	Hadoop HBase Weather data managing.		
Description			
Client	*unknown*	**Estimated Effort**	
Author	Sujan Duminda	**Drafted Date**	10/23/2016
Reviewed		**Reviewed Date**	
Version	0.1	**Last Modified Date**	10/23/2016
Status	Draft	**CR/SRS Ref**	

Name	Date	Reason For Changes	Ver
Sujan Duminda	10/19/2016	Initial Version	0.1

Overview

HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed File System), providing BigTable-like capabilities for Hadoop.

Getting Started with Apache HBase

Setup environment

Download cloudera virtual box version from below link
http://www.cloudera.com/downloads/manager/5-8-2.html version 5.8.2
Downloaded and Installed VirtualBox windows version from below link in order to run with linux virtual platform
https://www.virtualbox.org/wiki/Downloads version 5.1.8
Open Oracle virtualbox and drag the cloudera-quickstart-vm-5.8.0-0-virtualbox file into below window.

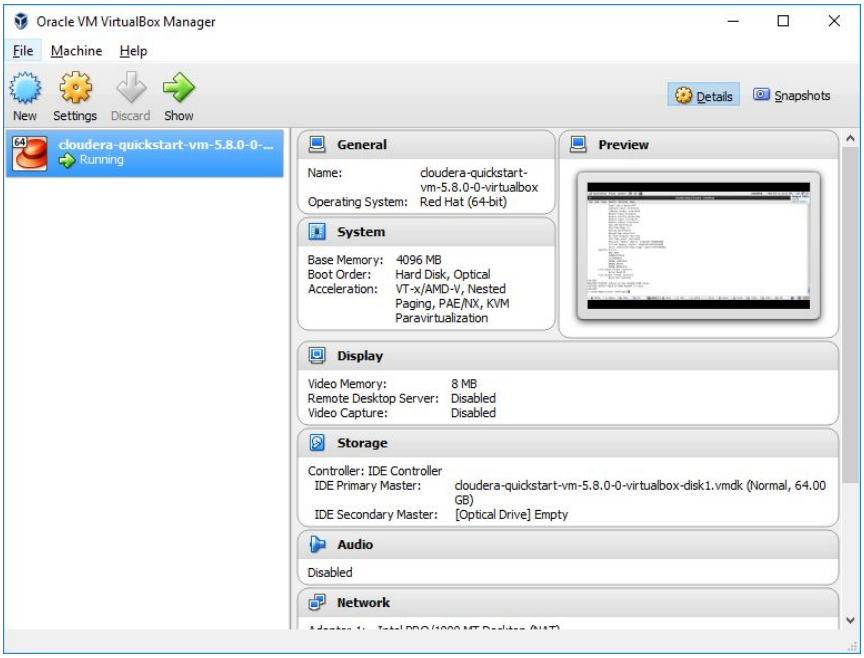


Figure 1 oracle virtual box

Run cloudera by clicking show button and virtual cloudera machine will be opened as below.

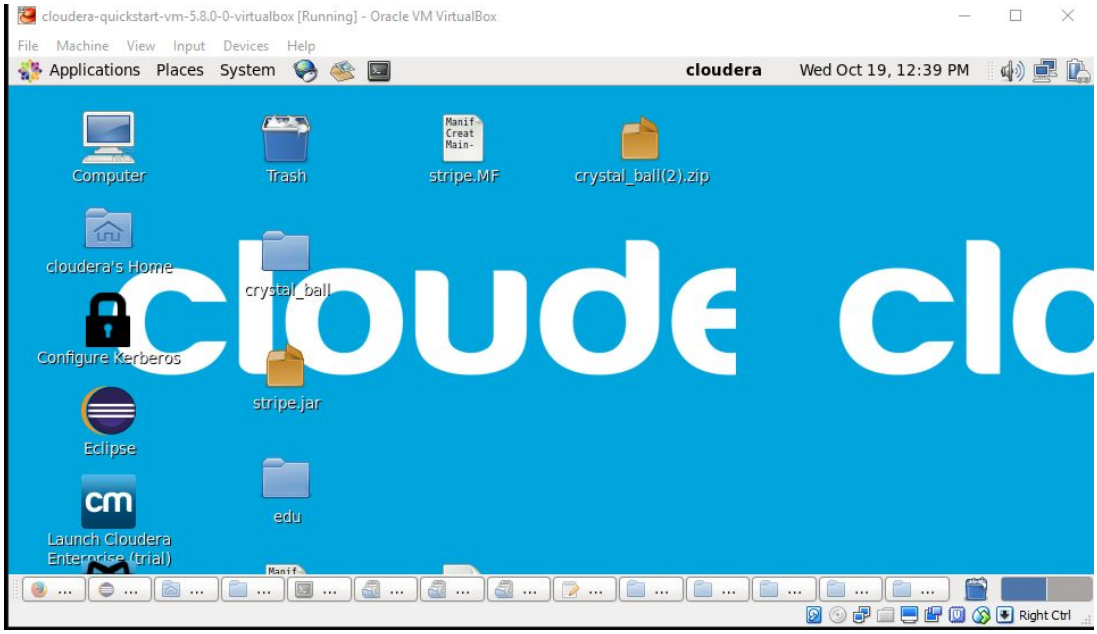
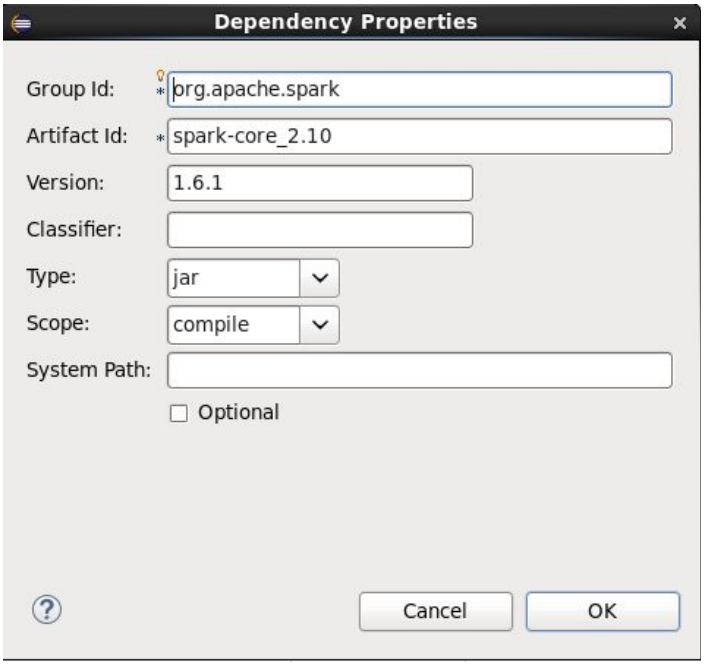


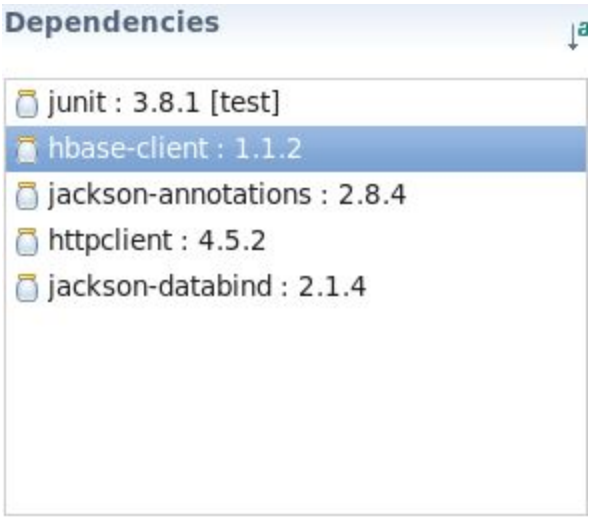
Figure 2 cloudera virtual platform

Create new project

Open eclipse in cloudera and create a new maven project.
Open the pom.xml and add spark jar file detail under dependency.
Use spark jar properties as below.



Other libraries



Project name : hbase-weather.
Project structure as below :

This project has been created for java 7. I
.

Problem :

- Get the weather information of three cities (NEWYORK, LONDON , COLLOMBO) time to time as user requested day and save them by allowing user to get and save each day of weather information.

Solution

I created Hbase nosql solution as below.

Step 1 - Hbase table structure

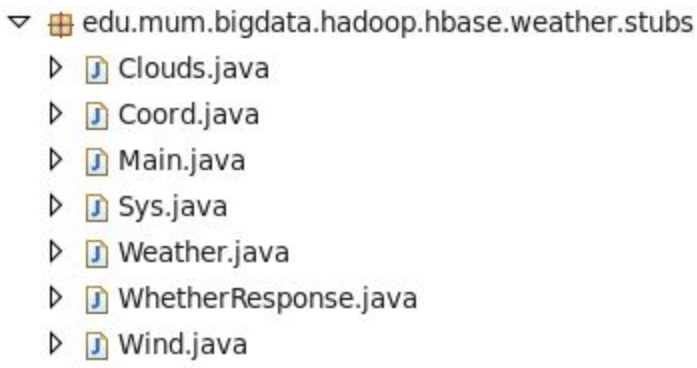
Newyork				London				Colombo			
c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4

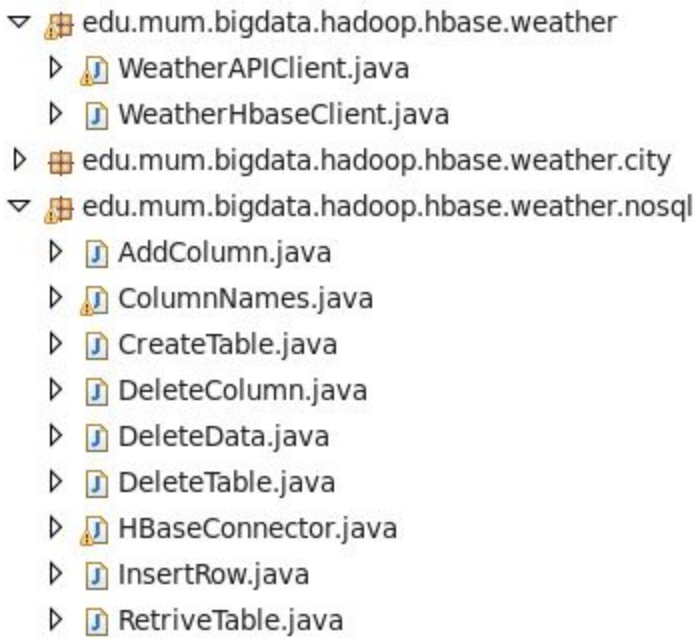
Column families -
New york
London
Colombo

Each column family has below column names
Description
Humidity
Temperature
Wind Speed

Note : Each row Id is save as day. User has to enter day as row id when run the application.

Step 2 - Java implementation





package structure of the project

First package - all weather api Json- java stubs
Second package - Two handlers of weather API and Hbase system
Fourth package - Hbase Nosql java classes

Connection Handler

Here Hbase function i have included as statics class and each class has sql java function with connection. This connection is handled by **HBaseConnector** class that it keep one connection until all tasks are completed. each sql function check whether connection is live, if is it closed then it get new connection thought **HBaseConnector** . Singleton implementation.

Eg : when create a class using create class object, It create through HBaseConnector's connection. If connection is lost , It create new connection.

```java
public class CreateTable {


    public static void createTable( String tableName ,List<String> columnFamilies)
    {
        try{
                HTableDescriptor tableDescriptor = new HTableDescriptor(TableName.valueOf(tableName));
               //column tableDescriptorfamily
                for(String family : columnFamilies)
                {
                tableDescriptor.addFamily(new HColumnDescriptor(family));
                }

                HBaseConnector.getConnection().getAdmin().createTable(tableDescriptor);
                System.out.println(" Table created ");
        }
        catch (IOException e) {
            e.printStackTrace();
        }
```

```java
public class HBaseConnector {


    private  static Connection connection = null;


    public static Connection getConnection() {

        if(connection==null ||connection.isClosed())
        {
            createConnection();
        }
        return connection;
    }
```

Main process

Main method first create table structure if table not exists
Get the argument as row id. .

```
public static void main( String[] args )
{
    //DeleteTable.deleteTable(ColumnNames.WEATHER);
    String rowId=args[0];

    if(!TableExist.tableExist(ColumnNames.WEATHER))
    {
    CreateTable.createTable(ColumnNames.WEATHER, ColumnNames.columnfamilies_weather);
    AddColumn.addColumn(ColumnNames.WEATHER, ColumnNames.DESCRIPTION);
    AddColumn.addColumn(ColumnNames.WEATHER, ColumnNames.TEMPERATURE);
    AddColumn.addColumn(ColumnNames.WEATHER, ColumnNames.WIND_SPEED);
    AddColumn.addColumn(ColumnNames.WEATHER, ColumnNames.HUMIDITY);
    }
    DeleteColumn.deleteolumn(ColumnNames.CONTACTS, ColumnNames.AGE);
    HBaseConnector.closeCoonection();


    WhetherResponse response = null;
    String description ;
    String humidity ;
    String windSpeed;
    String temperature ;
    for(String city : ColumnNames.columnfamilies_weather)
    {
        if (ColumnNames.NEWYORK.equals(city)) {
            response = WeatherAPIClient.getWhetherbyCity(
```

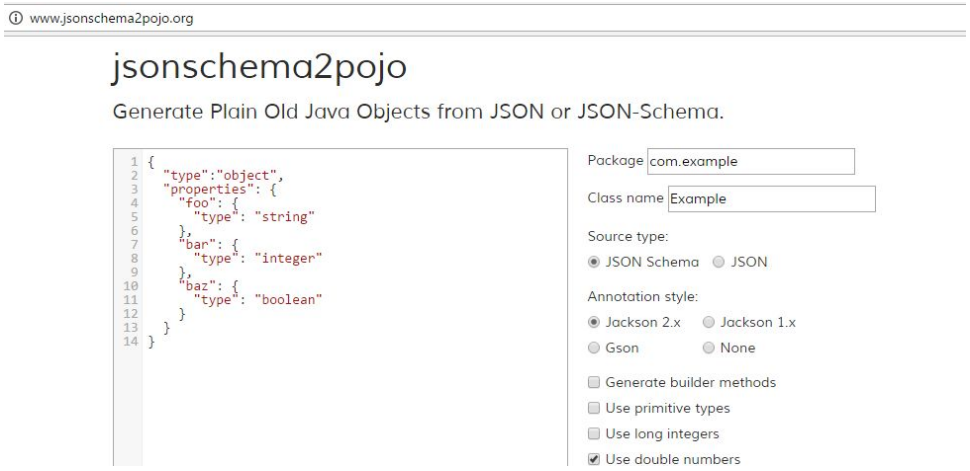
Weather ApI handler

After create the table structure if not exist , getting available table , then system send request to weather API using each city id and user id. Then Api send response as JSON format. Using Jakson libraries System get Json response as Java object. Request are handled by Apache Http client libraries.
Then using decoded response system send humidity , temperature , wind speed and description to Hbase.

```
WhetherResponse response = null;
String description ;
String humidity ;
String windSpeed;
String temperature ;
for(String city : ColumnNames.columnfamilies_weather)
{
    if (ColumnNames.NEWYORK.equals(city)) {
        response = WeatherAPIClient.getWhetherbyCity(
                ColumnNames.NEWYORK, "US");
    } else if (ColumnNames.COLOMBO.equals(city)) {
        response = WeatherAPIClient.getWhetherbyCity(
                ColumnNames.COLOMBO, "LK");
    } else if (ColumnNames.LONDON.equals(city)) {
        response = WeatherAPIClient.getWhetherbyCity(
                ColumnNames.LONDON, "UK");
    }
 description = "No descritopn";
 humidity = response.getMain().getHumidity().toString();
 windSpeed = response.getWind().getSpeed().toString();
 temperature = String.valueOf(Math.abs((response.getMain().getTemp()- 273.15)*100)/100);

for( Weather weather: response.getWeather())
{
    description = weather.getDescription();
```

This Json to java convertion is done by below link.
http://www.jsonschema2pojo.org/



Sample Json response

{"coord":{"lon":-0.13,"lat":51.51},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"base":"stations","main":{"temp":283.056,"pressure":1016.39,"humidity":76,"temp_min":283.056,"temp_max":283.056,"sea_level":1023.9,"grnd_level":1016.39},"wind":{"speed":5.25,"deg":78.0017},"clouds":{"all":92},"dt":1477261535,"sys":{"message":0.0141,"country":"GB","sunrise":1477204828,"sunset":1477241280},"id":2643743,"name":"London","cod":200}

Insert Data

When inset each data , system send decoded data to all cities column of Hbase.

```
for( Weather weather: response.getWeather())
{
    description = weather.getDescription();
}

    InsertRow.insertData(ColumnNames.WEATHER, rowId, city, ColumnNames.DESCRIPTION, description);
    InsertRow.insertData(ColumnNames.WEATHER, rowId, city, ColumnNames.TEMPERATURE, temperature);
    InsertRow.insertData(ColumnNames.WEATHER, rowId, city, ColumnNames.WIND_SPEED, windSpeed);
    InsertRow.insertData(ColumnNames.WEATHER, rowId, city, ColumnNames.HUMIDITY, humidity);

}
```

Al column names are used as constance as below.

```
public class ColumnNames {
    public static final String NAME = "name";
    public static final String EMAIL = "email";
    public static final String PHONE = "phone";
    public static final String AGE = "age";
    public static final String PERSONAL = "personal";
    public static final String OFFICE = "office";
    public static final List<String> columnfamilies_contact = Arrays.asList(PERSONAL,OFFICE);
    public static final List<String> columns_contact= Arrays.asList(NAME,EMAIL,PHONE);

    public static final String LONDON = "london";
    public static final String NEWYORK = "newyork";
    public static final String COLOMBO="colombo";

    public static final String DESCRIPTION= "description";
    public static final String TEMPERATURE = "tempereture";
    public static final String WIND_SPEED= "wind_Speed";
    public static final String HUMIDITY= "humidity";

    public static final List<String> columnfamilies_weather = Arrays.asList(LONDON,NEWYORK,COLOMBO);
    public static final List<String> columns_weather= Arrays.asList(DESCRIPTION,TEMPERATURE,WIND_SPEED,HUMIDITY);
```

Sample output result

This data is saved in Hbase tables and create file in output folder as well.

.

ROW ID :monday
tempereture - Celsius humidity - % wind_Speed - ms-1

london - description : few clouds
london - tempereture : 11.732000000000028
london - wind_Speed : 5.69
london - humidity : 67

newyork - description : clear sky
newyork - tempereture : 11.982000000000028
newyork - wind_Speed : 6.56
newyork - humidity : 84

colombo - description : broken clouds
colombo - tempereture : 25.75700000000001
colombo - wind_Speed : 3.66
colombo - humidity : 100

ROW ID :sunday
tempereture - Celsius humidity - % wind_Speed - ms-1

london - description : overcast clouds
london - tempereture : 9.906000000000006
london - wind_Speed : 5.25
london - humidity : 76

newyork - description : clear sky
newyork - tempereture : 14.706000000000017
newyork - wind_Speed : 6.0
newyork - humidity : 74

colombo - description : light rain
colombo - tempereture : 24.710000000000036
colombo - wind_Speed : 3.64
colombo - humidity : 100

ROW ID :tuesday
tempereture - Celsius humidity - % wind_Speed - ms-1

london - description : overcast clouds
london - tempereture : 9.906000000000006
london - wind_Speed : 5.25

london - humidity : 76

newyork - description : clear sky
newyork - tempereture : 14.706000000000017
newyork - wind_Speed : 6.0
newyork - humidity : 74

colombo - description : light rain
colombo - tempereture : 23.80600000000004
colombo - wind_Speed : 3.4
colombo - humidity : 100

ROW ID :wednsday
tempereture - Celsius humidity - % wind_Speed - ms-1

london - description : overcast clouds
london - tempereture : 9.906000000000006
london - wind_Speed : 5.25
london - humidity : 76

newyork - description : clear sky
newyork - tempereture : 14.706000000000017
newyork - wind_Speed : 6.0
newyork - humidity : 74

colombo - description : light rain
colombo - tempereture : 23.80600000000004
colombo - wind_Speed : 3.4
colombo - humidity : 100

Technical Details

Required technology

Linux platform cloudera 5.8.2
Oracle virtual box 5.1.8
Java 7
Eclipse platform
Hbase 1.1.2
Apache http client

Jackson - annotation
Jakson-databind

Review Comments