ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Analysis of Agriculture Commodity Prices using MapReduce Model

Roopam Dad[1], Nikhil Shanmugam[2], Prapti Singh[3], S.M Nalawade[4]

BE, Computer, Sinhgad Institute of Technology, Lonavala, India[1,2,3]

Professor, Computer Department, Sinhgad Institute of Technology, Lonavala, India [4]

Abstract: India is the fastest growing economy in the world. Agriculture is the backbone of country. Agriculture continues to be main stay of life for majority of the Indian population. It contributes around 25% of the GDP and employs 65% of the workforce in the country. Big Data can help agriculture sector to chance its phase and rise with the growing economy. It can provide economists and policy makers with insightsand help in makingdecisions and crucial policies. In this paper we are proposing a system which accumulates various commodity data from open government data sources, filter it by MapReduce steps as table functions and producing current and yearly trends that canhelps farmers selling their crops at best profitable prices and provides detailed analysis for various commodities and markets.

Keywords: Data Collection, Analysis, Agriculture, MapReduce.

I. INTRODUCTION

This paper proposes a system which inputs different commodity data sets from open government data sources and analyze it produce monthly, yearly trends, location based current and previous prices and production, comparison graph of commodities in different Markets/Cities/Market/States, price and production rise/fall graphs over time.
The proposed system has three major components:

1. Data collection Agents
2. Analyzer(MapReduce)
3. User Interface(Cross Platform Application) **Technology used :** Java, Bootstrap framework, Hadoop MapReduce.

II. DESIGN of SYSTEM

In this section we describe the general architecture of the proposed system. This section helps you understand how various components are interrelated to each other and how data is collected and processed to give desired output. Formal modeling is a process of writing and analyzing formal description of models and systems that represent real world process. It is technique to model complex phenomenon as mathematical entities so that rigorous analysis techniques can be applied on the models to understand the reality of complex phenomenon. System S is divided into three sub systems :

S={s1,s2,s3}
Where S1-> Data collection system
S2-> Data filtering and analytics system
S3-> Presenting data in rich format.

A. DATA COLLECTION SYSTEM :

Data from different sources is input to the data collection system.
Let different sources be T1,T2,….Tk.
For each Ti :
Ti={(k1,v1),(k2,v2)…..(kn,vn)}
Where (ki,vi) are key-value pairs.

Data from various sources can be collected with the help of web scrapping tool Mozenda, building specialized downloader using java which extracts data from websites and from creating surveys using open data source kit to target smart phone users.

Before understanding what data collection Agent does you need to know what are different data sets and from where they are collected.

1. DATA SETS :

Let us understand Data sets with set theory. Let C be a commodity data set.

C={C1,C2,C3.....Cn}, where Ci = commodity i.
Now, every commodity Ci can have specific values over time.

Ci={N,D, Q,M,P1,P2,P3,V}, where
N : name of Commodity, D : date of arrival in market, Q : quantity, M : name of Market,

P1 : minimum price, P2 : maximum price, P3 : modal price and V : variety.
Proposed system collects this data sets from year 2005-2015.

2. SAMPLE DATA SETS :

N	D	Q(quintals)	M	P1	P2	P3	V
onion	4/2/2013	42.2	pune	25	30	27	-
rice	5/2/2013	35.3	lonavala	50	70	60	-
potato	5/2/2013	13.5	kohlapur	20	25	22	-
corriander	6/3/2013	55.6	kohlapur	30	40	35	-
tomato	7/3/2013	56.9	mumbai	30	40	35	-
onion	8/3/2013	22.5	pune	20	30	25	-
onion	8/3/2013	20.6	nashik	26	28	27	-
cauliflower	8/3/2013	12.7	nagpur	50	80	65	-
lemon	9/4/2013	5.8	nagpur	15	25	20	-
oranges	12/4/2013	20.8	nagpur	100	140	120	-

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

3. AGENT :

Agent is a Java program specifically developed to collect data from open government data websites and www.argmarknet.nic.in. Bulk of commodity data has been collected for year 2005-2015. It also has feature which collects data on daily basis.

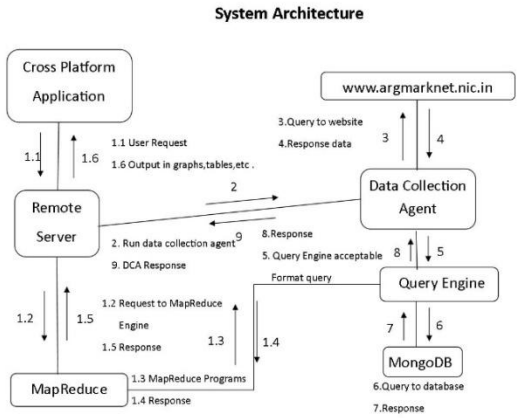

Figure 1: System Architecture

III. DATABASE AND STORAGE

Data collected from data collection agent is directly stored in master database. As collected data is mainly structured data, we are using Oracle RDBMS for storage of data. The Oracle RDBMS has support for the MapReduce paradigm for years through user defined pipelined table functions and aggregation objects.

Oracle provides connectors to allow external Hadoop programs to access data from traditional databases and to store Hadoop output on databases. Aster Data provides MapReduce extensions to SQL called SQL-MapReduce for writing MapReduce programs within the database.

IV. MAP REDUCE PROGRAMMING MODEL

Many NoSQL systems including MapReduce adopt the Key/Value pair as the data Model. A MapReduce computation takes a set of input Key/Value pairs and produce a set of output Key/Value pairs. One round of computation is generally divided into three phases : Map, Shuffle and Reduce.

Map : <k1,v1> --> {<k2,v2>,.....}. The map phase executes the user defined Mapper method to parse input pairs and produce a set of intermediate pairs.

Shuffle : {<k2,v2>,.....} --> {<k2,{v2,.....v2}>,.....}. The shuffle phase defined by the MapReduce library, groups all intermediate values associated with the same intermediate key together, so they are ready to be passed to the reduce phase.

Reduce : <k2,{v2,.....v2}> --> {<k3,v3>.....}. The reduce phase executes the user defined reducer Method to process the intermediate values associated with each distinct intermediate key.

V. MAP REDUCE STEPS AS TABLE FUNCTIONS

The shuffle phase of MapReduce is performed by the PQ (parallel query) Engine while both Map and Reduce phases are implemented by pipelined table functions. Pipelined table functions were introduced in Oracle 9i as a way of embedding procedural logic within a dataflow. Table functions take a stream of rows as an input and return a stream of rows as output. They can appear in FROM clause and act like a normal cursor. In the example below, Table_Func is a user defined table function.

```
INSERT INTO OutTable
SELECT * FROM TABLE
( Price_Display_Reduce ( :ConfKey ,
CURSOR (SELECT * FROM TABLE
(Price_Display_Map ( :ConfKey,
 CURSOR ( SELECT * FROM InTable))))))
```

The output from Price_Display_Map is partitioned by key and streamed to Price_Display_Reduce whose output is inserted into OutTable. Each table function performs one of the MapReduce phases.

In this way user writes a SQL queries that integrate the Map and Reduce functionalities in an intuitive way. All the computation details are hidden from users and handled by the Oracle parallel Query Engine. In addition, this approach greatly generalizes the MapReduce Model due to flexibility of SQL.

VI. DATA STORAGE ARCHITECTURE

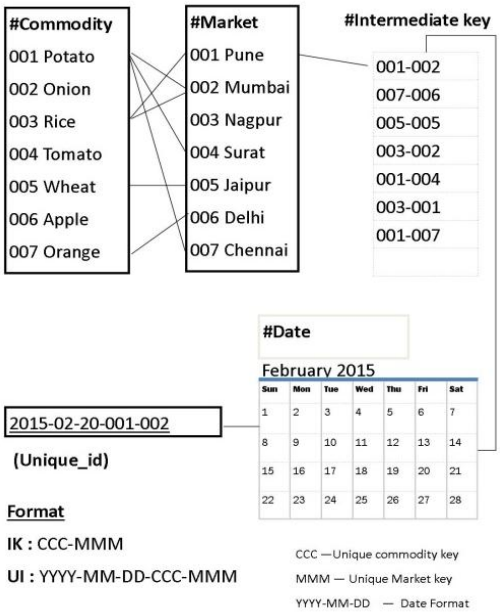

Figure 2 : Structure of the unique key.

A. *INTERMEDIATE KEY:*

There are two lists #Commodity and #Market. Commodity list contains all the commodity names with each having a unique id. Same case is with the market list that stores all market names. Commodity and Market lists generates a

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

combine Intermediate key . Intermediate key is in CCC-MMM format. Here CCC and MMM denotes unique commodity and Market id respectively.

Suppose user want to see potato price in Mumbai then a unique intermediate key will be generated as "001-002" where "001" is the id of potato in commodity list and "002" is the id of Mumbai in Market list. See Fig 2. Mapping is done between Commodity and Market list on user demand. This technique generates a unique id for every unique query.

A. UNIQUE ID :

Every data set is stored with a unique id in the database. If simplification is applied in data storage and its structure then it becomes easy to retrieve data no matter how complex the query is.

"YYYY-MM-DD-CCC-MMM" (date- commodityID - marketID) is the format of the unique id. Data type of Unique ID is string. A specific commodity with a specific market on a specific date with particular values such as its price, variety, quantity, quality will be stored with a unique id.

Let us understand by an example that how it is efficient to store and find data with the proposed unique key. Suppose we want to see price and quantity of commodity potato in Mumbai on 20/02/2015.

With this information first, an intermediate key will get generated as "001-002", now by adding date to it a unique key as "2015-02-20-001-002" is generated. This key directs cursor for where to look for the required data in the database. Figure 2 shows sample database key/value pairs.

{Key} {Values}

UNIQUE_ID yyyy-mm-dd-ccc-mmm	Min Price (Rs/kg)	Max Price	Avg Price	Quantity (quintals)	Variety
2013-01-01-005-005	25	30	27	42.2	-
2013-02-05-003-002	50	70	60	35.5	-
2013-02-05-001-007	30	40	35	17.8	-
.....	...	...	...	...	-
.....	...	...	...	...	-
2015-02-20-001-002	40	80	60	24	-

Figure 3:Different attributes of unique key and sample values at it.

VII. DATA FETCHING

With the assumed format of the unique key querying and fetching data gets simpler and time complexity gets reduced. Let us see it with some sample queries which system uses. Assume that there are 100 Markets and 100 commodities starting from 001 and ending at 100.

1. Single Market Multiple Commodities

Market name : Pune Code : 001
Commodity 1 Code : 001
Commodity 100 Code : 100
Date : 2015/02/20

So, first unique id value will be "2015-02-20-**001**-001" and last unique id value will be "2015-02-20-**100**-001". In above two unique ids, values except in bold are constants. So a pseudo query structure can be Select from table where key is like **2015-02-20-CCC-001** where CCC is between 001 and 100. This query selects required commodity data for the market.

Commodity	Min Price	Max Price	Quantity
Onion	20	30	43.2
Rice	60	80	100.5
….	….	….	….
….	….	….	….
Cotton	120	160	….

Table 1: Various commodities in Single market.

The above Table 1 shows all the commodities available in the chosen market.

2. Multiple Market Single Commodity

In Single Market Multiple commodity, commodity id is the variable id. In Multiple Market Single commodity market id is variable id. This query compares different attributes like price, production, quantity, demand, supply, of common commodities in different market.

Select from table where key is like **2015-02-20-097-MMM** where CCC is between 001 and 100.

Figure 4 shows price of potato in different markets across India. Similarly different attributes can be seen for various commodities in different markets.

1. Multiple Market Multiple Commodity

In Multiple Market Multiple Commodity both market id and commodity id will depend on the request from user and data is fetched according to the changes in those two sub keys. Date is kept constant. Query written below selects multiple commodities from multiple markets.

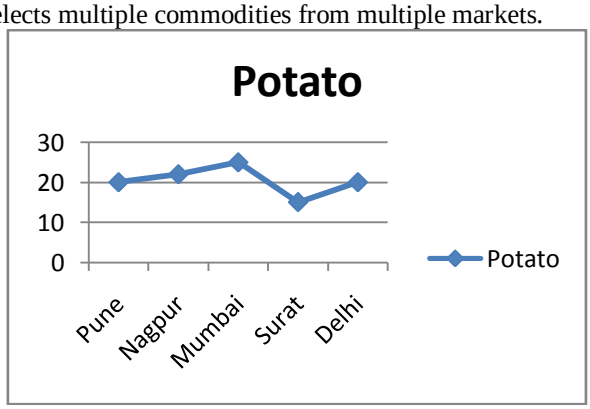

Figure 4: Price of commodity Potato in different markets.

DOI 10.17148/IJARCCE.2015.4416

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Select from table where key is like **2015-02-20-CCC-MMM** where CCC is between 001 and 100 and MMM is between 025 and 050.

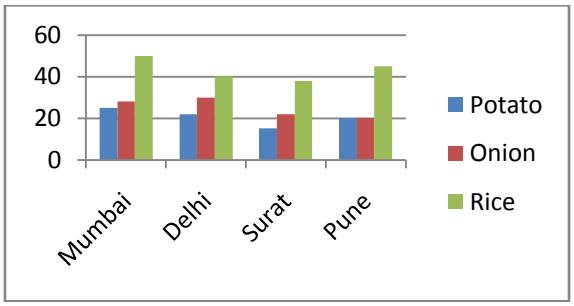

Figure 5: Prices of various commodities in different markets.

2. Single Market Single Commodity over time (Month/Year/Week)
Assume, start date : 2013/12/01 and end date: 2015/02/20. Market name : Pune Code : 001 commodity name : Rice code : 003.

For yearly,
First unique id will be "2013-**12-01-003-001**" and last unique id will be "2015-**12-31-003-001**". Resultant pseudo query will be :

Select from table where key is like YYYY-MM-DD-003-001 where YYYY is between 2013 and 2015.

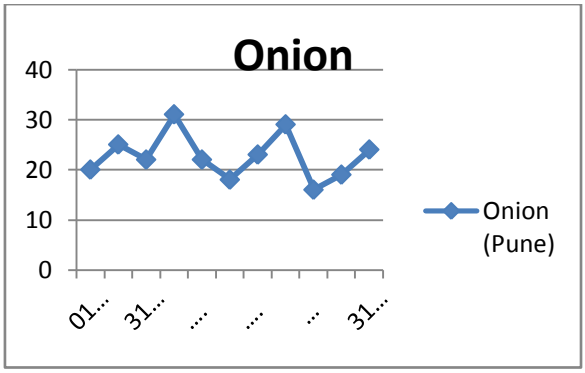

Figure 6: Change in price of onion overtime.

3. Multiple Market Single Commodity over time (Month/Year/Week)
Commodity name : Rice Code : 003

For Single month,
Let month be may,2014.
First unique id will be 2014-05-01-003-NNN and last unique id will be 2014-05-30-003-NNN. Resultant pseudo query will be :

Select from table Where key is like 2014-05-DD-003-NNN where DD is between 01 and 31 and NNN is between 020 and 040.

For couple of months,
Start Month : May 2014
End Month : December 2014
markets.
Select from table where key is like 2014-MM-DD-003-NNN where MM is between 05 and 12, DD is between 01 and 31 and NNN is between 020 and 040.

1. Multiple Market Multiple Commodity over time (Month/Year/Week)
Time period is fixed by user. User can set period as week month or year and then can look records for any month year which is in the system.
For a year,
Let year be 2014.

Select from table where key is like 2014-MM-DD-CCC-MMM where CCC is between 001 and 005 and MMM is 020,028,050.

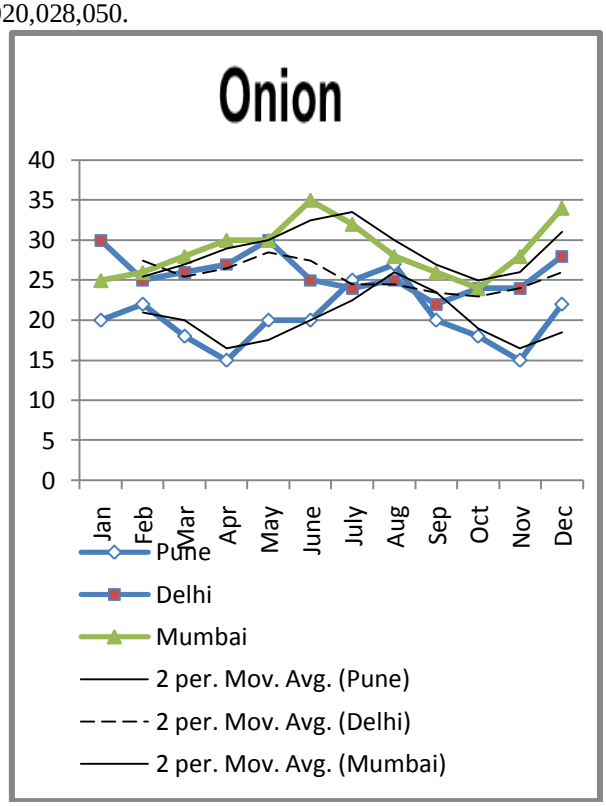

Figure 7: Change in price of onion overtime in different

VIII. USER INTERFACE

A cross platform application is developed using HTML5, CSS3 and JavaScript. Application is built upon Bootstrap framework using Intel XDK. This application helps user to see on demand current commodity prices in his/her locality, monthly and yearly trends of selected commodity, daily market prices, comparing prices of commodities between different cities/markets/states over time and help policy makers in making new and crucial policies and predicting market flow.

User interaction with application works as input Key/value pairs for MapReduce jobs. On the basis of received input values server executes on demand MapReduce Jobs. Output from MapReduce Jobs comes in Tabular Manner. This output is then converted into different line and bar graphs with the help canvas.js and Google chart APIs.

DOI 10.17148/IJARCCE.2015.4416

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

IX. CONCLUSION

The main goal for the system is to provide its users with accurate and in time commodity prices, helping farmers in selling their crops at right prices by providing them with current market trends and rates in their locality, giving him knowledge about best farming practices by providing him with useful agriculture information regarding climate, growing crops and land fertility.on various factors like demand and supply, prices over time, change in production rate and climate data is used to analyze production and price of various commodities and predicting future prices.

REFERENCES

[1]. Lingling Xu, "The Analysis of the Vegetables' Price Fluctuation with Cobweb Model" IJCRB December 2012

[2]. Franco Rosa, Michela Vasciaveo, "Agri-Commodity Price Dynamics: The Relationship Between Oil and Agricultural Market,"International Association of Agricultural Economists(IAAE) Triennial Conference, Foz do Iguaçu, Brazil, 18-24 August, 2012.

[3]. Raghava Rao Mukkamala, Abid Hussain and Ravi Vatrapu, "Towards a Set Theoretical Approach to Big DataAnalytics" Big Data (Big Data congress), IEEE 2014.

[4]. Anokwa, Y. et al. "Open Source Data Collection in the Developing World." Computer 42.10 (2009): 97-99. ©2009 Institute of Electrical and Electronics Engineers

[5]. "Vegetable Sub Sector Study to Understand the Supply Chain Coordination in Darrang, Barpeta and Kamrup District of Assam" A collaborative inhouse Study of CENTRE FOR MICROFINANCE & LIVELIHOOD November 2010

[6]. Virender Kumar, H.R. Sharma and Kamlesh Singh, "Behaviour of Market Arrivals and Pricesof Selected Vegetable Crops: A Study of FourMetropolitan Markets" Agricultural Economics Research ReviewVol. 18 July-December 2005 pp 271-290.

[7]. Big Data Strategy — Issues Paper, © Commonwealth of Australia 2013.

[8]. Patel, A.B, Birla, M and Nair, U "Addressing Big Data problems using Hadoop and Map Reduce" Engineering (NUiCONE), 2012 Nirma University International Conference.

[9]. "Harnessing Hadoop: Understanding the Big Data Processing Options for Optimizing Analytical Workloads"© Copyright 2013, Cognizant.

[10]. G.M.Nasira and N.Hemageetha, "Forecasting Model for Vegetable Price Using Back Propagation Neural Network "International Journal of Computational Intelligence and Informatics, Vol. 2: No. 2, July - September 2012.

[11]. Chris Eaton, Dirk Dieroos, Tom Deutsch and George Lapis, "Understandin Big Data". ISBN : 978-0-07-179053-6 MHID : 0-07-179053-5.

[12]. "Professional NoSQL", Copyright © 2011 by John Wiley & Sons, Inc., Indianapolis, Indiana. ISBN : 978-0-470-94224-6.

[13]. "Big Data for Dummies"Copyright © 2013 by John Wiley & Sons, Inc ISBN: 978-81-265-4328-1.