

A SURVEY OF HADOOP ECOSYSTEM AS A HANDLER OF BIGDATA

V.S.Narayana Bhagavatula¹, S.Srinadh Raju²,

S.Sudhir Varma³, Dr.G Jose Moses⁴

^{1,2,3,4} Associate Professor, Department of CSE, Raghu Engineering College, Visakhapatnam, (India)

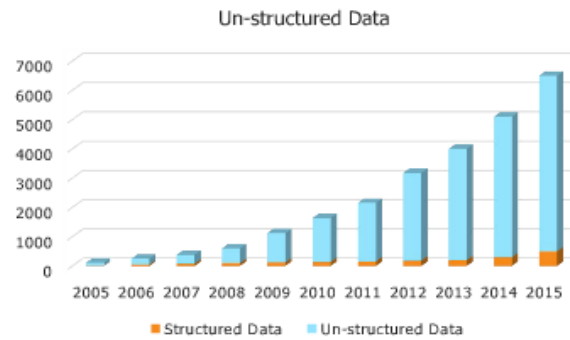
ABSTRACT

Big Data creates a problem to store, analyze and solving huge volumes of data and it cannot be solved with traditional database systems. To solve this problem of Big Data, we have different frameworks available like Hadoop, MongoDB, Cassandra etc., Apache Hadoop is a famous big data framework for handling large data when compared with other frameworks. In this paper we focus on how different components of hadoop ecosystem are built on top of HDFS (Hadoop Distributed File System) for providing high availability and reliability of data. We discuss different Hadoop Eco System Components like Flume, Sqoop, Pig, Hive and Hbase.

Keywords: *Big Data, Hadoop, Flume, Sqoop, Pig, Hive, Hbase.*

I INTRODUCTION TO BIG DATA

Traditional database systems are built only to handle structured data. The challenges with traditional database systems are storing, analyzing, processing, accessing and visualizing the data. In this modern era we receive huge amount of data coming from Social Media websites like Facebook, Twitter, Linked In, from different fields like Stock Market ,Airlines, from different Online Purchasing Portals like Amazon, FlipKart, e-bay, alibaba.com etc and data from emails like Gmail, yahoo mail, rediff mail etc. In Fig 1, we can clearly identify the growth of unstructured data [1]. Most of the web portal data are unstructured data and to analyze these data is very important for the organizations to improve their business, but analyzing these types of unstructured data is a tedious task with traditional database systems.



- By 2020, IDC (International Data Corporation) predicts the number will have reached 40,000 EB, or 40 Zettabytes (ZB)
- The world's information is doubling every two years. By 2020, there will be 5,200 GB of data for every person on Earth.

Fig 1: Growth of unstructured data.

To store this huge data on traditional systems is a difficult task and also to process the data takes more time with traditional systems because the data need to be stored in a single machine (distribution or parallel processing of data is not supported in traditional database systems). To solve this problem We have Hadoop framework that can process the data across different machines (Commodity hardware or systems) and also it supports wide varieties of data like structured data, semi structured data and Unstructured data.

RDBMS		HADOOP
Structured	Data Types	Multi and Unstructured
Limited, No Data Processing	Processing	Processing coupled with Data
Required On Write	Schema	Required On Read
Reads are Fast	Speed	Writes are Fast
Reads are Fast	Speed	Writes are Fast
Software License	Cost	Support Only Open source
OLTP Complex ACID Transactions Operational Data Store	Best Fit Use	Data Discovery Processing Unstructured Data Massive Storage/Processing

Fig 2: Comparison of Traditional Systems and Hadoop.

For example to handle 1 TB of data, let's assume normal database systems take 43 minutes, If we distribute the data across 10 Machines by using Hadoop framework, it takes only 4.3 minutes to process this 1TB of data.

II HADOOP FRAMEWORK

Hadoop [2] is a framework that allows distributed processing of large amounts of data across clusters of different commodity computer systems. Hadoop is discovered to support the characteristics like Scalability, Reliability, Availability and Flexibility. Hadoop framework consists of two components

- i) HDFS (Hadoop Distributed File System) – to store data and
- ii) Map Reduce [3]- to process the data

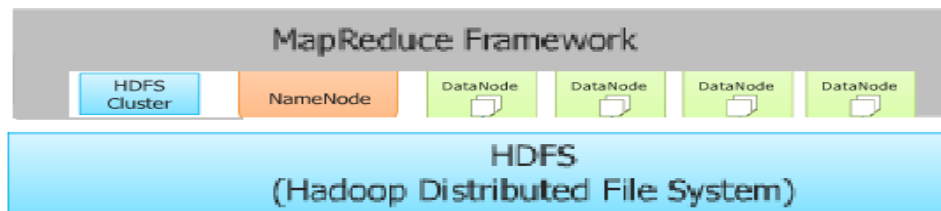


Fig 3: Hadoop framework

Map Reduce [4] Framework is constructed on Master/Slave Architecture where Master is represented with Name Node and Slave is represented with Data Node. The actual data is represented in Data node and the name node contains all the meta-data of data nodes and the cluster configuration information.

III HADOOP ECOSYSTEM

Hadoop Ecosystem is built with different components on top of hadoop framework to store and to ease the processing of data needed by the different users. These hadoop Eco system components are able to handle and analyze the data coming from different data sources like database systems ,OLTP systems, OLAP Systems and Web data written in scripting languages like Python, java pages, and Pearl etc.,

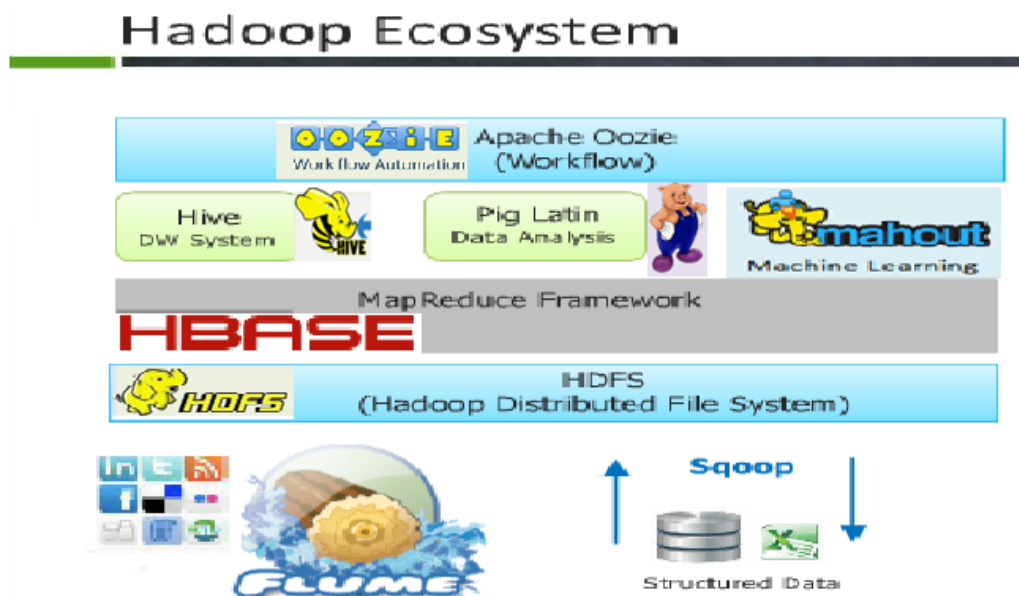


Fig 4: Components of Hadoop EcoSystem.

The different Components of Hadoop Eco System are Sqoop, Flume, Pig, Hbase, and Hive.

3.1 Sqoop

Sqoop [5] is a tool for efficiently transferring data between Hadoop file system and structured data sources such as relational databases. Sqoop imports individual database tables or entire database into Hadoop file system.



Sqoop uses different commands while loading the data from relational databases into hadoop file system such as Import, connect, target, query, columns, username and -P etc..

For example the following import command is used to load the data into HDFS from relational databases.

```

//***** for importing data into HDFS *****
//***** importing data from a table into hadoop *****//

/usr/lib/sqoop-1.4.4/bin/sqoop import --connect
jdbc:mysql://localhost/raghu --table emp -username
raghu -P --target-dir /user/raghu/sqoop/empl -ml

```

Annotations in the image:

- Green text: "for importing data into HDFS" with an arrow pointing to the first comment line.
- Orange text: "table name" with an arrow pointing to the "--table emp" parameter.
- Purple text: "user name" with an arrow pointing to the "-username raghu" parameter.
- Red text: "password" with an arrow pointing to the "-P" parameter.
- Red text: "Database name" with an arrow pointing to the "jdbc:mysql://localhost/raghu" part of the --connect parameter.

Fig 5: import Command in Sqoop

3.2 Flume

Flume[6] is a Hadoop Eco System component which provides services for handling the data from different data streamings like twitter efficiently and to load this stream of data into HDFS for analysis.

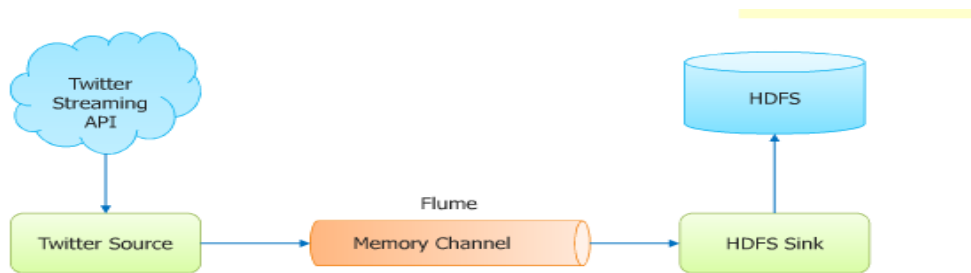


Fig: 6 Flume working model.

To import the data from twitter, Open the configuration file flume.conf and add the tweets or keywords copied from the twitter web pages into this configuration file.

```

TwitterAgent.sources.Twitter.consumerKey=Pw63cpjptT59ulP0zmT6w
TwitterAgent.sources.Twitter.consumerSecret= n8awrhwk0ui5W0wr3NLKf7S576DcILPk5Ddfp1LQUU
TwitterAgent.sources.Twitter.accessToken=1635433267-s0NAOXmRqm5y4UC2WV7HP0ui0E9fPZZ56eW095P
TwitterAgent.sources.Twitter.accessTokenSecret= CBKPGbJLwyJJ1jY4atf7iaiaR96Z1PmVvKF0i0XsP8E

TwitterAgent.sources.Twitter.keywords= Nabam, hadoop, scientist
    
```

Run the following command to extract the data from twitter pages.

```

]$. /flume-ng agent -n TwitterAgent -c conf -f /usr/lib/flume-ng/apache-flume-1.4.0-bin/conf/flume.conf
    
```

Fig 7: Configuration file setting in flume and importing tweets to HDFS.

3.3. Pig

Pig[7] is a high-level declarative language similar to SQL and can accept any kind of data like structured, semi structured and unstructured data, As it accepts any kind of data it got the name Pig. Pig is extended with UDF (User Defined Functions) Feature that accepts the code written in other languages like Python, Java Script, Ruby, PHP and Perl.

Pig is majorly used to reduce the development time of programs and also by using it the lines of code to execute the data are minimized.

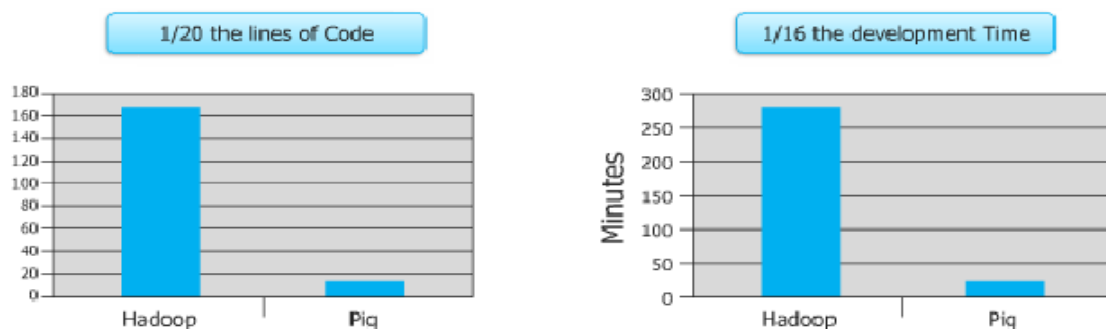


Fig 8: Comparison between mapreduce and pig with respective to time and LOC

Pig has its own scripting Language called Pig Latin, which can store all commands and can be executed sequentially. Pig can be accessed in 2 modes or shells i) local mode ii) HDFS mode

To use pig in local system mode we can use the following command

Pig -x local

To use pig in HDFS mode, we can use the following command

Pig -x mapreduce

Pig execution Environments in shell:

Pig can be executed in the following environments

- Interactive Mode (Grunt shell) –
By using this mode, we can run pig commands likes load, dump, group, cogroup, describe and store etc.
- Batch Mode (Script) –
By using this mode, we can run Apache Pig in Batch mode by writing Pig Latin script in a single file with .pig extension.
- Embedded Mode (UDF) –
By using this mode, we have the provision of defining our own functions (User Defined Functions) in programming languages such as Python, Ruby, Pearl and Java by using them in our script.

3.4. Hive

Hive[8] is a data warehouse package built on top of Hadoop. Hive is developed by Facebook to analyze several Terabytes of data. As most of the ETL developers are used to programming in SQL background, they came up with a language called HQL (Hive QL) which looks similar to SQL.

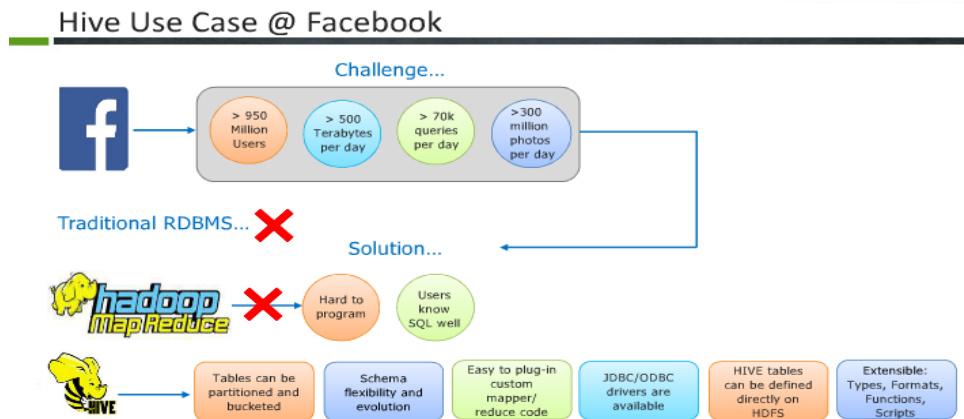


Fig 9: Workflow of Hive

Hive uses a data model for efficient processing of data with the following components

- i) Hive tables
- ii) Partitions
- iii) Buckets

Hive table:

Data can be stored using hive tables.

Partitions:

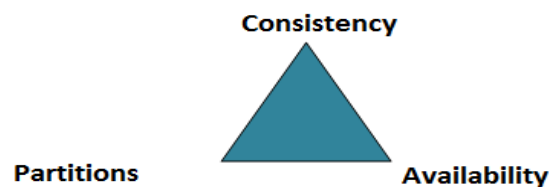
Partition is dividing the hive table data into groups based on Columns which makes it faster to access the data and to do querying on hive data.

Buckets:

Partitions are further sub divided into Buckets. Bucketing by using User-id or Column makes it easier to access. It is also faster.

3.5. Hbase

Hbase[9] is a Hadoop component that supports the features of NoSql database. It sits on top of HDFS and is developed based on Google’s Big data paper. It uses Key-value pair and is a column oriented data store. It is built on CAP theorem (that uses 3 characteristics –Consistency, Availability and Partition)



Hbase Commands:

- 1) Connecting to HBase.

We can connect to the hbase[10] shell by using the following command.

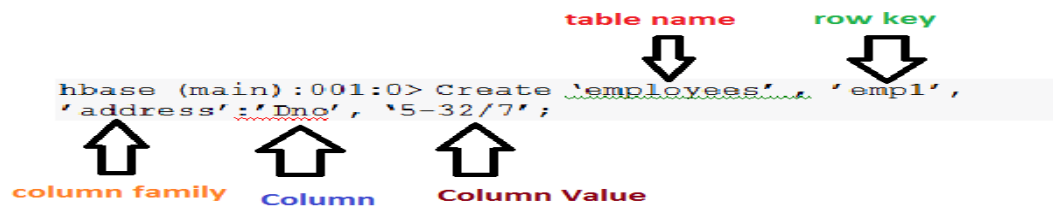
```
$ ./bin/hbase shell  
hbase (main):001:0>
```

- 2) Creating tables in HBase.

We can create the tables by using the keyword ‘Create’. When using the Keyword create, we need to specify the row key and the column family in the command and also the columns which can be defined under Column Family.

Example:

Query to create a table in HBase.



3) To see all tables in HBase

We use list command in Hbase to see all the tables in Hbase.

```
hbase (main):002:0> list
TABLE
employees
1 row(s) in 0.0230 seconds
=> ["employees"]
```

IVCONCLUSION

In this paper, we discussed how big data is able to solve the problem of handling large amounts of data. We saw the core component of hadoop like HDFS to store large data and map reduce for processing the data. Then we showed the different components of hadoop ecosystem like Sqoop, Flume, Pig, Hive and Hbase. These Ecosystem Components ease the process of storing data and processing it.





REFERENCE

- [1] IBM, "Big data at the speed of business," <http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>,
- [2] Apache Hadoop, <http://hadoop.apache.org> .
- [3] Wang G., Tang J. (2012). The NoSQL Principles and Basic Application of Cassandra Model. *Published in Computer Science & Service System (CSSS), 2012 International Conference. Print ISBN 978-1-4673-0721-5.*
- [4] X. Yang and J. Sun, "An Analytical Performance Model of MapReduce," in *Proc. IEEE CCIS*, Sep. 2011, pp. 306–310.
- [5] "Sqoop user guide", <http://sqoop.apache.org>
- [6] Pig home, <https://pig.apache.org>
- [7] "Flume" <https://flume.apache.org/>
- [8] How to process data with hive, hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/
- [9] Introduction to Hbase, <https://hbase.apache.org>
- [10] What is hbase? <https://www.ibm.com/software/data/infosphere/hadoop/hbase>

ACKNOWLEDGEMENT(S)

The authors would like to express heartfelt thanks to the management of Raghu Educational Institutions and its Chairman Sri.Raghu Kalidindi for supporting us to publish this paper as part of encouraging the faculty in research initiatives. We would like to extend our thanks to our colleague P.Siddhartha for his valuable advice and cooperation from time to time in successfully publishing this paper.

About the Authors

	<p>Mr.V.S.Narayana Bhagavatula is currently working as Assistant Professor in the department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam. He obtained his B.Tech (CSE) from ANU, Guntur and M.Tech (Database Systems) from SRM University, Chennai. He is a Lifetime Member of CSI. He did Certification (OCA) in Oracle database. He has presented and published papers in National and International conferences.</p>
	<p>Mr. S.Srinadh Raju is currently working as Associate Professor in the department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam. He obtained his B.Tech from University of Madras, Chennai, and M.Sc. (UK) from Leeds Metropolitan University, Leeds and M.Tech (CSE) from JNTUK. He has presented and published papers in National and International conferences.</p>
	<p>Mr. S.SudhirVarma is currently working as Assistant Professor in the department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam. He obtained his B.Tech (CSE) from Pondicherry University, Puducherry and M.Tech (Database Systems) from SRM University, Chennai. He has presented and published papers in National and International conferences.</p>
	<p>Dr. G. Jose Moses working as Associate Professor (Ratified by JNTUK), in the department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam. He is Approved Research Supervisor for JNTUK. He obtained his B.Tech (CSE) from JNTUH, M.Tech (CSE) from ANU, Guntur, PhD (CSE) from ANU, Rajahmundry. He has presented and published papers in National and International conferences. He is active member of various professional bodies. His research interests lies in Computer Networks, Cloud Computing and Data Mining</p>