www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11322-11331

Extracting Structue Data From UnStructured Data Through HiveQL

K. Balakrishna[1],Smt. S. Jessica Saritha[2], C. Penchalaiah[3]

[1] P.G.Scholor of Computer Science and Engineering Dept, JNTUACEP, Pulivendula, YSR Kadapa (District), Andhra Pradesh-516390, INDIA

Balakrishna.k208@gmail.com

[2] Assistant Professor in Department of CSE JNTUA College of Engineering, Pulivendula ,Andhra Pradesh, India,pin-516390.

sarithajntucep@gmail.com

[3]Dept. of Computer Science and Engineering, JNTUACEP, Pulivendula, YSR Kadapa (District), Andhra Pradesh-516390, INDIA

penchalaiah2000@email.com

Abstract: RDBMS can store structured data up to some GB of data. Processing of large data is very difficult to handle and also time consumption process. To overcome these problems made of using Hadoop. Apache Hadoop is a framework for big data management and analysis. The Hadoop core provides storing of structured, unstructured and semi structured data with the Hadoop Distributed File System(HDFS) and a simple MapReduce programming model to process and analyze data in comparable, the data stored in this distributed system. Apache Hive is a data warehouse built on top of Hadoop that allows you to query and manage large sets in scattered storage space using a SQL-like lingo call HiveQL, Hive translate queries into a series of MapReduce jobs. In existing system unstructured data stored in HDFS can't be retrieve into structured format through HiveQL. In this project It is converting twitter data into a structured format by using HiveQL with SerDe. HDFS can stores twitter data by using data streaming process.

Keywords: Big Data, Apache Hadoop, HDFS, MapReduce, Data Streaming process, HiveQL, SerDe.

1.Introduction:

Today, many organizations recognize that the data they draw together is a important reserve on behalf of perceptive their consumers, the routine of their business in the market place and the effectiveness of their transportation.

1.1 Hadoop:

The Hadoop[2] upbringing emerged as a gainful approach of effective among such huge datasets. It imposes a scrupulous programming model, called Map-Reduce[4], for infringement up division tasks into units that can be disseminated around a cluster of article of trade, server class hardware, thereby providing cost-effective, horizontal scalability. Underside this division model is a distributed file system called the Hadoop Distributed File System (HDFS)[3]. Even though the file system is"pluggable" there are now several commercial and open source alternatives.

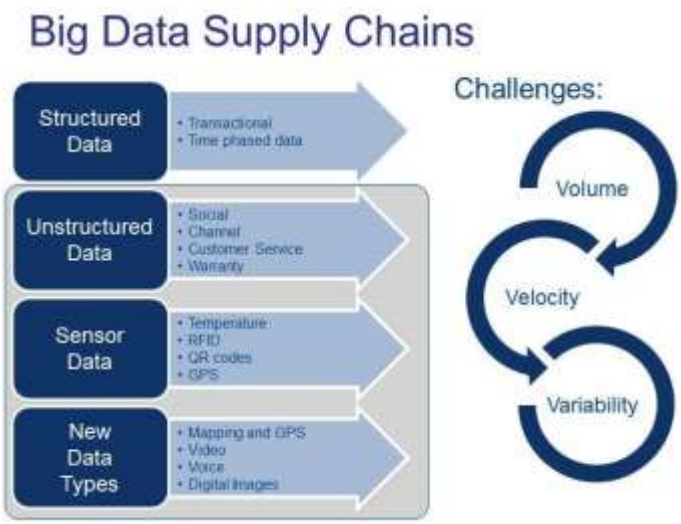

Figure 1 : Big Data Supply chain

1.2 Bigdata:

Big data[1] refers to data organism unruffled in ever-escalating volumes, at gradually more high velocities, and for a widening variety of unstructured formats and variable semantic contexts.

Big data describes any outsized body of digital in a row from the text in a Twitter feed, to the antenna in a row from developed tools, to in sequence about customer browsing and purchases on an online register. Big data can be chronological (maintaining stored data) or real-time (denotation streamed unswervingly from the resource).

For big data to make available actionable acumen or on the way, not only must the right questions be asked and data relevant to the issues be collected, the data must be easily reached, cleaned, analyzed, and then presented in a useful way.

1.2 Hive:

Hive[5] is most appropriate for data depot applications, where relatively static data is analyzed, express response period are not required, and when the data is not changing rapidly. Hive is not a full database. The design constraints and restrictions of Hadoop and HDFS impose limits on what Hive can do. The largest constraint is that Hive does not endow with record-level modernize, insert, nor delete. It can engender new tables from queries or amount produced query results to files. Also, because Hadoop is a batch-oriented system, Hive queries have higher latency, due to the start-up overhead for MapReduce jobs. Queries that would terminate in seconds for a long-established database take longer for Hive, even for reasonably miniature data sets. Finally, Hive does not make available connections.

1.3 Unstructured data:

The whole thing starting societal media posts and feeler data to email, images and web logs is on the increase at an exceptional pace. Here are just a few amazing information: Twitter[6] sees in relation to 175 million tweets every day and have further than 465 million accounts. 571 new websites are twisted every minute of each day. along with the humanity creates 2.5 quintillion bytes of data for each day on or after unstructured data sources resembling sensors, social medium posts and digital photos. plainly, unstructured data[7] is on the increase exponentially, along with government is refusal protection. What does that represent? “Unstructured” means just that -the essentials inside the data have no structure. For example, even a straightforward blog post have many elements embedded in it -the date and time it was posted, the content, embedded links, author, etc. That makes searching and analysis much more complicated than for structured data, like communication.

2. Existing Techniques and Approaches

In the earlier versions are using and giving out the structured and RDBMS[8] data. Structured data is in sequence well thought-out in immeasurable online databases that help to serve search results. Structured

data markup is a across the world collective verbal communication format that helps search engines understand and return the best results for users who are thorough for related data. Schema markup is the foremost type of shared markup vocabulary that Google is using to deliver results. Structured data[9] is a system of combination a name with a value that helps search engines catalog and index your pleased. Micro data is one outline of structured data that moving parts with HTML5. Schema.org is a development that provides a particular set of agreed-upon definitions for micro data tags. the structured data retrieving from the unstructured data earlier technologies are solving more complicated by by means of java and RDBMS[10].

3. Proposed System

Apache Hadoop eco-system is a framework for solving big data management and analysis. The Hadoop core provides storing of structured, unstructured and semi structured data by way of the Hadoop Distributed File System(HDFS) along with a unfussy MapReduce programming model to method and evaluate data in parallel, the data stored in this distributed system. Apache Hive is a data warehouse built on top of Hadoop. Query and manage large sets within distributed storage by means of a SQL[11similar to language call HiveQL, Hive translate queries into a series of MapReduce jobs. In existing system unstructured data stored in HDFS can't be retrieve into structured format through HiveQL. It is converting twitter data into a structured format by using HiveQL with SerDe[12]. HDFS can stores twitter data by using data streaming process. These unstructured data will be formatted into structured format using JSON Validator[13].

4. Methodology Approches

4.1 Storing and Processing the Unstructured data

Made of this consequence presentation, after that it depends on the amount of the cluster. For case in point in order to procedure 20 PB of data, (hypothetically) if Google has a 2000 node cluster, after that the every node will residence approximately 30 TB of data (10 for one set and the put your feet up of the 20 for the 2 replicas). And while this does occupies a lot of space, Hadoop uses inexpensive commodity hardware, and hence the storage gap costs are miniscule as compared to manufacturing evaluation hardware. So storing such amounts of data is very sufficient.

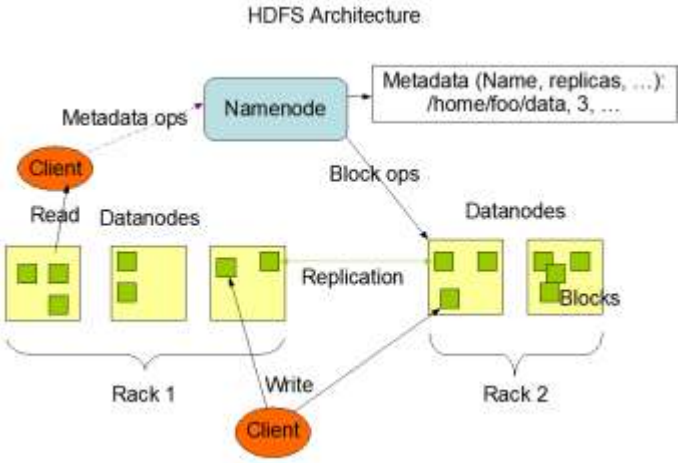

Figure 2 : HDFS Architecture

4.2 Replicating the data has its own uses.

1. To start with, it helps in increases the protection of the data. given that Hadoop[14] uses commodity hardware, its nodes are impediment to failures. So for example if one of the node goes down, it will in summit of fact bring down the replication of a confident block from 3 to 2. So the name node then gets the pre-replicated part of the blocks from any of the 2 remaining data nodes and replicates it all over over again on a different node, professionally bringing the imitation of that meticulous block rear to 3.

Now make up a situation where the duplication aspect is set to 1. One of the node fails and the portion of data stored in node is lost without end as the replication factor was 1 and we have no node from where it can be restored.

2. Secondly, having the same data in composite places enables Hadoop to progression the same data all together. This process is generally conceded out when the name node realizes that a particular node is not the theater well. So it starts the processing of the same data on a different healthier node and if the recovered node finishes the giving out quicker than the node which is not the theater well, the process on that node is killed and the node is no longer given whichever jobs intended for processing.

3. Lastly, Hadoop tries to stay behind to the strategy of bringing the giving out to data and not the other way encircling as it decreases the network transfer. So if a node entirely fails in the focus of a job, then Hadoop tries to run another instance of the job on a dissimilar node (if possible contain the similar data portion) so that no time is shattered in transferring the data chunk to another data node.

Hadoop normally breaks the file keen on chunks of 64 MB by default (although this charge can be distorted while loading the data to the HDFS or in the config file itself). If it breaks the data in between such that imperfect versions of the last line of data are present in two different blocks, then this is marked and handled while reading the data back. The data will be in use into as open starting place by using Apache Flume and also data will be discarded into HDFS. After storing unstructured data to Structure data using the JSON Validator[12] and HiveQL.

5. Implement and Results

5.1 Querying Semi-structured Data with Apache Hive

This is the in a sequence about analyzing Twitter data by means of some of the components of the Apache Hadoop ecosystem that are also accessible in CDH[15] (Cloudera's open-source giving out of Apache Hadoop and connected projects).

Flume[17] can be utilized to swallow data into Hadoop. However, that data is ineffective without some way to examine the data. Personally, It come from the relational world, and SQL is a language that speak fluently. Apache Hive provides an border that allows users to easily access data in Hadoop via SQL. Hive compiles SQL[11] statements into MapReduce jobs, and then executes them across a Hadoop cluster.

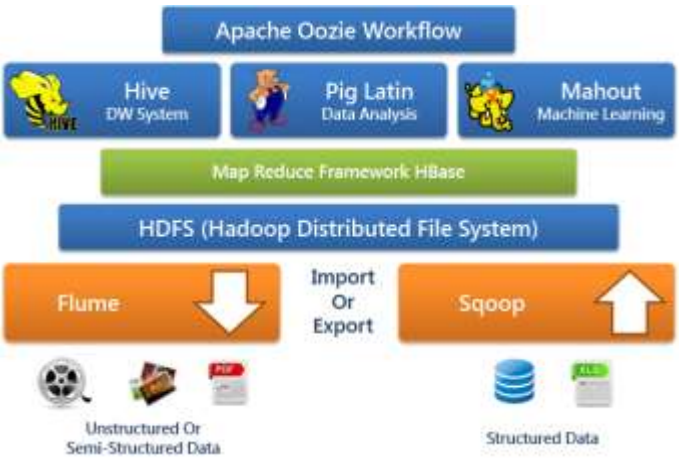

Figure 3 : Unstructured data processing

In this expose, It learn more about Hive, its strengths and weaknesses, and why Hive is the right choice for analyzing tweets in this request.

5.2 Characterizing Data

One of the first questions to ask at what time deciding on the right tool for the job is: "what does my data seem like?" If your data has a very exacting schema, and it doesn't turn sideways from that schema, maybe you should just be using a relational database. MySQL is just as free as Hive, and very effectual for commerce with well-structured data. However, as you start to try to explore data with less structure or with very high volume, systems like MySQL[18] become less useful, and it may become crucial to move out of the relational earth.

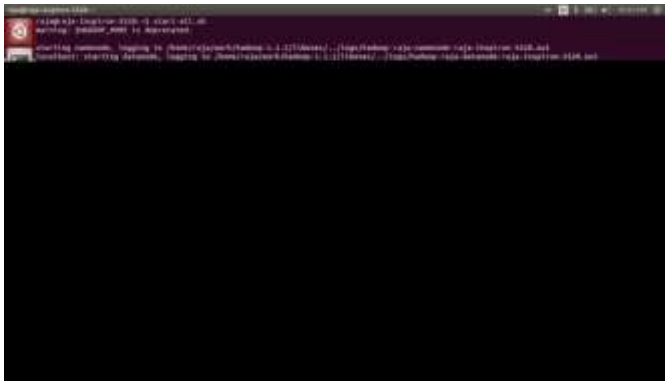

Figure 4: Start the Hadoop

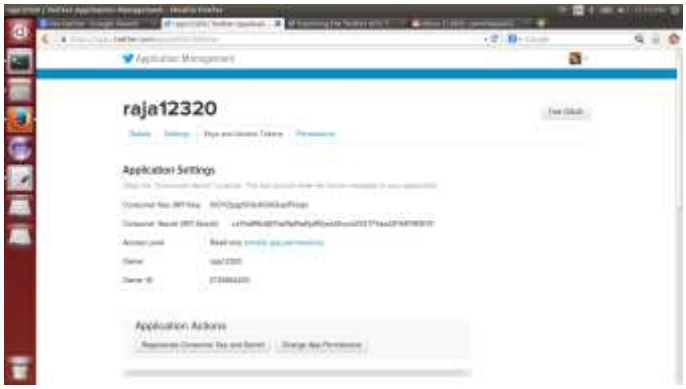

Figure 6: Setting the API-Keys

Unstructured, semi-structured, and poly-structured are all terms for data that doesn't fit well into the relational model. This is data like JSON, XML, RDF, or other sorts of data with a schema that may vary from record to record. What do we do with this data? Here's where Hive shines. Hive is tremendously effective for dealing with data that doesn't quite fit into the relational bucket, because it can procedure complex, nested types natively. Hive avoids the need for intricate transformation that might be otherwise necessary to handle this sort of data in a traditional relational system[10]. Hive can also gracefully handle records that don't strictly conform to a table's schema. For example, if some columns are absent from a particular record, Hive can deal with the record by treating missing columns as NULLs.

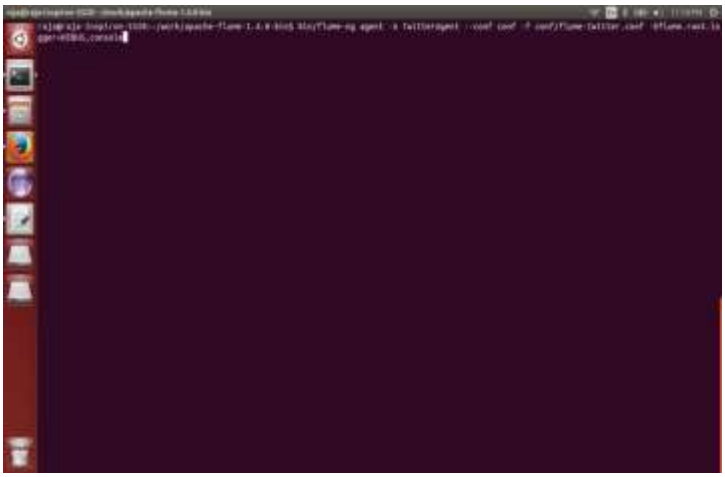

Figure 7: Start the flume command

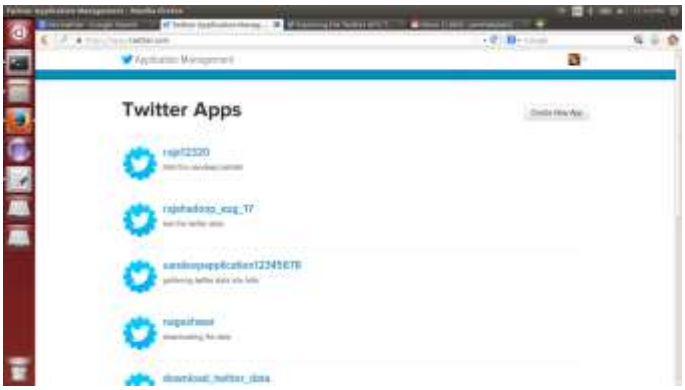

Figure 5: Create the Twitter Application

Figure 8: Stream the data from twitter by using Apache Flume

In the Twitter analysis example, It laden raw tweets into HDFS. Using the Twitter Streaming API, tweets are represented as JSON blobs.

It's fairly easy to see that there is a good bit of difficulty to this data structure. Since JSON can contain nested data structures,

it becomes very hard to force JSON data into a standard relational schema. Processing JSON data in a relational database would likely necessitate significant transformation, making the job much more cumbersome.

Looking at this particular bit of JSON, there are some very attractive fields: At the very top, there is a re-tweeted_status object, whose existence indicates that this tweet was retweeted by another user. If the tweet was not a retweet, you would not have a re-tweeted_status object at all. Tweets also contain an entities element, which is a nested structure. It contains three arrays, the elements of which are all nested structures in their own right, as can be seen in the hashtags array, which has two entries. How do you deal with a record like this in Hive?

5.3 Complex Data Structures

Hive has native maintain for a set of data structures that normally would either not exist in a relational database, or would necessitate definition of custom types. There are all the usual players: integers, strings, floats, and the like, but the interesting ones are the more exotic maps, arrays, and structs. Maps and arrays work in a fairly intuitive way, similar to how they work in many scripting languages:

Figure 9: Unstructured data after getting from the flume

Structs are a little more complex, since they are random structures, and a struct field can be queried much like an instance changeable in a Java class:

To store the data for a tweet, arrays and structs will be crucial.

By comparing the JSON objects from the tweet with the columns in the table, It can see how the JSON objects are mapped to Hive columns. Looking at the entities column, we can see what a mainly complex column strength look line:

```
1 entities STRUCT&lt;
2 urls:ARRAY&gt;,
3 user_mentions:ARRAY&gt;,
4 hashtags:ARRAY&gt;&gt;
```

entities is a struct which contains three arrays, and each individual array stores elements which are also structs. If we wanted to query the screen names of the first mentioned user from each tweet, It could write a query like this:

```
SELECT entities.user_mentions[0].screen_name FROM tweets;
```

If the user_mentions array is empty, Hive will just return NULL for that record.

The PARTITIONED BY section utilizes a feature of Hive called partitioning, which allows tables to be split up in different directories. By structure queries that involve the partitioning column, Hive can decide that certain directories cannot perhaps contain results for a query. Partitioning allows Hive to skip the processing of entire directories at query time, which can get better query presentation considerably.

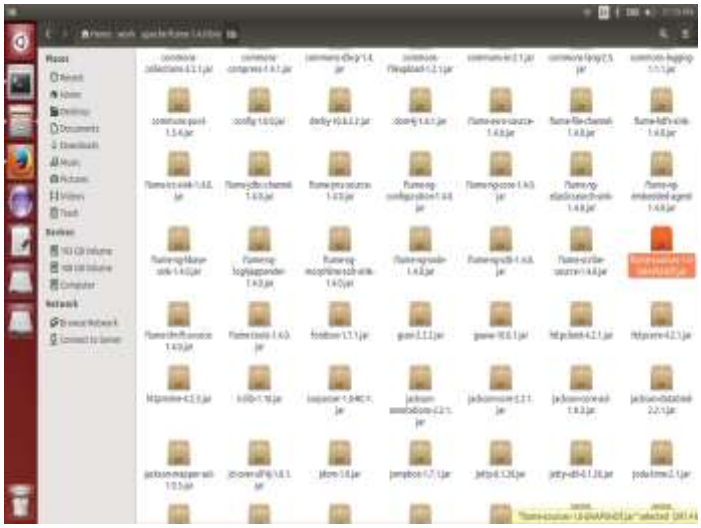

Figure 10: Creating the Jars of the flume and unstructured to structured

The LOCATION section is a obligation when by means of EXTERNAL tables. By non-payment, data for tables is stored in a directory located at /user/hive/warehouse/

However, EXTERNAL tables can identify an swap over location where the table data resides, which works nicely if Flume is being used to place data in a prearranged location. EXTERNAL tables also be different from regular Hive tables, in that the table data will not be uncomplicated if the EXTERNAL table is dropped.

The ROW FORMAT section is the most important one for this table. In simple datasets, the format will likely be DELIMITED, and It can specify the characters that terminate fields and records, if the defaults are not appropriate. However, for the tweets table, It specified a SERDE.

5.4 Serializers and Deserializers

In Hive, SerDe is an short form for Serializer and Deserializer, and is an interface used by Hive to conclude how to process a record. Serializers and Deserializers operate in opposite ways. The Deserializer interface takes a string or binary representation of a record, and translates it into a Java object that Hive can manipulate. The Serializer, on the other hand, will take a Java object that Hive has been working with, and turn it into something that Hive can write to HDFS. Commonly, Deserializers are used at query time to execute SELECT statements, and Serializers are used when writing data, such as from side to side an INSERT-SELECT statement. In the Twitter analysis[6] example, we wrote aJSONSerDe, which can be used to transform a JSON record into incredible that Hive can development.

5.5 Putting It All Together

By utilizing the SerDe interface, It can instruct Hive to understand data according to its intrinsic structure (or lack thereof). Since a SerDe is just a property of a Hive table, rather than the data, itself, They can also swap out SerDes as our data evolves. That suppleness allows us to choose the right tools for the job at hand, and to understand data in different ways. It makes Hive a spectacular choice for getting quick access to data of all types.

In the first post in this series, It could use Hive to find influential users. Let's look at some other queries It might want to write.

Geographic distributions of users can be attractive to look at. unluckily, the data they got from Twitter does not hold much of the geographic information necessary to plot really accurate locations for users, but it can use time zones to get a sense of where in the planet the users are. It can ask a question like, "Which time zones are the the majority active per day?"

Interestingly, more users are tweeting about the selected terms on the east coast, than the west coast. Europe also seems to be pretty paying attention in big data.

It can also formulate more complex queries to ask questions like "Which were the most ordinary hash tags?"

Not astonishingly, several of the terms that searched for when was collecting data show up here. The first term that shows up, which didn't search for, is job, followed by jobs. Cloudera's hiring, by the way. You may also notice the use of some non-standard SQL constructs, like LATERAL VIEW and EXPLODE. Lateral views are used when using functions like EXPLODE, which may manufacture more than one row of output for each row of input.

5.6 One Thing to Watch Out For…

If it looks like a bend, and it sounds like a duck, then it must be a duck, right. For users who are new to Hive, do not error Hive for a relational database. Hive looks a lot like a database, and you can interact with it very much like a database, but it should not be treated as such. Any query run in Hive is in fact executed as a sequence of MapReduce[4] jobs, which brings with it all of the performance implications of having to start up numerous JVMs[19].

Figure 11: Unstructured data will be formatted into Structured data formatting code

This income that all queries will have to pay a permanent setup cost, which will result in deprived presentation when organization lightweight queries. This fact makes Hive particularly nice for executing batch workloads. Like MapReduce[4], Hive shines brightest when working with massive data sets. However, it is important to realize that Hive queries may not have a response time that can be measured interactive, and Hive will likely not serve as a replacement for a traditional analytical database.

6. Future work

It proposed an approach which considers are done the Unstructured data into Structured data. It can be the theater cluster-setup with all the nodes working separately. Data will be processing very high speed. These Unstructured data Storing Problem solving it using cluster-setup. It can also implement Cloudera and Ambari.

6.1 Conclusion

As made per discussed some of the benefits and trade-offs of using Hive, and seen how to build a SerDe[12] to process JSON data, without any training of the data. By using the powerful SerDe border with Hive[5], It can process data that has a looser structure than would be possible in a relational database. This enables us to query and analyze traditional structured data, as well as semi- and even unstructured data. After this Unstructured data will be formatted into Structured data.

References

[1] Sachchidanand Singh, Nirmala Singh, “Big Data Analytics”, International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India.

[2] A. Bialecki, M. Cafarella, D. Cutting, and O. OSMalley, "Hadoop: A Framework for Running Applications on Large Clusters Built of Commodity Hardware," Wiki at http://lcene. apache. orglhop,

[3] T. White, "The Hadoop Distributed Filesystem," Hadoop: The Definitive Guide, pp. 41-73, GravensteinHighwaNorth, Sebastopol: O’Reilly Media, Inc., 2010.J.

[4] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Communications of the ACM, Vol. 51, Iss. 1, pp. 107-113, January 2008.

[5] (Online Resource) Hive (Available on:http://hive.apache.org/).

[6] A. Pak and P. Paroubek, “Twitter as a corpus for sentimen analysis and opinion mining,” in Proceedings of LREC, vol. 2010.

[7] http://www.webopedia.com/TERM/U/unstructured_data.html.

[8] Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). RDBMStoNoSQL:Reviewing Some Next-Generation Non-Relational Database_s . *International Journal of Advanced Engineering Science and Technologies*, *11*(1), 15-30.

[9] (Online Resource) http://structureddata.org/

[10] Rabi Prasad Padhy, Manas Ranjan Patra and Suresh Chanadra , "RDBMS to NoSQL: Reviewing Some Next-Generation Non-re.

[11] Rick Cattell : Scalable SQL and NoSQL Data Stores. http://cattell.net/datastores/Datastores.pdf

[12] https://cwiki.apache.org/confluence/display/Hive/SerDe (online

[13] http://jsonlint.com/(online Resource)

[14] T. White, "The Hadoop Distributed Filesystem," Hadoop: The Definitive Guide, pp. 41-73, GravensteinHighwaNorth, Sebastopol: O'Reilly Media, Inc.,

Online Resources:

[15] http://www.cloudera.com/content/cloudera/en/search.html.hbas

[16] http://hortonworks.com/

[17] http://flume.apache.org/

[18] http://www.tutorialspoint.com/mysql/

[19] http://www.artima.com/insidejvm/ed2/

[20] https://hadoop.apache.org/

[21] https://www.mapr.com/products/apache-hadoop

[22] http://www-01.ibm.com/software/data/infosphere/hadoop/

Online Videos:

[23] https://www.youtube.com/watch?v=xWgdny19yQ4

[24] https://www.youtube.com/watch?v=zgu91lfvmJ8

[25] https://www.youtube.com/watch?v=NvrpuBAMddw

[26] https://www.youtube.com/watch?v=ht3dNvdNDzI

[27] https://www.youtube.com/watch?v=Pn7Sp2-hUXE

[28] https://www.youtube.com/watch?v=ri3dqc-rt5s

[29] https://www.youtube.com/watch?v=-fwhVrkH0v0

[30] https://www.youtube.com/watch?v=uEjeMb81cQ0

Author's Profile

Bala Krishna.K was born in Andhra Pradesh, India. He received the B.Tech Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapur branch, India in 2013 and M.Tech Degree in also same branch and University. His research interests are in the area of Big Data Analytics, Dataminig and Cloud Computing.

Smt. S.Jessica Saritha M.tech.,[Ph.D] is currently working as an Assistant Professor in Department of CSE JNTUA College of Engineering, Pulivendula ,Andhra Pradesh India Her Research interests are Data mining and distributed computing.

Penchalaiah.C was born in AndhraPradesh, India. He received the B.Tech Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapur branch, India in 2011 and M.Tech Degree in also same branch and

University. His research interests are in the area of SemanticWeb and Big Data Analytics.