DATA SCRAPING FROM VARIOUS DATA RESOURCES

[1] Moparthy. RajyaLashmi, [2]Nimmala. Mrudula, [3]Makinaboina. Divya, [4]Manukonda. Sri Sai Lakshmi
[1]Bachelor of Technology, [2] Bachelor of Technology, [3] Bachelor of Technology,[4] Bachelor of Technology
[1]Computer Science and Engineering,
[1]Vasireddy Venkatadri Institute of Technology, Guntur, India

Abstract : Big data contains large data sets that are so voluminous and complex that traditional data processing applications are inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating. The first five important dimensions to big data are characterized as: Volume, Variety, Velocity, veracity and volatile. Big data is generated from multiple sources like social networking sites, sensors, pdf documents. As the big data generated from multiple sources, it is heterogeneous in nature. Big data includes structured, unstructured and semi-structured data. For analyzing the structured data we have Traditional Data Base Management (TDBM). Traditional data bases are inadequate to maintain unstructured and semi-structured. To manage semi-structured data NoSql (Structured Query Language) data bases can be used. Before performing analytics the data should be captured from multiple resources, which is one of the major challenge. Data Scraping is a technique that extracts desired data from big data sources. Big data can be abstracted or scraped by using JSOUP (Third party libraries in Java), web crawlers. The scraped data can automatically loaded to hive engine with the help of flume tool. So in this project we are going to capture big data from various big data sources and then perform analytics on the data which have been scraped from multiple sources.

IndexTerms - **TDBM, No Sql, Big Data Analysis, Data Scraping, JSOUP, flume tool.**

I. INTRODUCTION

Data scraping is a process of extracting data from the internet through various methods. The internet is a universally ccessible resource for millions of people. As the usage of internet has commonly increased in everywhere there is highly growth in competition between the organizations in their business. Data scraping helps in automation of Data through the use in various techniques. The usage of Data scraping have been in many fields and beneficial for weather data monitoring, website change detection, research, web data integration, contact scraping and online price comparison.. In this project we will go through the tools and techniques used in scraping and its impact on the social networks. Through Data scraping services unstructured data are converted into structured data which can be stored and verified in a centralized data bank. The aim is to collect, store and analyze data. The data analysis is very much needed in a society to extract any information and transforming it into a format helpful to interpret. Thus, Data scraping services have a direct influence on the outcome which is needed from the data collection. Data extraction is the process of transforming the useful content on websites into valuable business assets. There are several Data extracting software that has emerged in the market which helps to address this problem. The software aids in extracting structured content from a web page and exposes the required services as APIs and makes it usable for further processing. It is necessary to know the available technologies in the market today. The available technologies that are related may be in different languages written such as java, python, php etc. The benefits of this are beyond the limitations of the users. Since there is rise in new online business through internet this has an adverse effect on the consumers as well. Online marketing analyst use Data scraping methods to grab some information from other competitors such as emails, targeted keywords and links and also traffic source. The scraping techniques are used for personal as well as commercial usage. All the techniques available have its own pros and cons to overcome this here is need to have a clear idea on the usage of these techniques in social networking.

II. BIG DATA ANALYSIS

'BIG DATA' IS ALSO A DATA BUT WITH A HUGE SIZE. 'BIG DATA' IS A TERM USED TO DESCRIBE COLLECTION OF DATA THAT IS HUGE IN SIZE AND YET GROWING EXPONENTIALLY WITH TIME. IN SHORT, SUCH A DATA IS SO LARGE AND COMPLEX THAT NONE OF THE TRADITIONAL DATA MANAGEMENT TOOLS ARE ABLE TO STORE IT OR PROCESS IT EFFICIENTLY.

Examples of 'Big Data'-
1. The New York Stock Exchange generates about one terabyte of new trade data per day.
2. Statistic shows that 500+terabytes of new data gets ingested into the databases of social media site Facebook, every day. This data is mainly generated in terms of photo and video uploads, message exchanges, putting comments etc.
3. Single Jet engine can generate 10+terabytes of data in 30 minutes of a flight time. With many thousand flights per day, generation of data reaches up to many Petabytes.

Categories Of 'Big Data':
Big data' could be found in three forms:
1. Structured
2. Unstructured
3. Semi-structured

Structured
Any data that can be stored, accessed and processed in the form of fixed format is termed as a 'structured' data. Over
The period of time, talent in computer science have achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, now days, we are foreseeing issues when size of such data grows to a huge extent, typical sizes are being in the rage of multiple zetta byte.

Unstructured

Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, unstructured data poses multiple challenges in terms of its processing for deriving value out of it. Typical example of unstructured data is, a heterogeneous data source containing a combination of simple text files, images, videos etc. Now a day organizations have wealth of data available with them but unfortunately they don't know how to derive value out of it since this data is in its raw form or unstructured format.

Semi-structured

Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in XML file.

Characteristics of 'Big Data':

(i)Volume – The name 'Big Data' itself is related to a size which is enormous. Size of data plays very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon volume of data. Hence, 'Volume' is one characteristic which needs to be considered while dealing with 'Big Data'.

(ii)Variety – The next aspect of 'Big Data' is its variety. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Now days, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. is also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

(iii)Velocity – The term 'velocity' refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data. Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks and social media sites, sensors, Mobile devices, etc. The flow of data is massive and continuous.

(iv)Variability – This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

III. Disadvantages of Traditional Database Systems

Traditional data systems, such as relational databases and data warehouses, have been the primary way businesses and organizations have stored and analyzed their data for the past 30 to 40 years. Although other data stores and technologies exist, the major percentage of business data can be found in these traditional systems. Traditional systems are designed from the ground up to work with data that has primarily been structured data. Characteristics of structured data include the following:

1. Clearly defined fields organized in records. Records are usually stored in tables. Fields have names, and relationships are defined between different fields.

2. Schema-on-write: that requires data be validated against a schema before it can be written to disk. A significant amount of requirements analysis, design, and effort up front can be involved in putting the data in clearly defined structured formats. This can increase the time before business value can be realized from the data.

Every year organizations need to store more and more detailed information for longer periods of time. Increased regulation in areas such as health and finance are significantly increasing storage volumes. Expensive shared storage systems often store this data because of the critical nature of the information. Shared storage arrays provide features such as striping (for performance) and mirroring (for availability). Managing the volume and cost of this data growth within these traditional systems is usually a stress point for IT organizations. Examples of data often stored in structured form include Enterprise Resource Planning (ERP), Customer Resource Management (CRM), financial, retail, and customer information.

Difficulty Of Traditional Systems Towards Big Data

The reason traditional systems have a problem with big data is that they were not designed for it.

Problem—Schema-On-Write: Traditional systems are schema-on-write. Schema-on-write requires the data to be validated when it is written. This means that a lot of work must be done before new data sources can be analyzed. If you look at data structures from social media, they change on a regular basis. The schema-on-write environment is too slow and rigid to deal with the dynamics of semi structured and unstructured data environments that are changing over a period of time. The other problem with unstructured data is that traditional systems usually use Large Object Byte (LOB) types to handle unstructured data, which is often very inconvenient and difficult to work with.

Solution—Schema-On-Read: Hadoop systems are schema-on-read, which means any data can be written to the storage system immediately. Data are not validated until they are read. This enables Hadoop systems to load any type of data and begin analyzing it quickly. Hadoop systems have extremely short business latency compared to traditional systems. Traditional systems require schema-on-write, which was designed more than 50 years ago. A lot of companies need real-time processing of data and customer models generated in hours or days versus weeks or months. The Internet of Things (IoT) is accelerating the data streams coming from different types of devices and physical objects, and digital personalization is accelerating the need to be able to make real-time decisions. Schema-on-read gives Hadoop a tremendous advantage over traditional systems in an area that matters most, that of being able to analyze the data faster to make business decisions. When working with complex data structures that are semi-structured or unstructured, schema-on-read enables data to be accessed much faster than schema-on-write systems.

Problem—Cost of Storage: Traditional systems use shared storage. As organizations start to ingest larger volumes of data, hared storage is cost prohibitive.

Solution—Local Storage: Hadoop can use the Hadoop Distributed File System (HDFS), a distributed file system that beverages local disks on commodity servers. Shared storage is about $1.20/GB, whereas local storage is about $.04/GB. Hadoop's HDFS creates three replicas by default for high availability. So at 12 cents per GB, it is still a fraction of the cost of traditional shared storage.

Problem—Cost of Proprietary Hardware: Large proprietary hardware solutions can be cost prohibitive when deployed to process extremely large volumes of data. Organizations are spending millions of dollars in hardware and software licensing costs while supporting large data environments. Organizations are often growing their hardware in million dollar increments to handle the increasing data. New technology in traditional vendor systems that can grow to petabyte scale and good performance are extremely expensive.

Solution—Commodity Hardware: It is possible to build a high-performance super-computer environment using Hadoop. One customer was looking at a proprietary hardware vendor for a solution. The hardware vendor's solution was $1.2 million in hardware costs and $3 million in software licensing. The Hadoop solution for the same processing power was $400,000 for hardware, the software was free, and the support costs were included. Because data volumes would be constantly increasing, the proprietary solution would have grown in $500k and $1 million dollar increments, whereas the Hadoop solution would grow in $10,000 and $100,000 increments. When you look at any traditional proprietary solution, it is full of extremely

Problem—Complexity: When you look at any traditional proprietary complex silos of system administrators, DBAs,application server teams, storage teams, and network teams. Often there is one DBA for every 40 to 50 database servers. Anyone running traditional systems knows that complex systems fail in complex ways.

Solution—Simplicity**:** Because Hadoop uses commodity hardware and follows the "shared nothing" architecture, it is a platform that one person can understand very easily. Numerous organizations running Hadoop have one administrator for every 1,000 data nodes. With commodity hardware, one person can understand the entire technology stack.

Problem—Causation: Because data is so expensive to store in traditional systems, data is filtered and aggregated, and large volumes are thrown out because of the cost of storage. Minimizing the data to be analyzed reduces the accuracy and confidence of the results. Not only are accuracy and confidence to the resulting data affected, but it also limits an organization's ability to identify business opportunities. Atomic data can yield more insights into the data than aggregated data.

Solution—Correlation: Because of the relatively low cost of storage of Hadoop, the detailed records are stored in Hadoop's storage system HDFS. Traditional data can then be analyzed with nontraditional data in Hadoop to find correlation points that can provide much higher accuracy of data analysis. We are moving to a world of correlation

because the accuracy and confidence of the results are factors higher than traditional systems. Organizations are seeing big data as transformational. Companies building predictive models for their customers would spend weeks or months building new profiles. Now these same companies are building new profiles and models in a few days. One company would have a data load take 20 hours to complete, which is not ideal. They went to Hadoop and the time for the data load went from 20 hours to 3 hours.

Problem—Bringing Data to the Programs: In relational databases and data warehouses, data are loaded from shared storage elsewhere in the datacenter. The data must go over wires and through switches that have bandwidth limitations before programs can process the data. For many types of analytics that process 10s, 100s, and 1000s of

terabytes, the capability of the computational side to process data greatly exceeds the storage bandwidth available.

Solution—Bringing Programs to the Data: With Hadoop, the programs are moved to where the data is. Hadoop data is spread across all the disks on the local servers that make up the Hadoop cluster, often in 64MB or 128MB block increments. Individual programs, one for every block, runs in parallel (up to the number of available map lots, more on this later) across the cluster, delivering a very high level of parallelization and Input/Output Operations per second (IOPS). This means Hadoop systems can process extremely large volumes of data much faster than traditional systems and at a fraction of the cost because of the architecture model. Moving the programs (small component) to the data (large component) is an architecture that supports the extremely fast processing of large volumes of data.

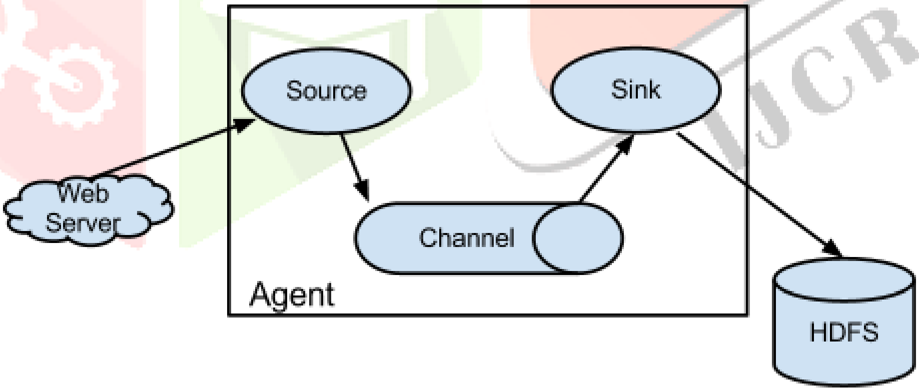
3.1 Mechanism

For data scraping we need to visit a web site. We have to give the url specified and then we have to execute the code. The data obtained is comma separated saved file. To get the desired data, we need to preprocess the obtained data for our convenience. This was done by the flume tool and the data will be stored in hive engine.

3.2 Softwares used

Flume tool

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application. It is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log data, events (etc...) from various web serves to a centralized data store.



Jsoup

Jsoup is *a* java html parser. It is a java library that is used to parse HTML document. Jsoup provides api to extract and manipulate data from URL or HTML file. It uses DOM, CSS and Jquery-like methods for extracting and manipulating file.

Code to scrape the data

```
import java.io.IOException;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
public class FirstJsoupExample{
public static void main( String[] args ) throws IOException{
Document doc = Jsoup.connect("url
here").get();
String title = doc.title();
System.out.println("title is: " + title);
}
}
```

Hive Engine

The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL. In simple words, people who aren't strong in JAVA but still wants to run map-reduce programs can use HIVE , as HIVE is just a SQL based mechanism. If one knows the basic SQL queries, he can work with HIVE.

IV. RESULTS AND DISCUSSION

4.1 Conclusion

So, data scraping can be done to produce the effective results for acquiring the better results and to maintain the time efficacy. So in this project we are going to capture big data from various big data sources and then perform analytics on the data which have been scraped from multiple sources. Even though techniques useful there are some challenges faced that may be such as the high volume of Data scraping can cause regulatory damage to the pages. Scale of measure the scales of the Data scraper can differ with the units of measure of the source file thus making it somewhat hard for the interpretation of the data. The Level of source complexity in case if the information being extracted is very complicated Data scraping will also be paralyzed.

IV. ACKNOWLEDGMENT

SQL	- Standard Query Language.
HDFS	- Hadoop Distributed File System.
TDMS	- Traditional Data Base Systems.

REFERENCES

[1] https://www.ibm.com/analytics/hadoop/bigdata-analytics
[2] http://www.cisstat.com/BigData/CIS-BigData_04_Eng%20%20ISTAT%20Web%20scraping-HICP.pdf
[3] https://www.javatpoint.com/jsoup-tutorial
[4] http://www.pearsonitcertification.com/articles/article.aspx?p=2427073&seqNum=2
[5] http://www.kdu.ac.lk/proceedings/irc2015/2015/com-020.pdf
[6] https://digitalcommons.georgiasouthern.edu/cgi/viewcontent.cgi?article=1227&context=honors-theses
[7]https://www.sciencedirect.com/science/article/pii/S1877050914011181/pdf?md5=7880f2f228cf382878ec3ae4d530375e&pid=1-s2.0-S1877050914011181-main.pdf&_valck=1