www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 6 Issue 2 Feb. 2017, Page No. 20379-20382
Index Copernicus Value (2015): 58.10, DOI: 10.18535/ijecs/v6i2.38

A Unified Bigdata Analysis Platform Using Hadoop Technology

Suma M.R, Sanjana P, Bharath Kumar S
Assistant Professor, Dayananda Sagar Academy of Technology And Management, Bengaluru, Karnataka, *India*
sumamrvp@gmail.com
B.E, Computer Science and Engineering, R N S Institute of Technology,
Bengaluru, Karnataka, *India*
sanjanaprasanna088@gmail.com
[B.E], Electronics and Communication Engineering, R N S Institute of Technology,
Bengaluru, Karnataka, *India*
bharathtm12@gmail.com

Abstract- Big data is prevalent in both industry and scientific research applications where the data is generated with high volume and velocity it is difficult to process using on-hand database management tools or traditional data processing applications. Some techniques have been developed in recent years for processing large object data on cloud, such as Cloud Analytics. However, these techniques do not provide efficient support for parallel processing and cluster technology.

Big data platforms often need to support emerging Data sources and applications while accommodating existing ones. Since different data and applications have varying requirements, multiple types of data stores (e.g. file-based and object-based), frequently co-exist in the same solution today without proper integration. Hence cross-store data access, key to effective data analytics, cannot be achieved without laborious application Re-programming, prohibitively expensive data migration, and/or costly maintenance of multiple data copies.

We address this vital issue by introducing a first unified big data platform over heterogeneous storage. In particular, we present a prototype joining Apache Hadoop Map Reduce and Flume technology. A retail data analysis application using data of real Twitter application is employed to test and showcase our prototype. We have found that our prototype achieves 50% data capacity savings, eliminates data migration overhead, and offers stronger reliability and enterprise support. Through our case study, we have learned important theoretical lessons concerning performance and reliability, as well as practical ones related to platform configuration. We have also identified several potentially high-impact research directions.

I. INTRODUCTION

Distributed computing is a field of computer science that studies distributed systems. Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

In order, to store and process huge amount of data we use Apache Hadoop framework in our platform. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Hadoop was developed by Dough Cutting, to store as well as process huge data. Hadoop core mainly consists of two parts they are: HDFS (Hadoop distributed file system) for storage of data. MapReduce programming model, for processing large data set framework that works on distributed computing.

II. PROBLEM STATEMENT

Big Data has become one of the most transformative IT developments of the decade with unprecedented amounts of granular data being generated and collected for analysis across a wide range of sources. The problem is, this abundance of data can also be a double-edged sword.

Despite its potential – and the abundance of tools, platforms and solutions that promise to make it useful – the ability to find and act on truly valuable insights still eludes many companies.

Even with massive computing power and virtually infinite cloud storage capacity, most businesses still have a very slow, difficult time when it comes to uncovering legitimate insights to use to their advantage. The root of the problem is that, in most IT organizations, the integration and data management functions have historically operated in entirely separate silos, despite their intimate dependency.

In order to realize the full potential of the Big Data promise and succeed in a data-driven future, it’s critical that integration and data management be brought together in a *data-centric* (rather than app-centric) approach. It calls for a new model – Data Platform as a Service or DPaaS – that leverages the power of the cloud to fully unify the integration and data management functions into a single, cohesive system.

DOI: 10.18535/ijecs/v6i2.38

Unified Data Analysis Platform (UDAP) helps enterprises to get actionable insights from their data faster, thereby, significantly reducing the time taken to transform that data into dollars. UDAP can make 'Dark Data' extinct and reduce the span of the entire analysis life cycle by turbo charging the existing data infrastructure with the power to consolidate internal and external unstructured data sources, and unifies them with the existing datasets.

III. SYSTEM DESIGN

Huge amount of data can come through various sources it can be from structured sources (data from RDBMS which is already stored in table form) or unstructured sources like the data coming in from Twitter (live data streaming).Then through various data connector engines like Flume and sqoop data transfer takes place from various to HDFS(Hadoop distributed file structure) which is then polished and processed through MapReduce programming model, consisting of various prebuilt programming packages or API's ,this makes up the core of our platform.

And finally, we built the charts after the data is being processed using tools like Tableau, Am charts even write Java programs (jsps) to build business Analytics.

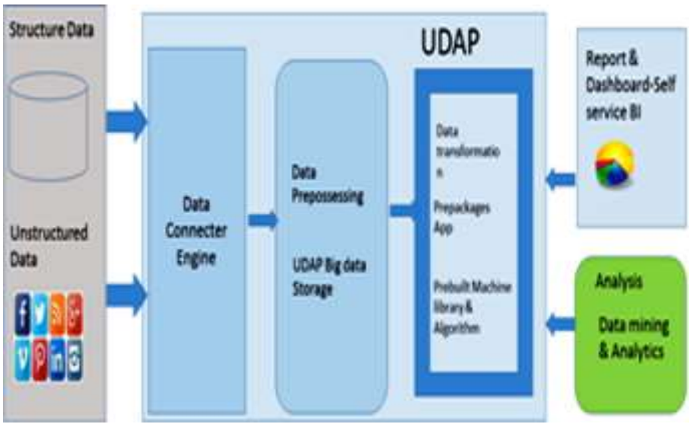

Figure 1: UDAP design architecture

Requirements

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Floppy Drive	: 1.44 Mb.
Monitor	: 15 VGA Color.
Mouse	: Logitech.
Ram	: 8 GB
Operating System	: Windows 7/UBUNTU.
Coding Language	: Java 1.7 and Hadoop 2.0.1
IDE	: Eclipse indigo
Database	: HDFS

IV. IMPLEMENTATION

Data Streaming from Twitter can be done in two ways:

1. Unstructured data extraction
2. Structured data extraction

Unstructured data extraction can be done using Apache Flume. Unstructured data helps to extract the live streaming data Apache Flume is one way to bring data into HDFS. Apache Flume is a distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of log data.

At the most basic level, flume enables applications to collect data from its origin and send it to a resisting location, such as HDFS. At a slightly more detailed level, flume achieves this goal by defining dataflow consisting of three primary structures: source channel sink

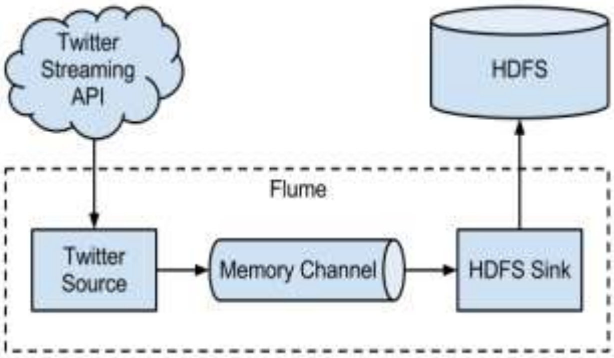

Figure 2: Analyzing twitter data with Apache Hadoop

Source: A source is the part of flume that connects to a source of data, and starts the data along its journey through a flume dataflow. A source process events and moves them along by sending them into a channel. Sources operates by gathering discrete pieces of data, translating the data into individual events, and then using the channel to process the events one at a time, or as a batch.

Channel: Channels act as a pathway between the sources and sinks. Events are added to channels by sources, and later removed from the channels by sinks. Flume dataflow can actually support multiple channels, which enables more complicated dataflow, such as fanning out for replication purposes.

Sinks: The final piece of flume dataflow is the sink. Sinks take events and send them to a resisting location or forward them on to another agent. In the twitter example, the utilized HDFS sink, this writes events to a configured location in HDFS.

Structured data extraction can be done using twitter 4j API .Structured data helps to extract already tweeted data. To connect and extract data from twitter using twitter APIs, the first step is to create a twitter app and get it approved. Pre-requisite to this step is to have a twitter username and password. For creating an application we need to connect https://apps.twitter.com/.With the twitter username and password, there would be a button to create a new application. After creating a new application, one need to download twitterr4j jar and import it in eclipse and finally use the authentication details in the program.Twitter4j is an unofficial API for accessing twitter API data, but it is simple yet very powerful. Then we write a map reduce program in java to implement the data extraction.

APACHE HADOOP MAP REDUCE MODEL FOR ANALYTICS:

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly- available service on top of a cluster of computers, each of which may be prone to failures. Hadoop was developed by Dough Cutting, to store as well as process huge data. Hadoop core mainly consists of two parts they are:

- HDFS (Hadoop distributed file system) for storage of data.(figure 3)
- MapReduce programming model, for processing large data sets(figure 4)

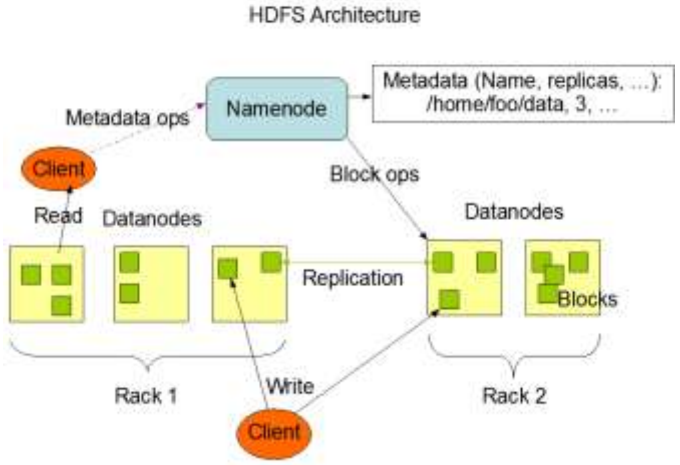

Figure 3: HDFS architecture

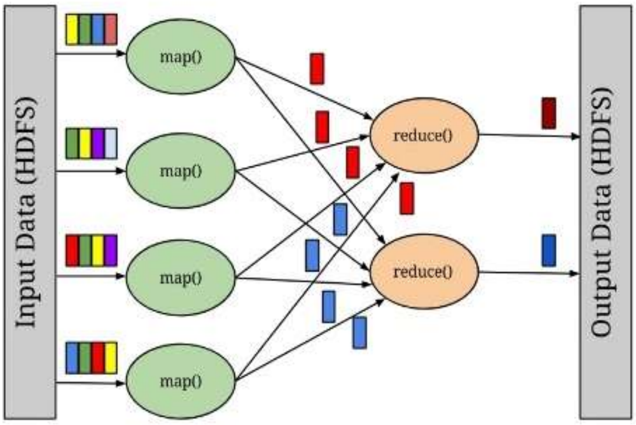

Figure 4: MapReduce program model

THE MAP REDUCE ALGORITHM

The job execution starts when the client program submit to the Job Tracker a job configuration, which specifies the map, combine and reduce function, as well as the input and output path of data. The Job Tracker will first determine the number of splits (each split is configurable, ~16-64MB) from the input path, and select some Task Tracker based on their network proximity to the data sources, then the Job Tracker send the task requests to those selected Task Trackers.

Each Task Tracker will start the map phase processing by extracting the input data from the splits. For each record parsed by the “Input Format”, it invokes the user provided “map” function, which emits a number of key/value pair in the memory buffer. A periodic wakeup process will sort the memory buffer into different reducer node by invoke the “combine” function. The key/value pairs are sorted into one of the R local files (suppose there are R reducer nodes).

When the map task completes (all splits are done), the Task Tracker will notify the Job Tracker. When all the Task Trackers are done, the Job Tracker will notify the selected Task Trackers for the reduce phase. Each Task Tracker will read the region files remotely. It sorts the key/value pairs and for each key, it invokes the “reduce” function, which collects the key/aggregated Value into the output file (one per reducer node).

Map/Reduce framework is resilient to crash of any components. The Job Tracker keep tracks of the progress of each phase and periodically ping the Task Tracker for their health status. When any of the maps phase Task Tracker crashes, the Job Tracker will reassign the map task to a different Task Tracker node, which will rerun all the assigned splits. If the reduce phase Task Tracker crashes, the Job Tracker will re-run the reduce at a different Task Tracker.

After both phases completes, the Job Tracker will unblock the client program.

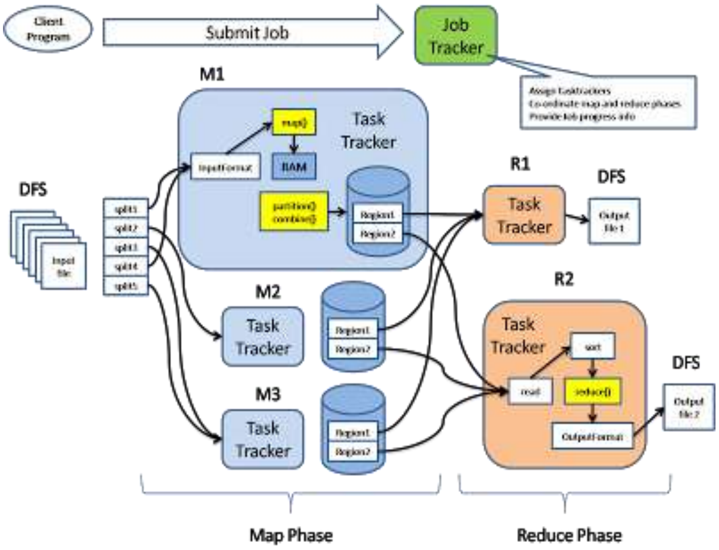

Figure 4: MapReduce architecture

WEB APPLICATIONS FOR DASHBOARD:

The user interface is implemented using Tomcat 6 server and java servlet programs (JSP).

Tomcat 6 consists of a nested hierarchy of components. Containers are components that can contain a collection of other components. The below diagram display how the Tomcat architecture looks, some of the components can be contained multiple times are denoted by a symbol that has multiple profiles, including Connector, Logger, Valve, Host and Context.

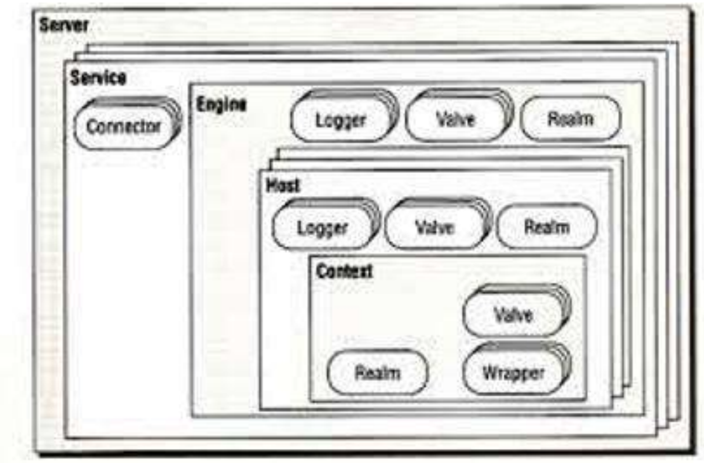

Figure 6: Apache Tomcat Architecture overview.

Java Servlet Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>. A Java Servlet Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Java Servlet Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI. Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files. JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.

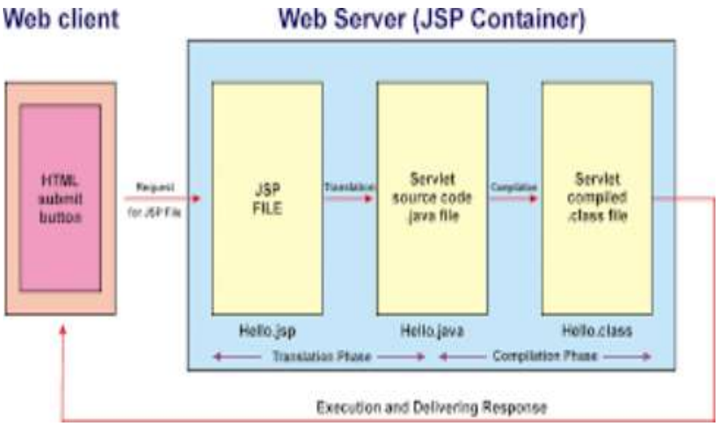

Figure 7: JSP architecture

RESULTS

Finally, after analyzing the data we obtained the results separately for structured as well as unstructured data. For structured data, we obtained the table in accordance to the query keyword typed. We, basically get the name, commented tweets, date and time of the tweeted data in the form of structured table. For unstructured data, the retail store Name as well as the category has to be given, in accordance to that live tweets can be extracted in the form of table, showing the username ,tweeted data and time. For the above data extracted, our platform produces pie chart depicting the number of tweets coming in for different retail stores.

V. CONCLUSION

We can conclude that, our Unified Data Analysis Platform is to support Business users to expedite the information discovery process and analyze business data.

It encompasses all the functional modules to process operational data as well as big data. It also provides point to point user interface, we report visualization, and predictive analytics using low cost commodity hardware.

Though, our Platform covers most of industry requirements, it will always open doors for future enhancements especially in structured data integration and data Visualization through various tools.

References

[1] Rue Zhang IBM Res. - Almaden, San Jose, CA, USA Hildebrand, D. ; Tiwari, “In unity there is strength: Showcasing a unified big data platform with MapReduce Over both object and file storage” Big Data (Big Data), 2014 IEEE International Conference on 27-30 Oct. 2014,pp no-960-966.

[2] M. Mesnier, G. R. Ganger, and E. Riedel, “Object-based storage,”Communications Magazine, IEEE, vol. 41, no. 8, pp. 84–90, 2003.

[3] M. Boehm, S. Tatikonda, B. Reinwald, P. Sen, Y. Tian,D. Burdick, and S. Vaithyanathan, “Hybrid parallelization strategies for large-scale machine learning in system ml,”*PVLDB*, vol. 7, no. 7, pp.553–564, 2014.

[4] E. R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan,J. E. Gonzalez, M. J. Franklin, M. I. Jordan, and T. Kraska,“MLI: an API for distributed machine learning,” in 2013IEEE 13th International Conference on Data Mining (ICDM),Dallas, TX, USA, December 7-10, 2013,2013, pp. 1187–1192.

[5] Gaurav Kumar, Rajpal Sharma,” Analytical Study of Sense Amplifier”, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X Volume 4, Issue 5, May 2014

[6] Jyoti hooda, Sarita ola, Manisha saini, “Design and Analysis of a low Power CMOS Sense Amplifier for Memory Application”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, April 2013.

[7] Parita Patel, Sameena Zafar and Hemant Soni, “Performance of Various Sense Amplifier Topologies in sub100nm Planar MOSFET Technology”,International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 3, Issue 2, March – April 2014 ISSN 2278-6856

[8] Swati Anand Dwivedi, “Low Power CMOS Design of an SRAM Cell with Sense Amplifier”,International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-6, February 2012.