IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 5, Ver. VI (Sep. - Oct. 2016), PP 42-47
www.iosrjournals.org

Hadoop based Self-Service Data Integration Platform

Sandeep H S[1], Kumar Sambhav[1]
[1]*Research & Development, Techieventures Technologies Pvt. Ltd., Bangalore, Karnataka, 560029, India*

Abstract: *The Data Integration platform is a Web based Application which allows Business users or users with minimal knowledge of Data Blending/Mining to integrate multiple master and transactional source systems to a staging area and data-marts to suffice the BI & Analytics needs of an organization. Data from heterogeneous sources in structured & semi-structured form are cleaned and aggregated depending upon specific business metrics and then stored into Hive/HBase (Databases of Hadoop ecosystem). Analytics algorithms can be further executed on these data-sets on demand. The Platform is capable of connecting to heterogeneous sources (RDBMS, NoSQL, Cloud Systems & Flat-file storages) from the web user Interface. Hadoop based ETL scripting tool, SQOOP and Flume has been used to connect to perform the Extraction, Transformation & Loading of huge amount of data from source systems to the Hive based Reporting Data-mart. Visualization platforms like Tableau or HTML based Visualization frameworks like Bootstrap/Chart.js connects to Hive to build reports/dashboards to enable data-driven decision making.*

Keywords: *Self-Service Data Integration, Hadoop based Analytics, Business Intelligence, SQOOP ETL, Flume ETL, Web-based Data Integration Platform.*

I. Introduction

In the present era analyzing enterprise data to derive insights on various facets of a business is of utmost importance. Often business organizations face challenges in doing the same because of the data sitting in multiple silos, which prevent business decision makers from obtaining a holistic view of the business. [1] The situation becomes more challenging when an organization acquires and merges with multiple entities because the Enterprise Resource Planning & other business systems of the new subsidiaries are often not compatible with the existing data systems. [2] Under such instances, organizations invest massively in the on-boarding & integration of these data systems. However, the process is effort intensive which delays the Business Intelligence and Analytics outcome and spikes the IT Infrastructure budget for Data Engineering as well.

We have a novel approach towards solving this problem which accelerates the process of integrating new data systems with the existing reporting data marts with minimal/negligent involvement of the Data Engineering team. The easy-to-use web user interface allows analysts to prepare data for the business reporting and dashboard building in no time. The application has been designed, developed & deployed on premise for a leading Supply Chain Solutions service provider of India. However, the future scope of this application is aimed towards a cloud based solution where it will be hosted on the cloud to connect to the data-sources residing within an organizations' firewall, hence assuring the data confidentiality and security.

II. Existing Business Intelligence Reporting Architecture Overview

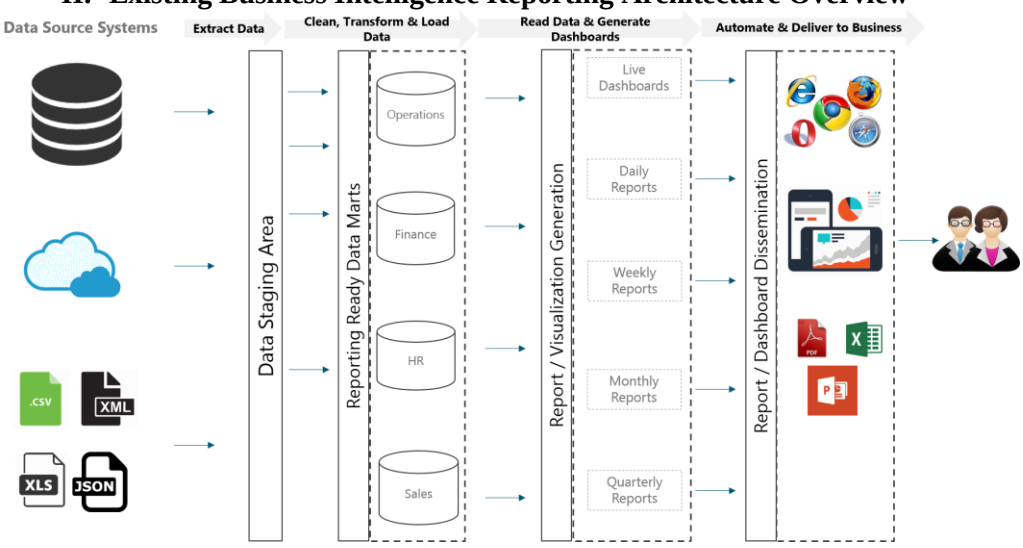

Fig. 1 Existing BI setup

The existing data-to-visualization setup of our client is as represented in Fig. 1. This setup is aligned with the typical legacy Business Intelligence framework which involves substantial involvement of the Data Engineering team and is also not scalable to handle bulk semi-structured & unstructured data in the volume of Terabytes. The components are elaborated in the following sub-sections [2].

2.1 Data Staging Area

The Data Staging Area is temporary location where data from the source systems are extracted and dumped. A staging area is mainly required to manage the anomaly that may be caused due to the variation in timings. In other words, all required data must be available before data can be integrated into the Reporting Data-marts. Due to varying business cycles, data processing cycles, hardware and network resource limitations and geographical factors, it is not feasible to extract all the data from all source databases at exactly the same time.

2.2 Reporting Ready Data Marts

A data mart is the layer of the data warehouse environment that is used to provide access to the data to the end-users. It is a subset of the data warehouse and is ideally owned by a specific business function or team. Whereas data warehouses have an enterprise-wide depth, the information in data marts pertains to a single department. In some deployments, each department or business unit is considered the owner of its data mart including all the hardware, software and data. Data undergoes the required cleansing and transformation before loading it into the tables of the data marts. Reporting ready data mart tables' stores data at an aggregated or summarized level depending on the detail at which the business metrics or Key Performance Indicators (KPIs) needs to be represented. Summarization of the data also helps in optimizing performance of the reports and dashboards.

2.3 Report/Visualization Generation

Data is consumed in form of reports or dashboards, where the data is represented in graphical form for the ease of comprehension by the business users. Appealing, intuitive, interactive and insightful visualizations are of utmost need to enable data driven decision making. Proprietary or Open Source visualization platforms are often used by analysts to generate such reports or dashboards for the business users. The Visualization platforms are capable of connecting to the reporting data marts using compatible Open Database Connectivity (ODBC) drivers to read data and represent the same in form of graphs and charts.

2.4 Report/Dashboard Dissemination

Report or Dashboard dissemination involves automation of the periodic data refresh of the reports/dashboards and modes of delivering the updated versions to the business users. The refresh periods may be instantaneous, hourly, daily, weekly, bi-weekly, monthly, quarterly, etc. depending upon the needs of business. However, it is important to have an automated mechanism in place which ensures timely delivery of dashboards updated accurately, in various forms, to users under different business groups. Dissemination also empowers a failover redundancy mechanism ensuring zero downtime in delivery.

III. Hadoop based Self-Service Data Integration Platform Architecture

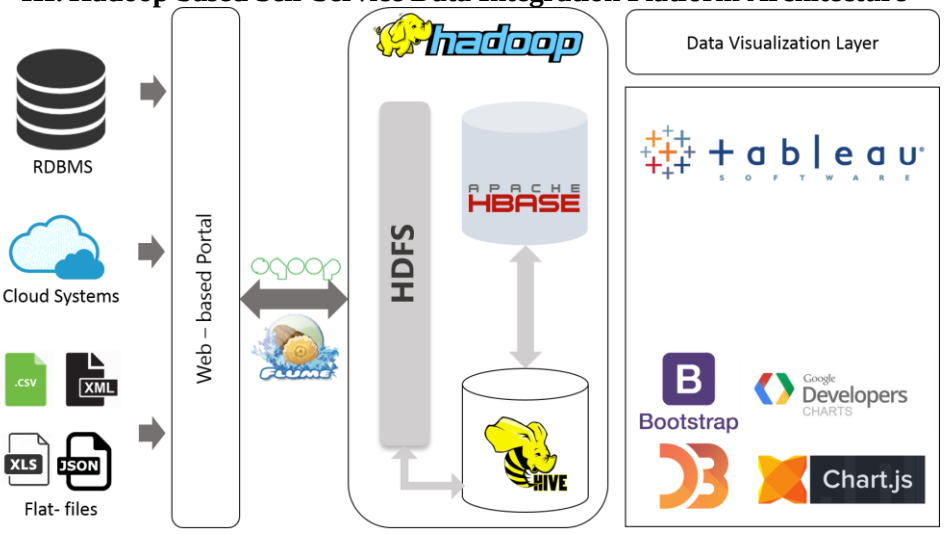

Fig 2. Hadoop based Data Integration Platform

The modern Hadoop based Data Integration Platform is a highly agile, robust, versatile and scalable version of the components and processes as mentioned in section [2]. This architectural components are elaborated in the following sub-sections [3].

3.1 Heterogeneous Data-sources

Data originates from variety of source systems and it is difficult to ensure the homogeneity of the source systems because different business systems are acquired and/or adopted over a period of time, and this is a function of the changing business models. For e.g. 15 years back CRM tools were a luxury for the Marketing teams of the leading organizations but today cloud based CRM solutions like Salesforce, Marketo etc. is a necessity. Data-sources can be structured databases like RDBMS or semi structured like NoSQL or unstructured flat files. It can also be on premise or on the Cloud. Our Data Integration platform is versatile enough to connect to the market leading solutions which plays around the abovementioned areas. The end-users can connect to a wide variety of data sources by the click of few buttons. Our scope of work also lies in customization and implementation of data connectors for a much wider range of solutions.

3.2 SQOOP

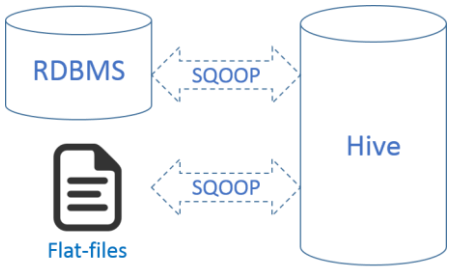

Fig 3. Data transfer from RDBMS & Flat-files to Hive

Sqoop is a tool designed to perform ETL operations on data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases. It is provided by the Apache Software Foundation. SQOOP data extraction, transformation and loading jobs are created in form of scripts. We have certain pre-defined scripts that has been written in alignment to our client's business needs of data extraction and loading. For data transformation we dynamically generate the SQOOP commands based on the data manipulation components used and configurations made by the user on the web layer. This ensures Job design/customization as well as source-destination schema mapping with minimal effort.

3.3 Flume

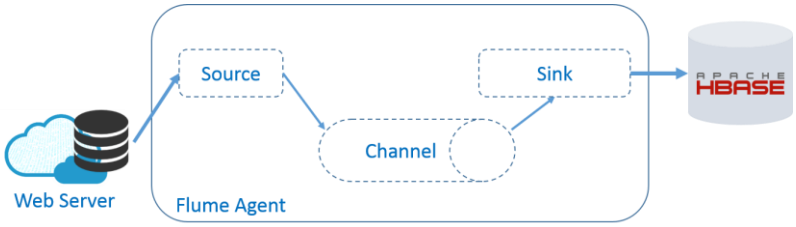

Fig 4. Data transfer from Cloud Data-sources to HBase

Apache Flume is a Hadoop based component used for collecting, aggregating and transporting large amounts of streaming data such as log files, clickstream data, etc. from various sources to a centralized data store.[3] Apache Flume is used to store the data in to any of the centralized stores like HBase & HDFS. Along with the log files, Flume is also used to import huge volumes of event data produced by social networking sites like Facebook and Twitter, and e-commerce websites like Amazon and Flipkart. In our solution, we used Flume to extract live Cloud CRM feeds from Sales Force to be aggregated and pushed to HBase.

3.4 Web – based Portal

The web – based portal is a J2EE Hibernate MVC based application which runs on Apache Web Server and uses MySQL database to maintain its meta-data. The presentation layer (User Interface) of the web application is built using Bootstrap UI framework for device responsiveness. The web based portal allows users to connect to different data-sources and perform operations like Schema Replication/Designing, Job Designing, Job Scheduling & Job Tracking/Management as explained in section [4]. The events of UI Objects are used to

trigger the SQOOP/Flume scripts for ETL processes. The user interface enables the user to perform operations on heterogeneous data sources by clicking of the buttons and HTML form elements, thereby completely eradicating the need of writing any kind of SQL or data extraction/manipulation scripts.

3.5 Hadoop 2.0

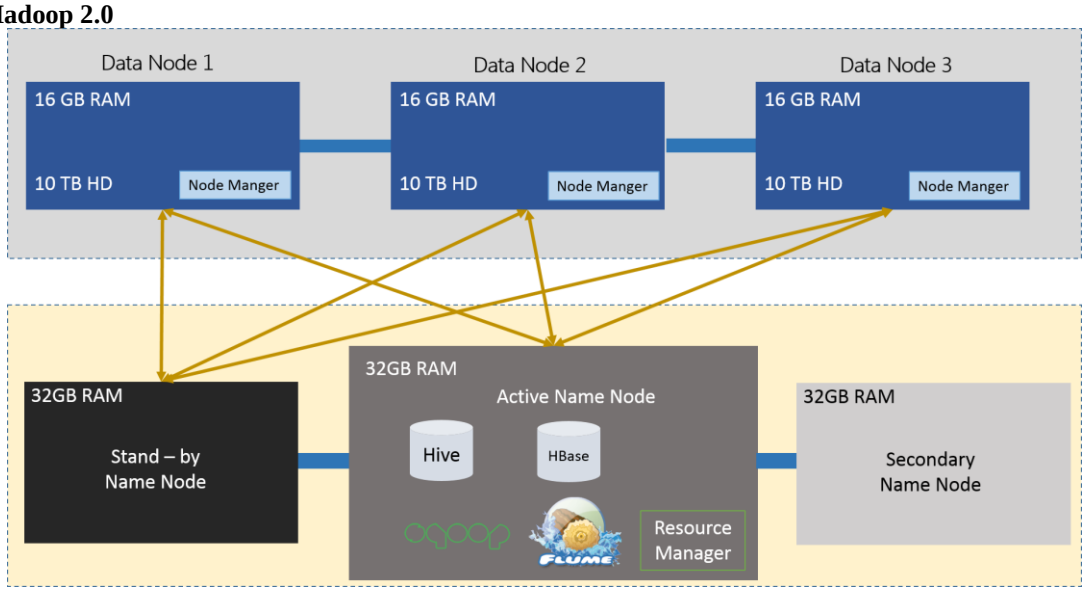

Fig 5. Hadoop 2.0 Configuration Architecture

Apache Hadoop is an open source software platform for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Hadoop services provide for data storage, data processing, data access, data governance, security, and operations. Hadoop Distributed File System (HDFS) is a core component of Apache Hadoop and is designed to store large files with streaming data access patterns, running on clusters of commodity hardware [4]. HDFS follows the master-slave architecture and it has the following elements: Name-node, Data-node and Block. Name-node acts as master server. Two of the most important advances in Hadoop 2 are the introduction of HDFS federation and the resource manager YARN. HDFS is the Hadoop file system and comprises two major components: namespaces and blocks storage service. The namespace service manages operations on files and directories, such as creating and modifying files and directories. The block storage service implements data node cluster management, block operations and replication. [5] In Hadoop 1, a single Name-node managed the entire namespace for a Hadoop cluster. With HDFS federation, multiple Name-node servers manage namespaces and this allows for horizontal scaling, performance improvements, and multiple namespaces. The implementation of HDFS federation allows existing Name-node configurations to run without changes. For Hadoop administrators, moving to HDFS federation requires formatting Name-nodes, updating to use the latest Hadoop cluster software, and adding additional Name-nodes to the cluster. HDFS federation brings important measures of scalability and reliability to Hadoop. YARN, the other major advance in Hadoop 2, brings significant performance improvements for some applications, supports additional processing models, and implements a more flexible execution engine.

YARN is a resource manager that was created by separating the processing engine and resource management capabilities of MapReduce as it was implemented in Hadoop 1.[7] YARN is often called the operating system of Hadoop because it is responsible for managing and monitoring workloads, maintaining a multi-tenant environment, implementing security controls, and managing high availability features of Hadoop.

3.6 Data Visualization Layer

Once all the required data has been staged, cleansed, aggregated & loaded into the Hive & HBase, it is important for the data to be represented in form of charts/visualizations to make it consumable for business decision-making. Analysts uses sophisticated and user-friendly data visualization tools like Tableau to visualize the data whereas some other business units may use JavaScript based Charting APIs like Google Charts/ Chart.JS to visualize data.

IV. Components of Data Integration Platform

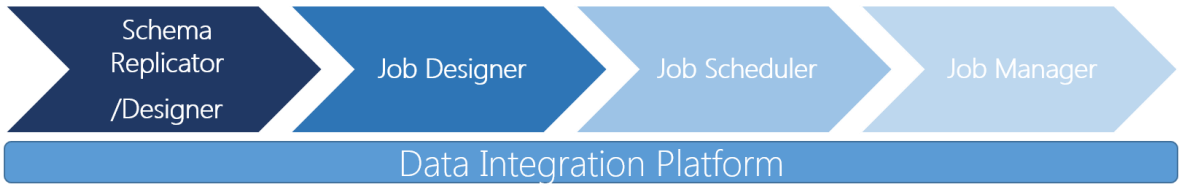

Fig 6. The Data Integration Platform Process Flow

The Data Integration Platform is formed of four major components as shown in figure 6. These however follow a process flow in order to integrate heterogeneous sources to the Hadoop based storages, Hive & HBase. The components are explained in the following sub-section [4].

4.1 Schema Replicator/Designer

The uniqueness of this application lies in its versatility of connecting to different source systems, determining the schema/metadata and its ability to seamlessly replicate/create the same schema into Hive with the compatible data types and constraints of the source systems, without having the need of creating the schema in the destination separately using SQL queries, etc.

4.2 Job Designer

The Job designer component, just like any other ETL tool allows the end user to create Data Integration Workflows by dragging & dropping objects using a simple web interface. The Job Designer consists the following data blending objects,

Swap – Helps swap between Rows and Columns. Transpose & Cross-tab.
Sorting – Helps sorting the data based on multiple criteria
Lookups – Helps perform calculations within a column based on the reference row of the same column.
Mathematical Aggregations – Performs operations like Sum, Average, Max, Min, Percentage, Modulus, etc.
Logical Conditioning – Performs conditional filtering & modification of data
Summarize – Helps in implementing Group By operation on a dataset.
String Operations – Helps in performing different operations like Trimming White Space, extracting a Substring, Matching a substring, Find & Replace, etc.
Joins – Helps join multiple data sets based on conditions both row-wise or column-wise. Hence it can be used for implementing a Primary Key–Foreign Key relationship and append data one below the other, as well.
The above list can be exhaustive and customized depending upon the customer needs but we have implemented the above functionalities which suffices the day-to-day data blending needs of our customer.

4.3 Job Scheduler

The Job Scheduler allows end users to create a recurrence run time of the work flows created using the Job Designer. This component is a form on the web interface which allows the user to select the recurrence schedule for a particular job from a set of drop downs. On scheduling, a Cron job gets created which keeps track of the reference schedule depending on system time of the web server. The Cron job triggers the work flow on the scheduled time. Every job/work-flow needs to be scheduled at least for the first time.

4.4 Job Manager

The Job Manager provides a tabular view of different job status and their run time metrics on the web based user interface. This also helps the user to monitor the success/failure of the jobs in production. This component also provides an option to instantly trigger a failed job by the click of a button. In addition, it also generates a log file for the failed jobs, which helps the user to do a root-cause analysis to resolve the cause of failure.

V. Conclusion

This solution has enabled our customer to overcome the dependencies on the IT Data Engineering team for data acquisition required for business reporting or analysis. It has reduced the resourcing cost of data engineering by $1/10^{th}$. Hence, we have enabled a self-service data integration platform with the capability of Big Data processing at an optimized cost of implementation & ownership. The scope of our future work lies in the seamless integration of the solution with multiple data sources. [6]However, we also understand that data transformation is a function of the source data and the desired business metric, which differs across industries.

Therefore, we also intend to bring variety in data transformation tools which will make the solution compatible across industries.

References

Proceedings Papers:

[1] Patrick Mader, Jane Cleland-Huang, "From Raw Project Data to Business Intelligence", *IEEE Software*, vol. 32, no. , pp. 22-25, July-Aug. 2015, doi:10.1109/MS.2015.92

[2] Shweta Malhotra, M. N Doja, Bashir Alam, and Mansaf Alam, Data integration of cloud-based and relational databases, 2015 IEEE International Conference, Electronic ISBN: 978-1-4673-6792-9, **DOI:** 10.1109/ICSCTI.2015.7489542.

[3] Yi Shen, Shengsheng Shi, Haitao Wang, Wu Wei, Chunfeng Yuan, and Yihua Huang, Parallel Approach and Platform for Large-scale Web Data Extraction, 2013 IEEE International Conference, Electronic ISBN: 978-1-4799-3261-0, **DOI:** 10.1109/CBD.2013.24.

[4] Niall Gaffney, Christopher Jordan, Tommy Minyard, and Dan Stanzione, Building Wrangler: A transformational data intensive resource for the open science community, 2014 IEEE International Conference, **Electronic ISBN:** 978-1-4799-5666-1, **DOI:** 10.1109/BigData.2014.7004480.

[5] Liutong Xu, Kai Jin, and Hongqiao Tian, MRData: A MapReduce-based Tool for Heterogeneous Data Integration, 2010 IEEE International Conference, Electronic ISBN: 978-1-4244-7670-1, **DOI:** 10.1109/ISME.2010.252.

[6] Paul T. Ward, "The transformation schema: An extension of the data flow diagram to represent control and timing", *IEEE Transactions on Software Engineering*, vol. 12, no. , pp. 198-210, Feb. 1986, doi:10.1109/TSE.1986.6312936

Books:

[7] Srinath Perera, Thilina Gunarathne, Hadoop MapReduce Programming, Packt Publishing, 2013