Health and Insurance Data Analysis Architecture

Health and Insurance Data Analysis is Data Hub where analysis for Medicine and Insurance Policies data is done on Big Data platform . At least 10% of the Healthcare insurance payments are attributed to fraudulent claims. Worldwide this is estimated to be a multi-billion dollar problem. Fraudulent claims is not a novel problem but the complexity of the insurance frauds seems to be increasing exponentially making it difficult for the healthcare insurance companies to deal with them.

Need for Insurance data analysis :

- Parallel Data Processing that is unconstrained.
- Provide storage for billions and trillions of unstructured data sets.
- Fault tolerance along with high availability of the system.

Long the domain of sophisticated data processing applications, the insurance industry is today awash in data about customers, trends, and competitors. The volume of data available to businesses from both internal systems and external channels has led to a new category of application known as “Big Data”. For insurers, the benefits of using analytical applications that tap into the Big Data stream are significant. These applications can provide information to enhance sales, marketing, and underwriting; operational activities that reduce costs; and strategies to better understand and reduce risk.

Deploying Big Data applications and the Big Infrastructure to support them, however, entails a high level of complexity. Thus far only a small percentage of top tier insurers have these systems in place. Partly the delay is due to the complexity of this new type of infrastructure

Benefits :

- Integrated, Hadoop solution for the enterprise
- Faster time to deployment
- Automated, consistent, dependable deployment and management
- Simplified operation that can be quickly learned without systems administration experience

Architecture :

The cluster was designed in such a manner so that source Oracle could be transformed in Hive platform which is integrated with HBase.

The below architecture shows the flow of data from source Oracle to target Hive.

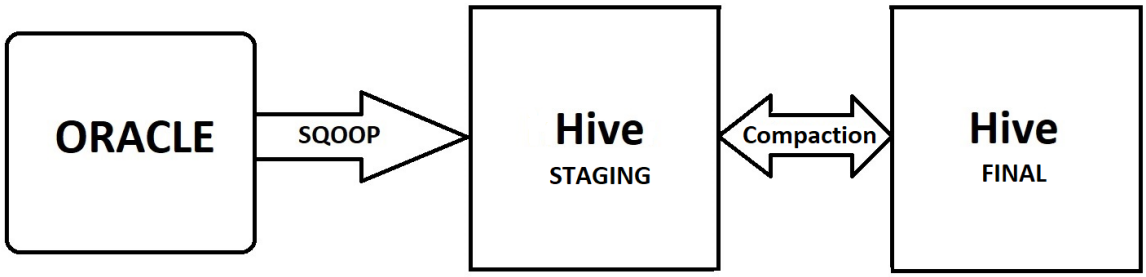

Fig. Health and Insurance Data Analysis Architecture

The source data which is in Oracle is transferred to HBase tables via SQOOP . This Hbase is coupled with Hive . So the target tables will be stored in HBase while the same will be reflected in Hive table.

Airflow scheduler will be used to trigger the jobs on timely basis.

Cluster Architecture

Data Warehouse was designed keeping in mind to store data for 1 year. It has a cluster which consists of 9 Nodes, namely 3 NameNodes and 6 DataNodes.

NameNodes are configured high on memory keeping processing speed in mind. Configuration of them are as follows:

RAM:- 512 GB and HDD:- 3.4 TB each. (PNN,ANN,RM)

Total RAM = 512 GB * 3 NN = 1.5 TB

Total HDD = 3.4 TB * 3 NN = 13.2 TB

One of the namenode is the Active(Primary NameNode) which will be responsible for assigning jobs to the datanodes. The second will be the Passive (Secondary NameNode). The last one will be acting as resource manager for Allocation of resources to the application(s)/jobs(s)

Datanodes are configured high on space. So they would have lot disk space than that of the NameNodes . Configuration of them are as follows:

RAM:- 256 GB and HDD:- 9.4 TB each.

Total RAM = 256 GB * 6 DN = 1.5 TB

Total HDD = 9.3 TB * 6 DN = 55.8 TB

Services and their Versions:

1. HDFS : 2.7.2
2. SQOOP : 1.4.6
4. Hive: 0.14 (hive-1.1.0-cdh5.4.0)
5. AirFlow:
6. Spark: 2.1.1

Table Structure

There 2 types of tables to be loaded into Hive target tables they are Transactional and Reference . Transactional tables are those where there is update in data continuously . References tables are used as reference and loaded once in a day.

There are 80 Transactional tables and 30 Reference tables.

The initial bulk load will consist of 25-30 TB data. Incremental data would be 200 GB per day.

Transactional Tables:

1. Billing
2. Claim
3. FamilyDetails
4. Patient
5. Staff

Reference Tables:

1. Address
2. City
3. Country
4. Department
5. Disease
6. Doctor
7. EAddress
8. Hospital
9. Policy
10. State
11. Test

Optimization Techniques

HIVE OPTIMIZATION

1. Partition
2. Bucketing
3. Cost based optimizations
4. Use Of Order by
5. Use of Sort by
6. Use by limit function
7. Paralellism
8. Vectorization
9. Hive indexing

SQOOP OPTIMIZATION

1. - - direct
2. - -boundary query- -
3. Avoid joins while importing data

Roles and Responsibilities

• Import Data into the Hive from various Relational Databases (Oracle, Sybase, AS400, SQL Server) using Sqoop.
• Write Java MapReduce Job to Ingest EBCDIC, XML Files into Hive Warehouse.
• Write Hive DDL to Create Hive Table for Optimize Query Performance.
• Ingest Flat Files like Delimited, Fixed Length, etc. into Hive Warehouse.
• Raise JIRA for Infrastructure, Platform issues.
• Engage with BA to understand the requirements clearly.
• Define all the possible Test Cases along with the Test Data.
• Comes up with missing scenarios and get them clarified with BA.
• Implement Data Validation, Quality Checks, Profiling.
• Perform UAT of Big Data implementation.
• Involved in Collecting Business Requirements from Business Users, Translate into Technical Design (Data Pipelines and ETL workflows).
• Involved in import data from various RDBMS into HDFS using Sqoop which includes Incremental Load to populate Hive External Table and vice-versa.
• Involved in import data from various RDBMS into Hive Tables which includes Queries using Sqoop.
• Developed MapReduce in Java for ETL workflows like XML, Fixed Length.
• Designed both Managed and External Hive Tables and Defined static and dynamic partitions as per requirement for optimized performance on production datasets.
• Worked with various File Formats like Text File, SequenceFile, ORC Files, Avro Files and various Compression Formats like Snappy, bzip2.
• Written Hive Queries for Data Analysis to meet the business requirements.
• Built reusable Hive UDF libraries for business requirements which enabled users to use these UDF's in Hive Querying.
• Involved in developing shell scripts and automated data management from end to end integration work.

Interview Questions and Answers

1. Explain your project and its flow.
2. What is the size of cluster and team ?
3. What is the configuration of the nodes used in cluster ?
4. Explain YARN/SPARK architecture.
5. What if the spark job job doesn't get containers ?
6. What if the Sqoop job fails in between ?
7. What are the difficulties faced during this project .
8. What did you use for security in cluster ?
9. What did you code in shell script ?
10. How did you remove duplicates in Hive ?
11. What Transformations were used in Hive ?
12. How the jobs were triggered ?
13. What was the data about? Columns and mapping
14. What were the joins you used in Hive ? Do you any other joins ?
15. What were your Roles and Responsibilities ?
16. How many mappers were used?
17. Optimiztaion techniques used in the Project.
18. How many mappers were used in your sqoop project ?
19. Do you have experience in Spark?
20. How did you verify whether the correct data was imported ?
21. Why did not you made joins while importing through Sqoop?