International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.30
October-2014,
Pages:5935-5938

www.ijsetr.com

Data Modeling Considerations in Telecommunications using Hadoop

PRASADU BEZAVADA[1], SULAKSHANA CHEPPALLI[2]

[1]PG Scholar, Bharat Institute of Engineering and Technology, Ibrahimpatnam, Ranga Reddy, Telangana, India,
E-mail: prasadbezavada@gmail.com.

[2]Associate Professor, Bharat Institute of Engineering and Technology, Ibrahimpatnam, Ranga Reddy, Telangana, India,
E-mail: sulakshana.cheppalli@biet.ac.in.

Abstract: We are living in an information age and there is enormous amount of data that is flowing between systems, internet, telephones, and other media. The data is being collected and stored at unprecedented rates. There is a great challenge not only to store and manage the large volume of data, but also to analyze and extract meaningful information from it. There are several approaches to collecting, storing, processing, and analyzing big data. The main focus of the paper is to draw an analogy for data management between the traditional relational database systems and the Big Data Techniques .Aim of this project is finding the business insights of current user records data (i.e data cards usage records). And get the benefits for business growth. The parameters to be considered for analysis are

1. Daily user count and bytes transmitted on a particular time slot.
2. Area wise business(usage) share in the total business
3. Since every network owner will be depending on partners to get the service where they does not have the service tower.

Keywords: Data Manipulation (DML), User Defined Column Transformation Functions (UDF), User Defined Column Transformation Aggregation Functions (UDAF).

I. INTRODUCTION

From case1: We can find the exact what time more users using the network and what time more downloads and uploads happening. Based on that, they can concentrate tower capacity enhancements. If the tower is underutilized then they can reduce the tower capacity.

From case2: They can concentrate the area where they can invest more to get the more users.

From case3: Find out the areas of partner leading and try to improve the owner tower installations.

All above activities currently happening using data warehousing technologies. But it is more expensive and time consuming. To help better in this area (Telecommunications), we are using the Hadoop and Hadoop Eco-systems.

Data creation is occurring at an unprecedented rate. In 2010, the world generated over 1ZB of data; and by 2014, we have generated 7ZB of data. IBM estimates that every day 2.5 quintillion bytes of data are created – so much that 90% of the data in the world today has been created in the last two years. Increasingly large numbers of embedded sensors, smart phones, PCs, and tablet computers connected to network are generating enormous amounts of data. This data creates new opportunities to "extract more value" for the areas that it is needed. We have entered the age of "Big Data." Just as this data is generated by people in real time, it can be analyzed in real time by high performance computing networks, thus creating a potential for improved decision-making. The International Data Corporation (IDC) believes organizations that are best able to make real-time business decisions using Big Data solutions will thrive, while those that are unable to embrace and make use of this shift will increasingly find themselves at a competitive disadvantage in the market and face potential failure.

II. RELATED WORK

Big Data: Human Generated Data is emails, documents, photos and tweets. We are generating this data faster than ever. Just imagine the number of videos uploaded to You Tube and tweets swirling around. This data can be Big Data too. Machine Generated Data is a new breed of data. This category consists of sensor data, and logs generated by 'machines' such as email logs, click stream logs, etc. Machine generated data is orders of magnitude larger than Human Generated Data. Before 'Hadoop' was in the scene, the machine generated data was mostly ignored and not captured. It is because dealing with the volume was NOT possible, or NOT cost effective.

Where does Big Data come from?

- Original big data was the web data -- as in the entire Internet! Remember Hadoop was built to index the web. These days’ Big data comes from multiple sources.
- Web Data -- still it is big data.

- **Social media data:** Sites like Facebook, Twitter, LinkedIn generate a large amount of data Click stream data: when users navigate a website, the clicks are logged for further analysis (like navigation patterns). Click stream data is important in on line advertising and E-Commerce
- **Sensor data:** sensors embedded in roads to monitor traffic and misc. other applications generate a large volume of data
- Connected Devices: Smart phones are a great example. For example when you use a navigation application like Google Maps or Waze, your phone sends pings back reporting its location and speed (this information is used for calculating traffic hotspots). Just imagine hundres of millions (or even billions) of devices consuming data and generating data.

Hadoop is an open-source implementation of Google's distributed computing framework (which is proprietary). It consists of two parts: Hadoop Distributed File System (HDFS), which is modeled after Google's GFS, and Hadoop MapReduce, which is modeled after Google's MapReduce. MapReduce is a programming framework. Its description was published by Google in 2004 [http:// research.google.com/ archive/mapreduce.html]. Much like other frameworks, such as Spring, Struts, or MFC, the MapReduce framework does some things for you, and provides a place for you to fill in the blanks. What MapReduce does for you is to organize your multiple computers in a cluster in order to perform the calculations you need. It takes care of distributing the work between computers and of putting together the results of each computer's computation. Just as important, it takes care of hardware and network failures, so that they do not affect the flow of your computation. You, in turn, have to break your problem into separate pieces which can be processed in parallel by multiple machines, and you provide the code to do the actual calculation. But of course, Hadoop really shines when you have not one, but rather tens, hundreds, or thousands of computers. If your data or computations are significant enough (and whose aren't these days?), then you need more than one machine to do the number crunching. If you try to organize the work yourself, you will soon discover that you have to coordinate the work of many computers, handle failures, retries, and collect the results together, and so on. Enter Hadoop to solve all these problems for you. Now that you have a hammer, everything becomes a nail: people will often reformulate their problem in MapReduce terms, rather than create a new custom computation platform.

III. DATA PROCESSING WITH HIVE

The size of data sets being collected and analyzed in the industry for business intelligence is growing rapidly, making traditional warehousing solutions prohibitively expensive. Hadoop[3] is a popular open-source map-reduce implementation which is being used as an alternative to store and process extremely large data sets on commodity hardware. However, the map-reduce programming model is very low level and requires developers to write custom programs which are hard to maintain and reuse. In this paper, we present Hive, an open-source data warehousing solution built on top of Hadoop as shown in Fig.1. Hive supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into map-reduce jobs executed on Hadoop. In addition, HiveQL supports custom map-reduce scripts to be plugged into queries.

The language includes a type system with support for tables containing primitive types, collections like arrays and maps, and nested compositions of the same. The underlying IO libraries can be extended to query data in custom formats. Hive also includes a system catalog, Hive-Metastore, containing schemas and statistics, which is useful in data exploration and query optimization. In Facebook, the Hive warehouse contains several thousand tables with over 700 terabytes of data and is being used extensively for both reporting and ad-hoc analyses by more than 100 users. Hive provides a SQL-like query language called HiveQL which supports select, project, join, aggregate, union all and sub-queries in the from clause. HiveQL supports data definition (DDL) statements to create tables with specific serialization formats, and partitioning and bucketing columns. Users can load data from external sources and insert query results into Hive tables via the load and insert data manipulation (DML) statements respectively.

HiveQL currently does not support updating and deleting rows in existing tables. HiveQL supports multi-table insert, where users can perform multiple queries on the same input data using a single HiveQL statement. Hive optimizes these queries by sharing the scan of the input data. HiveQL is also very extensible. It supports user defined column transformation (UDF) and aggregation (UDAF) functions implemented in Java. In addition, users can embed custom map-reduce scripts written in any language using a simple row-based streaming interface, i.e., read rows from standard input and write out rows to standard output. This flexibility does come at a cost of converting rows from and to strings. We omit more details due to lack of space. For a complete description of HiveQL see the language manual.

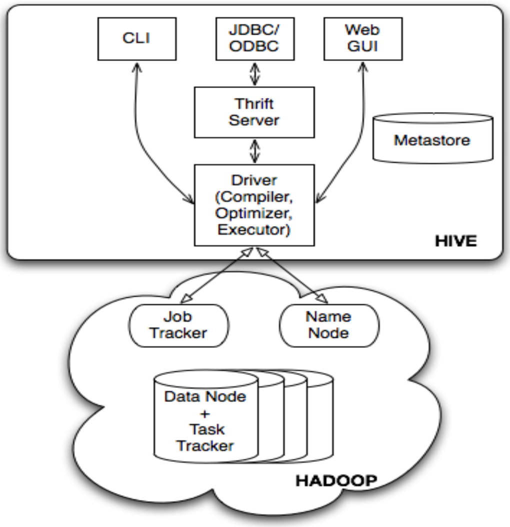


Fig.1. Hive Architecture.

A. Our Data Processing Task With SQOOP

Sqoop is a tool designed to transfer data between Hadoop and relational databases. You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS. Sqoop automates most of this process, relying on the database to describe the schema for the data to be imported. Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance. This document describes how to get started using Sqoop to move data between databases and Hadoop and provides reference information for the operation of the Sqoop command-line tool suite. This document is intended for:

- System and application programmers
- System administrators
- Database administrators
- Data analysts
- Data engineers

In the telecommunications industry, a single household is often comprised of different individuals who have each contracted with a particular service provider for different types of products, and who are served by different organizational entities within the same provider. These customers communicate with the provider through various online and offline channels for sales- and service-related questions, and in doing so, expect that the service provider be aware of what’s going on across these different touch points.

IV. PROPOSED WORK

In This paper we can find the exact what time more users using the network and what time more downloads and uploads happening. Based on that, they can concentrate tower capacity enhancements. If the tower is underutilized then they can reduce the tower capacity. They can concentrate the area where they can invest more to get the more users. Find out the areas of partner leading and try to improve the owner tower installations is as shown in Fig.2.

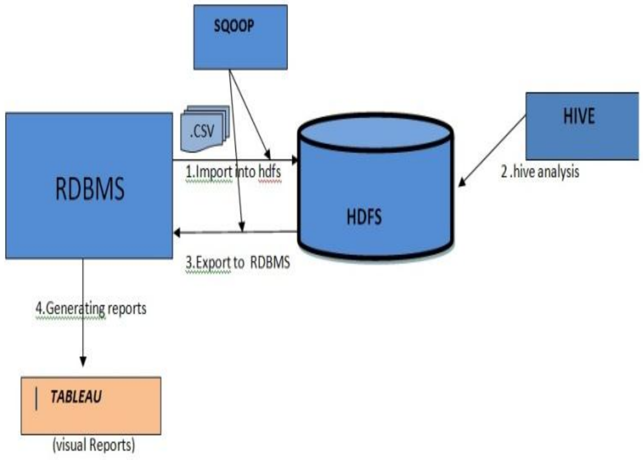
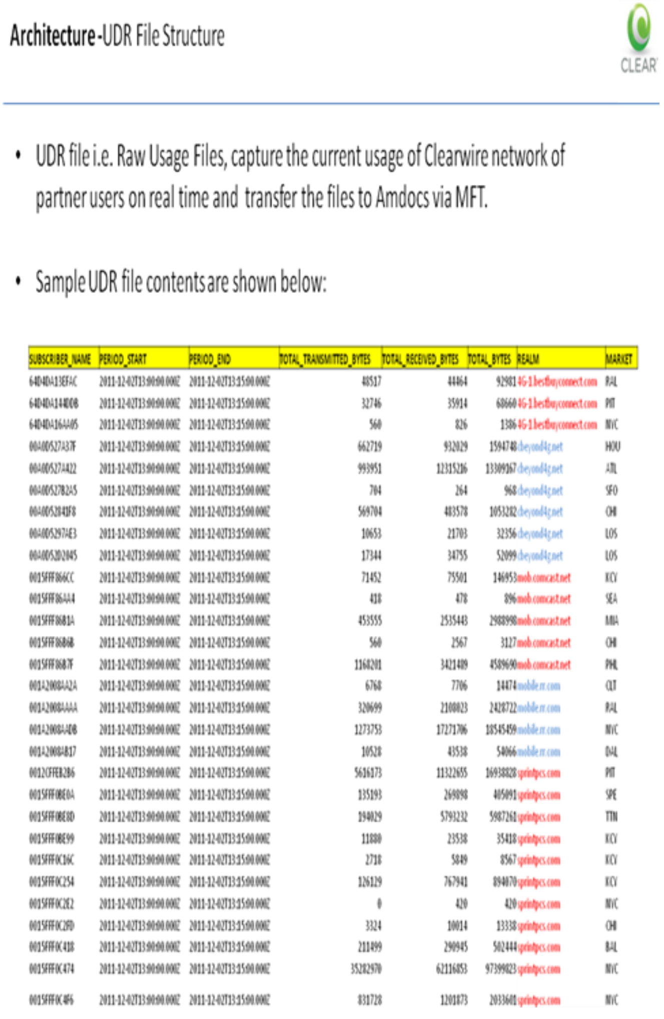


Fig.2. Proposed Architecture.

V. EXCREMENTAL RESULT

Input values give for Testing:



Architecture-UDR File Structure

CLEAR

- UDR file i.e. Raw Usage Files, capture the current usage of Clearwire network of partner users on real time and transfer the files to Amdocs via MFT.
- Sample UDR file contents are shown below:

SUBSCRIBER_NAME	PERIOD_START	PERIOD_END	TOTAL_TRANSMITTED_BYTES	TOTAL_RECEIVED_BYTES	TOTAL_BYTES	REALM	MARKET
64D4DA13EFAC	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	40517	44464	92981	4G-1.bestbuyconnect.com	RAL
64D4DA144D0B	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	32746	35914	68660	4G-1.bestbuyconnect.com	PIT
64D4DA164A05	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	560	826	1386	4G-1.bestbuyconnect.com	NYC
00A0D527A37F	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	662719	932029	1594748	cbeyond4g.net	HOU
00A0D527A422	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	993951	12315216	13309167	cbeyond4g.net	ATL
00A0D527B2A5	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	704	264	968	cbeyond4g.net	SFO
00A0D52841F8	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	569704	483578	1053282	cbeyond4g.net	CHI
00A0D5297AE3	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	10653	21703	32356	cbeyond4g.net	LOS
00A0D52D2045	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	17344	34755	52099	cbeyond4g.net	LOS
0015FFF866CC	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	71452	75501	146953	mob.comcast.net	KCY
0015FFF86A44	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	418	478	896	mob.comcast.net	SEA
0015FFF8681A	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	453555	2535443	2988998	mob.comcast.net	MIA
0015FFF868AB	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	560	2567	3127	mob.comcast.net	CHI
0015FFF8687F	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	1168201	3421489	4589690	mob.comcast.net	PHL
001A2008AA2A	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	6768	7706	14474	mobile.rr.com	CLT
001A2008AAAA	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	320699	2108023	2428722	mobile.rr.com	RAL
001A2008AAD8	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	1273753	17271706	18545459	mobile.rr.com	NYC
001A2008AB17	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	10528	43538	54066	mobile.rr.com	DAL
0012CFFEB2B6	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	5616173	11322655	16938828	sprintpcs.com	PIT
0015FFF08E0A	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	135193	269898	405091	sprintpcs.com	SPE
0015FFF08E8D	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	194029	5793232	5987261	sprintpcs.com	TTN
0015FFF08E99	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	11880	23538	35418	sprintpcs.com	KCY
0015FFF0C16C	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	2718	5849	8567	sprintpcs.com	KCY
0015FFF0C254	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	126129	767941	894070	sprintpcs.com	KCY
0015FFF0C2E2	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	0	420	420	sprintpcs.com	NYC
0015FFF0C2FD	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	3324	10014	13338	sprintpcs.com	CHI
0015FFF0C418	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	211499	290945	502444	sprintpcs.com	BAL
0015FFF0C474	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	35282970	62116853	97399823	sprintpcs.com	NYC
0015FFF0C4F6	2011-12-02T13:00:00.000Z	2011-12-02T13:15:00.000Z	831728	1201873	2033601	sprintpcs.com	NYC

The result is as shown in Fig.3.

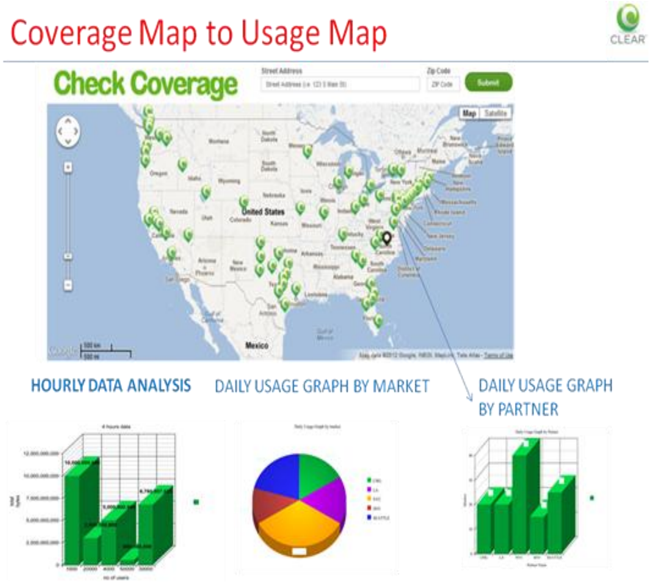


Fig.3.result.

VI. CONCLUSION

We found the business insights of current user records data (i.e data cards usage records). And get the benefits for business growth. The parameters to be considered for analysis and gave them the results like Daily user count and bytes transmitted on a particular time slot, Area wise business(usage) share in the total business and Since every network owner will be depending on partners to get the service where they does not have the service tower. We solved the Problem Statement present in existing system in this paper.

VII. REFERENCES

[1] Big Data and Cloud Computing: Current State and Future Opportunities 2013.

[2] D. Agrawal, S. Das, and A. E. Abbadi. Big data and cloudcomputing: New wine or just new bottles? PVLDB, 3(2):1647–1648,2010.

[3] D. Agrawal, A. El Abbadi, S. Antony, and S. Das. Data ManagementChallenges in Cloud Computing Infrastructures. In DNIS, pages1–10, 2010.

[4] P. Agrawal, A. Silberstein, B. F. Cooper, U. Srivastava, andR. Ramakrishnan.Asynchronous view maintenance for vlsddatabases. In SIGMOD Conference, pages 179–192, 2009.

[5] S. Aulbach, D. Jacobs, A. Kemper, and M. Seibold.A comparison offlexible schemas for software as a service. In SIGMOD, pages881–888, 2009.

[6] “Understanding Hadoop Clusters and the Network.” Available at http://bradhedlund.com. Accessed on June 1, 2013.

[7] Sammer, E. 2012. Hadoop Operations. Sebastopol, CA: O'Reilly Media.

[8] “HDFS High Availability Using the Quorum Journal Manager.” Apache Software Foundation. Available at http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithQJM.html. Accessed on June 5, 2013.

[9] “Hadoop HDFS over HTTP - Documentation Sets 2.0.4-alpha.” Apache Software Foundation. Available at http://hadoop.apache.org/docs/r2.0.4-alpha/hadoop-hdfs-httpfs/index.html. Accessed on June 5, 2013.

[10]“Yahoo! Hadoop Tutorial.” Yahoo! Developer Network. Available at http://developer.yahoo.com/hadoop/tutorial/. Accessed on June 4, 2013.

[11]“Configuring the Hive Metastore.” Cloudera, Inc. Available at http://www.cloudera.com/content/cloudera-content/cloudera-docs/CDH4/4.2.0/CDH4-Installation-Guide/cdh4ig_topic_18_4.html. Accessed on June 18, 2013.

[12]Kestelyn, J. “Introducing Parquet: Efficient Columnar Storage for Apache Hadoop.” Available at http://blog.cloudera.com/blog/2013/03/introducing-parquet-columnar-storage-for-apache-hadoop/. Accessed on August 2, 2013.