International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 6, Issue 6, June 2017 **ISSN 2319 - 4847**

Cognitive Data Partitioning and Analysis using Hive and Neo4j Server

Mr. Swapnil A. Kale [1], Ms. Madhurika S. Patil[2]

[1]Lecturer, Dept. of Information Technology, Government Polytechnic, Gadge Nagar, Amravati, Maharashtra - 444603

[2]Student, Dept. of Computer Science, SRM University, Kattankulathur Campus, Chennai, Tamilnadu - 603203

ABSTRACT

Data processing archetypes have been customized with the advent of new techniques in the area of database management. The management of bulky datasets has posed a big question in front of enterprises. The major issue arises when specific information needs to be driven out of large database. Along with time restrictions, the organizations have to deal with the issues of competence, performance and prominent infrastructure expenditure in the data processing in the competitive surroundings. Using Hadoop and MapReduce architecture these large companies could trounce the predicament of retrieving appropriate information from large data repository. This paper is focused on processing of raw information using partitioning technique of hive, getting the relevant details out and analyzing it using Neo4j server.

Keywords: Hadoop, Big data, MapReduce, Hive, Neo4j

1. INTRODUCTION

Amplified quantity of data is flowing into modern-day business entities and because of this hastily mounting magnitude of data being produced the working environment is now considered as data centric. The centre of attraction here is how to manage this huge quantity? Organizations in their business environment encircled by stakeholders have to develop techniques to tackle this situation. Thus, in this scenario upcoming business entities are finding ways to undertake data handling problem by the use of technology like Hadoop [1]. Previously applications were developed to handle only steady schema with conventional databases that had the benefit of transactional processing and holds a very bulky part of the needed information. Traditional databases are not optimized and the point to consider is high-performance investigation, velocity, scalability and accessibility [2].

Effectual managing and examination of extensive data creates a motivating and vital challenge. After successful execution of big data analytics in the area of social media most of the other sectors are also looking forward to it for analysis purpose [4]. Hadoop can be considered as a momentous constituent of the socio-economical transformation taking place globally, and it is the tremendously vibrant development of progressively more dominant and omnipresent information technology. Innovation in this database technology ground has been a noteworthy vehicle for the revolution of the present-day economy and the rising of an “interrelated economy”.

In this new era of information technology dragging out the required and useful data from huge bulk is the critical task because of many reasons including exponential data growth in size and types and velocity of its generation. Generally, 90% of the worldwide data at present has been produced over the past two years [3]. As a result, the quantity of information and data present with the organizations for analysis is blowing up [6].

2. HIVE PARTITIONING TECHNIQUE

Hive permits users to convert simple and usual Map-Reduce tasks into queries. HiveQL is a language used for firing queries over the database includes a type system with support for tables containing primary data types, collections and nested combination of maps. Hive query language (HiveQL) is developed to run over Hadoop’s MapReduce framework, but covers up intricacy from the developer, and it has certain valuable extensions that are supportive for batch processing systems [7]. Figure shows working of hive queries, the Thrift server is a service to allow remote client to submit requests and communicates with services running on the server. A driver which is combination of compiler, optimizer and executer can be helpful in giving output of fired queries in web GUI as well as command line interface as required. All metadata for hive tables i.e. table definitions and data mappings are stored in metastore.

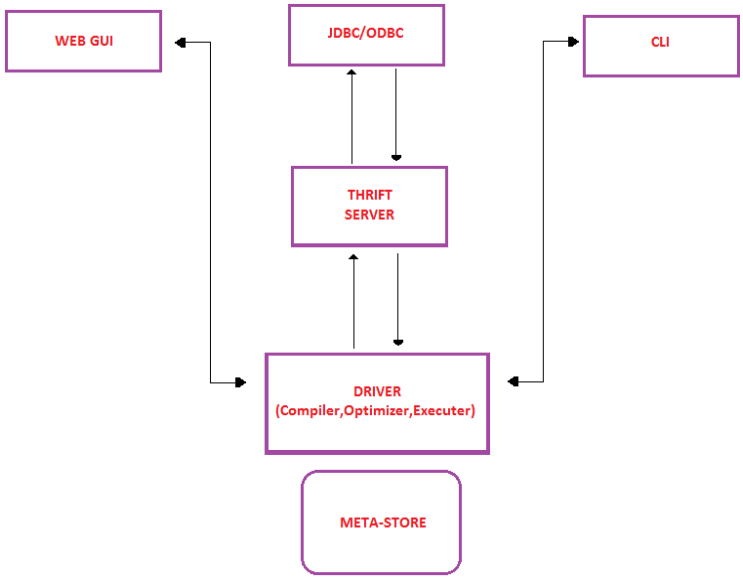

Figure 1 Working of Hive Queries

Hive is an excellent tool for firing queries on big datasets, particularly datasets that involve complete table scans. But pretty frequently there are occasions where users are required to sort the data present in particular columns. A plain query in Hive would read the whole dataset although we have used a where clause in query as a filter. This turn out to be a troublesome task for executing MapReduce tasks over a big tabular data [5].

The issue can be addressed by employing partitioning in Hive. Hive makes it extremely simple to apply partitions when the table is created, by means of automatic partitioning scheme. In this case the data present in Hive table is divided across multiple partitions where each partition represents to a particular value / values of partition column / columns and it is stored in a sub-directory inside the table's directory on HDFS (Hadoop Distributed File System). When the query is fired on the table, where appropriate, only the requisite partitions of that table are queried, whereby reducing the time required for the query processing and I/O. Here data is taken in a traditional format like .csv file initially and then loaded into the hive table. Inside the table partitions are applied.

Normally, Hive users are familiar with the area of the data they are dealing with. With the help of this information they can identify the common columns which are queried frequently. Here they would recognize columns with small cardinality which can be used to systematize data with the help of partitioning feature of Hive. However in case of non-partitioned tables, Hive will have to read all the files in a table's data directory and then apply filters on it. This is sluggish and pricey—particularly in cases of large tables.

3. RELATED WORK

Initially data file in .csv format is loaded in HDFS i.e. the initial data loading is done into a non-partitioned table. From a non-partitioned table data is loaded in a partitioned table. The statement written would be an insert statement for a partitioned table where the columns on which partition is to be made are specified at the last in select clause.

The end-result of this series of queries gives partitions on which specific queries could be fired and results can be obtained quickly because the unnecessary scanning of complete database table is avoided. In the next stage to do the meaningful feature extraction, data from hive is loaded in the neo4j server through a java program. It would generate a networked database where relationships among the data nodes is established and in a large dataset specific interlinks could be identified.

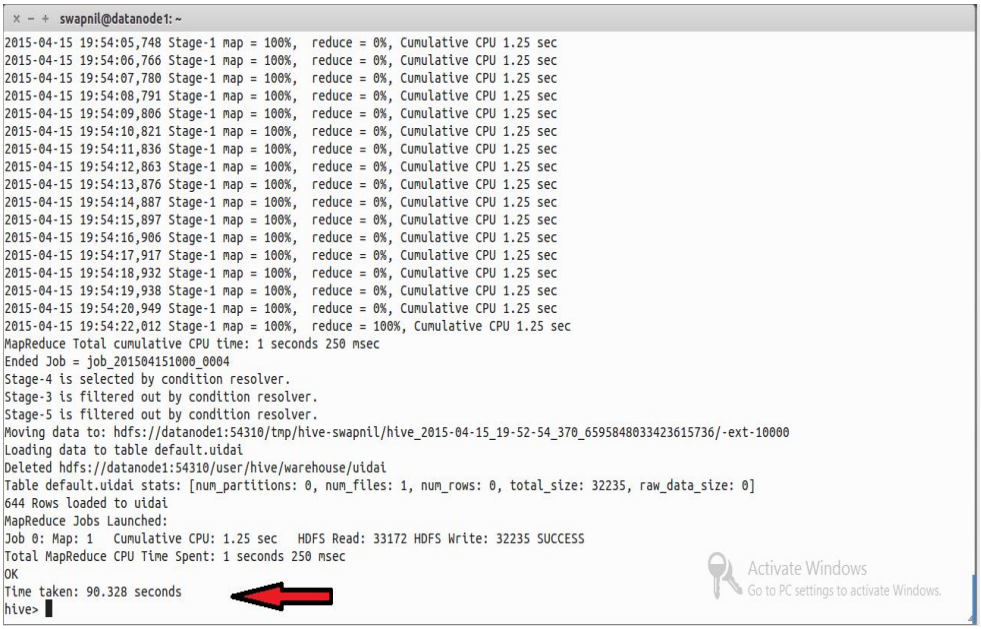

Figure 2 Hive Partitioning

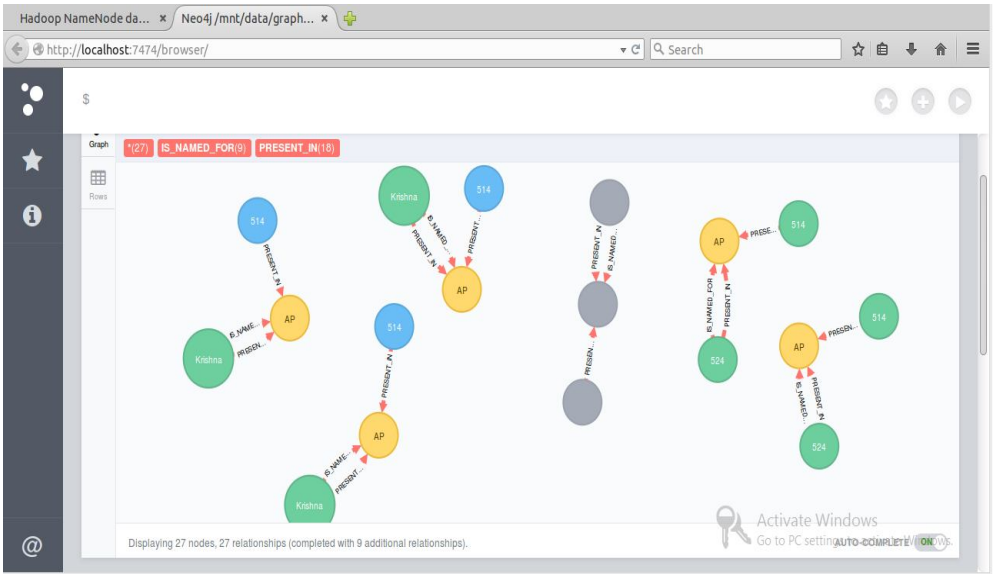

Figure 3 Graph Representation in Neo4j server

4. CONCLUSION

In the vast growing technology era current serving database management applications need to be modified to a larger extent. Handling of structured data and sharing data between applications is a difficult yet a challenging task. All over world many countries are working on ways to take out requisite information from data generated in various domains. The combination of Hive and Neo4j can be a major plus in the area of data retrieval, analysis and can be beneficial for decision-making and also for providing better service. The query processing time can be reduced to a subtle extent in case of large databases and hence it is provide a cost effective solution for data analysis at complex levels.

References

[1] Kemp Little LLP, "BigData – Legal Rights and Obligations", January2013.

[2] Rajagopalan M.R and Solaimurugan vellaipandiyan "*Big data framework for national E-governance plan*" ICT and Knowledge Engineering (ICT&KE), 2013, 11th

InternationalConferenceon DOI:10.1109/ICTKE.2013.6756283 Publication Year: 2013, Page(s): 1–5 IEEE CONFERENCE PUBLICATIONS
[3] A. Lampitt, "Hadoop: Analysis at massive scale", in *InfoWorld*, pp. 8-12, winter 2013.
[4] Hadoop at Twitter https://blog.twitter.com/2010/hadoop-twitter
[5] Kim Rose, How Facebook uses Hadoop and Hive http://hortonworks.com/big-data-insights/how-facebook-uses-hadoop-andhive/
[6] http://engineering.linkedin.com/hadoop
[7] Thusoo, A. ; Sarma, J.S. ; Jain, N. ; Zheng Shao ; Chakka, P. ;Ning Zhang ; Antony, S. ; Hao Liu ; Murthy, R. "Hive -Petabyte scale data warehouse using Hadoop" in Data Engineering (ICDE), 2010 IEEE 26th International conference on DOI: 10.1109/ICDE.2010.5447738 Publication Year:2010, Page(s):996-1005

AUTHOR

Swapnil Kale received the M.E. in Computer Science and Engineering from Prof. Ram Meghe Institute of Technology & Research, Amravati, Maharashtra, in 2015. From 2011, he has been working as a Lecturer in the Department of Information Technology, Government Polytechnic Amravati, Maharashtra. He is also working as a mentor of Big Data Projects for students from various institutes and Universities.