Using Open Source NoSQL technologies in Designing Systems for Delivering Electric Vehicle Data Analytics.

Mr. Vamshi Krishna Bolly, Purdue University

Vamshi K Bolly is a Master of Science student at Purdue University, West Lafayette, Indiana. His research work is related to big data technologies and is concentrated around designing systems for performing electric vehicle data analytics. Prior to his graduate studies, he has about 4 years of industry experience as a software professional.

Dr. John Springer, Purdue University, West Lafayette, IN, USA

John Springer is an Associate Professor in Computer and Information Technology (CIT) at Purdue University and the Lead Scientist for High Performance Data Management Systems at the Bindley Bioscience Center at Discovery Park. Additionally, he is the chair of the Data Management Curricular Subcommittee in CIT. Dr. Springer's discovery efforts focus on distributed and parallel computational approaches to data integration and analytics, and he serves as the leader of the Discovery Advancements Through Analytics (D.A.T.A.) Laboratory.

Dr. J. Eric Dietz, Purdue University

Dr. Dietz's research interests include optimization of emergency response, homeland security and defense, energy security, and engaging veterans in higher education. A professor in Computer and Information Technology and a Director in Purdue's Discovery Park, Dr. Dietz is responsible for the catalysis of the Purdue's homeland security research, increasing the impact of Purdue research on society, and organizing interdisciplinary projects within the university. Prior to his current responsibilities, Eric was on loan from Purdue to Governor Mitch Daniels to serve as the founding Executive Director for The Indiana Department of Homeland Security, a new state agency of over 300 people responsible for emergency planning, training, fire and building safety, and disaster response for 6.2 million Indiana residents. During this period, Eric led Indiana's response to 7 Presidential Major Disasters and Emergency Declarations which included restoration and recovery of critical infrastructure. Eric also led the creation of the Indiana Intelligence Fusion Center and the Indiana Fire Training System both new government functions that were created with new laws and funding. Retiring as a Lieutenant Colonel from the U.S. Army in 2004, Dr. Dietz led a number of Army Acquisition and research programs throughout his career including power systems, chemical sensors and command and control systems. An Indiana native, Eric was graduated in 1984 from Rose-Hulman Institute of Technology after earning a bachelor of science in chemical engineering. He also earned a master's of science from Rose-Hulman Institute of Technology in 1986 and a PhD in chemical engineering in 1994 from Purdue University.

Using Open Source NoSQL technologies in Designing Systems for Delivering Electric Vehicle Data Analytics.

Abstract

Advances in scientific research related to electric vehicles have led to generation of large amounts of data. This data is mainly logger data collected from various sensors in the vehicle and stored as flat files. It is predominantly unstructured and non-relational in nature, also called Big Data. Analysis of such data needs a high performance information technology infrastructure that provides superior computational efficiency and storage capacity. It should be scalable to accommodate the growing data and ensure its security over a network. This research proposes an architecture built on Hadoop to effectively support distributed data management over a network for real-time data collection and storage, parallel processing, and faster and random read access for information retrieval for decision-making.

This system provides a simplified way of extracting from sensor data loggers and transforms this raw data into classified buckets. Once imported into a data store, the system supports data analytics over the data for knowledge discovery, and these analytics can help understand correlations between parameters under various circumstances. This system provides scalability to support data accumulation in the future and still perform analytics with less overhead, and its design can be employed to other fields with similar kind of data analytic challenges. Overall, open problems in data analytics are taken into consideration and a low-cost architecture for data management is proposed.

Introduction

Data mining has made significant progress in various fields of scientific research for discovering new knowledge. The research data accumulated for these purposes is a product of experimentation, simulation and analysis of systems[1]. As this data is predominantly collected from heterogeneous sources, it tends to be multi-dimensional and unstructured in nature[1]. This kind of large scale unstructured multi-dimensional data is known as Big Data[1]. This kind of data requires enriched systems that can help integrate it and allow querying on it for information retrieval. Also, such data needs human analysis and interpretation for identifying any underlying patterns and valuable information[1].

In recent times, plug-in electric vehicle (PEV) research is gaining prominence, attributing to the increasing concerns about global warming and for addressing transportation needs of the future[2]. This continuous advances in PEV research has resulted in generation of large volumes of data, which is collected by numerous sensors and data acquisition systems[2]. Such data provides tremendous opportunities for data mining to discover knowledge that supports future research. There is a need to study the operational characteristics of PEVs and analyze their usage trends to make them an efficient and trustworthy transportation alternative. To realize this potential, researchers need an infrastructure that provides superior computational efficiency and storage capability for PEV data. The main challenge here is to design a system that can effectively help derive knowledge from such datasets. This paper presents an approach towards designing a system using open source NoSQL technologies for meeting the needs and challenges in PEV data analytics.

This is interdisciplinary research that uses the computational capabilities provided by open source NoSQL tools and uses it for performing electric vehicle (EV) data analytics. It aims at building a mechanism for data extraction from sources, loading it into a repository and querying it for relevant information using open source tools such as Apache Hadoop[3]. The repository is designed to handle unstructured data from multiple sources and seamlessly integrate the information. Essentially, the system aims to accomplish the following:

- Effectively scrub the data and prepare for loading.
- Bulk import into the repository.
- Create tables in the appropriate data store for querying.
- Mine the data using data stores.
- Interpret the results to identify any correlations between attributes.

There are numerous benefits in having a centralized data repository for performing data analytics. It can improve the understanding of the underlying systems and can help in development of new tools for supporting future advances in the field of PEVs. The main benefits envisaged from this system are the following:

- Improve the read performance for all types of user queries.
- Support multiple query systems of the users.
- Perform random and parallel processing of the complete dataset.
- Reduce the costs associated with the data repository.
- Generate a scalable data storage solution for supporting data growth in the future.
- Flexibility to add different types of related data during analysis.

Project Plug-IN

The data being analyzed is retrieved from project Plug-IN[4], which is undertaken by Smart Grid network to evaluate the deployment of PEVs as an efficient transportation solution. It works to promote an ecosystem of PEVs powered by an efficient charging infrastructure accessible to users[4]. It aims to solve the business, technical and usability issues related to driving an electric vehicle and envisage a commercial scale deployment of PEVs for transportation needs[4].

As part of the project, technical and driver related data was collected from commercially driven PEVs. This research focused on the data collected from Think City[5] EVs from Indianapolis area. The raw dataset was a set of comma separated value (CSV) files that contained information about various attributes recorded by an EV. EV data loggers recorded this data during real-time commuting conditions over a period of time. The first line of each file provided information about the contained data columns. The major part of the data was unstructured and needed an efficient repository design to mine for useful information.

Role of Hadoop

Open source NoSQL technologies were the first choice for the repository due to their capabilities of handling large scale unstructured data. Hadoop is one such emerging Java-based open source data processing system for distributed computing clusters that provide a platform for data analytics[6]. It is part of the Apache Software Foundation and is designed to support data-intensive distributed computing. Additionally, it enables applications to operate on multiple computing nodes and work with large amounts of data. The two important components that enable these characteristics are the Hadoop Distributed File System[6]

(HDFS) layer and the MapReduce (MR) parallel processing layer. The Hadoop architecture inherently provides support for managing the computing nodes in a cluster and works on reducing the traffic latency between server nodes[2,6].

HDFS is a native distributed file system implementation for storing large files and provides ready availability of data to all the nodes in a cluster. It is used to store all the application data and distribute these data blocks throughout the cluster and provides reliable access to data in the cluster during quick computations. It stores application data in a persistent way and dynamically manages data replication, availability and distribution across the nodes[7].

MapReduce is a programming framework for applications to process large amounts of data in parallel on distributed computing clusters[8]. In MapReduce, a map function reads the data and produces many key-value pairs. A reduce function takes these key-value pairs and consolidates all the values corresponding to a given key. The map and reduce functions have the capability of running independently on each key-value pair, extracting large amounts of inter-connected data[8].

Data Preparation and Analysis

A preliminary task was to analyze the dataset and understand the available information. The preparation of this required removal of inconsistencies in the data and erroneous entries. Also, the duplicate data had to be eliminated to ensure optimum storage. Another important attribute that needed preprocessing in the dataset is Timestamp. It had to be appropriately converted to the compatible format of the underlying data repository. Otherwise, It has to be saved as a string inside a table and has to be interpreted using UDF methods to convert it into a timestamp.

Once, the dataset is scrubbed and prepared for loading, it had to be appropriately imported into the data repository. The data repository consisted of a multi-node computational cluster with Hadoop installed on each node. The cluster configuration consisted of a master node and multiple worker nodes. The master node consisted of a NameNode and a JobTracker. The NameNode kept track of data within each of the worker nodes. The JobTracker breaks a processing job request into smaller tasks and allocates it to all worker nodes. Similarly, each worker node has a DataNode and TaskTracker. The DataNode stores the data and retrieves it on request from the master node. The TaskTracker runs the MR tasks assigned to the worker node and returns the result to the master node.

Firstly, the data files are imported into HDFS for effective use by the components inside Hadoop. Apache HBase[9] is used as a database for storing the data as it provides faster and random reads on data and is scalable to host large tables. It is a distributed column-based data store with an easy to use Java APIs for client access and supports both linear and modular scaling. The source CSV files are bulk imported into HBase database using the ImportTSV and CompleteBulkLoad utilities. ImportTSV will first create an indexed StoreFile, which is then bulk loaded into a HBase table using CompleteBulkLoad utility. As this utility is to import tab separated value (TSV) files, an appropriate column separator is defined in the command options to handle CSV files. As it is a column-oriented database, each table will have a column family under which each column is defined and should have a column defined as HBASE_ROW_KEY to uniquely identify each row in the table. Moreover, the source code of these utilities is readily available and can be customized as per the user needs and requirements.

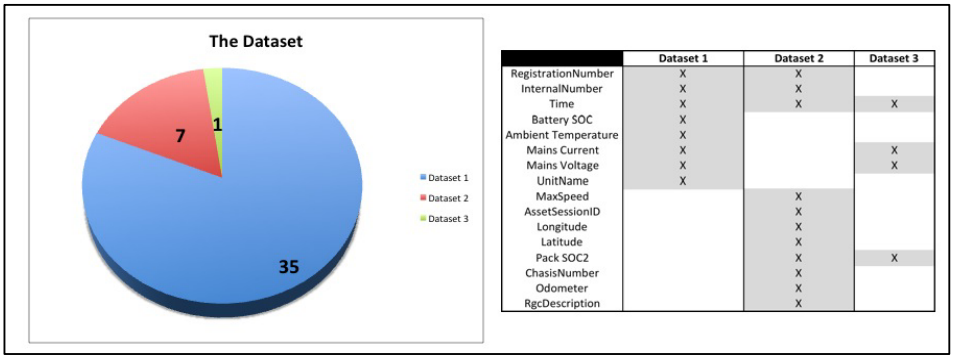
```
public class ReadTable {

   public static void main(String[] args) throws IOException {

      Configuration masterConfig = HBaseConfiguration.create();
      HTable masterTable = new HTable(masterConfig, "TABLE_NAME");

      //Write output to a file
      FileWriter fstream = new FileWriter("OUTPUT_FILEPATH");
      BufferedWriter out = new BufferedWriter(fstream);

      //Object to scan the table
      Scan s = new Scan();
      s.addColumn(Bytes.toBytes("COLUMN_FAMILY"), Bytes.toBytes("COLUMN_NAME"));

      //Interface for client-side scanning
      ResultScanner rs = masterTable.getScanner(s);
      try {
         //Read each result from the ResultScanner
         for (Result r = rs.next(); r != null; r = rs.next()) {
         byte[] columnObj = r.getValue(Bytes.toBytes("COLUMN_FAMILY"),
            Bytes.toBytes("COLUMN_NAME"));
         String columnValue = new String(columnObj);

         //Write the output into the buffer
         out.write(columnValue);
         out.newLine();
         }
      }
      finally {
      // Make sure to close the ResultScanner when done!
      rs.close();

      // Make sure to close the HTable when done!
      masterTable.close();

      // Make sure to close the buffered writer when done!
      out.close();
      }
   }
}
```

Figure 1: Sample Java program to read from a HBase table.

To perform data reads from the repository, appropriate and easy to use Java APIs are provided for programmatic access. Figure 1 shows a sample program to read data from HBase table and store it in an output file. It demonstrates the flexibility of performing reads on a table and writing the results in user defined formats. Also, this helps the user in customizing the data reads from the table and improves the query times. This inherently gives more control over the data to the user.

Also, HBase supports the MapReduce processing framework. Hence, queries can be done using mapper jobs for mapping the necessary attributes for the analysis. These mapped key-values can be consolidated using reducer jobs to calculate the needed results. The reducer jobs can be programmed to run a user defined algorithm on the mapped data and generate results. These result files can be loaded into any analytic tool and interpreted accordingly. The ability to self-program the query gives more control over the retrieved data and avoids complicated joins and manipulations of the read data. The flexibility of formatting the resulting output will improve the compatibility of this system with most analytic tools in the market.

Analysis on the dataset



	Dataset 1	Dataset 2	Dataset 3
RegistrationNumber	X	X	
InternalNumber	X	X	
Time	X	X	X
Battery SOC	X		
Ambient Temperature	X		
Mains Current	X		X
Mains Voltage	X		X
UnitName	X		
MaxSpeed		X	
AssetSessionID		X	
Longitude		X	
Latitude		X	
Pack SOC2		X	X
ChasisNumber		X	
Odometer		X	
RgcDescription		X	

Figure 2: Overview of the dataset

Initially, the data from files was analyzed to identify the attributes recorded by the EVs as shown in figure 2. Then, it was categorized based on the containing attributes. This helps in querying on the relevant dataset and save read times. To effectively meet the project requirements, the base fields useful for analysis were first identified. The important base fields recorded by each car in the dataset are,

- Vehicle Registration Number.
- Unique Session ID for the event recording (a session starts by wakeup and ends by going to sleep of the data logger).
- Time of the recording (in YYYY/MM/DD HH:MM:SS).
- Latitude and Longitude of vehicle GPS location (in degrees).
- MaxSpeed of the vehicle between recordings (in km/hr).
- Odometer reading in meters.
- State of charge (SoC) of the battery in % capacity remaining for this charging cycle.
- Ambient temperature (in degrees F).
- Battery mains current drawn in Amps.
- Battery mains voltage in V.

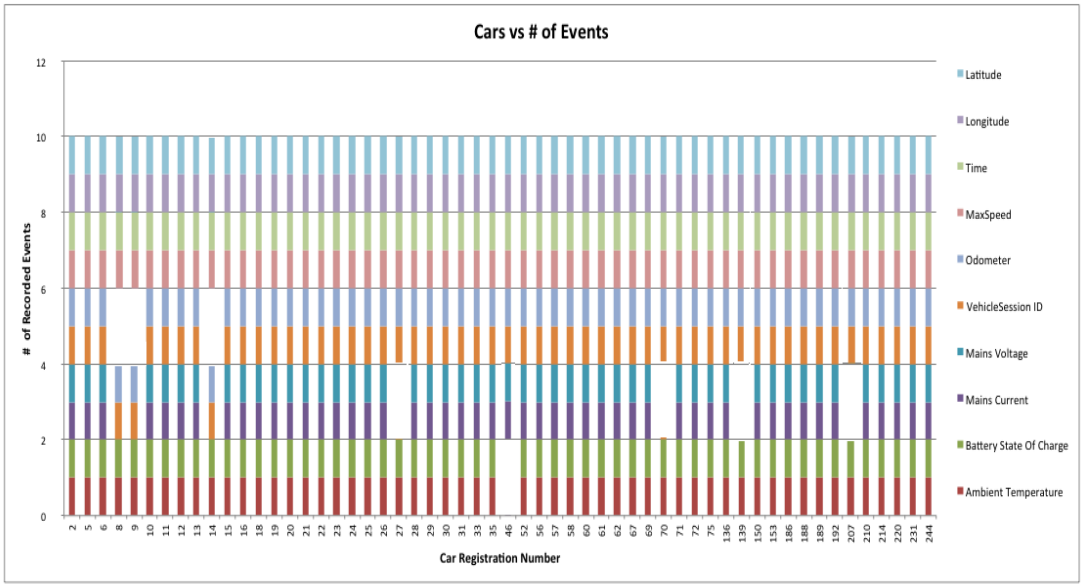


Figure 3: Types of events recorded in the dataset

Further analysis was done to identify the number of EVs recording the base data field. This can provide an insight into the type of data available for each EV. We observed the following (figure 3):

- 56 vehicles (each with a registration number plotted on X-axis) recorded various attributes.
- 8 vehicles recorded only 8 of the base fields (as shown by empty spaces in the chart).
- In total, 48 vehicles consistently recorded all the 10 base fields in the dataset.
- This information can be used as reference to avoid querying for a particular attribute on vehicles that are not recording it.

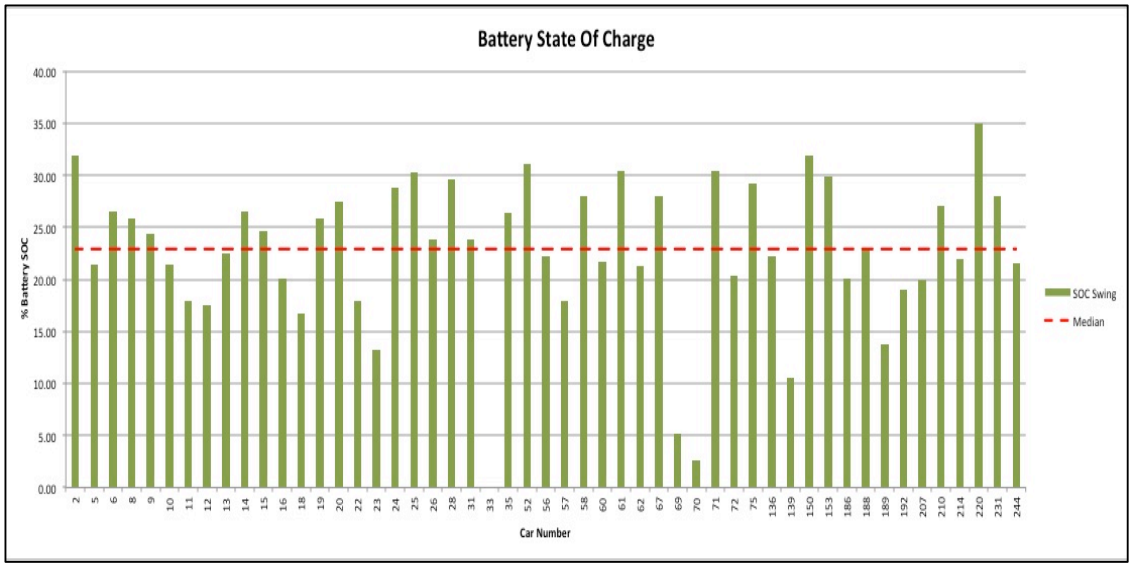


Figure 4: The battery usage trend among all the EVs

Figure 4 shows a statistical analysis done on the data to identify extent of battery capacity used by each EV. The SoC swing is defined as the difference between the maximum SoC and the average SoC recorded during all of the drive sessions of an EV. The dotted line is the median SoC swing value for all the participating EVs. This can help identify the vehicles that are using the battery extensively among all the participating EVs. Accordingly, the vehicles can be clustered into groups and any further analysis can be directed towards the relevant group to improve query effectiveness.

The above analysis demonstrates the capability of the system in exploring an unstructured dataset related to PEV drive cycles and understanding its structure. This can help plan the queries that can be done on the available data for deriving any hidden patterns and correlations between parameters.

Conclusion

The main purpose of this project has been to build a system that helps in facilitating EV data analytics. Hadoop is used as the platform for storing data and to perform analysis. This setup is configured over a cluster of multiple computing nodes each using HDFS for storing the datasets from the EV sensor loggers. MR jobs are used to perform parallel data processing and are programmable based on the user needs.

The main benefits conceived from the implementation of this repository are the following:

- Simplify the storage of data from heterogeneous sensors at one place and perform integrated data analysis.
- Real-time information retrieval and processing of large volumes of data to open up new possibilities.
- Statistical modeling and analysis of sensor data to identify hidden patterns, associations or relationships between parameters.
- Harness the data to identify factors that decide the strategic locations of PEV charging infrastructure.
- An opportunity for engineers to validate hypotheses related to components in an EV, using the available data in the repository.
- Engineers can develop new models to explain behavioral characteristics of an EV battery, through extensive analysis of sensor data.
- Anticipate and address the technical and practical challenges that emerge with the deployment of PEVs.
- Test new business models and gather essential information and rationale to support their implementation with better predictability.
- Enable user's capability to pre-process the data during the query as per their requirements.

Recent trends in computational engineering strongly advocate a non-relational model of data storage[10]. The need for providing efficient and scalable databases is the main contributing factor in this transition. Even though, the RDBMS model continues to exist, NoSQL database technologies will keep evolving to better address the storage and performance requirements[10]. This project demonstrates the capability of using Hadoop in designing a repository to handle scientific data effectively. The numerous capabilities of this system make it easier to manage unstructured data generated by PEV's data acquisition systems. From exploring the dataset to discovering the contained knowledge, this system is capable meeting the user requirements. The flexibility of customizing the data loading and data reading utilities of the system provides a greater control to the user.

Visualization is an important factor in harvesting information from data. This repository can be connected to a capable business intelligence (BI) tools available in the market, to visualize the data and analyze it. Tableau software[11] is one such tool to support visualization of data from a Hadoop repository. It supports various flavors of Hadoop distributions available in the market and ensures harnessing of information with minimum latency. The in-memory analytical engine of Tableau extracts data from a Hadoop cluster and supports faster and ad hoc reporting. It uses native connectors to plug into a Hadoop repository, which in-turn use Hive[12] for establishing the connection. Hive provides a structure to conveniently query the data inside HDFS using a SQL-like language called HiveQL[13]. Nevertheless, there are many visualization tools in the market that help analyze big data as per user needs and capabilities.

The possibilities of this system design is not just for performing EV data analytics but can be generalized to projects involving the handling of unstructured and massive datasets. These capabilities can be employed in other fields of engineering research involving similar data challenges. Moreover, this system uses open source tools that are readily available in the market at lower costs and have strong support from their respective developer communities. The field of engineering involves various types of experiments that generate huge amounts of unstructured data that needs human analysis for deriving any useful information. The interdisciplinary research done in the field of engineering can benefit from implementing

such open source systems for data analytics. This kind of system can help integrate data from different fields of engineering and identify any relationships and patterns between them. The knowledge discovered from such analytic systems can greatly influence new developments and can provide leverage to field of engineering education.

References

1. A. Cuzzocrea, I. Y. Song, and K. C. Davis. "Analytics over large-scale multidimensional data: the Big Data revolution," in *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*, on pages 101-104, ACM, October, 2011.
2. S. Amjad, S. Neelakrishnan, and R. Rudramoorthy. "Review of design considerations and technological challenges for successful development and deployment of plug-in hybrid electric vehicles," in *Renewable and Sustainable Energy Reviews*, *14*(3), on pages 1104-1110, 2010.
3. The Apache Software Foundation, "Apache Hadoop," http://hadoop.apache.org, February, 2014.
4. "A commercialization project of Energy Systems Network," http://www.energysystemsnetwork.com/project-summary-benefits, April, 2012.
5. "Why Think City?," http://thinkev.leftbankcompanies.com/why-think-city, December, 2013.
6. J. Shafer, S. Rixner, and A.L. Cox. "The Hadoop distributed filesystem: Balancing portability and performance," in *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium*, on pages 122-133, IEEE, 2010.
7. G. Attebury, A. Baranovski, K. Bloom, B. Bockelman, D. Kcira, J. Letts, T. Levshina, C. Lundestedt, T. Martin, W. Maier, H. Pi, A. Rana, I. Sfiligoi, A. Sim, M. Thomas, and F. Wuerthwein. "Hadoop Distributed File System for the grid," in *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, on pages 1056-1061. IEEE, October, 2009.
8. R. P. Padhy. "Big Data Processing with Hadoop-MapReduce in Cloud Systems," in *International Journal of Cloud Computing and Services Science (IJ-CLOSER),* 2(1), on pages 16-27, 2012.
9. The Apache Software Foundation, "Apache HBase," http://hbase.apache.org, February, 2014.
10. R.P. Padhy, M.R. Patra, and S.C. Satapathy. "RDBMS to NoSQL: Reviewing some next-generation non-relational databases," in *International Journal of Advanced Engineering Science and Technologies*, *11*(1), on pages 15-30, 2011.
11. "Tableau Software and Hadoop Analysis," http://www.tableausoftware.com/solutions/hadoop-analysis, December, 2013.
12. The Apache Software Foundation, "Apache Hive," http://hive.apache.org, February, 2014.
13. "Apache Hive Language Manual," https://cwiki.apache.org/confluence/display/Hive/LanguageManual, October, 2013.