Formal Specification of a Truck Geo-Location Big-Data Application

Ayman Naseem, Nadeem Akhtar, Malik Saad Missen
Department of Computer Science & IT, Baghdad-ul-Jadeed Campus
The Islamia University of Bahawalpur
Bahawalpur, Pakistan

***Abstract*—In the last few year's social networks, e-commerce, mobile commerce, and sensor networks have resulted into an exponential increase in data size. This data comes in all formats i.e. structured, un-structured and semi-structured. To efficiently extract useful information from these huge data sources is important. This information can play a central role in making future decisions and strategies. A truck geo-location big-data application integrated with formal model is proposed. The truck geo-location data is un-structured and it is accessed and manipulated by Hadoop query engine. Labelled transition system based formal model of the application is proposed to ensure safety and liveness properties of correctness.**

Keywords—Big-data; Formal methods; Correctness properties; Safety; Liveness; Internet-of-Things (IoT); MapReduce; Hadoop Distributed File System (HDFS); Finite State Processes (FSP); Labelled Transition System (LTS)

I. INTRODUCTION

During the last few years, with the increasing use of sensor networks and social networks, the data size has increased very rapidly. This huge amount of data is not easy to maintain and modify, and is of no use if not treated in an organized way. Conventional techniques are not sufficient for getting useful information from these huge amounts of data. Big-data sources are social networks, e-commerce, mobile commerce, and sensor networks. Relational database is a structured way of representing data. Big-data can be structured, un-structured, or semi-structured; and data-type cannot be easily defined. Therefore, managing and processing un-structured data is different than structured data. Big-data cannot be processed and managed by using traditional algorithms and techniques. It is huge in size and is generated at a much faster rate than traditional data. In order to efficiently process big-data, appropriate processing capabilities, proper hardware, with efficient algorithms are required. For processing, analyzing, and transferring big-data large space and high bandwidth is required. Data science deals with the huge amount of data i.e. data retrieval, data processing and data manipulation.

The first and the most challenging task is to ensure correctness of the methodology used for the extraction and manipulation of big-data i.e. verification of the process of data extraction and analysis. The step of verification is important and it affects the whole process of data analysis. The verification of data extraction methods and techniques can be made more reliable by introducing formal methods and techniques. Formal methods and techniques can be applied for accessing, searching, and storing big-data. Statistical analysis can be performed on big sets of data to extract useful information for future decision making, and proposing future roadmaps and strategies.

Why Big-Data?

With the advent of social media, e-commerce, and Internet of Things, use of sensor devices have increased and the data size are huge. The data generated can be dig up and proper meaning can be extracted for making future planning. Big-data extraction and analysis make a business follow pro-active approach. A pro-active business analyzes the current and past situation of business and also compare the changes made by other businesses and prepare the new business policy to be launched. Big-data analysis helps a business to make better future strategies, policies, and decisions. Each big-data set contains huge amounts of data and correct methods are required for the extraction and analysis of these big-data sets.

Why Formal methods?

Formal methods are based on mathematical logics, proofs and formulas. These methods give precise and accurate answers. Use of formal methods on each step of data analysis minimize significantly the probability of errors.

II. MOTIVATION

Data is the most important part of any organization. There are huge sets of sensor data, data size are huge and data growth is fast. Big-data has a tremendous scope, and by performing statistical analysis of huge quantities of data; future forecasts can be made; better strategies can be proposed; and businesses can be improved.

The use of formal methods for specification and verification of the process of data extraction is very recent. Now big-data is an important aspect of data science, and formal verification has its own importance. There is a need to propose correct methods for the analysis and design of big-data based applications. In this paper, formal specification and verification for big-data based truck-geolocation application is proposed.

III. PROBLEM STATEMENT

Formal modeling and verification of a system that generates, extracts, and analyzes massive sets of data. This work addresses the following research questions:

RQ-1*: Why correctness is important in the process of big-data storage, retrieval, and analysis?*

RQ-2*: How the process of big-data storage, retrieval, and analysis can be formally verified?*

RQ-3*: How the process of big-data storage, retrieval, and analysis can be made correct?*

A truck geo-location system is formally modelled in the form of a labelled transition system and verified. Queries are applied to massive sets of data and extracted data is analyzed. For this purpose Apache Hadoop is used. Apache Hadoop is an open source framework for the processing of big-data in a distributed environment. Hadoop has two main and distinct features. These features are Hadoop Distributed File System (HDFS) and MapReduce. HDFS is a distributed file system which provides scalable and reliable storage medium. It stores huge amount of data and is reliable. It can easily deal up to 200 PB of data and can easily be scalable up to 4500 severs. MapReduce consists of two further functions of Map and Reduce. Statistical analysis of big-data based truck geo-location system is proposed, then formal specification and verification of the methodology is proposed.

IV. State of the Art

A. Formal methods

Formal methods have a mathematical foundation. The fundamental use of formal methods is in ensuring correctness. They provide a concise and precise representation and proofing of a software model. Formal methods and techniques use algorithms, logics, predication, proofs, propositional calculus and first-order predicate calculus.

A formal language is an alphabet of symbols and a set of grammar rules used to construct well-formed formulas from the alphabet [18]. A broad view of formal methods includes all applications of (primarily) discrete to software engineering problems. Formal application involves modeling and analysis where the models and analysis procedures are derived from or defined by an underlying mathematical precise foundation [19]. Abstract State Machines, B-method, Event-B and Colored Petri Nets are some examples of formal languages. Colored Petri-net is a mathematical and graphical modeling technique widely used for specification and verification purpose [20].

Formal verification focuses on safety and liveness properties of correctness. Both of these two properties are critical to assure correctness. They complete each other (i.e. safety or liveness alone is not sufficient to ensure system correctness). Liveness property relates with the execution and working of a model and is concerned with a program eventually reaching a good state. Safety property assures that nothing bad will happen in the model. Safety is concerned with a program not reaching a bad state [21]. Labelled Transition System (LTS) [21] is a finite state machine which specifies and verifies the functional behavior. It models a system in the form of states and transitions. It consists of all the states which a component can reach and all the possible transitions that it may perform. It mechanically checks that a system is satisfying all the mentioned properties.

B. Big-data

Big-data is huge in size that use special processing algorithms and methods that are capable of processing peta-bytes of data within finite time. Big-data has to be analyzed for making future strategies and decisions. Analyzing the data in minimum time is important. Data is the most important component in computer science. Data science is the field which came in to existence after the peta and zeta-bytes of data. Data science is rebranding of computer science and applied statistical skills [1].

Data size is increasing day by day due to Internet-of-Things (IoT), social media, and e-commerce. IoT enables the physical objects to connect wirelessly and communicate with one another using sensors. These devices share million bytes of data within few seconds. This communication generates huge amounts of data continuously. IoT are a major source of big-data, communicating massive amounts of streamed information from billions of Internet-Connected Objects (ICO’s) [2]. With the use of IoT, the number of challenges have also increased. These challenges range from capturing and storing data; processing and analyzing the captured data and managing communication in such a way that those users can seamlessly search, find and utilize their data [3].

Big-data analyzed the data to find and extract out the useful information. The patterns and trends of data have to be studied for analyzing and gaining knowledge. The whole designed process of finding patterns in big sets of data is data mining. In data mining the exact groups and patterns of data are searched and extracted [4]. According to NIST [5] in big-data the data volume, acquisition speed, data representation limits the capacity of using traditional relational methods to conduct effective analysis or data which may be effectively processed with important horizontal zoom technologies.

Big-data requires new technologies and architecture for its processing and analysis purpose, and it cannot be processed and managed using traditional relational database models [6]. In a big-data based system the data generation speed is more than its storing and capturing speed [7], therefore data size is huge in every dimension, and normal algorithms are not suitable for data processing. Big-data is a comprehensive term for all the huge and complex sets of data which are not able to store, process and curate under traditional means [8]. Essence of big-data is not only massive data processing, but also optimization of real world knowledge which is extracted from such a huge data [9]. [10] Define big-data as 3 V’s model. The 3 V’s are its *volume*, *velocity* and its *variety*. [11] Define big-data as 4 V’s and these are *volume*, *variety*, *velocity* and *value*. *Complexity* can also be added to these 4 V’s. Complexity means that storing and capturing of big-data is not easy. The fast speed and huge size make big-data complex to process.

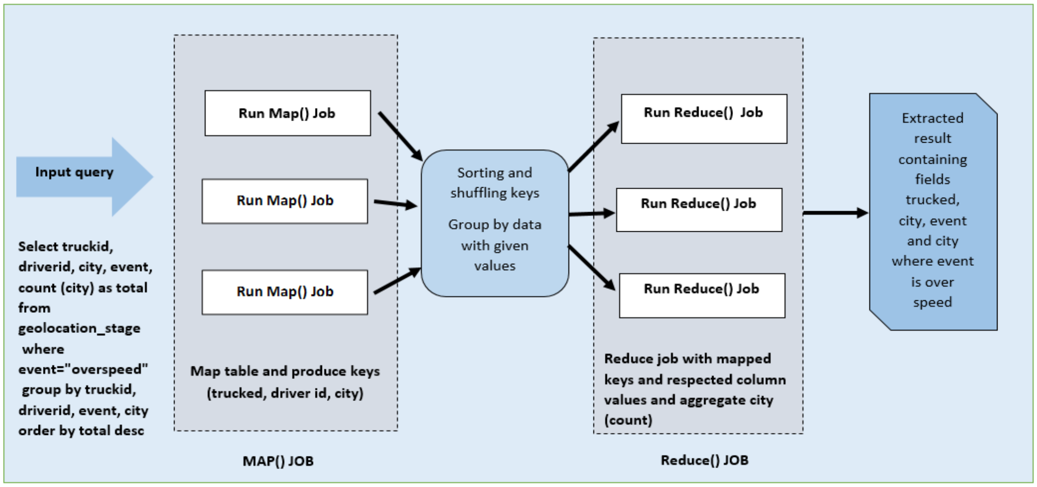

Input query
Select truckid, driverid, city, event, count (city) as total from geolocation_stage where event="overspeed" group by truckid, driverid, event, city order by total desc
Run Map() Job
Run Map() Job
Run Map() Job
Map table and produce keys (trucked, driver id, city)
MAP() JOB
Sorting and shuffling keys
Group by data with given values
Run Reduce() Job
Run Reduce() Job
Run Reduce() Job
Reduce job with mapped keys and respected column values and aggregate city (count)
Reduce() JOB
Extracted result containing fields trucked, city, event and city where event is over speed

Fig. 1. Parallel model for Data Extraction and Formal Verification

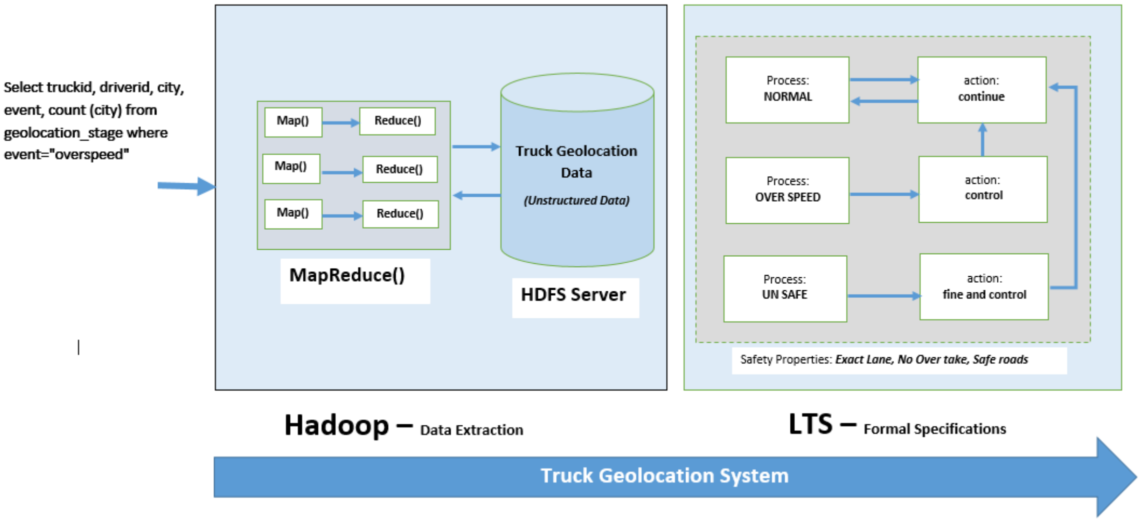


Fig. 2. Data mapping and reduction

Big-data describe the voluminous amount of structured, unstructured (i.e. data from social media, sensors, research, online shopping, scientific application and surveillance) and semi-structured (i.e. xml) [12]. Big-data defines a set of data that actually grows exponentially, that is too raw and too unstructured and difficult to process by the conventional methods [13]. Big-data not only handles and utilizes large amounts of data but also utilizes various types of data including unstructured data and attributes [14]. Major challenges in big-data environment is its security and privacy. The process of analyzing data by applying theorems, procedures, and algorithms for finding patterns and to extract accurate required knowledge is known as data analytics. Analyzing data is more challenging than data organizing, managing, storing, and processing [15]. The scalability of the data analytics process i.e. whether data analytics process scales as data sets increases is a major issue [16]. Data has no meaning and possess very low value by itself. Many organizations are collecting different types of data of very high volume, at a very high velocity. This collected big-data is analyzed using big-data analytics. As a result, organizations can get deeper knowledge about their business and customers' behavior, eventually making better forecasting and decisions [17].

MapReduce plays a vital role in the processing and analysis of big-data. Its features of fault-tolerance, scalability, and simplicity make it useful for processing big-data by using its two main functions of map() and reduce(). The explosive growth of big-data processing imposes a heavy burden on computation, storage, and communication.

V. Big-Data Based Truck Geo-Location System

A. *Proposed Approach*

The main purpose of data extraction is to give data a proper meaning. Some of the techniques which are followed to analyze big-data are Association Rule Learning, Data Mining, and Cluster Analytics. All of these techniques are adopted with the aim to analyze data to get the desired results.

Model checking is applied to get *correctness*, *preciseness*, and *accuracy*. They can be applied to any extent to any step in the data analysis process. A model has been proposed in which formal specification for the truck geo-location system is proposed. A user generates a query and graph for the extracted results. Finite State Processes based Labelled Transition System verifies the processes of data extraction and check all possible paths.

B. Requirement Elicitation and Data Extraction

The main purpose of modeling a verified system is to eliminate the errors and chances of failure at very initial level. This system is proposed to enhance the use of formalism in every step of big-data analysis. A query without verification leads to false extraction of data or may disturb the whole process of data analytics from the start of procedure. The goal of big-data analytics is to provide the useful and required information from the huge sets of data. This information helps to make statistical analysis which leads to future strategies and decisions. The formally verified big-data analytics provide the true means of data extraction and decisions based on the extracted results.

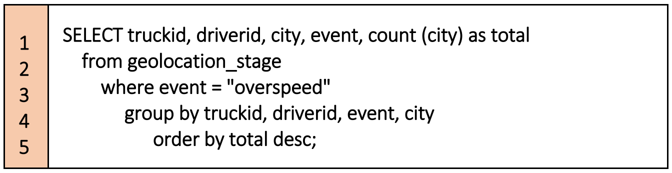
Hadoop divides the input stream into smaller independent chunks which are processed in a concurrent manner. The MapReduce paradigm sorts the output of the mapped data. Hadoop works with a number of components. These components work together to facilitate the whole procedure. Major components of Hadoop are HDFS (Hadoop Distributed File System), MapReduce, Ambari, Yarn, and Hive. HDFS provides access to application data and it’s very useful for the applications that contain huge data sets. It is like a distributed file system and it provides an environment of shared resources that can be accessed simultaneously. It provides a fault-tolerant system, and it is particularly designed to be deployed on low cost hardware. It has a common Master-Slave architecture. It has a single, master name-node which not only manages the whole file system but also controls the access to file system by different clients. It also manages its slave-nodes which are known as data-nodes. Data-nodes are responsible for managing data storage related to the system. Data-nodes also perform various tasks including creation, deletion and replication of data blocks as instructed by the name-node. Second major component of Hadoop is MapReduce. MapReduce is responsible for parallel processing of huge data sets. It works in a reliable and fault-tolerant manner. MapReduce framework divides a huge set of data into independent chunks. These chunks are then processed by map tasks. After processing the framework sorts the output. This sorted output becomes the input for the Reduce task. Reduce task perform the summarizing operations on the data.

C. Formal modelling

The proposed model focuses on the use of formal specifications with Hadoop. A query which is deployed on Hadoop system is modelled in LTS. The possible states and transitions of data extraction process is modelled. All the conditions mentioned in query can be represented as processes and actions of FSP which generates LTS. These processes are constituted by series of actions that define behavior. Once the query is modeled, the next step is to execute it using the Hadoop framework. Hortonworks is one of the implementation of Hadoop. The query is deployed on Hadoop using Hortonworks. This query generates the required set of results from the bulk of data.

Big-data analysis is important as it helps in future predictions, future policies and business decisions. Future policies for the trucks travelling in different cities of a state can be proposed.

The data sets basically contain two large log files. First log file named geolocation contains all the data related to the location with the related truck. It contains 10 fields including *truck id*, *driver id*, *event*, *latitude*, *longitude*, *city*, *state*, *velocity*, *event_ind* and *idling_ind*. It contains huge amount of data sets of related trucks and area. The other log file named trucks contains all the data sets related to the trucks that are travelling in these areas. It contains more than 90 fields containing truck id, driver id, model and the gas consumption by every truck from January 2009 to June 2013. Both of log files are actually linked with truck id and driver id. Each driver has its own truck which can travel in any area. Event field in geolocation actually represent the speed limit of truck. It has five instances normal, over speed, unsafe following distance, unsafe tail distance and lane departure. This information can help to figure out the truck that are doing over speed, and can also make an analysis about the trucks and the respective truck drivers. Analysis can also be done by taking the longitude and latitude to trace the exact location which is travelled most of the times. This analysis helps to find which road is frequently used and it also helps to make new policies while planning and making new road in a specific area.

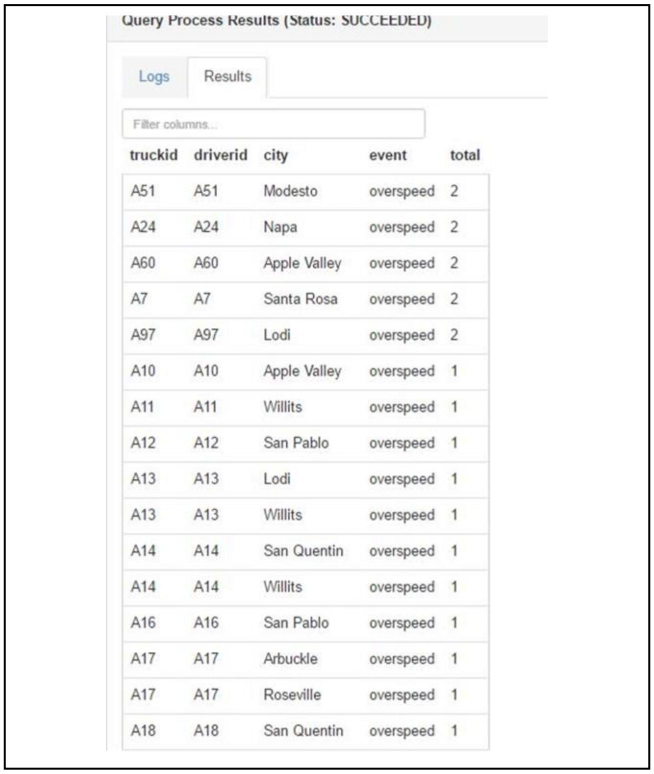


```
SELECT truckid, driverid, city, event, count (city) as total
   from geolocation_stage
      where event = "overspeed"
          group by truckid, driverid, event, city
               order by total desc;
```

The event field in log file geolocation mentioned different events occurred during driving. This includes five instances. A query can be applied to find out which truck is doing over-speed maximum time and at which location. This information can further be used to take an action against the drivers who are doing in an unsafe manner. Future security measures can be applied on the generated data.

This query fetches the truck id, diver id, city, event and total city count where drivers did over speed. The data is arranged in the descending order and grouped by truck id, driver id, event and city.

The actual process of Map and Reduce can also be monitored and displayed in the Hortonworks. This shows how the data is fetched using MapReduce() function. The log files contain thousands of rows and huge amount of data. From this data the required precise information is extracted.



Query Process Results (Status: SUCCEEDED)

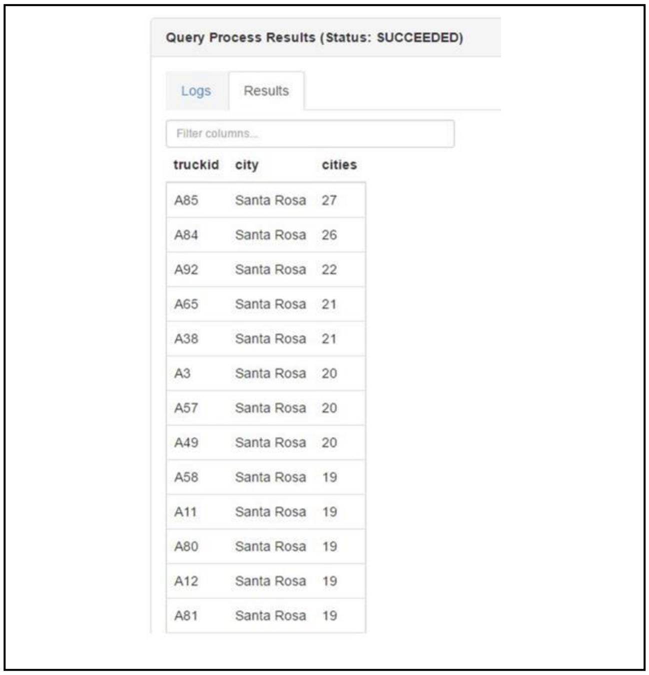
truckid	driverid	city	event	total
A51	A51	Modesto	overspeed	2
A24	A24	Napa	overspeed	2
A60	A60	Apple Valley	overspeed	2
A7	A7	Santa Rosa	overspeed	2
A97	A97	Lodi	overspeed	2
A10	A10	Apple Valley	overspeed	1
A11	A11	Willits	overspeed	1
A12	A12	San Pablo	overspeed	1
A13	A13	Lodi	overspeed	1
A13	A13	Willits	overspeed	1
A14	A14	San Quentin	overspeed	1
A14	A14	Willits	overspeed	1
A16	A16	San Pablo	overspeed	1
A17	A17	Arbuckle	overspeed	1
A17	A17	Roseville	overspeed	1
A18	A18	San Quentin	overspeed	1

Cities visited maximum times

The query extracts the total number of times a city is visited. It shows the city with maximum visits at the top. It shows which truck visited which specific city with total number of visits. This information facilitates the development of new roads infrastructure. Cities with the maximum truck visits can be shortlisted, therefore useful information like if more roads are needed to that specific city can be found out.

```
SELECT truckid, city, count(city) as cities
   from geolocation_stage
      group by truckid, city
          order by cities desc;
```

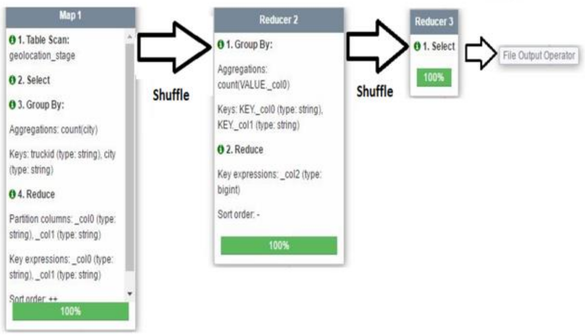
The query fetches the truck id, city and count (i.e. total number of visits) of a particular city by a specific truck. The result is arranged in descending order with respect to count and also the data is grouped by truck id and the city so that each row will depict which truck visited which city and how many times.



Query Process Results (Status: SUCCEEDED)

truckid	city	cities
A85	Santa Rosa	27
A84	Santa Rosa	26
A92	Santa Rosa	22
A65	Santa Rosa	21
A38	Santa Rosa	21
A3	Santa Rosa	20
A57	Santa Rosa	20
A49	Santa Rosa	20
A58	Santa Rosa	19
A11	Santa Rosa	19
A80	Santa Rosa	19
A12	Santa Rosa	19
A81	Santa Rosa	19

The MapReduce operation performed in the query is also visualized in a map or tree structure to show the actual operations performed on the query. It displays each step of Map() and Reduce() operations performed on the data.


Map 1
1. Table Scan: geolocation_stage
2. Select
3. Group By:
Aggregations: count(city)
Keys: truckid (type: string), city (type: string)
4. Reduce
Partition columns: _col0 (type: string), _col1 (type: string)
Key expressions: _col0 (type: string), _col1 (type: string)
Sort order: ++
100%
Shuffle
Reducer 2
1. Group By:
Aggregations: count(VALUE._col0)
Keys: KEY._col0 (type: string), KEY._col1 (type: string)
2. Reduce
Key expressions: _col2 (type: bigint)
Sort order: -
100%
Shuffle
Reducer 3
1. Select
100%
File Output Operator

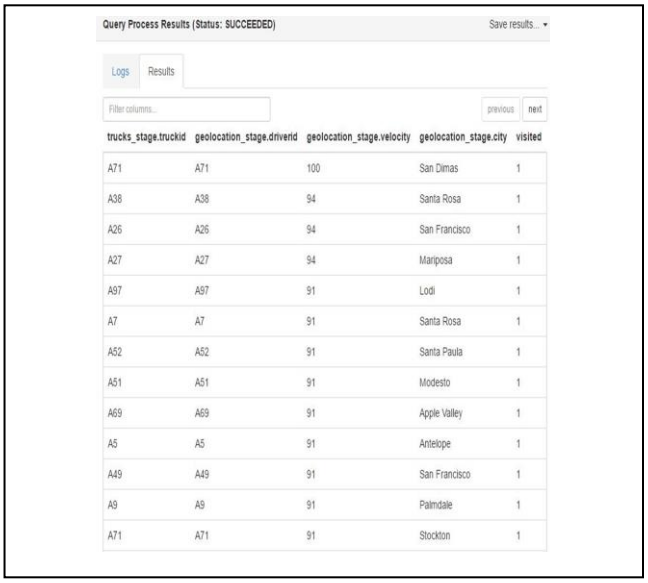
Graph can also be made to analyze quickly from the extracted data that can give a complete understanding of data. In the graph, it can be seen clearly which truck is visiting which city for how many times.

Truck moving with the highest speed

```
SELECT trucks_stage.truckid, geolocation_stage.driverid,
  geolocation_stage.velocity, geolocation_stage.city,
    count (geolocation_stage.city) as NoOfVisits
      from trucks_stage, geolocation_stage
        where trucks_stage.truckid = geolocation_stage.driverid
        group by trucks_stage.truckid, geolocation_stage.driverid,
          geolocation_stage.velocity, geolocation_stage.city
              order by geolocation_stage.velocity desc;
```

This query identifies the truck that is moving with highest speed. It shows the respective drivers. It helps to make complete analysis about the truck drivers by showing their trucks, velocity and the cities they visited

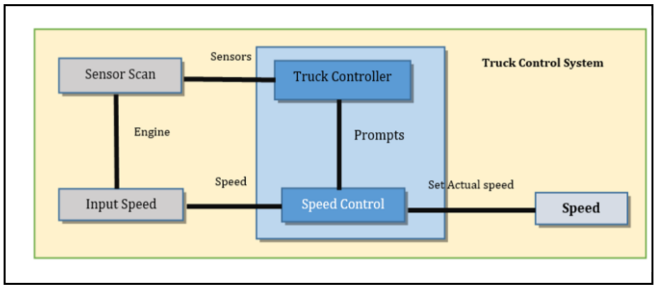
The query joins geolocation data and truck data by using 'when' condition with truck id. As truck id is making a link between the two. After joining the two tables, the query fetches the mentioned fields (truck id, driver id, velocity and city). This result leads to a complete analysis about truck drivers, their velocity and cities where they travelled. This result of query is basically focusing on the information of truck drivers.



trucks_stage.truckid	geolocation_stage.driverid	geolocation_stage.velocity	geolocation_stage.city	visited
A71	A71	100	San Dimas	1
A38	A38	94	Santa Rosa	1
A26	A26	94	San Francisco	1
A27	A27	94	Mariposa	1
A97	A97	91	Lodi	1
A7	A7	91	Santa Rosa	1
A52	A52	91	Santa Paula	1
A51	A51	91	Modesto	1
A69	A69	91	Apple Valley	1
A5	A5	91	Antelope	1
A49	A49	91	San Francisco	1
A9	A9	91	Palmdale	1
A71	A71	91	Stockton	1

Labelled Transition System based verification

Verification ensures system correctness. The use of labelled transition system ensures correctness of the truck geolocation system. As these trucks are moving in different cities, a model has been proposed that checks and verifies the working of trucks and maintains the pattern on which they are moving. A model based on labelled transition system is proposed. The model is constituted of processes and actions. Well-defined requirements lead to a well-defined model. This formal model specifies the properties of liveness and safety. *Liveness property* assures that system will work and *safety property* assures that nothing bad will happen in the model. The process SAFETY assures that the truck geolocation system is working.

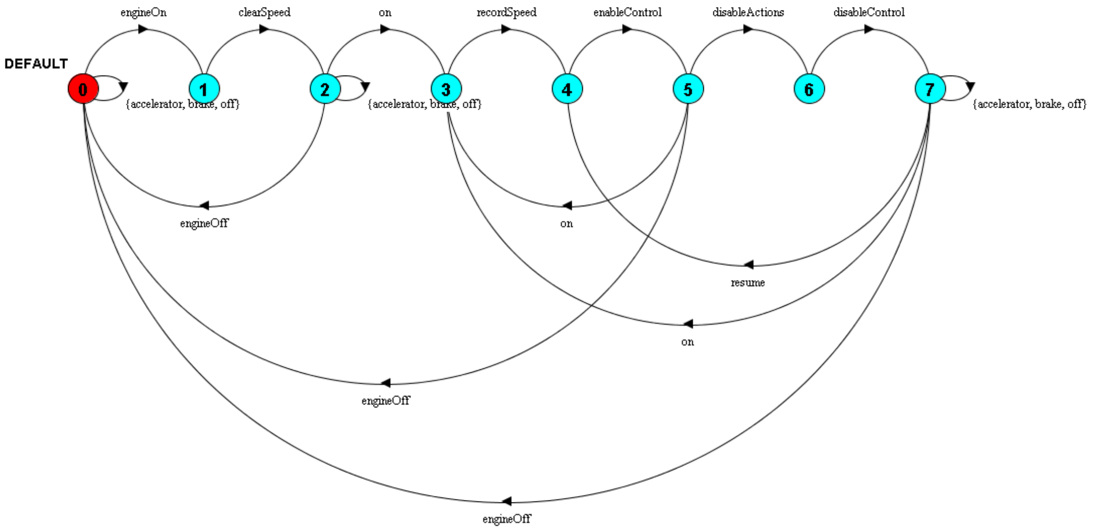


Model - Structure and actions

Model - Design, Process and Actions Truck Controls model

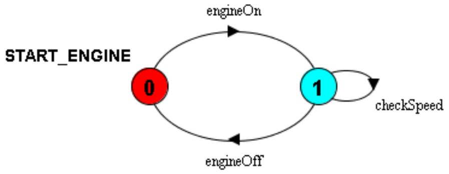
```
////  Engine Control Actions  ////
set ControlActions = {off, brake, accelerator}

TRUCK_CONTROLLER = INACTIVE,
INACTIVE = (engineOn -> clearSpeed -> ACTIVE
                  | {ControlActions} -> INACTIVE),
ACTIVE = (engineOff -> INACTIVE
                  | on -> recordSpeed -> enableControl -> MOVING
                  | {ControlActions} -> ACTIVE),
MOVING = (engineOff -> INACTIVE
                  | disableActions -> disableControl -> STANDBY
                  | on -> recordSpeed -> enableControl -> MOVING),
STANDBY = (engineOff -> INACTIVE
                  | resume -> enableControl -> MOVING
                  | on -> recordSpeed -> enableControl -> MOVING
```

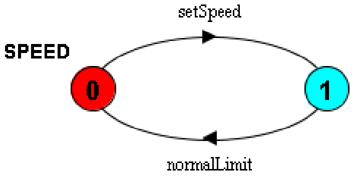
```
                | {ControlActions} -> STANDBY).
```

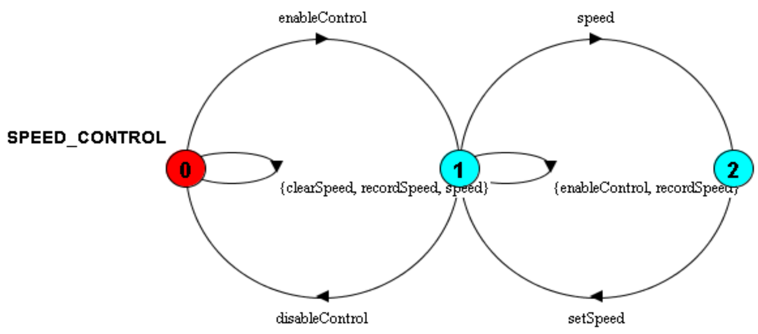


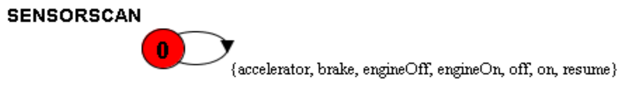
Truck Speed Sensors model

```
//// Truck Sensors ////
set Sensors = { engineOn, engineOff, on, off, resume, brake, accelerator }
//// Engine States ////
set Engine = { engineOn, engineOff }
//// Scan the Sensors ////
SENSORSCAN = ({Sensors} -> SENSORSCAN).
//// Start Truck Engine ////
START_ENGINE = (engineOn -> CHECK_SPEED),
//// Check Truck Speed ////
CHECK_SPEED = (checkSpeed -> CHECK_SPEED | engineOff -> START_ENGINE).
SPEED = (setSpeed -> normalLimit -> SPEED).
//// Truck Speed Control  ////
SPEED_CONTROL = DISABLED,
DISABLED = (  {speed, clearSpeed, recordSpeed} -> DISABLED
             | enableControl -> ENABLED ),
ENABLED = (  speed -> setSpeed -> ENABLED
           | {recordSpeed, enableControl} -> ENABLED
           | disableControl -> DISABLED).
```



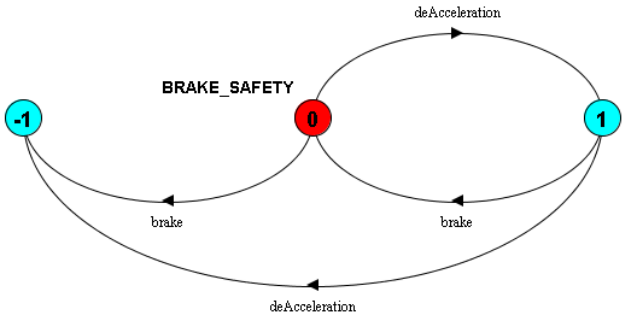


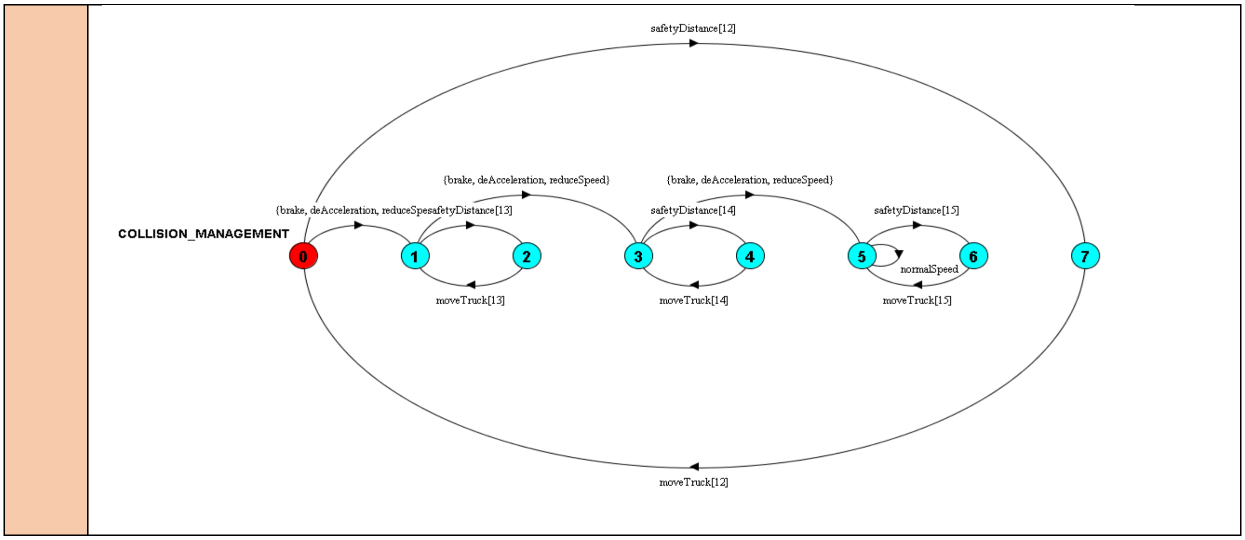

SPEED_CONTROL
0
1
2
enableControl
speed
{clearSpeed, recordSpeed, speed}
{enableControl, recordSpeed}
disableControl
setSpeed



Truck Safety properties – Brake safety and Collision avoidance properties

```
property BRAKE_SAFETY = (deAcceleration -> brake -> BRAKE_SAFETY).

set DecreaseSpeed = {reduceSpeed, brake, deAcceleration}
const MinDist = 12   /// minimum safety distance
const MaxDist = 15   /// maximum safety distance
range R = MinDist..MaxDist

COLLISION_MANAGEMENT = COLLISION_AVOIDANCE[MinDist],
COLLISION_AVOIDANCE[r:R] =
      ( safetyDistance[r] -> moveTruck[r] -> COLLISION_AVOIDANCE[r]
| when(r >= MinDist && r < MaxDist) {DecreaseSpeed} -> COLLISION_AVOIDANCE[r+1]
| when(r == MinDist) {DecreaseSpeed} -> COLLISION_AVOIDANCE[r+1]
| when(r == MaxDist) normalSpeed -> COLLISION_AVOIDANCE[r]).
```





VI. Results and Discussion

State space evaluation checks all the possible paths in a system and guarantees a system with zero chances of failure. Big-data based applications generates huge and massive amounts of data that is used for making future predictions, so it is necessary to verify such applications using formal methods which will ensure the correct working of model.

Therefore, in order to have reliable data labelled transition system based model-checking technique is used.

VII. Summary and Contributions

A truck geolocation system based on big-data is considered. This system is based on large sets of data of goods transport trucks and their routes. This data includes the *truck velocity*, *cities visited*, *fuel consumption* and *the driving pattern of each truck driver*. The information is stored in two log files named *geolocation* and *trucks*. These files contain the data that are of huge size and cannot be used directly to get any information. Furthermore this system of trucks was not verified at all. Queries are applied on the data on the basis of our requirement. For making analysis and big-data Hadoop framework is used.

In order to verify the working pattern of truck system, a labeled transition system is proposed. Each possible state is checked. Analyzing and verifying the whole system by using labelled transition system based model checking generate accurate results which can be further used in analysis.

The data generated from a big-data application is in semi-structured, or non-structured form, getting useful information from this data is important. This analysis extracts information required for making future predictions.

Big-data applications are important for large business organizations to make future predictions. It's important to verify the data extraction and analysis method by using model checking. For giving the exact meaning to data, data needs to be extracted on the bases of required criteria. After extraction useful analysis could be performed on the data sets. These analysis can be used to make future reports and policies. Businesses can predict about their future policies, and make strategies regarding their future roadmaps. An important aspect of the use of big-data technique is the formal verification of the methodology used for *data extraction*, *data analysis*, and *information retrieval* from this data.

References

[1] D. Kumar, "Data Sciences overtakes Computer Science?", ACM Inroads,Vol.3, No. 3, September 2012.

[2] A. Zaslavsky, "Internet of things and ubiquitous sensing", Computing Now, vol. guest editor's introduction, 2003.

[3] Sulayman K. Sowe, Takashi Kanata, Mianxiong Dong, Koji,Zettsu, "Managing Heterogeneous Sensor Data on a Big Data Platform: IoT Services for Data intensive science", IEEE 38th Computers, Software and Applications Conference workshops, 2014.

[4] Olga Kurasova, Virginijus Marcinkeviˇcius, Viktor Medvedev, Aurimas Rapeˇcka and Pavel Stefanoviˇc, "Strategies for Big Data Clustering", IEEE 26th International Conference on Tools with Artificial Intelligence, 2014.

[5] Big Data and the Internet of the Things, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Web Link: http://rdi2.rutgers.edu/sites/rdi2/files/img/Greer_Rutgers_BigData_Apr_2014.pdf, [accessed on November 5, 2015].

[6] Min Chen, Shiwen Mao, Yunhao Lin, "Big Data: A survey", Springer Science+Business Media New York 2014, January 22, 2014.

[7] Olga Kurasova, Virginijus Marcinkeviˇcius, Viktor Medvedev, Aurimas Rapeˇcka and Pavel Stefanoviˇc, "Strategies for Big Data Clustering", IEEE 26th International Conference on Tools with Artificial Intelligence, 2014.

[8] Erin-Elizabeth A Durham, Andrew Rosan and Robert W. Harrison, "A model architecture for Big Data Applications using Relational Databases", IEEE International Conference on Big Data, Washington D.C, 2014.

[9] Takafumi Nakanishi, "Toward a Realization of Knowledge Creation Grid for Big Data Era", IIAI 3rd International Conference on Advanced Applied Informatics, 2014.

[10] Doug Laney, "3D Data Management: Controlling Data Volume, Velocity and Variety", Application Delivery Strategies by META Group Inc. (2001, February 6), p. 949 Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

[11] Palak Gupta and Nidhi tayagi, "An Approach towards Big Data: A Review", International Conference on Computing, Communication and Automation (ICCA), 2015.

[12] Konstantinos Slavakis, Georgios B. Giannakis and Gonzalo Mateos, "Modeling and Optimization for Big Data Analytics", IEEE signal Processing Magazine, September 2014.

[13] Janusz Wielki, “Implementation of Big Data in organization-Possibilities, Impediments and challenges”, Proceedings of the 2013 federated Conference on Computer Science and Information System, PP. 985-989, 2013.

[14] Matturdi Bardi, ZHOU Xianwei and Li Shuai, Lin Fuhong, “Big Data Security and Privacy: A Review”, Big Data Cloud and Mobile Computing, China Communications, Supplement No.2, 2014.

[15] Alfredo Cuzzocrea, “Privacy and Security of Big Data: Current Challenges and Future Research Perspectives, PSBD’14, November 7, 2014, Shanghai, China, 2014.

[16] Stephen Kaisler, Frank Armour, J. Alberto Espinose and William Money, “Big Data: Issues and Challenges Moving Forward”, 46th Hawaii International Conference on System Science, 2013.

[17] Ankur Balar, Nikita Malviya, Swadesh Prasad, Ajinkya Gangurde, “Forecasting consumer behavior with innovative value proposition for organizations using Big Data Analytics”, IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[18] Perry Alexandra, “Best of both worlds Combining formal and semi-formal methods in software engineering”, IEEE Potentials, 1995.

[19] Levenson NG, “Formal methods in software engineering”, IEEE transactions in Software Engineering, Vol 16. No. 19, 1990.

[20] Hossam A. Gabbar, “Fundamentals of Formal Methods”, Modern Formal Methods and Applications, Springer Science and Business Media, Chapter 1, 2006.

[21] Jeff Magee, Jeff Kramer “Safety and Liveness Property”, Concurrency: state models & Java programs, Willey, Chapter 7, 2006.