A REAL-TIME DATA STREAM PROCESSING MODEL FOR A SMART CITY APPLICATION LEVERAGING INTELLIGENT INTERNET OF THINGS (IOT) CONCEPTS

A Thesis Presented to the Department of Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of

MASTER of Computer Science

By

Yakubu Mulikatu Ibrahim

Abuja, Nigeria

December, 2017

CERTIFICATION

This is to certify that the thesis titled “A real-time data stream processing model for a smart city application leveraging intelligent internet of things (IoT) concepts” submitted to the School of Postgraduate Studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of Master's degree as a record of original research carried out by Mulikatu Yakubu Ibrahim in the Department of Computer Science.

A REAL TIME DATA STREAM PROCESSING MODEL FOR A SMART CITY APPLICATION LEVERAGING INTELLIGENT INTERNET OF THINGS (IoT) CONCEPT

By

Mulikatu Yakubu Ibrahim

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED _____

Supervisor, Professor. Ekpe Okorafor

_____

Head, Department of Computer Science

APPROVED _____

Chief Academic Officer

9th December 2017 _____

Date

ABSTRACT

Due to the vast amount of data that is being generated by the sensors through the smart devices in smart cities, streams of data must be processed in real time to gain insight quickly and to make decisions that are in most cases critical and time sensitive. The difficulty is diminished by using big data methods such as Cassandra, Hadoop, Kafka and Spark to perform real-time stream processing in an Internet of Things (IoT) environment, such as traffic monitoring in a smart city environment. Among the different dimensions that improve the quality of life of people in a smart city, one of the very important one is transportation. Intelligent Traffic Monitoring System (ITMS) in a smart city, monitors traffic by detecting and displaying what is occurring on a particular road. In this thesis, a real-time data stream processing model was developed and used data streaming trends to monitor traffic in an ITMS.

Keywords: Smart cities, Real-time processing, Intelligent Traffic Monitoring System, real-time data stream processing model

ACKNOWLEDGEMENT

In the name of Allah, most gracious, most merciful, my gratitude goes to Allah subhaanahu wataála for seeing me through this thesis. I also extend my gratitude to my lovely parents, Mallam Yakubu Ibrahim Dakata and Mallama A'isha Yakubu for their support.

I also extend my sincere gratitude to African University of Science and Technology (AUST), Abuja for giving me the opportunity to have access to their exceptional facilities and faculties as I studied for this degree.

My appreciation goes to the African Capacity Building Foundation for giving me a scholarship to study my master's degree. I thank you for your support.

Additionally, I wish to extend my gratitude to the staff of AUST.

I express my gratitude and appreciation to my thesis supervisor, Professor Ekpe Okorafor for his sincere support, encouragement and guidance throughout the research period and for giving me the strong support to do more research on my thesis. I wish to extend my gratitude to the Head of Department of Computer Science, Professor Amos David for guiding us throughout course work and making us learn the basics of Database and Information systems. I also wish to send my gratitude to all the lecturers that taught us including Professor Ekpe Okorafor, Professor Ben Abdallah, Professor Lehel Csato, Professor Hamada, Professor Cohen, etc. Thank you all for taking the time to teach us to the best of your ability and bringing out the best in us.

I would like to use this opportunity to express my greetings to everyone who supported me throughout the course of my project. I am thankful for their aspiring guidance, invaluably constructive criticisms and advice during the thesis. Further, I would like to express my greetings to my friends and course mates, in particular, Latifat, Habibah, Ruqayya, Nkiru, Joseph, Nura, Jabir, Luqman, Habib, Abdul and all the others.

It also gives me great pleasure to express my profound gratitude and greetings to Muhammad Kabiru Momoh for his immense support towards achieving my aim, may Allah continue to bless you, guide you, and protect you.

DEDICATION

This thesis is dedicated to my parents. May Allah continue to bless you, guide and protect you all.

Table of Contents

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

Quintillion bytes of data are being generated daily and handling this enormous amount of data is becoming more tedious every day. These bytes of data are generated by people using devices such as mobile phones, laptops, smart devices and these devices are connected to the internet so as to be able to identify themselves to other devices. These devices are found everywhere in a smart city (Gehlot, 2016). According to (Santana, Chaves, Gerosa, Kon & Milojicic, 2016), a Smart City is a city in which social, business, communication, and technological aspects are supported by Information and Communication Technologies such as intelligent IoT and data collection sensors to improve the experience of the citizen within the city. To achieve that, the city provides public and private services that operate in an integrated and sustainable way. The bytes of data generated by the sensors must be used to make data – driven decisions as they are generated in real time proactively. As the sensors sense environments continuously, data streams generated must be processed in real-time to gain insight quickly because the data generated are in many cases critical and time sensitive. Smart cities are built on the Internet of Things. According to (Hahanov, 2015), the Internet of Things (IoT) is a new paradigm which involves exchange of data between different things through devices without human arbitration, automated collection, processing and analysis of large amounts of data, generated by sensors in an IoT environment. (Al Nuaimi, Al Neyadi, Mohamed & Al-Jaroodi, 2015) said that big data systems will store, process, and mine smart city applications information in an efficient manner to generate information to improve and enhance different smart city services. In addition, big data will help decision-makers who will use data generated by sensors, to plan for any expansion and extension such as making data-driven decisions in smart city services, resources, or areas.

The IoT environment has three components namely sensors which senses the movement of objects, actuators and embedded communication hardware; a middleware, which analyses and stores data and information generated by the hardware; and a presentation layer, in which users access, view, manipulate, and visualize data extracted from the hardware (Santana et al., 2016). There is a wide range of services and applications which covers fields such as transportation (intelligent road networks, smart mobility, smart traffic lights, smart parking systems, connected cars and public transport), public utilities (smart electricity, water and gas distribution), education, technology, health and social care, public safety (Radek Kuchta, Kuchta & Kadlec, 2014). This thesis will focus on transportation which is an important part of a smart city and will cover all areas of transportation.

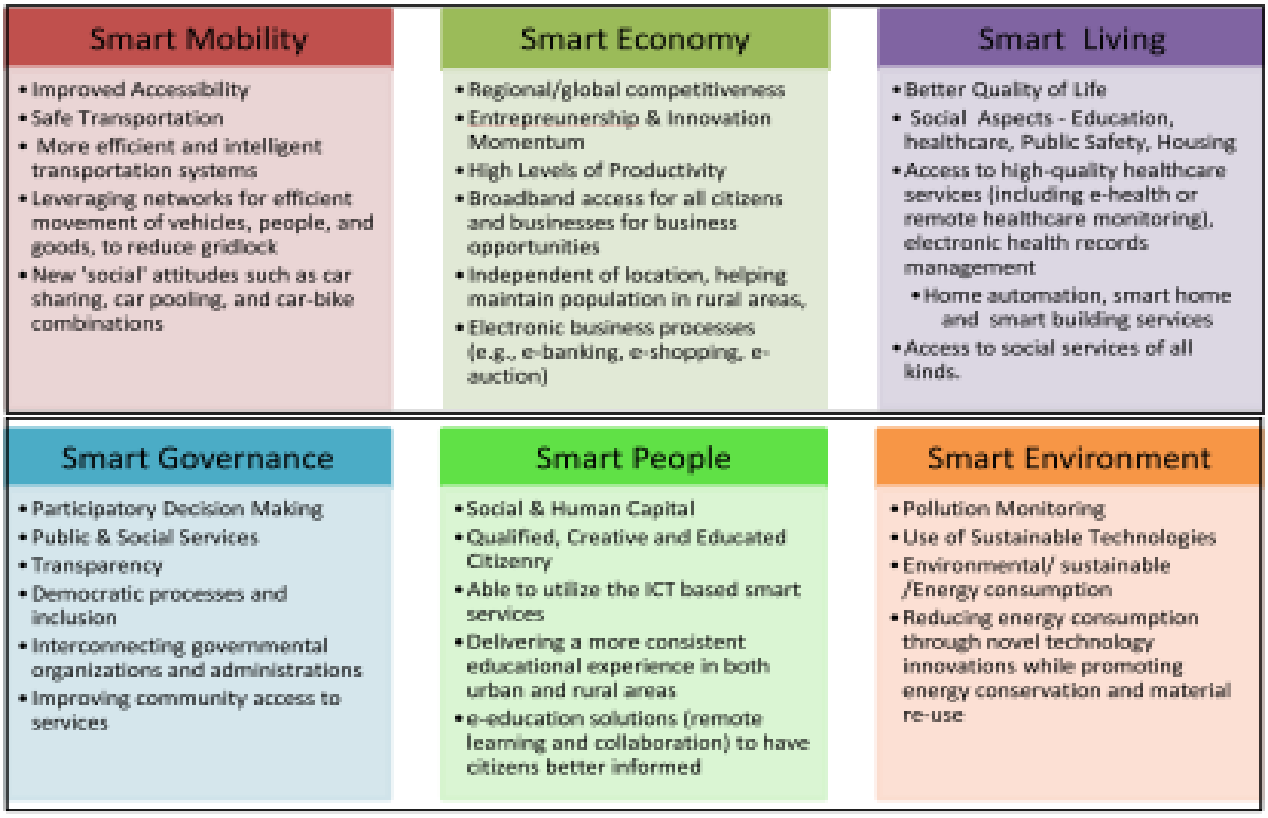

Figure 1.1: Areas of smart city applications

(To & Cited, 2016)

One of the problems that most cities face is inefficient and ineffective traffic management. On a daily basis, the world's population increases which leads to congestion on the roads because of the escalating population of people migrating to urban areas. The ITMS makes life easier by leveraging the IoT concepts to monitor the traffic; reduce the traffic congestion on the road; avoid traffic jam; improve public transportation; and provide better services within the city such as the smart traffic signals, traffic monitoring and control, and smart traffic lights. In the traffic monitoring system, if an accident occurs, an alert is sent out immediately and the remote monitoring system provides instant updates on the situation, drivers can receive warnings on their Global Positioning Systems (GPS) and through connected road signs, and traffic lights can adjust automatically to control the traffic appropriately in order to manage traffic flow and prevent traffic jams. The data generated by the sensors are used to monitor the traffic flows to determine how the traffic jam can be prevented. Based on the data generated, if it reports to the road safety, and if an accident occurs frequently on a particular road, it sends a message to the medical center to send an ambulance to the location in real time and this can save the life of a citizen if something is done immediately. To perform the real time data stream processing model, big data methods such as Apache Cassandra and some other sub projects of Hadoop such as Kafka and Spark are used. These methods enable data driven decisions to be made. Apache Cassandra is an open source No SQL (not only SQL) database used for distributed processing. Hadoop services provides for data storage, data processing, data access, data governance, security, analytics and operations ("What is Apache Hadoop_," n.d.). According to (*Apache Kafka*, n.d.), Kafka is a solution to the real-time problems of any software solution, that is, to deal with real-time volumes of information and route it to multiple consumers quickly so that streaming data can be processed immediately as they are received.

Apache Spark is a fast, in-memory data processing engine with a classic and expressive development APIs to allow data workers to masterly execute streaming, machine learning or SQL workloads, graph processing that require fast continual access to dataset. In real time data stream processing, the data needs to be processed as fast as possible, and to achieve this, a fast platform is needed ("What is Apache Spark," n.d.). After the big data methods have been applied, then the data is used to make data driven decision to determine actions to be taken when an event occurs and also to predict probable future occurrences and outcomes. In Prathilothamai, Lakshmi & Viswanthan (2016), the sensor data collected was converted to Comma Separated Values (CSV) files to retrieve the count of the vehicles that passed through and the speed of the vehicles within a particular duration. After the retrieval of these parameters, the file is uploaded onto Apache Spark and used to predict the traffic congestion status which is categorized as high congestion, medium congestion and low congestion. The main component of Apache Spark is the Spark SQL, where queries are processed on the sensor data (CSV file) to detect the number of vehicles, human count, and traffic status. In the proposed system, a message broker known as Kafka is used as a collector for monitoring events and as a tracker of users' consumption of data streams processing in real time. Kafka is a distributed publish-subscribe system that is designed to be fast, reliable, scalable, and durable. Apache Spark is a fast and general engine for big data processing, with built-in modules for streaming SQL, spark streaming and it is it is faster than map although they largely perform the same task. Apache Spark performs at a rate that is hundred times faster than map and is therefore a more suitable alternative. The proposed system will be used to monitor the traffic congestion and to decide which actions to take when there is traffic congestion, actions to take when vehicles are disobeying the traffic rules and signals and to regulate traffic flow to prevent traffic jams.

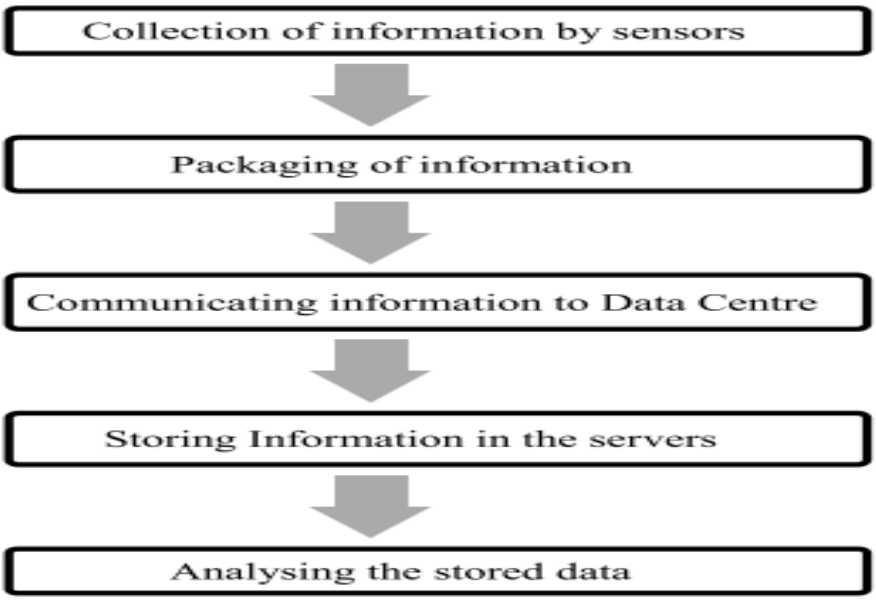

Figure 1.2: Processing and storing data
(Gehlot, 2016)

1.1 Problem Statement

In real-time data stream processing models, data driven decisions can be made in real time if real time streaming data is available in order to test the performance of the system. In an IoT environment such as for smart city application, the challenge does not lie in the ability to generate vast amount of data but the problem is that as the data is received, it needs to be processed rapidly because the data generated are in most cases, critical and time sensitive. To model a system that will not only generate the data in real time but will be able to determine how to use the data as it is received and process the data accordingly. Another problem lies in ensuring that the data generated by the sensors is not lost. Data driven decisions can be enhanced if data can be replicated for fault tolerance so that no data is lost. However, big data methods such as Cassandra, apache Kafka, Spark, and Zookeeper will be leveraged to perform real-time data stream processing.

The main concern is to ensure that no data is lost and that data displayed can be monitored as well as used to determine succeeding processes.

1.2 Objectives

The main objective of this thesis is to demonstrate the ability to leverage big data methods such as Cassandra, Kafka, Zookeeper and Spark to perform real-time stream processing; to ensure that no data is lost ;monitor displayed data; determine succeeding data processes and monitor road traffic within an IoT environment for a traffic monitoring system in a smart city environment. The expected result is the integration of Kafka and Spark to perform real-time data stream processing, to process the data by Apache Spark, and to forward the data to the database. A big data tool is expected to be used to query the data from the database.

1.3 Thesis Organization

This thesis is organized as follows: Chapter One introduces the topic, defines the problem, sets out expectations as well as objectives, and also attempts to clarify what the thesis aims to achieve. Chapter Two presents the literature review in which the concepts of a Smart City are presented ensuring that the different areas of a smart city are explained, the real-time data streaming and analytical model for making data driven decisions for the ITMS is described, the IoT environment is explained and related works that leverage Intelligent IoT concepts are introduced. Chapter three introduces the big data methods such as Cassandra, Hadoop, Apache Kafka, Zookeeper, and Apache Spark which are used to implement the system and demonstrates the ability to leverage big data methods for the ITMS. Chapter Four explains how the real-time streaming model is implemented using the tools described in Chapter Three. The work is concluded in Chapter Five.

CHAPTER TWO
LITERATURE REVIEW

2.1 Smart City Concepts

The combination of social, physical and Information Technology infrastructure to improve the quality of services and enhance the citizens' quality of life is termed 'Smart City'. It allows for real-world urban data to be collected through software systems like sensors, server substructure, network infrastructure, and client devices, implements solutions, with the support of instrumentation and interconnection of sensors, actuators, and mobile devices. According to (Policy & Division, 2015) 'Smart city is a city that monitors and integrates conditions of all of its critical infrastructures including roads, bridges, vehicles, waste management, tunnels, rails, subways, airports, sea-ports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services and enhance the quality of life to its citizens'. Smart cities can be seen as systems with flows of energy, services, people and financing. Moreover, urban planning is closely related to the business, economic and social metabolism of communities. Identification, integration and optimization of different energy, transport and data flows in city planning, monitoring, controlling, making decisions and city management are crucial to creating sustainable smart environments (Highl, n.d.). Some areas of smart city applications are as follows:

2.1.1 Smart Mobility

In the mobility era, transportation must move people and goods faster, seamlessly, and in a proper way, in urban and IoT environments. The environment needs intelligent substructures that are able to process the vast amount of information collected in real time and data stream, and provide the most proficient transportation services to

businesses, technologies and citizens alike. Providing transportation in a way that realizes the smart mobility concept requires building a network for the coordination of transportation companies or entities that collect, process and analyse information from the various entities that operate in the city; and supply each entity with information they can use to optimize and utilize the overall system (Okuda, 2012). All around the world, people are combining cities. Fifty-three percent of the population currently lives in urban areas and, by 2050, this is expected to reach 67 percent. Countless studies have shown that most cities are badly designed and are not able to cope with underlying transportation constraints, resulting in congestion within cities as the population grows.

2.1.2 Smart Grid

A *smart grid* is an electrical *grid* that includes a wide variety of operational and energy measures including *smart* meters, *smart* appliances and combining renewable energy resources with non-renewable energy resources to manage energy consumption. It is a renovated electrical grid system that uses information and communication technology and networks of physical devices to collect and act on available data, and process the data (such as information about the behaviors of suppliers and consumers) in an automated fashion to add some value (Al Nuaimi, Al Neyadi, Mohamed & Al-Jaroodi, 2015).

2.1.3 Smart Buildings

A smart building is a smart network with a central computer used for programming its environment, devices, and building appliances. A smart building is one that achieves significant energy savings by taking advantage of improved technology and materials in terms of structure, appliances, electrical systems (“What is a Smart Building_ _ Building Efficiency Initiative _ WRI Ross Center for Sustainable Cities,” n.d.).

It combines building electricity usage with motion sensor lights which can switch off automatically when a room is empty; detect when there is a leaking pipe using smart meters; keep track of electricity usage through a smart electric meter, and generate alerts when it reaches a specified threshold.

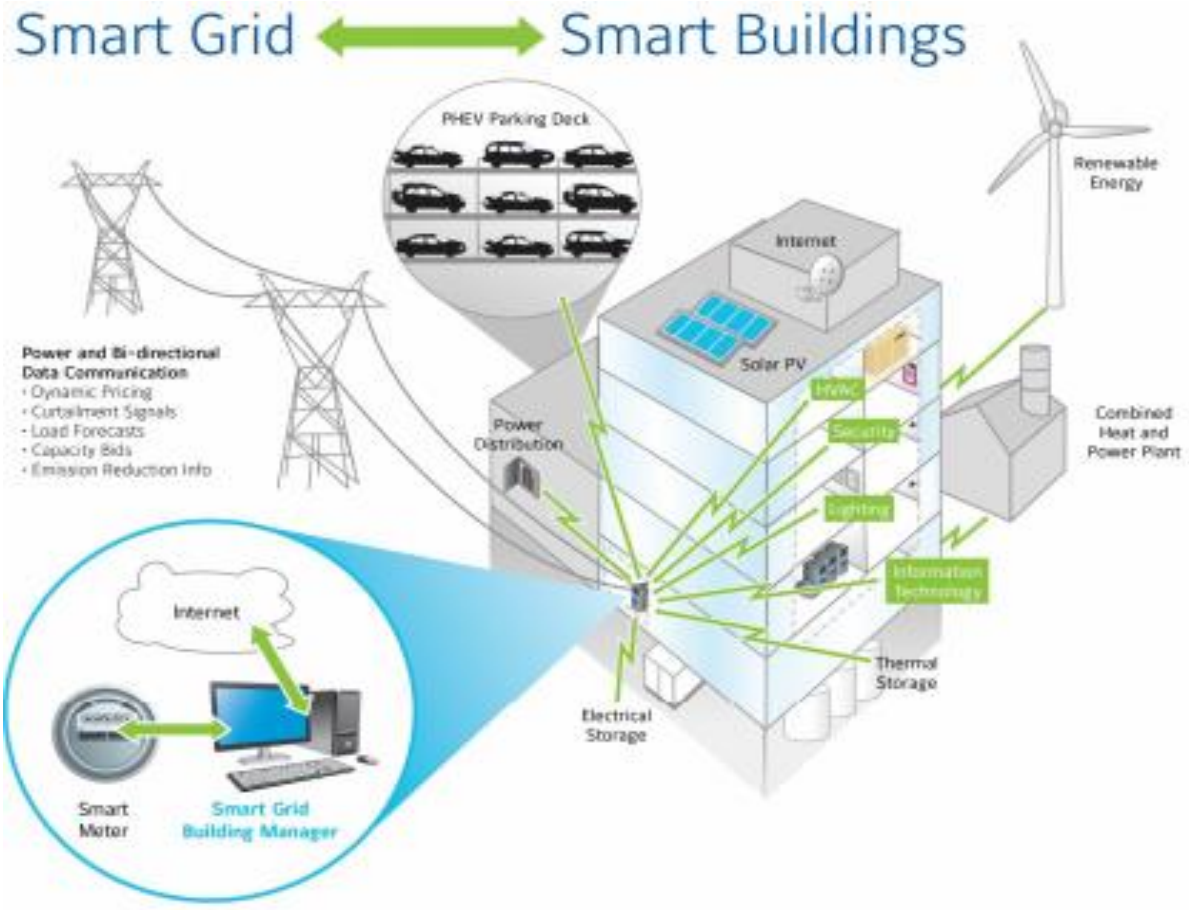

Figure 2.1: Smart Building concepts

("What is a Smart Building_ _ Building Efficiency Initiative _ WRI Ross Center for Sustainable Cities," n.d.)

2.1.4 Smart Water

Smart water makes more informed decisions, protects city's water supply and prevents water waste in the city using data from detecting water pressure, temperatures, and leaks (Universal & Platform, n.d.)

2.1.5 Smart goods

Smart goods produces real-time city event information, leverages GPS locations and combines with user profiles to find a good parking spot whilst considering the driver's sport interest, event starting time, and tickets purchased or seat numbers, etc.

2.1.6 Smart Industry

It provides the means of easier grasp of transport and logistics flows, not only for one industry, but also for multiple industries.

2.1.7 Smart Lightning

It economizes time for maintenance crews and reduces fuel costs involved in driving around to find and replace broken light bulbs and to lighten the environment in real-time.

2.1.8 Smart Waste Management

By using sensors and connectivity in waste bins to monitor the level of the rubbish inside, collection routes can be improved so that the bins are emptied when they need to be in real-time, even if that means some bins are emptied twice a day or even three times a day and saves citizens the effort of checking it daily and others only every few days. This produces cost savings, reduces CO_2 emissions from the collection trucks, and inflates citizen atonement as waste bins are no longer overflowing and will no longer be kept over a long period before disposal.

2.1.9 Intelligent Traffic Monitoring Systems

The Intelligent Traffic Monitoring System (ITMS) improves traffic flow using traffic signals and connected vehicles to monitor the traffic systems, the number of vehicles and congestion roads. The diagram below shows how the ITMS works within a smart city.

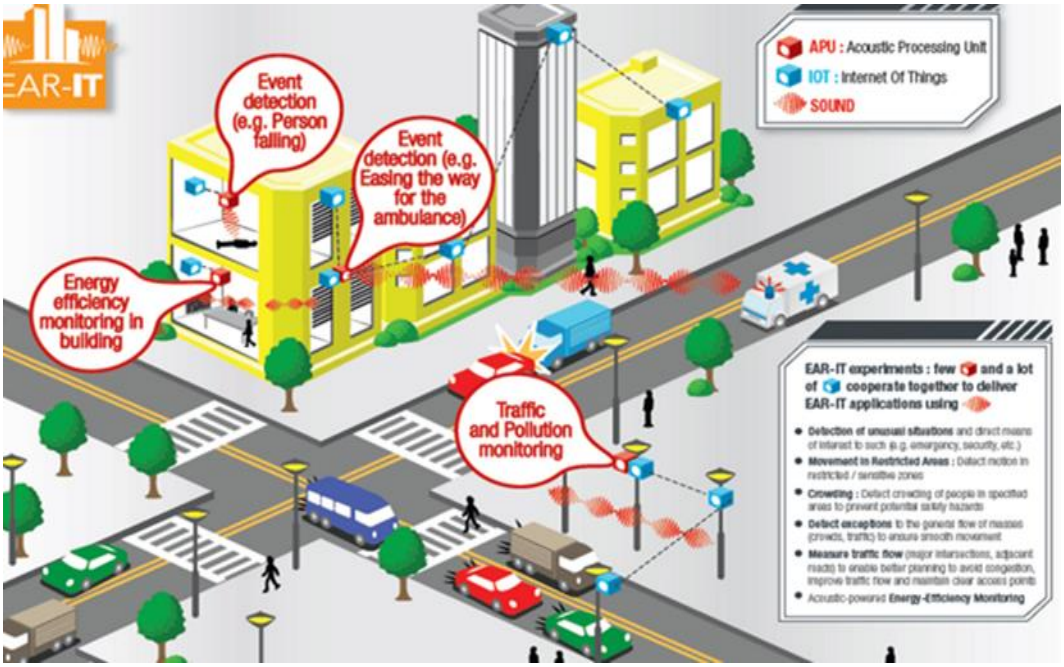

Figure 2.2: Traffic monitoring system

2.1.10 Smart Energy Management

Smart energy management can result in energy savings for users and enterprises alike. From connected thermostats in the home, such as Hive, to cosmopolitan systems that manage heating, ventilation, and air conditioning (HVAC) for shopping malls, IoT is already making an extraordinary impact on abating energy costs.

2.2 Architecture of Smart Cities

The following are used as cases of the smart city and the collection of the use cases makes up the smart city.

2.2.1 Urban Area

An area where there is human habitation, movements of people and high population density in the environment.

2.2.2 Dense and heterogeneous devices

Devices sense data from different sources to generate data. The sensing layer has a varied set of IoT nodes that are scattered across an urban area. These nodes collect data continuously and concurrently about various activities occurring in the physical environment in a smart city. An IoT node comprises sensors, microchips, power supply, and network elements. IoT nodes are categorized into 2 different sections based on their operating conditions:

- **Constrained node:** These nodes operate in a low-power environment. They have low processing power, low speed and a low data-transfer rate.
- **Unconstrained node**: These nodes have no operational constraints in terms of power consumption, processing rate, and data-transfer rate (Vasundhendra Badami. *Architecture of smart cities*, n.d.).

2.2.3 Types of Data

This represents the types of data that is generated by the sensors.

2.2.4 Communication Techniques

It represents different techniques that are used by different devices to send the data to a centralized server or to communicate with other devices. Smart city systems typically have billions of IoT nodes that are stretched across the city. These IoT nodes should be addressed independently. This is made possible with IPv6, which provides a 128-bit address field.

2.2.5 Control Centre Server

All the data agglomerates in a centralized server. It guards communication with Servers and Recorders.

The Control Centre Server (CCS) is a Windows based service that is connected to network computers, returning Recorder status to the CCS application and instructions to the Recorder.

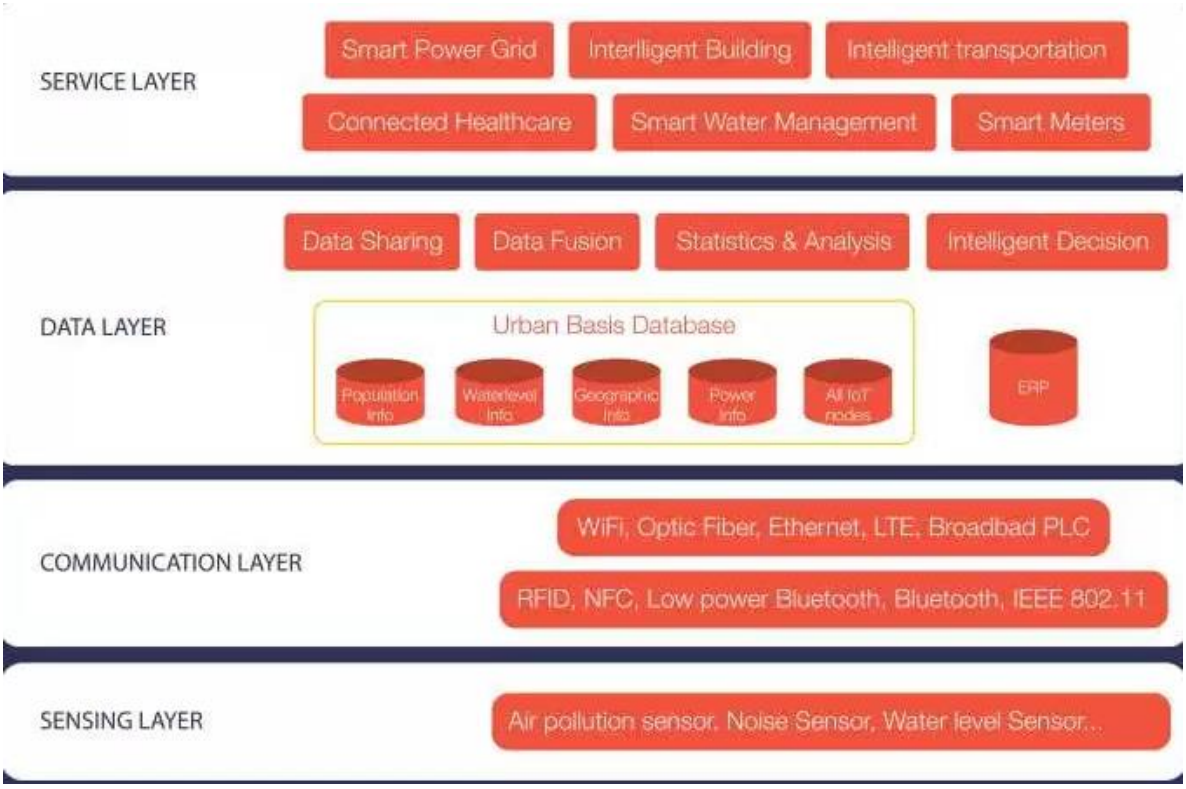

Figure 2.3: Architecture of Smart Cities

2.3 IoT for Smart Cities

In 2016, 22.9 billion devices were connected to the Internet and these devices produced data continuously in real-time and by the year 2017, 100 zettabytes of data were be uploaded onto the Internet. According to (Paper, 2015), IoT is a concept where an object is assigned to an IP address and through that IP address the device is identifiable on Internet and can exchange data with other devices.

A team from the International Telecommunications Union defined IoT as a global infrastructure for the information society, which makes advanced services by interconnecting (physical and virtual) things based on previous information and evolving interoperable information and communication technologies

IoT is an environment in which objects, animals, and people are provided with a unique identifiers and the ability to transfer data to a network without involving human to human or human to computer interactions but only device to device. IoT is applied to control sensors and actuators responsible for repossessing information from smart city environment. It plays an important role in smart city applications. In fact, a smart city cannot exist without the IoT. According to *Smart Cities Are Built On The Internet Of Things* (n.d.), Intelligent IoT ensures that transportation systems have improved capacity, enhance travel experiences and make moving anything safer, more efficient, more reliable and more secure. The local police, emergency services and other government services can use these sensor networks with smart traffic management in order to improve citywide visibility to help alleviate congestion and rapidly respond to incidents in real time. IoT is the digital representation of real things in the Internet and state of real environments is available in real time. Through actuators it is possible to influence the real world based on actions in the digital world. In an embedded system, sensors, actuators and radio interfaces makes up the wireless sensor node and wireless sensor nodes makes up the wireless sensor network. The wireless sensor network uses internet connectivity to get the IoT proactively (Cities, 2015).

2.4 IoT Architecture

In a nutshell, the (IoT) is a concept of networking of devices to be able to identify and communicate with other devices. The devices are constrained devices such as Radio Frequency Identification (RFID) sensors.

To process data from IoT devices, data generated by sensors are captured and stored in a distributed database to build up the volume of the Big Data. These devices generate data at a high rate in real time and continuously in different formats. The IoT architecture consists of three main blocks which are:

2.4.1 Message Queue/Stream processing block

This block takes the input data and performs the processing functions depending on the application requirements. It does buffering, filtering and complex online processing over streams of data produced. The message queue block can produce real-time data or the ingestion output for downstream components so as to gain insight quickly.

2.4.2 The Database Block

The database block takes the data from the message queue/stream processing block and provides structured, fine grained and low latency to the data points. Due to the nature of the data, the database block is a Not Only SQL (NO SQL) solution able to accommodate sparse data, with auto sharing and horizontal scale out properties.

2.4.3 The Distributed File System Block

It takes data from either the Database block or directly from the Message queue/stream processing block and performs batch jobs over the entire data set. It includes combining the data from IoT devices with other sources or potentially unstructured ones.

2.5 Challenges of IoT

The following are some challenges faced in an IoT environment.

- Lack of innovate ways to send;

- Connectivity;
- Power;
- Security and privacy; and
- High Complexity.

2.6 Traffic management system

The problem of traffic jams in cities is not an easy problem to address. It is necessary to improve transportation systems and provide people with alternative means so that they may be less dependent on their personal vehicles. In addition, it is also necessary to stimulate these other alternative modes of transportation, to improve road safety, people's security and walkability through the streets (Lakshminarasimhan, 2016). Traffic management system consists of the traffic monitoring, Smart Parking System, Smart Street Lights, and public transport. While searching for parking spot, there is an excessive consumption of fuel, wastage of time, and congestion on roads due to different vehicles waiting to identify for the parking spot.

2.6.1 Smart Parking System

In the smart parking system, sensors in the parking areas communicate occupancy of parking slots to the parking area receiver in order to determine where to park the next incoming vehicle. An application is also provided to show all the nearby parking spaces, notify the users on how many parking slots are available and where the vehicle can be parked or should be parked to avoid congestion. The benefits of the Smart Parking System are that it saves fuel, time and also it avoids traffic on the road because drivers do not spend excessive time looking for parking space.

2.6.2 Smart Street Lights

Smart traffic lights use excessive energy and in case of light failure, pedestrians are inconvenienced. They use motion detection sensors implanted on the street lights to vary luminosity of LED lights, detect malfunctioning lights and notify the service station. The intensity of light is also controlled using weather sensors. This ensures avoidance of mishap on roads, and also allows smooth flow of traffic and reduces traffic jams. Sensors and cameras are used to detect congestion at traffic signals and traffic lights are changed automatically when there is congestion on the road. Zebra crossing automation is also enabled. Sensors detect over speeding and cameras quickly record the registration number of the vehicle in real time. Whenever bad conditions are detected on the road, sensors detect it and display warning messages on the sign boards accordingly. They also transmit information to the nearest health center in case an accident occurs and the sensors send information to the smart automobiles approaching the affected area.

2.6.3 Public Transport

The ease of access to public transport is improved and this encourages uses of public transport. It shows a map with details of buses on a particular route; shows details for a particular route number, and communicates the status of the next bus to arrive in real time. The number of vacant seats available on a vehicle and can be accessed on mobile applications. Smart cards can be used for payment.

2.6.3.1 Benefits of Traffic management systems

- Less Traffic Jams and congestion leads to lesser fuel consumption;
- Road safety is improved;
- Smooth flow of traffic;
- Convenient for both vehicles and pedestrians;

- Saves power by controlling lights;
- Rate of accidents roads is reduced; and
- Saves life due to enhanced and proactive accident reporting.

2.6.3.2 Challenges of Traffic management systems

- Data is generated in motion and the dynamic data generated by the sensors are of huge volume and needs to be handled in real time proactively to yield efficient and accurate results in terms of performance.
- The processing of the huge volume of data generated by the sensors may result in lack of accuracy and efficiency because streaming data is generated continuously.
- Data is generated from different sources which could be structured, semi-structured, and unstructured which means that managing the variety of data that comes with this speed tends to be difficult.

2.7 Real-Time Data stream processing model

Data is captured from sensors and stored for processing. The amount of data generated by the sensors tends be too large for processing such that it cannot be handled by the file handling systems. So better ways are utilized to process the vast amounts of data recorded from the sensors in real time by stream processing which ensures that real-time analytics are being leveraged. To implement real-time data analytics, the system must be scalable and fault tolerant whilst ensuring low latency and high availability. Stream Analytics is a processing engine that can process real-time events in real time either from an individual data stream or from multiple streams. The events are recorded from the sensors and a variety of sources.

2.8 Related works

2.8.1 Cost effective road traffic predictive model using Apache Spark

(Prathilothamai, Lakshmi & Viswanthan, 2016)

This paper proposes a cost effective model to predict traffic jams and to inform any members of the public who are planning to use the road system about the current traffic conditions. Traffic monitoring needs to be processed in real time due to huge volume of data generated and traffic prediction is analysed using the current technologies such as Apache Hadoop and Apache Spark frameworks. Spark was used because it processes 10 Terabytes of data within half a second. Since data needs to be processed in real time, road traffic can be predicted within half a second and it helps to reduce road traffic delays. The proposed system uses vehicle count and speed to predict t traffic conditions which are communicated to the users within the city. The system shows the integration of sensors with Apache Spark used for interpreting the semantics of data which improves the overall structure of traffic prediction system. In this paper, Timely Prediction of Road Traffic Congestion by processing of sensor data in Apache Spark is the main approach, it exploits Apache Spark and its application in real-time data processing as a way to annotate and interpret the data. The work is done by:

- Presenting the prediction system by means of a parallel processing of sensors as well as video data;
- Focusing on prediction by using ontology; and
- Alerting the public about traffic conditions. Thus, the revealing part focuses towards the evidential approach that will be able to solve the above-mentioned risk.

The severity of the road traffic is predicted by using Apache Spark which increases the accuracy in prediction. Initially data was collected using ultrasonic as well as passive infrared sensors. After processing the collected sensor data, the dataset provides the speed of the vehicle, actual count of the vehicle and duration during which the vehicle passed through the sensor area. Sensor data are then converted to a comma separated value (CSV) file. The CSV file is then processed in Apache Spark in order to predict traffic congestion. The prediction of the road traffic is classified as High Congestion, Medium Congestion, and Low Congestion. The paper also focuses on the idea of traffic prediction based on mainly two factors namely the speed of vehicles and the number of vehicles. If sensors detect that the vehicles are moving at low speeds which gradually decrease, it is inferred as high traffic congestion. Whereas, if the vehicles are passing through the sensor area at speeds that are more than the minimum speed and speeds do decrease it is inferred as low traffic congestion.

The sensor data is first collected, converted to a CSV file to retrieve count of the vehicles passed, and the speed of the vehicles within a particular duration. After the retrieval, it is loaded into Apache Spark and used to predict traffic congestion. The system architecture is explained in Figure 2.4 below.

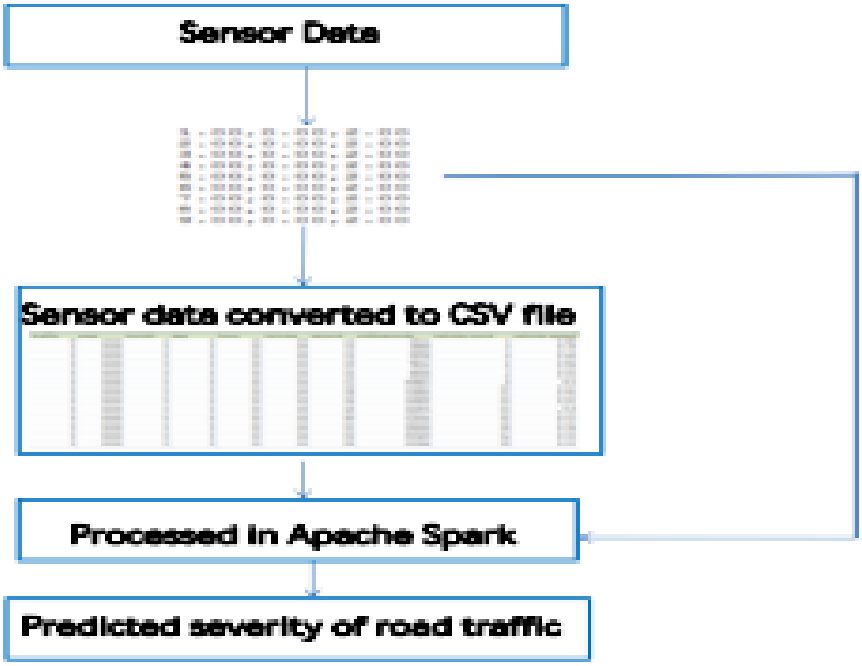

Figure 2.4: Architecture of the system to predict road traffic congestion.

The vehicle speed is calculated as

Speed $\sum_{i=1}^{n}(Xi - Yi)$

Where Xi is the beam separation distance, and Yi is the time taken to traverse the beams. The summation of all the beams that is projected from the sensor and reflected back to sensor are given by the above formula. That is, the speed is calculated by observing the time at which the vehicle crosses each of the emitted beams.

In order to predict the traffic by classifying it as low, medium and high traffic, a classification algorithm is used in Apache Spark, which classifies the data set into different classes. A decision tree is then used to predict the model from the classes. Decision tree acts as a predictive model in decision analysis which can be used to explicitly and visually represent decision making.

The decision tree is applicable where traffic congestion must be predicted and categorized into different classifications. For processing the Decision tree first the spark MLlib library is imported into spark which a machine is learning library provided by Apache spark. Further the CSV file is loaded where it is processed by using Decision tree. The limitation of this paper is that it does not guarantee that the data generated by the sensor is reliable and that no data is lost as it is transmitted to its destination.

2.8.2 Advanced traffic management system using IoT

(Lakshminarasimhan, 2016)

This paper analyses the ever-growing urban population around the globe and discusses the traffic systems in densely populated cities. Furthermore, an advanced traffic management system is proposed which was implemented using IoT. In the traditional traffic management system, ineffective traffic lights with predefined timers are used with manual control by police officers, without taking account of real-time traffic data for consideration. It can happen that a green light is granted to an empty lane while a lot of cars are lined up at a red light on the other lanes because the same time interval of green lights is granted to every lane. The system consists of a circuit embedded in each vehicle in commutation. The users can interact with the system either through a wired or wireless connection of their smart phone with mounted board. The system uses Radio Frequency identification (RFID) which plays a vital role in the research paradigm of IoT. Instead of using Global Positioning System (GPS), it uses Local Positioning System (LPS) for locating a vehicle. Data analysis involves implementation of big data analytics. The basic functionalities of the system include:-

- Smart traffic light control system that works dynamically based on the concentration of vehicles in a specific region;

- Parking space identification and allotment system, with the placement of spatial sensors in parking lots that communicate availability of spaces with the regional workstations; and
- Anti-theft system that automatically retrieves the location of a stolen vehicle and automatically disrupts the functioning that vehicle.

Big data analytics is used to process the huge amounts of data received from the vehicles. The data generated by the sensors uses Hadoop map reduce tool which drastically reduces the data traffic which would occur when a single centralized control unit is used for analysing the data from each vehicle. Hierarchical clustering and density-based clustering techniques are used to process the data. Frequent pattern mining is used in order to derive results to enable efficient traffic management. The increase in number of cars on streets is not only the reason for traffic problems to appear but also lack of planning of the amount of cars can also cause the traffic problem. The conventional data processing systems are faced with lack of efficiency and accuracy and the traffic information management system using traditional data processing technology cannot meet the rapid growth of data.

This paper focuses on the communication between vehicles, processing unit and traffic lights. In this case, each vehicle acts as eye, which transmits traffic data. The communication is established by a socket programming over Wi-Fi connection, so ports are like ‘mouth’ and ‘ear’ of sender and receiver consecutively. Vehicles ‘name’ each other by calling RFID Reader name. One of the contributions of IoT to human operations in this scenario is that it can replace humans in doing tedious exhaustive work. He also proposed a supervised learning technique used for constantly improving the system’s information processing. I supervised learning algorithms; the basis of classification is training the classifiers using some training data.

Based on the training methods and characteristic of the classifiers, five major algorithms can be used to classify the data which are, k-nearest neighbor, decision trees, Na'ive Bayes, Logistic regression, and support vector machines. Computation of average velocity is done to determine the speed of vehicles on the road.

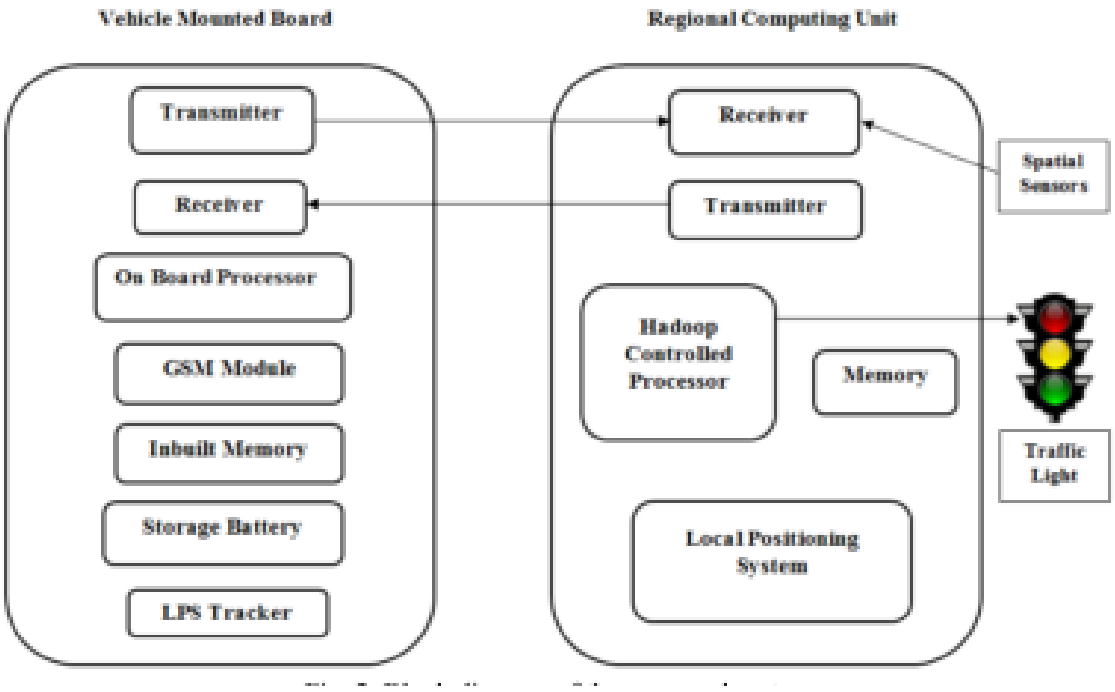

Figure 2.5: Advanced Traffic management system
(Lakshminarasimhan, 2016)

2.8.3 Smart traffic light in terms of the Cognitive Road Traffic Management System (CTMS) based on the IoT

(Hahanov, 2015)

This paper describes a CTMS based on the IoT approach. The telecommunication technologies which are used for the system development are analysed. In the paper, smart traffic light integration was proposed as a replacement of the existing traffic lights.

Real-time public traffic analysis allows to deliver the optimal amount of transport for a given number of passengers during the peak hours and to avoid empty transport movement on the roads. This not only results in saving money, but improving passengers' comfort level. The system proposed generates control signals based on the information obtained during the analysis of data obtained from cars and road sensors as well as alternative sources of user data such as social networks, the result of opinion polls, and others. In real-time, the CTMS is part of the e-government model which excludes operating errors caused by the human factor.

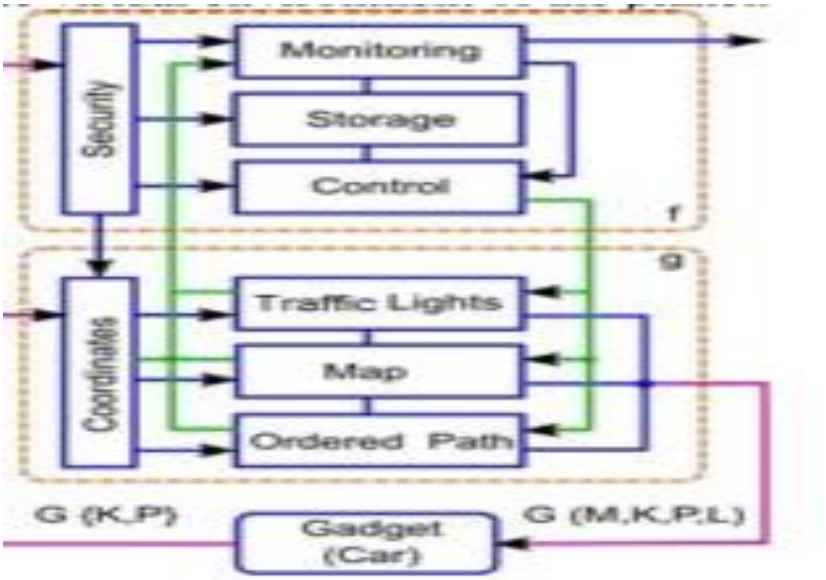

Figure 2.6: Cognitive road traffic management system structure
(Hahanov, 2015)

2.8.4 Traffic accident analysis using neural networks and decision trees

(Technology & Science, 2014)

This paper employed artificial neural networks and decision tree analysis techniques to discover new knowledge from historical data about accidents in one of the Nigerian busiest road in order to reduce carnage on highways.

The data were organised into categorical and continuous data through which the continuous data were analysed using the Artificial Neural Networks technique and the categorical data was analysed using the decision trees technique. Experimental results reveal that between the machines learning paradigms considered, Decision Tree approach outperformed the Artificial Neural Network with a lower error rate and higher accuracy rate.

2.8.5 Big Data Analytics Architecture for Real-Time Traffic Control

(Amini & Prehofer, n.d.)

Comprehensive and flexible architecture based on distributed computing platform for real-time traffic control is based on systematic analysis of the requirements of the existing traffic control system. In the architecture that was proposed, Big Data analytics engine informs the control logic. The system uses the distributed streaming platform Kafka and big data tool for the stream processing and building data pipelines. This paper gives an overview of both the existing system ITMS and the big data analytics approach. Kafka is a built-in mechanism hence it can tolerate hardware faults example failing computing machines.

CHAPTER THREE
METHODOLOGY

This section discusses the methods used to implement the thesis; the tools used to implement the thesis; how it was implemented. The data set of the traffic monitoring system in a smart city environment is discussed. The methods used to perform real-time data stream processing model are discussed as well as how IoT is leveraged; how the traffic data is been monitored; and what the data is used for.

3.1 Apache Kafka

Kafka is a distributed streaming platform and a publish-subscribe messaging system. It is a platform used for collecting and delivering large volumes of data, the actual time it is captured. Kafka is scalable, durable, reliable and fast. It is used to send or ingest data over a cluster. Once the data is captured from the sensors, Kafka transmits the data to the system and also sends the data over the cluster for processing. Apache Kafka is a publish-subscribe messaging platform implemented as a distributed commit log, suitable for both offline and online message streaming. Kafka is a solution to the real-time problems of any software solution to deal with real-time data and route it to multiple consumers quickly. Kafka provides seamless integration between information of producers and consumers without blocking the producers of the information and without divulging the identity of subscribers to producers. Kafka is used to collect the data in real time before it will be analysed. In a very basic structure, the Kafka producer publishes transmits to a Kafka topic, which is created on a Kafka broker acting as a Kafka server and then the consumers then subscribe to the Kafka topic to consume the data from the producers. Kafka ensures reliability.

3.2 Apache Spark

Apache Spark is used to perform data analytics. Once the data is collected from the sensors, it is processed. The analysis of the data is called data analytics. Since it is a fast, it is suitable for the application because a fast platform is required to process the data as soon as it is received. The size of data in a traffic monitoring system of a smart city application means that it needs to be processed in real time to obtain efficient results immediately. Spark works more efficiently than the Map Reduce model and it supports some features that the Map Reduce does not support such as interactive queries and stream processing. One of the features that Spark supports which Map Reduce does not support is the ability to run computations in memory. By supporting these workloads in the same engine, Spark makes it easy and inexpensive to combine different processing types, which is important in production data analysis pipeline. It can be accessible, and it offers API's in Python, Java, Scala, and SQL and is very efficient in built-in Libraries like Spark streaming, ml-lib, Spark SQL and spark core. It is used to combine and integrate with some big data tools to perform its stream processing and analytics. One big advantage of Spark over Map Reduce is that it has a compatible integration with other application tools which means that Spark can be integrated with other tools like Kafka to perform real-time applications. For instance, if an application that uses machine learning tools needs to be used, machine learning will be used to classify the data in real time as it is captured from streaming sources. Data scientists play an important role in doing the data science. Machine learning is not part of data science. Spark Core is used for task scheduling to ensure which job task is to be scheduled first. Spark SQL introduces the schema of the data it processed

3.3 Spark Streaming

To perform data streaming, Spark streaming is used to process the continuous data because the data generated is huge and needs processing as it is received because

sometimes it is time sensitive and critical. Spark streaming leverages Spark Core's fast scheduling capability to perform streaming analytics.

3.4 Real-time Integration of Apache Kafka with Apache Spark

As discussed, Kafka is a publish-subscribe messaging system used in messaging or ingesting data. It is used to fetch and collect the data from servers through and integrates with Spark to perform the data analytics. After the data is been processed, it is pushed to the database for storage. Real-time is the actual time that an event happens.

3.5 Cassandra

- Apache Cassandra is a free, open source, distributed data storage system that differs sharply from relational database management systems. Apache Cassandra is a scalable (Not-only NoSQL) database. Cassandra's technical roots can be found at companies recognised for their ability to effectively manage big data such as Google and Facebook. Some of the application use cases that Cassandra excels in include:
- Real-time, big data analysis;
- Time series data management and control;
- High-velocity device data consumption;
- Media streaming and data streaming management (e.g., music, movies);
- Unstructured data analysis;
- On- line shopping and web retail;
- Real-time data analytics; and
- Applications that utilize web services.

3.6 Spring boot

Spring is a very popular Java framework for building and developing web and enterprise applications. It is a framework used to facilitate and accelerate application development, and uses jQuery to push data to the web socket.

3.7 Architecture of the proposed system

In an IoT environment, smart devices are attached everywhere so as to generate data to make data driven decisions. In a smart city environment, to implement Traffic monitoring, vehicles are connected to capture information regarding their speeds and hence determine vehicles causing congestion thereby enabling the system to work. Data is captured by the sensors; then it is fetched from the sensor by Apache Kafka which is a distributed streaming platform. Apache Kafka captures the data from the sensors as soon as it is generated. It pushes the data to the brokers, the Kafka server and replicates the data to avoid data loss and also ensures that the data is been delivered for processing. Apache Spark is a fast and general engine which is used for big data processing. It subscribes the data from the Kafka topics and performs the stream processing. Data is generated randomly by the application. The attributes of the vehicle are: - Vehicle_id, longitude, latitude, type of vehicle, speed, time, etc. Spark performs the stream processing by counting the number of vehicles on a particular road. It then saves the data in the Cassandra tables. Cassandra is a NoSQL data base which is used for storage processing. The data which is processed by Apache Spark is now persisted to the Cassandra tables. Then the Cassandra tables are used to push the data to the traffic monitoring dashboard which displays the information of the vehicle. The key space and tables in Cassandra were created using the cqlsh command. The traffic monitoring dashboard is developed using the spring boot. Part of the information it displays is the vehicle count, the time, the type of vehicle, and the route. The diagram of how it works is shown below in Figure 3.1.

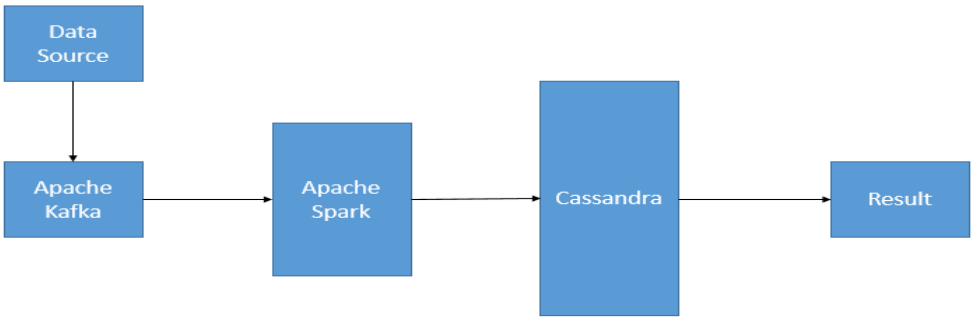

Figure 3.1: Architecture of the proposed system

- The System is categorized into three (3) parts namely:
- The Producers;
- The Consumers; and
- The Result processed.

3.8 The Producers

The application generates data randomly by itself so it acts as a real-time IoT platform. Vehicle type and the route is defined first comprising of three routes and five types of vehicles being used. The type of connected vehicles found in a city are: truck, bus, car, taxi, private car. A random class is then used to generate values of the attributes of these connected vehicles. The attributes of these vehicles are the Vehicle_id, longitude and latitude, time stamp, speed and fuel level. So, in the first stage, where the data is produced each connected vehicle sends event information randomly while moving in a particular route.. The producer acts as a data simulator application where data is produced. Once it produces the data, Apache Kafka now generates and captures the data for stream processing. The data is captured by the producer, pushed to the topics for storage and waits for the brokers to replicate the data to avoid data loss. Then, the data is now consumed by the consumer application.

3.9 The Consumers

This is where the Spark streaming is done. Spark is the consumer which subscribes the data from the topics for streaming. As soon as the data is received, Spark streaming processes it to gain insight quickly and keep pace with generation rates. Once it is done with the processing, it is stored on the Cassandra database and waits for the Dashboard to use it. The consumers try to calculate the vehicle count. To achieve this, it saves the previous records which are already processed and maps it with the current one which is been processed. The master node in the spark cluster assigns tasks to the worker nodes and the tasks are run on the worker node on the Spark cluster for processing and once it is done, it is saved to the Cassandra database and immediately it is sent to the traffic dashboard to display the information that is required. This data that has been processed contains the latitude and longitude which is needed to detect the actual route. Java Random class is used to generate values of the attributes of the vehicles. Data is generated randomly by the producer application.

3.10 The results

The dashboard displays the data that is has been processed by the consumer application and displays it on a web page. It also updates the information every 3 seconds.

3.11 Java

Java programming language is the language that we used to develop the application. Java was used because it is an object-oriented language, a high-level language, platform independent and it is robust and architecture neutral.

CHAPTER FOUR
IMPLEMENTATION AND RESULTS

This chapter explains how the implementation was done, the challenges faced, and how the expected results were achieved after the implementation was done. It also describes how traffic is monitored in a smart city environment. The data streaming model is also explained, showing the manner in which it works and also how Apache Kafka performs data processing. The dashboard of the traffic monitoring system is also explained.

4.1 Introduction

The Cassandra is launched; then Zookeeper is started because Kafka cannot work without Zookeeper starting perfectly. Also, the Kafka and the zookeeper servers are started. Then Spark is started. In Spark worker node and master nodes must all be started because that is where the jobs are submitted for processing. Also Spark-submit is used to submit all the jobs to the Spark cluster for stream processing.

4.2 Apache Zookeeper

The following diagram shows how the Zookeeper is started (Figure 4.1).

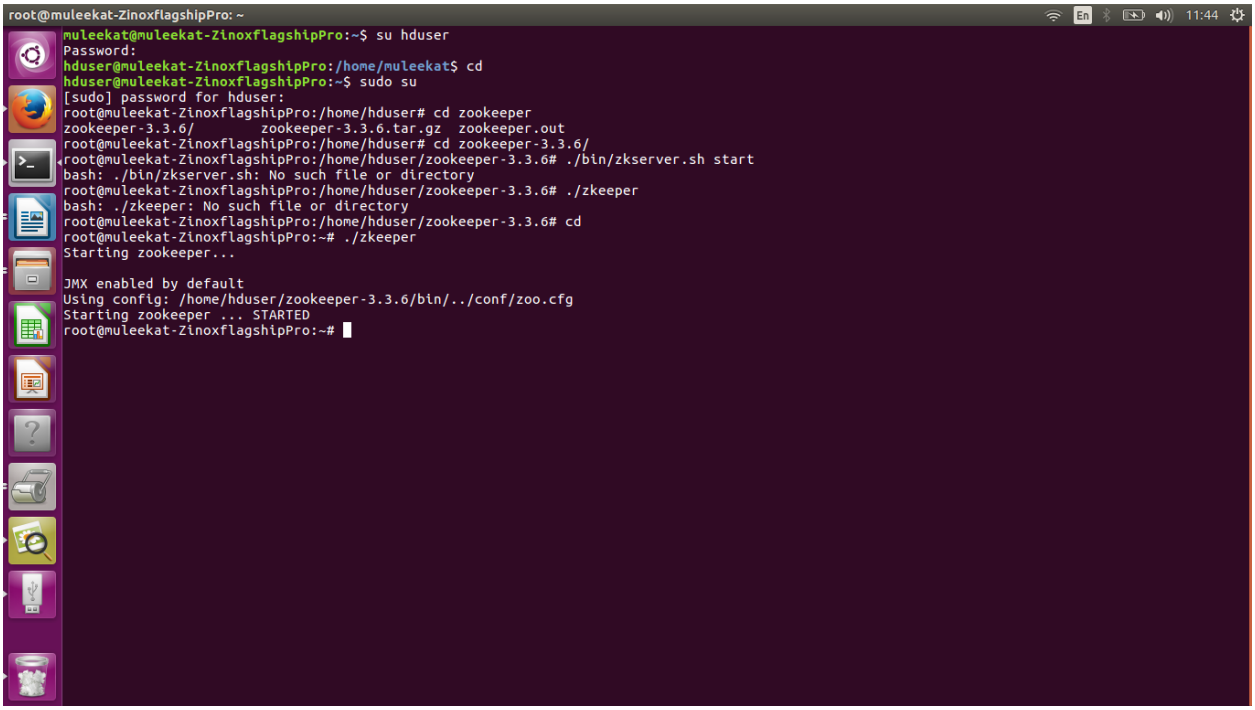

Figure 4.1: Zookeeper must be started before Apache Kafka

Apache Zookeeper is used to maintain the server that enables a highly distributed coordination. Kafka uses Zookeeper because it can coordinate the tasks across a distributed system.

4.3 Kafka and Zookeeper servers

After starting Apache Zookeeper, the Zookeeper and the Kafka servers are started so as to be able to perform the real-time integration. Kafka produces the messages. In our system, Kafka acts like a simulator application to produce the data and pushes it to topics. In this case, the Kafka topic for this application was created immediately when Kafka servers and Zookeeper servers were started. The following diagrams describe how the Kafka server and Zookeeper servers are started (Figure 4.2 and 4.3).

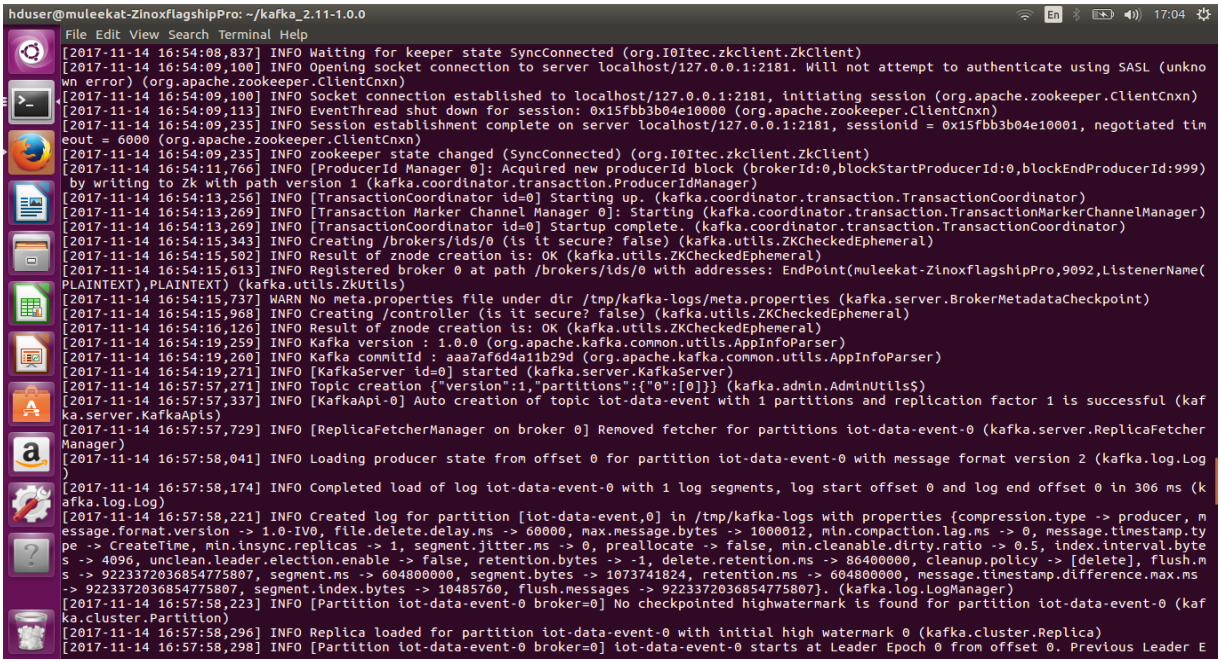

Figure 4.2: Zookeeper server

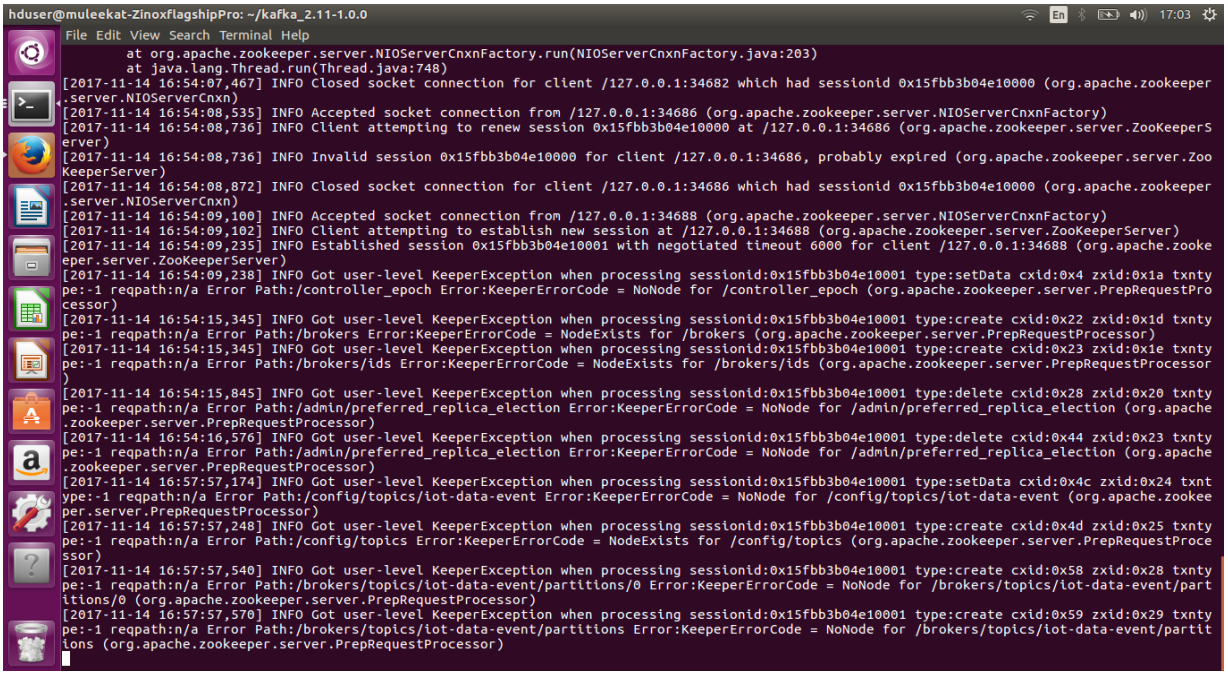

Figure 4.3: Kafka server

4.4 Spark and Cassandra

The Cassandra is started using the cqlsh command and it starts automatically. In order to ensure that jobs are submitted for processing, it is important to make certain that Spark-shell is running. And to verify that all the services that are running, the jps command is used to check. Figure 4.4 below depicts this process.

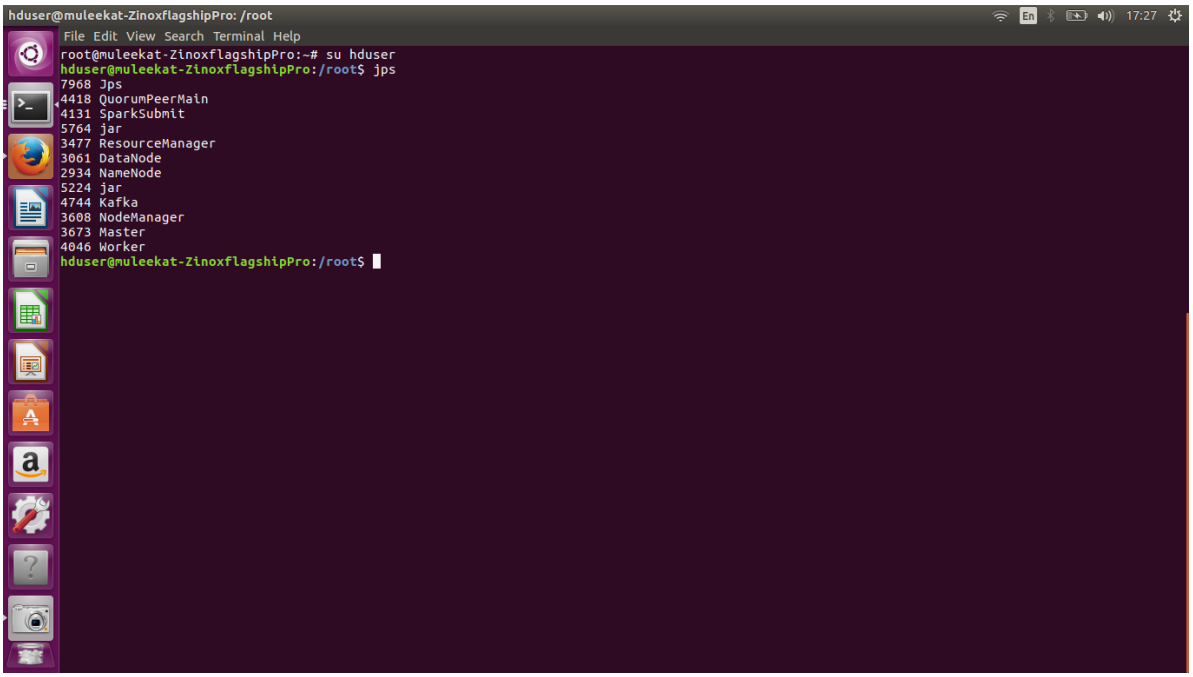

Figure 4.4: Services started

4.5 Kafka producer

Kafka producer produces the data in Java script object notation format which is known as the (JSON). JSON is an open standard file format that provides a method to store files in a readable format. Once everything is launched, the Kafka producer starts producing the messages in streams and saves everything to the Kafka topics. As it is saved, the consumer subscribes from the topics to process the data.

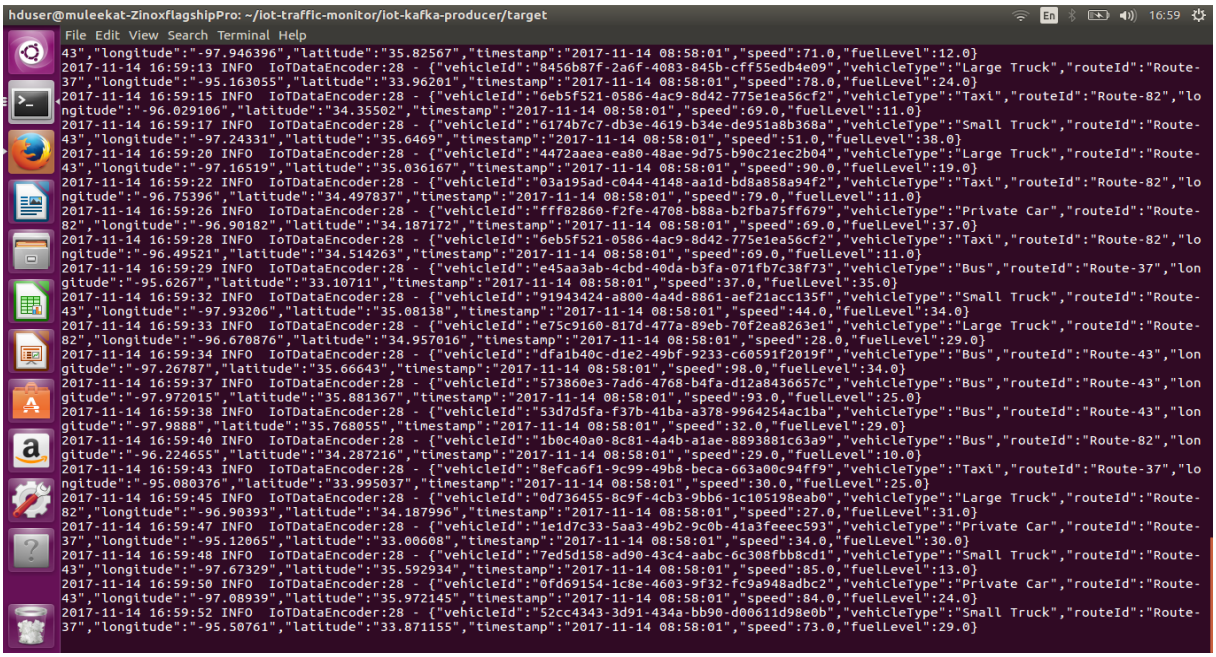

Figure 4.5: Kafka Producer

4.6 Spark Streaming

Apache Spark consumes the messages from the topics and starts processing immediately. After each process, it pushes everything to the Cassandra database. The Cassandra database forwards the results processed by Kafka to the Spring boot. Spring boot develops the interface where the results are displayed known as the dashboard (see Figure 4.6).

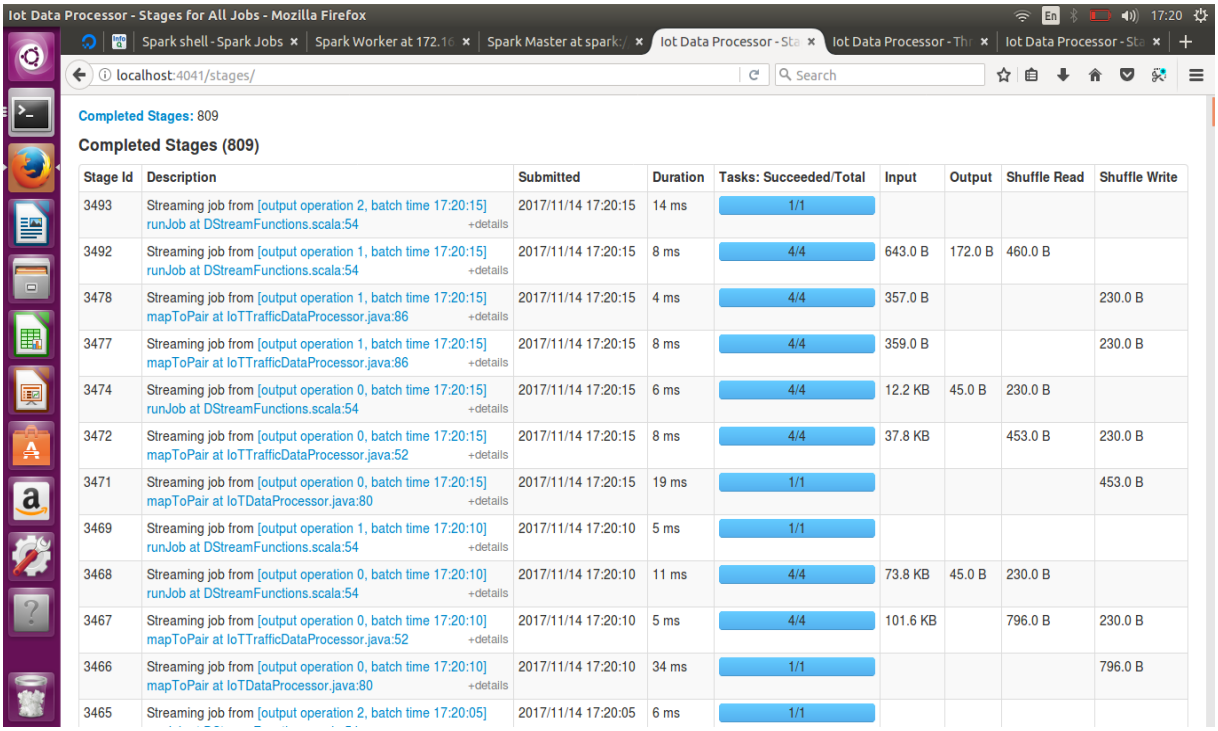

Figure 4.6: Completed Jobs

Figure 4.6 shows the number of completed jobs, the time spent in processing the job, and the tasks completed successfully.

Figure 4.7 overleaf shows the Spark Resilient distributed data also known as RDD which is the representation of a set of data spread across the machines and allows in-memory processing and ensures that data is computed on different nodes on the Spark cluster.

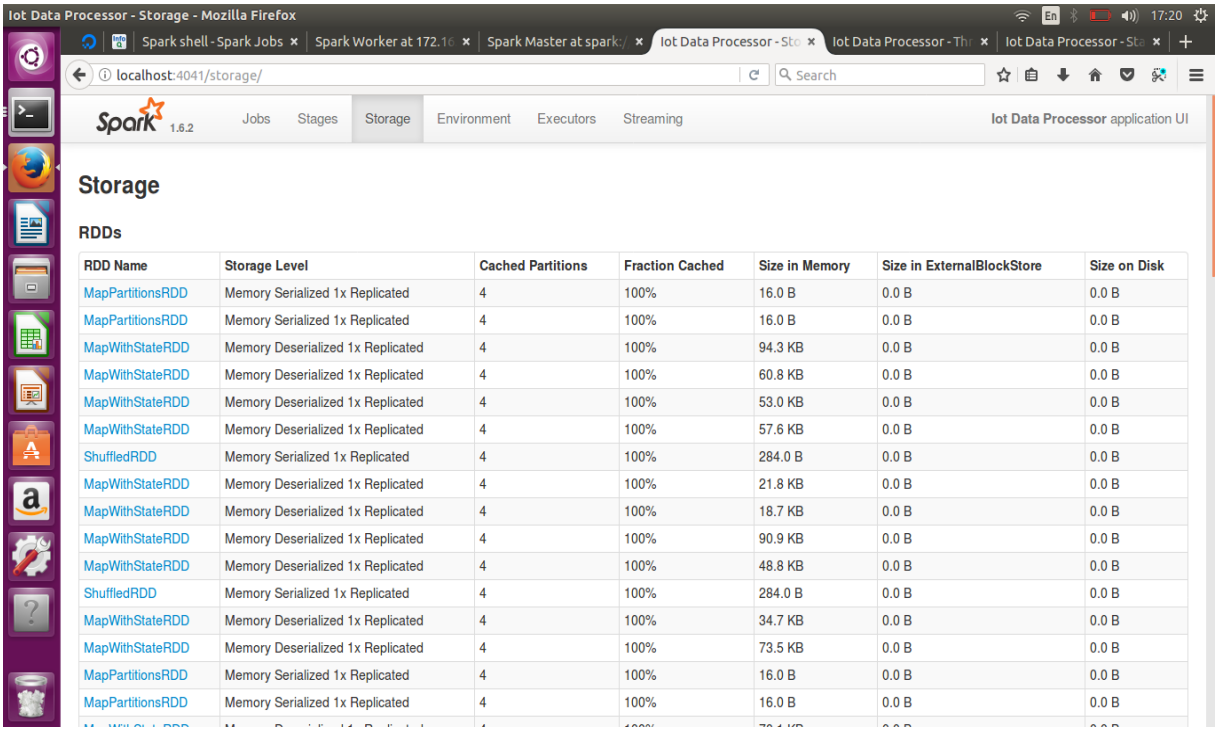

Figure 4.7: The Resilient Distributed data of the Spark

Figure 4.8 below shows the completed Batches.

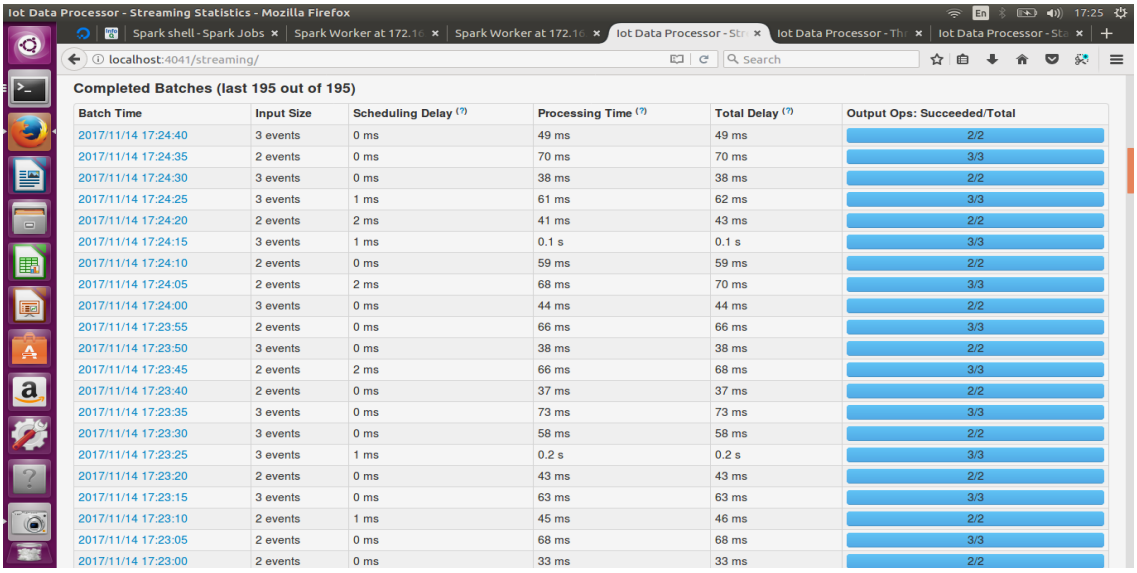

Figure 4.8: Completed Batches

4.7 Streaming statistics

Figure 4.9 depicts the Spark streaming, the running batches, the time and date it started, and also the completed batches. After this process, the results are queried from the database.

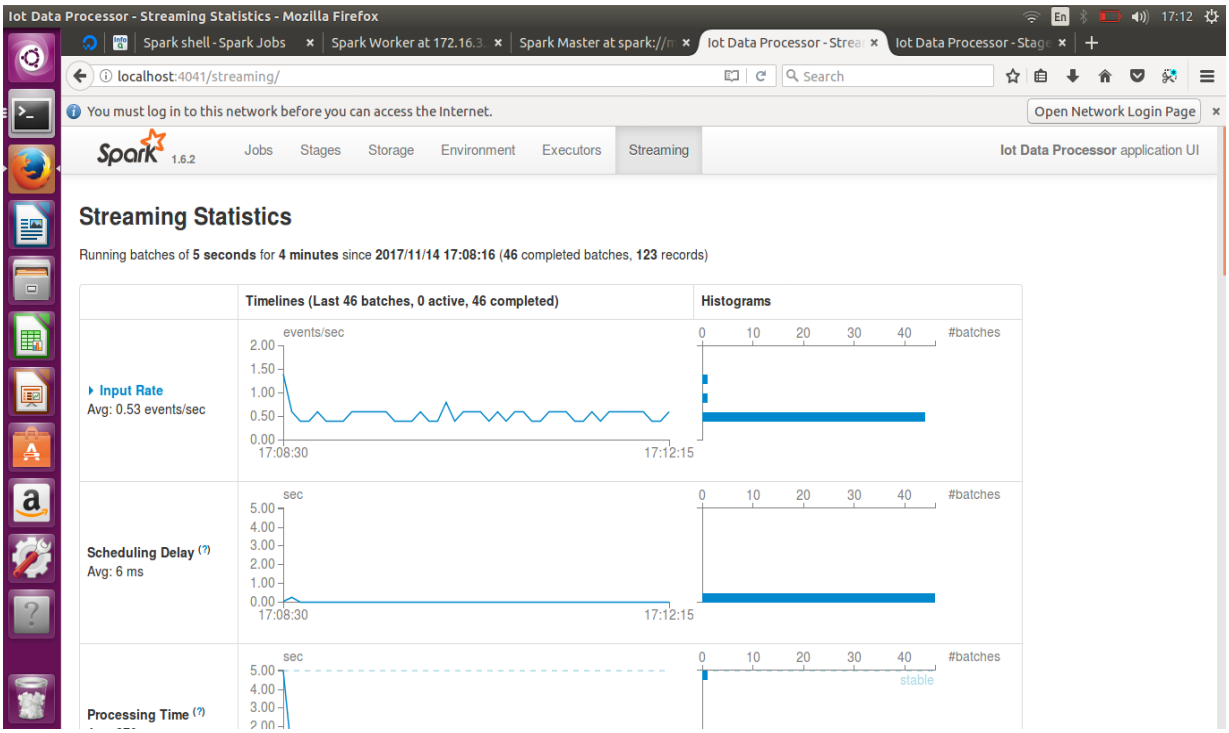

Figure 4.9: Spark streaming statistics

Figure 4.10 overleaf gives the results of implementation. (Number of vehicle type on a particular route.)

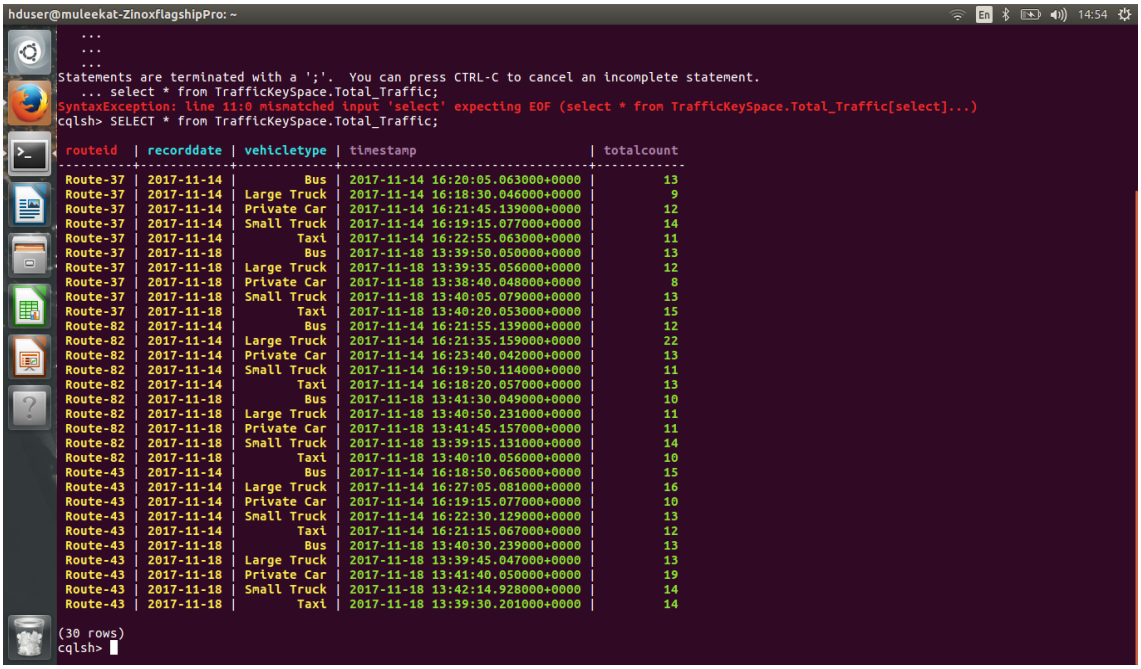

Figure 4.10: Results of implementation. (Number of vehicle type on a particular route.)

4.8 CHALLENGES

The challenges faced were to start learning new tools of the big data methods which was not easy and also in acquiring a computer with an appropriately sized a RAM to install the program. We were not able to get a compatible system which lead to a system crash and meant we had to start all over again.

CHAPTER FIVE
SUMMARY, CONCLUSION AND RECOMMENDATION

Chapter Five summarizes the work done on the thesis, the conclusion of what has been achieved and how it was achieved. It also discusses about what should be done in the future and the method to follow to achieve these recommendations.

5.1 Summary

The thesis tries to monitor the movement of traffic by using the connected cars in the IoT environment. The objectives achieved from the implementation are that, we were able to perform real-time data stream processing using Apache Kafka with Spark and also, to process data in the IoT environment using the information captured from the connected cars. The information from the cars was captured and used to make data driven decisions. It tries to monitor the movement of the vehicles, the longitude and latitude which is used to calculate the distance, and processes the data as soon as it is received using Apache Spark.

The proposed system achieved the aim of performing real-time streaming integration of Apache Kafka with Spark which captures the data from the IoT devices, and also does the processing instantaneously.

5.2 Conclusion

In an IoT environment such a smart city with a traffic control system in , the problem lies in monitoring road traffic in order that life may be comfortable for the citizens within the city.

The objective of this thesis was to use data generated from the IoT devices to monitor traffic and to make decisions using the data captured from the sensors instantaneously because the data are at most times, critical and time-sensitive.

This approach was taken by understanding how the smart devices work in an IoT environment, understanding how to generate data and capture the data from these devices and also to be able to process the data captured instantaneously. The approach was achieved by using Kafka which is a distributed streaming platform and a messaging system to publish messages for streaming.

Firstly, Kafka topics were created which was used to store the messages, then the brokers or the Kafka server was used to replicate the messages in order to avoid message loss and ensure that messages were delivered successfully for processing by Spark streaming. Spark streaming, the consumer fetches the messages from the topics and processes immediately. Also, it tries to understand the nature of the messages in order to be able to process the data.

Secondly, Spark processor processes the data and pushes the data to the Cassandra database. In the Cassandra database, key spaces were created in which the messages that were processed were stored and key space is used to define data replication on nodes. Further, the spring boot was used to capture the processed messages from the database and display these messages on the dashboard for monitoring.

5.3 Future work

It is recommended that future work on traffic monitoring should use the Kaa IoT platform to run the application. Kaa IoT is a highly flexible, multi-purpose, 100% open-source middleware platform for implementing complete end-to-end IoT solutions, connected applications, and smart products. It generates real-time data from smart devices to develop IoT applications. The data displayed on the monitoring dashboard should be used to avoid traffic congestion and prevent road accidents in the city. It

should also use decision tree or random forest to predict probable future events and should compare the two methods to determine which will work faster and better.

REFERENCES

Al Nuaimi, E., Al Neyadi, H., Mohamed, N., & Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications*, *6*(1), 25. https://doi.org/10.1186/s13174-015-0041-5

Amini, S., & Prehofer, C. (n.d.). *Big Data Analytics Architecture for Real-Time Traffic Control*, (Tum Llcm).

Apache Kafka. (n.d.).

Architecture of smart cities. (n.d.).

Caliri, G. V. (n.d.). *Introduction to Analytical Modeling*.

Chong, M. M., Abraham, A., & Paprzycki, M. (1997). *Traffic accident analysis using.*

Cities, S. (2015). *IoT What is IoT*?

Corporation, I. B. M. (2013). Hive © 2013.

Gehlot, R. (2016). *Storage and Retrieval of Data for Smart City using Hadoop*, *3*(5), 85–89.

Hahanov, V. (2015). *Smart traffic light in terms of the Cognitive road traffic management system (CTMS) based on the Internet of Things*. Volodymyr Miz PhD student at Kharkov National University of Radio.

Highl, C. H. (n.d.). *Smart city*.

Lakshminarasimhan, M. (2016). *IoT Based Traffic Management System Advanced Traffic Management System Using Internet of Things*, (March), 0-9.

Okuda, T. (2012). *Smart Mobility for Smart Cities*, *61*(3), 141–146.

Paper, C. (2015). *Scope and challenges of Internet of Things: An Emerging Technological Innovation Scope and challenges of Internet of Things*. (February).

Policy, I. T. U., & Division, T. W. (2015). *Smart Cities Seoul: a case study*, (February 2013).

Prathilothamai, M., Lakshmi, A. M. S., & Viswanthan, D. (2016). *Cost Effective Road Traffic Prediction Model using Apache Spark*, *9*(May). https://doi.org/10.17485/ijst/2016/v9i17/87334

Radek Kuchta, R. N., Kuchta, R., & Kadlec, J. (2014). Smart City Concept, Applications and Services. *Journal of Telecommunications System & Management*, *3*(2), 1–8.

https://doi.org/10.4172/2167-0919.1000117

Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. (2016). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *arXiv Preprint arXiv:1609.08089*, (October). Retrieved from http://arxiv.org/abs/1609.08089

Smart Cities Are Built On The Internet Of Things. (n.d.).

Technology, I. J. I., & Science, C. (2014). *Traffic Accident Analysis Using Decision Trees and Neural Networks*, (January), 22–28. https://doi.org/10.5815/ijitcs.2014.02.03

To, N. O. T., & Cited, B. E. (2016). *Issues Paper On Smart Cities and Infrastructure*. Advance Unedited Draft, (January).

Universal, H. P. E., & Platform, I. (n.d.). *Smart cities and the Internet of Things*.

What is Apache Hadoop_. (n.d.).

What is Apache Spark. (n.d.).

What is a Smart Building_ _ Building Efficiency Initiative _ WRI Ross Center for Sustainable Cities. (n.d.).