Accepted Manuscript

Urban data management system: Towards Big Data analytics for internet of things based smart urban environment using customized Hadoop

Muhammad Babar, Fahim Arif, Mian Ahmad Jan, Zhiyuan Tan, Fazlullah Khan

PII: S0167-739X(18)32109-5
DOI: https://doi.org/10.1016/j.future.2019.02.035
Reference: FUTURE 4788

To appear in: *Future Generation Computer Systems*

Received date : 1 September 2018
Revised date : 17 January 2019
Accepted date : 20 February 2019

Urban Data Management System: Towards Big Data Analytics for Internet of Things based Smart Urban Environment using Customized Hadoop

Muhammad Babar[a,b], Fahim Arif[a], Mian Ahmad Jan[c,*], Zhiyuan Tan[d], Fazlullah Khan[c,]

[a]*Software Engineering Department, National University of Sciences and Technology Islamabad, Pakistan*
[b]*Department of Computing and Technology, Iqra University, Islamabad Campus*
[c]*Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan*
[d]*School of Computing, Edinburgh Napier University, United Kingdom*

Abstract

The unbroken amplification of a versatile urban setup is challenged by huge Big Data processing. Understanding the voluminous data generated in a smart urban environment for decision making is a challenging task. Big Data analytics is performed to obtain useful insights about the massive data. The existing conventional techniques are not suitable to get a useful insight due to the huge volume of data. Big Data analytics has attracted significant attention in the context of large-scale data computation and processing. This paper presents a Hadoop-based architecture to deal with Big Data loading and processing. The proposed architecture is composed of two different modules, i.e., Big Data loading and Big Data processing. The performance and efficiency of data loading is tested to propose a customized methodology for loading Big Data to a distributed and processing platform, i.e., Hadoop. To examine data ingestion into Hadoop, data loading is performed and compared repeatedly against different decisions. The experimental results are recorded for various attributes along with manual and traditional data loading to highlight the efficiency of our proposed solution. On the other hand, the processing is achieved using YARN cluster management framework with specific customization of dynamic scheduling. In addition, the effectiveness of our proposed solution regarding processing and computation is also highlighted and decorated in the context of throughput.

Keywords: Big Data Analytics, Smart City, Internet of Things, Hadoop.

1. Introduction

With the passage of time, the technological growth has revolutionized the generation of data [1]. Unlike the landline phones of earlier ages, the availability of smart phones has made our lives smarter. We used to have floppy disks for data storage, however, the same data are now stored at the cloud. A huge amount of data is generated by each action performed using the mobiles phones [2]. The introduction of smart cars in the transportation industry has increased the scale of data generation. These cars have a number of sensors to record every happening event in the context of a vehicle's functionality. Thus, the volume of data has increased exponentially. Besides, the generated data is not in a structured form [3]. Internet of Things (IoT) plays an essential role in the evolution of data. IoT connects the physical objects with the Internet and makes the objects smarter. IoT is the organization and arrangement of interconnected machines, objects and computing platforms to transmit data over a particular network. IoT has changed the entire digital world and is the main reason behind the evolution of data. It is predicted that there will be 50 billion physical devices integrated in the Internet by 2020 [4].

IoT forms the base of smart urban setup and its services. These services include but are not limited to transportation, smart parking, healthcare, waste management and smart grid [5, 6, 7, 8]. Smart urban is not only about the integration of IoT and ICT, but also about a voluminous amount of data produced in these environments. This huge collec-

*Corresponding author
Email address: mianjan@awkum.edu.pk (Mian Ahmad Jan)

tion of data is used for intelligent decision making in the context of smart governance. The main objective of smart urban is to improve the quality of life and the effectiveness of urban services and operations. At the same time, it conforms various requirements with regard to social and environmental facets [9]. The benefits of smart urban are persuasive in case of acquiring a huge amount of gigantic data from IoT and ICT-based environments. A smart city chooses among the best available technologies and skills to solve urban challenges such as, air pollution, loss of mobility due to traffic congestion, energy inefficiency, and city crimes. The worldwide smart urban returns are predicted to rise up to 88.7 billion USD by 2025 from 36.8 billion USD in 2016 [10].

In recent years, Big Data has established a notable impetus from industry, governments, and research societies. Big Data can be defined as an expression that covers utilization of practices to collect, pre-process, process, compute, analyze, extract, and visualize huge data in a practical time slot, which is unavailable to standardized technologies of ICT [11, 12, 13, 14]. This definition can be interpreted as a relative term describing the circumstances for existence of Big Data. The first point of this definition indicates different dimensions of data to be termed as Big Data V's in which the major dimensions are volume, velocity and variety [14]. Development and expansion of smart urban and smart production is a foremost anxiety in this era, where the smart city and industry can be developed using Big Data analytics and IoT [15]. Presently, there is an eagerness regarding the potentials possessed by innovative and wide-ranging sources of data to understand, manage, control and administer the cities in a smarter way [16]. The constituency of the smart cities and societies is totally dependent on Big Data as the latter plays a pivital role [17]. It is argued that Big Data for the most part, is being generated by sensors in the IoT-based environment. It symbolizes a deep revolution in the categorization of data that is analyzed to determine happening events in the cities [18]. A comprehensive research has been undertaken to highlight the existing works on Big Data and IoT for smart city development and efficient decision making [19].

Hadoop is comprised of a number of small subprojects, i.e., elements, that belong to its infrastructure category for distributed computing [20, 21]. There are essentially two major components of Hadoop. The first one is a storage Hadoop Distributed File System (HDFS) [22] to store the Big Data of diverse structure classways. The second component is the processing unit that allows parallel processing and is based on Map Reduce programming paradigm along with cluster resource management in a distributed environment. The other sub-projects of Hadoop offer complementary operations. The open source framework can store a huge quantity of data and can run a number of applications on several clusters of commodity hardware. By default, it is an underlying storage mechanism using Hadoop framework. HDFS makes it possible to store diverse range of huge datasets. It maintains the log file regarding the metadata i.e., stored data. Moreover, HDFS is a counterpart of the Google File System (GFS) [23]. Similar to HDFS, GFS is a distributed and partition file system that is chunk-based to maintain the fault-tolerance property by data replication and partitioning. It is the fundamental storage layer or space of cloud computing platform, provided by Google. The Hadoop provides the paramount data management necessities that are verified by various proposals using Hadoop platform for a large-scale network. Hadoop is highly scalable and cost-effective, which is justified by proposing methods for speedy event discovery on an enormous quantity of data.

In this paper, we propose an urban data management system using Hadoop to address the issues of Big Data analytics. The proposed scheme is Hadoop-based architecture that deals with data loading and processing. The proposed scheme is comprised of two different parts. The first part is responsible for transferring and storing the Big Data in Hadoop and the second part deals with the data processing. The major contributions of this paper are as follow.

1. Initially, a data ingestion utility is customized for loading the data efficiently into Hadoop. The utility loads the data in parallel, which helps to ingest the data quickly in order to achieve efficient working of the overall system.
2. The HDFS architecture is customized with regard to replicas and block size in order to avoid the network overhead while loading the data into Hadoop. Thus, our proposed customization of Hadoop system architecture assists the parallel data loading.
3. A novel utilization of Hadoop latest description is proposed that is based on YARN (Yet Another Resource Negotiator). The proposed

YARN-based solution facilitates the system architecture to provide efficient processing of Big Data being generated by smart devices in an IoT environment.

4. Finally, extensive simulation is conducted using Apache Hadoop by considering authentic and reliable datasets produced in simulated scenario of the smart urban. The simulation results reveal that the proposed architecture is feasible for analyzing huge datasets generated in an IoT-based smart environment.

The rest of this paper is organized as follows. A review of related works is presented in Section 2. The proposed architecture is presented in Section 3. Section 4 elaborates the analysis and results. Finally, Section 5 presents the conclusions of the proposed work.

2. Related Work

The growth of smart urban attracts the concentration of researchers and scientists in the course and direction of a proficient architectural devise. A typical smart city design can present a variety of returns. In addition, a large variety of work related to Big Data analytics and IoT from theoretical to a complete set of processes are being enclosed by the smart city. At present, a number of research groups are functional to develop different solutions to illustrate a broad design for smart city, based on Big Data analytics and IoT. Moreover, a variety of proposals have been proposed that pursue thorough experimentation and simulations, based on the test beds, to conquer the issues regarding the analysis of Big Data generated in the IoT-based smart city environments. To establish the prospective benefits of Big Data analytics for smart cities, SmartSantander test bed in North Spain was designed [24, 25], where the analysis related to a particular season, traffic, temperature, and working days were performed to describe a network with several interacting entities. Likewise, a smart city architecture was proposed from data viewpoint [26].

Yet Another Resource Negotiator (YARN) is the brain of Hadoop responsible for the core activities [27]. It is responsible for cluster management in Hadoop later description. It performs all the processing actions by scheduling tasks and allocating the resources. It is comprised of two major units, i.e., resource manager and node manager. YARN was introduced in the latest description of Hadoop [28, 29]. It detaches the main operations of resource management, job tracker, and job scheduling to a separate daemon. The basic idea is to encompass an inclusive resource management controller and per-application one specific application master to separate the cluster management and core processing [28]. Those applications that require write once and read many times get the most utilization out of Map Reduce (MR) programming paradigm [30]. In Hadoop framework, there are various programming languages available that support MR programming paradigm such as Ruby, Java, and Python [31, 32]. There are some basic classes for Map Reduce processing which are provided by various programming languages. In an MR program, there are two functions performed, i.e., Map() and Reduce(). The Map function carries out actions like grouping, filtering and sorting while Reduce summarizes and aggregates the result by mapping. The input/output of the MapReduce are in the key-value pair (K, V) format.

In [33], a 3-tier architecture was proposed for smooth communication among heterogeneous connected devices across a ubiquitous platform. To build up the physical execution of a large-scale IoT infrastructure in a Santander city, a scheme on a variety of test bed components was proposed [25]. In the literature, it is stated that the *things* can be linked and communicated via Internet to be utilized for different applications [34]. The Internet vision can also be assumed as 'Ubiquitous IoT' [35] that is close to the idea of social association model. One of the complementary approaches for smart urbans to conquer the mobility issues is to spearhead the scientific bound with the Big Data [36]. Similarly, the Big Data management is considered the key for smart grid management [37].

Various platforms and solutions are designed for describing the combination of IoT and social network [38, 39]. The purpose of the web cannot be underestimated to connect various devices [40]. Each picky application needs multifaceted amalgamation effort, and consequently practical capability, endeavor and instance which avoids the consumers from producing small premeditated applications using sensor networks. There are numerous works performed to manage an enormous amount of data and offer IoT services. There are numerous difficulties in Big Data management, however, difficulties due to sophisticated IoT environment starts to be extremely precious learning in research [13]. In addition, various solutions are used for implementation to deal with Big Data in the context of offline and online enormous data coming from IoT. Big Data from the linked things can be analyzed with the help of various storage services [41]. These storage services improve data scalability, accessibility, flexibility, and compliance. Connecting IoT with social network, the concept of Big Data is kept side-wise. The Big Data and IoT are becoming very popular to incorporate other disciples. For instance, advanced machine learning methods, e.g. deep computation, has accomplished the efficient recital for Big Data feature learning [42, 43, 44]. Similarly, the computation method of deep convolution facilitated noteworthy improvement in Big Data feature learning [45, 46]. This is because, IoT and Big Data have an extremely influential relationship to work together as these are the main sources of smart urban.

3. Proposed Methodology

Our proposed urban data management system is composed of various layers. The data analytics in smart urban applications is performed at layers, based on different architectures. These layers include source systems and data collection, data loading and processing, and results utilization as depicted in Figure 1. The description of these layers along with the proposed architecture is provided in the following subsections.

3.1. Source Systems and Data Collection

Data collection is the first layer of our proposed architecture. This layer is responsible for acquisition and organization of data. These tasks are performed prior to data processing and computation. A practical smart city does not merely contain an impressive amount of data but involves multifaceted and wide-ranging computation. The apprehension of a smart city implementation depends on all aspects of data and computation due to their unavoidability. A smart city concept endeavors to make the most effective use of residential resources to diminish traffic clogging, to offer proficient healthcare facilities, to perform environmental convenience, weather and forecast judgment, and to carry out the water and electricity management. Data acquisition is a practice to sample the signals that measure the real-world scenarios and transform the results into digital values that can be operated by a digital machine, e.g. a computer.

The acquisition functionalities are performed by different data acquisition systems that transform the analog data into digital form. The data acquisition is a cumbersome and difficult task due to the massive amount of data produced by inhabitants of the smart cities. Therefore, the apprehension of the proposed architecture initiates with the wide-ranging data acquisition that is not a part of the proposed scheme. We assume that the data is obtained by the concerned smart city development departments. These departments extract the data from the society by deploying heterogeneous sensors inside the city that are responsible for gathering the real-time data from the environment. A set of centers (containing the smart community development departments) is connected to the proposed system by providing the datasets of their corresponding departments such as, Set_1, Set_2, ..., Set_n. In addition, each set further contains k number of nodes, i.e., MN_1, MN_2, ..., MN_k, and is mathematically expressed as:

$$Data = \sum_{i=1}^{n} Set_i. \qquad (1)$$

where, Set $= \sum_{i=1}^{k} MN_i$

3.2. Data Loading and Processing

This layer is responsible for two different tasks, i.e., data loading and data processing. The data loading to Hadoop, also known as data ingestion, is performed using a multiple attribute criteria model that includes customized replica mechanism, customized block-size, and customized tool. The HDFS divides and saves large files into small chunks, i.e., blocks. The default size of a block is 128MB, but the proposed size is larger, i.e., 256MB. This selection is due to the volume of input datasets. The preference of a smaller size would create too many data blocks that may increase the metadata. This in turn will increase the overhead. In addition to block size, the number of replicas are also taken into consideration while loading the data. The replica mechanism makes the actual size of a dataset several times larger that requires more time for loading. The default size of the number of replicas in Hadoop is 3, but the proposed replicas are configured to 2. The replication process is used to copy the actual data blocks several times, which is a time consuming process. Therefore, we propose a customized replication factor. The configured and customized replication improves the performance of

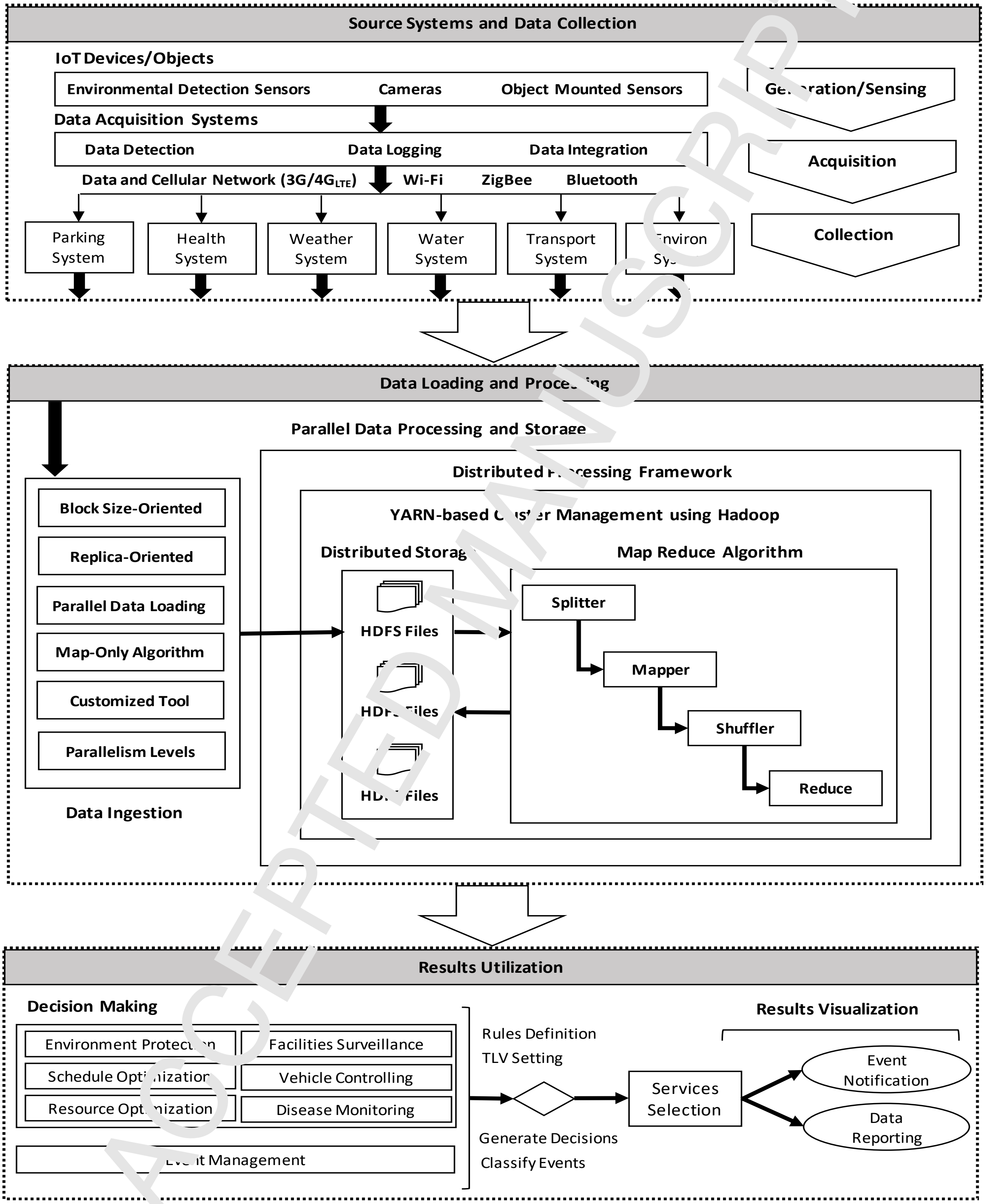

Figure 1: Proposed Urban Data Management System

data loading in the context of time consumption. Moreover, the Apache Sqoop tool is used to load the data. The Sqoop offers parallel data loading, using the map-only algorithm. In addition, it also offers customization, a mechanism to improve its efficiency. The Sqoop is also preferred because of its openness nature.

The proposed architecture uses Sqoop connectors that provide connectivity to external resource systems. Data movement between external systems and Sqoop is made possible with the assistance of these connectors. The relational databases differ with respect to the dialect somehow, otherwise, they are designed with SQL standard in general. This variation in dialect brings challenges when come up to data transfers crossways different systems. These challenges are overcome with Sqoop connectors. There connectors are available for proper functioning of a variety of accepted sources. Each connector is familiar with its linked DBMS to interact. The generic Java Database Connectivity (JDBC) connector also provides the communication to any of the databases that have the support for this connector. Initially, the dataset being moved is partitioned into various segments and a map-only job of MapReduce programming paradigm is initiated with individual mappers that are responsible and accountable for transferring a segment of this dataset, as shown in Figure 2.

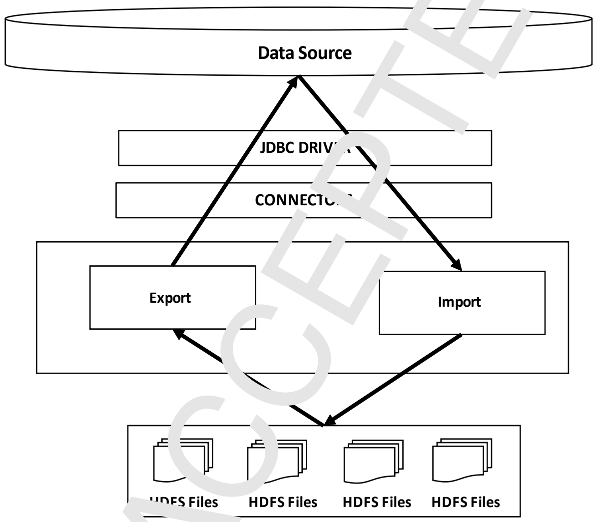

Figure 2: Data Loading using Apache Sqoop

Every record of the dataset is handled in a protected way since Sqoop uses metadata to deduce the data types. As far as the export of data using Sqoop is concerned, the export tool extracts the data from HDFS to external sources. Upon submission of our task/job, it is mapped into Map tasks which bring the lump of data from HDFS and exports them to a structured data destination. Integrating all these exported chunks of data, we receive the whole data at the destination, i.e., RDBMS (Oracle/MYSQL/ SQL Server). Reduce stage is necessary in case of data aggregation, however, Sqoop only imports (to HDFS) and exports (from HDFS) data and does not carry out any other data aggregation. Map job initiates mappers (multiple) depending on a pre-defined number. Every mapper job is allocated with a piece of data file to be loaded to HDFS (imported) for Sqoop import. Sqoop slices the input among different mappers uniformly to obtain lofty performance and efficiency. Next, every mapper makes connection and communication with the DBMS using JDBC and extracts the division of data allocated by Sqoop to corresponding arguments, provided in Command Line Interface (CLI). Integrating Sqoop with the proposed architecture automates most of the practices depending on DBMS to define the schema, i.e., metadata, that is to be loaded to HDFS (imported).

The utilization of Sqoop enables various features in our proposed architecture such as incremental load, full import, parallel and equivalent import/export, compression, easy migration, design independence, automatic code generation and extensible backend support. In addition, as Sqoop is based on MR programming paradigm, it is also managed by YARN-based cluster management scheme, as shown in Figure 3. A specific level of parallelism is achieved because the default level does not provide efficient results when the dataset size is smaller than 1GB. Therefore, a specific formula is devised to deal with this issue.

Data processing, on the other hand, is performed using MapReduce programming framework for parallel processing of large datasets. Hadoop splits the input file into chunks of identical sizes, i.e., input splits. For optimization, the split size is typically equal to the block size of HDFS. A particular map task is created for every split that executes the map function described by user for every record (row) in the split. The RecordReader is used to make the records as a pair, i.e., key-value. The map task is usually run by Hadoop on a particular node where the split lives in for a better performance. This

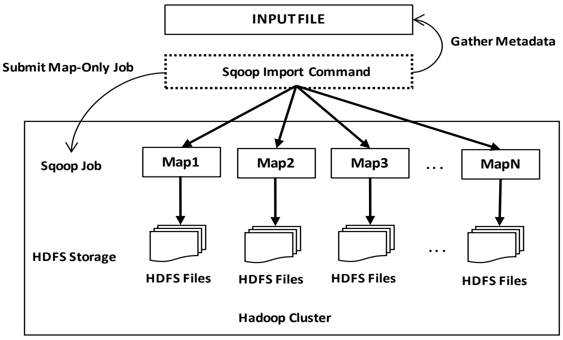

Figure 3: Apache Sqoop Operation

phenomenon is known as optimization of data locality. If the three nodes (when the replication is 3) hosting the duplication (replication) of task split are in use, scheduler is responsible to search for a free slot of the map. The operation of Map phase is shown in Figure 4.

The results of the map tasks are not written to the HDFS. Instead, they are written to the local storage where the mappers actually exist. The reduce task input is usually the results from several mappers (map tasks). Reduce tasks do not encompass the benefit of the property of data locality. Thus, the saved and shuffled map results have to move crossways the system to that particular position where reduce job is executing, and to the node where they would combine and then handed over to user-described reduce job. The result of the reducer (reduce function) is saved in HDFS. The overall working of the Reduce phase in the MapReduce process is shown in Figure 5.

Using MapReduce programming paradigm, we propose an algorithm with its application on pollution dataset. The proposed algorithm is used to collect the values of different gases at certain time (of the day) that cause the environmental pollution. The graphical representation of the proposed MapReduce algorithm is shown in Figure 6. The Map function of the proposed algorithm takes the line offset as key and the values of entire row as value. The timeStamp as key and the required associate values are emitted as value by the map function. The Reduce function groups the required associated values against each timeStamp and compares them with the Threshold Limit Value (TLV).

As MapReduce performs operations in two stages, i.e., map and reduce, a different mapper and

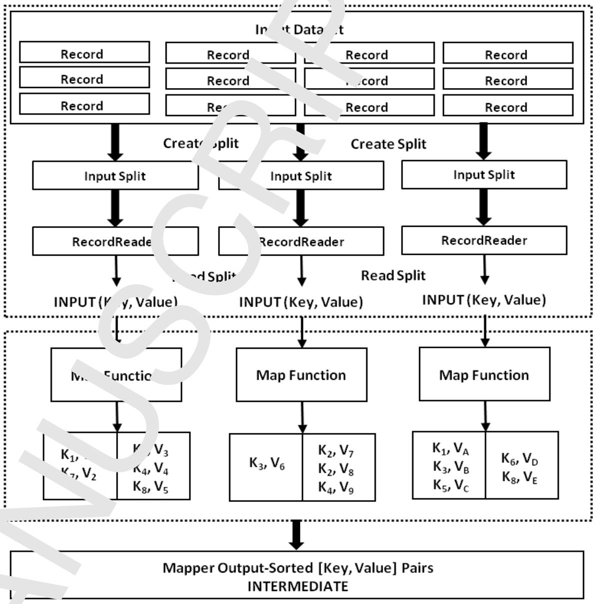

Figure 4: Mapping Process

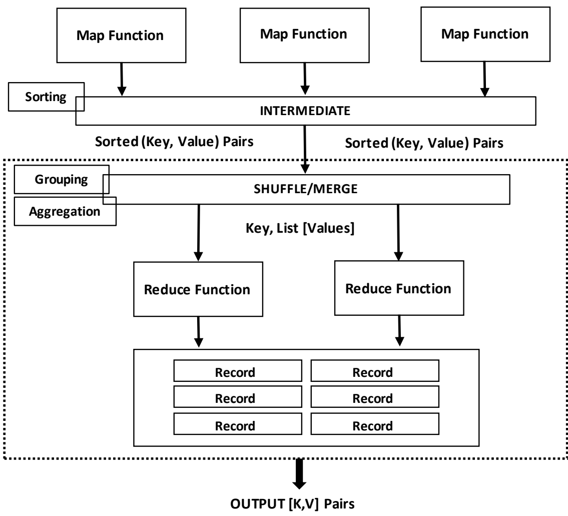

Figure 5: Reduce Phase

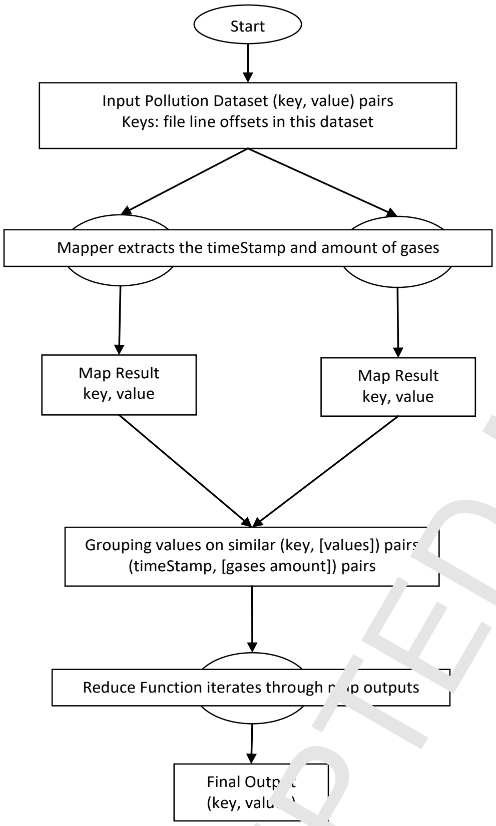

Figure 6: Proposed MR Algorithm for Pollution Dataset

a reducer are proposed for the proposed algorithm. The Map function of the proposed algorithm takes the line offset as key and the values of entire row as value. The timeStamp as key and the required associate values are emitted as value by the map function. The Reduce function groups the required associate values against each timeStamp and compares with the TLV. The mapper of the pollution dataset is given as Algorithm 1. The mapper task emits the timeStamp as key and the quantity of ozone, particulate_matter, carbon_monoxide, sulfur_dioxide, and nitrogen_dioxide as value. The Algorithms 1 is implemented using Mapper class of Java programming language.

Algorithm 1 Mapper for Pollution Dataset

procedure
BEGIN
Input:
  key: line-offset
  $value := row\_containing\_pollution\_data$
Output:
  $key : timeStamp$ ▷ value will be the sequence of all posting at a particular time
  $value : gases\_values$
  $date, all\_gases\_values := line.split('\backslash t')$
  ▷ line splitting
  $key := date$
  $value := ozone$
  $value.append(particullate\_matter)$
  $value.append(carbon\_monoxide)$
  $value.append(sulfur\_dioxide)$
  $value.append(nitrogen\_dioxide)$
  $emit(key, value)$
END

Similarly, the reducer of the parking dataset is given as Algorithm 2. The reduce task emits a row containing values of ozone, particullate_matter, carbon_monoxide, sulfur_dioxide, and nitrogen_dioxide against each timeStamp. The Reducer algorithm is implemented using Reducer class of Java programming language.

The proposed Big Data analytics architecture is based on the latest description of Hadoop distributed and parallel processing framework. The latest description of Hadoop framework is embedded with YARN and is responsible for cluster resource management and data processing. Unlike classical MapReduce (earlier description of Hadoop), YARN basically separates the process-

Algorithm 2 Reducer for Pollution

procedure
BEGIN
Input:
    key: timeStamp
    $value := row\_containing\_amount\_of\_gases$
Output:
    $key : timeStamp$
    $value : amount\_of\_ozone, particullate\_matter, carbon\_monoxide, sulfur\_dioxide, nitrogen\_dioxide$
    final [ ]
    for each_gas at timeStamp **do**
        final.append (ozone, particullate_matter, carbon_monoxide, sulfur_dioxide, nitrogen_dioxide)
    End for
    $key := timeStamp$
    $value := final$
    $emit(key, value)$
END

ing components and resource management. The YARN-based solution is not restricted to MapReduce. We preferred the YARN because of the limitations and issues in traditional MapReduce which are mainly allied to resource usage and utilization, scalability, and workload support, unlike to MapReduce. The YARN is also modified to improve the efficiency.

3.3. Results Utilization

This layer is responsible for decision making and producing and communicating events. As it exists on top of the proposed architecture, therefore, it is the moderator between processing unit and the end user. The decision and event management unit of this layer is used to classify the events and generate the decisions. The smart decisions illustrate the decision, based on ontology, that is utilized to unicast the results (events) and the consequent sections differentiate high-level and low-level events. The departmental level stores high-level events while the low-level events are now transferred further down the level. Figure 7 represents the structure (layered) that includes departmental, services, and subservices level. The self-sufficient results are unicasted to the division unit in order to send the decisions to the corresponding smart city development departments such as, smart traffic control department and smart heal department. Later, the communication channel is further identified based on their services, i.e., smart traffic management, smart accident control etc. Afterwards, the decisions are sent to the corresponding lowest level subservices identified as road congestion control, accident location management etc. The decisions are thoroughly analyzed and based on proper analysis, the notifications are generated. Finally, the notification component determines the specific recipient, based on any generated event. Accordingly, it informs the user with the produced event for its execution. Assume, the sensors implanted in a city observe a street congestion. The ontology determines the respective departmental event according to the decision message, i.e., street congestion. The event is unicasted to the smart traffic control department at the application level. The departmental level determines the service event component as traffic congestion. Successively, the produced event is sent to the subservice level, and finally, the event is notified to the individual receiver via the notification component.

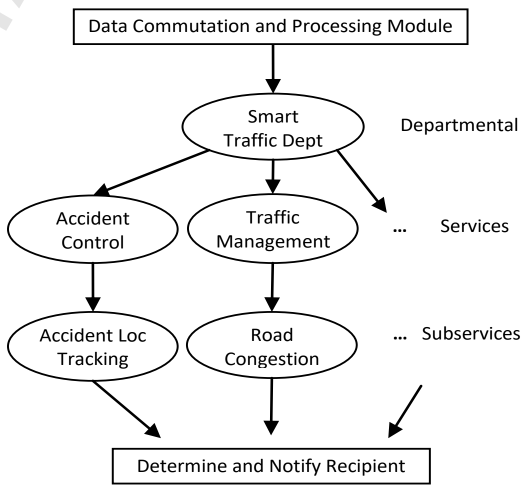

Figure 7: Event Generation Process

For the evaluation of different datasets, various thresholds are set and different rules are defined. These rules are used by our proposed algorithms for data processing. The threshold is a specific value or limit and is denoted by TLV. This value is set for each dataset such as temperature TLV for fire detection, water consumption TLV for alarming water level, traffic transport TLV for traffic congestion and so forth. TLV can be a specific integer value or in the form of percentage such as, 90 cubic liters, 12 vehicles, 90% pollution etc. The TLVs are the boundary limits for different actions to be performed. The decision making and event generation is totally based on this TLV. Similarly, various rules are also defined based on the corresponding TLVs. These rules are if/then statements that are based on pre-defined TLVs for decision making.

4. Data Analysis and Results

In this section, the detailed analysis and discussion on obtained results using our proposed architecture are taken into consideration. Analysis are carried out on an authentic and reliable dataset to evaluate the proposed architecture using different designed algorithms. The proposed design is free from open issues and exclusively depends on the processing of previous data.

4.1. Implementation Detail and Data Source Information

The implementation of our proposed system is performed using cluster of Hadoop on Ubuntu 16.04 LTS operating system along with Apache Sqoop utility. In addition, corei5 processor with a RAM of 8GB is utilized and operated for implementing the proposed solution. Moreover, the MapReduce algorithms are implemented using Java programming language with predefined mapper and reducer classes. The datasets are attained from a valid and reliable source that are accessible and authenticated. It consists of pollution dataset that contains information about different toxic gases such as ozone, carbon monoxide, sulfur dioxide, nitrogen dioxide, and so forth [47]. The pollution data is annotated (semantically) datasets for the CityPulse EU FP7 project. Moreover, this data is licensed under Creative Commons Attribution 4.0 International License. These datasets are freely available online [47]. Moreover, these datasets are used in a variety of research [48, 49, 50]. These solutions, i.e., [48, 49, 50] are provided with regard to smart city data management. The pollution data is measured and collected using Air Quality Index (AQI) metric (total of 449 observations). The data is available and accessible in raw form of CSV (Comma Separated Values) and semantically interpreted format using the information model of CityPulse. The time frame of this dataset is August 2014 - October 2014. The values for carbon_monoxide, sulfur_dioxide, nitrogen_dioxide, ozone and particulate_matter levels

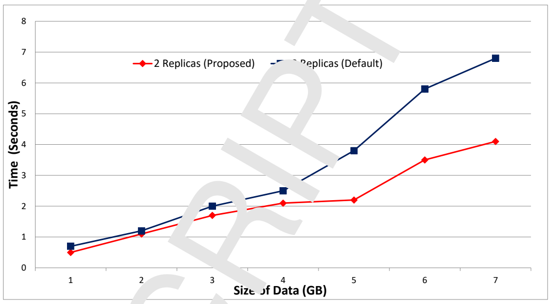

Figure 8: Customized Replicas

of index have been given according to API (Air Pollution Index)

4.2. Data Loading and Ingestion Results

The data loading time difference is not noticeable when the data size is small. Due to replica mechanism, the data loading time is quite noticeable when the dataset size is large. The question that arise is the size of a specific dataset, i.e., its threshold value. The threshold for dataset size is a value that is the equivalent size of dataset from where the data loading time difference is noticed. To find the threshold, we measure the performance of data loading using test datasets of different sizes. The threshold of a dataset size is the value where time variation tends to be greater than 0. When the variation is greater than 0, significant changes occur. As Hadoop might be occupied by other jobs running by some other users, we may get dissimilar time to load the same size of dataset, twice. Therefore, the time variation equivalent to threshold value can be described as a specific range in order to overcome the said issue such as, 0 to 6 seconds, to discover the threshold. The thresholds for different parameters are established using the results of the same experiments. Taking data loading utility into consideration, the threshold is up to 900MB (dataset file size) where the impact of data loading time starts, as shown in Figure 8. As the figure shows, up to 1GB of dataset file does not generate any difference even if an automated data loading technique is used. The efficiency is achieved when the dataset size is greater than at least 900 MB.

Similarly, Figure 9 highlights and demonstrates that the threshold value for replica mechanism is 1.7GB.

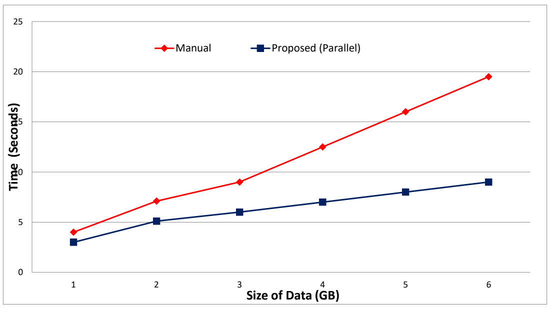

Figure 9: Customized Tool

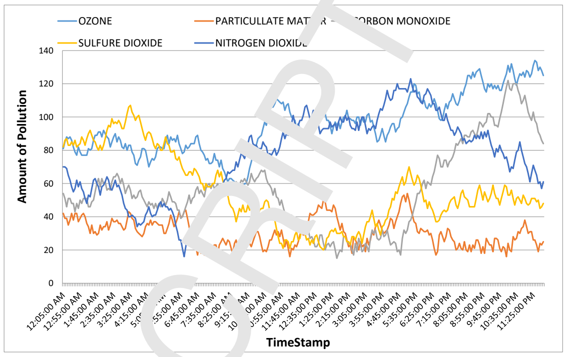

Figure 11: Pollution Amount at Different TimeStamp

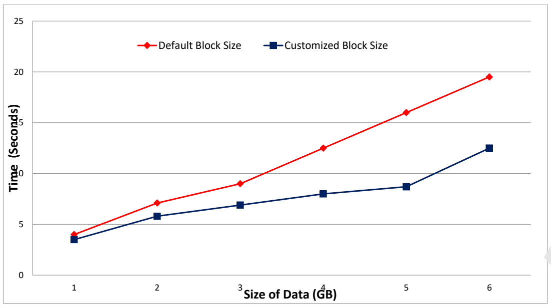

Figure 10: Dataset Size Threshold for Block Size

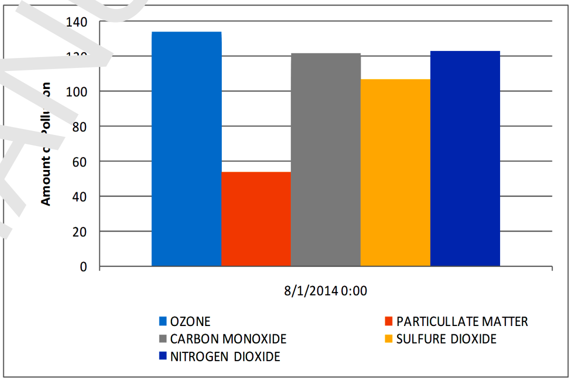

Figure 12: Pollution Amount of a Specific Day

In a similar fashion, Figure 10 demonstrates the threshold for customized block size of HDFS which is 1GB.

4.3. Result Discussion on MR Algorithm for Pollution Dataset

As the industrial, transportation, and domestic appliances' usage increase, the production of pollution increases radically. To control and manage the pollution and its consequences, the pollution data of Aarhus city is investigated. The amount of different gases at different time of the day is collectively shown in Figure 11. This figure demonstrates the amount of Ozone, Particullate Matter, Carbon Monoxide, Sulfur Dioxide, and Nitrogen Dioxide gases. In Figure 12, the horizontal axis represents different time intervals from 00:00:00 AM to 12:00:00 PM, while the vertical axis represents the amount of different gases in different colors. It is noticed that Particullate Matter has less quantity throughout the entire day while the amount of other gases fluctuates at different time of the day.

In addition, the pollution of a specific day, i.e., August 1, 2014, is also shown and highlighted in Figure 12.

Moreover, the individual amount of Ozone, Particullate Matter, Carbon Monoxide, Sulfur Dioxide, and Nitrogen Dioxide is also demonstrated separately. The amount of Ozone is specifically demonstrated in Figure 13. The Ozone amount is noticed towards a higher side during the day time as compared to midnight and dawn.

Similarly, the amount of Particullate Matter at different time of the day is demonstrated and highlighted in Figure 14. The vertical axis of this figure shows the amount of Particullate Matter against different timestamps at horizontal axis. The amount of Particullate Matter is less observed as compared to Ozone amount, as shown in Figure 14. Furthermore, it is observed that the amount of

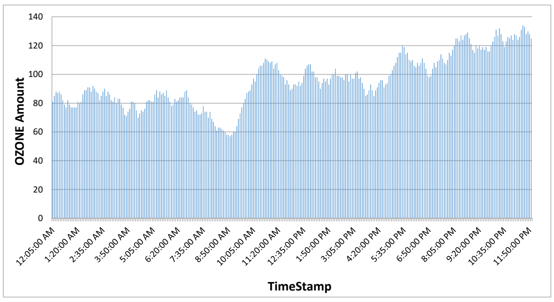

Figure 13: Ozone Amount at Different Time

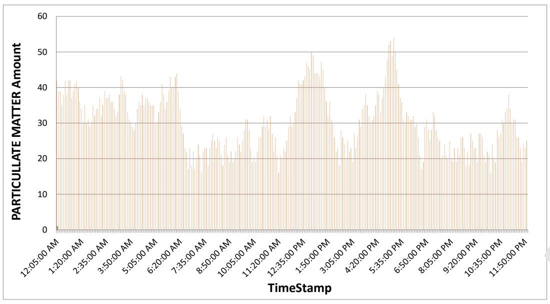

Figure 14: Particullate Matter Amount at Different Time

Particullate Matter is almost half from 6:00 AM to 8:00 AM as compared to the time interval between 12:00 AM and 5:00 AM.

Likewise, Figure 15 shows the amount of Carbon Monoxide in the air during different time intervals of the day. A smaller amount of Carbon Monoxide in the air is observed between 12:00 PM and 4:00 PM and a higher amount at the night between 9:00 PM and 12:00 AM. Figure 15 also demonstrates the amount of Carbon Monoxide in the air at peak during 10:00 PM at night.

In a same way, the Sulfur Dioxide amount in the air at different time intervals is demonstrated in Figure 16. The amount of Sulfur Dioxide in the air decreases exponentially during the time interval between 4:00 AM and 11:00 AM.

Finally, Figure 17 demonstrates the amount of Nitrogen Dioxide in the air at different time intervals of day. It is observed that the amount of Nitrogen Dioxide in the air is increased exponentially between 5:00 AM and 9:00 AM.

It is noticed that the pollution is predominantly

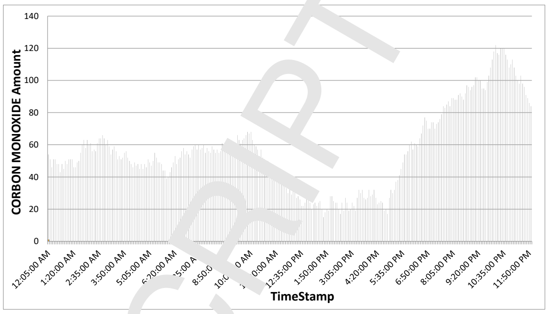

Figure 15: Carbon Monoxide Amount at Different Time

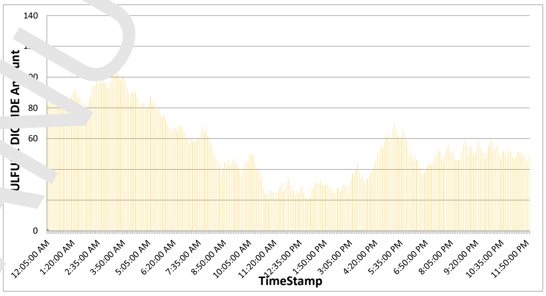

Figure 16: Sulfur Dioxide Amount at Different Time

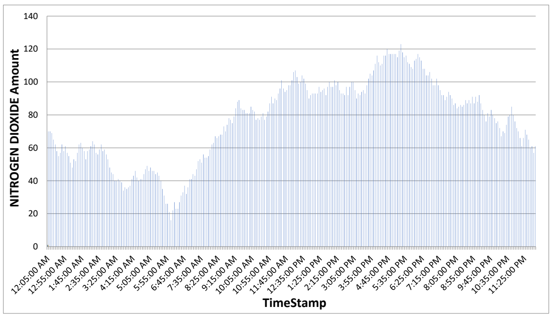

Figure 17: Nitrogen Dioxide Amount at Different Time

higher at different times of the day. The decision can be taken by the weather and forecast and health departments to circulate a messages among citizens to take precautionary measures while visiting the polluted areas. In addition, the citizens or patients can be facilitated to opt for precautionary steps (such as wearing mask etc) to avoid pol-

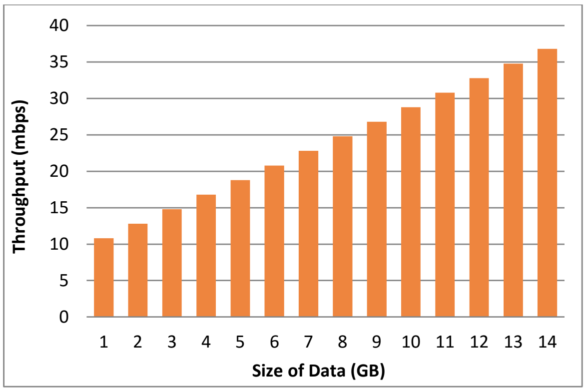

Figure 18: Throughput of proposed Solution

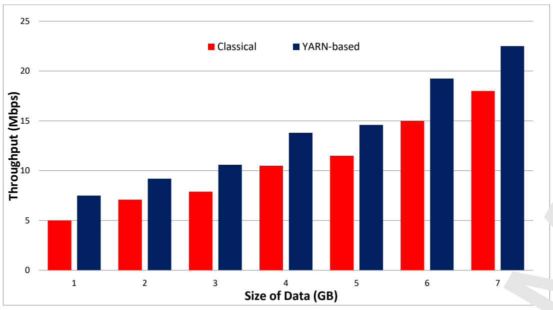

Figure 19: Throughput Comparison with Classical MR

lution. Furthermore, the concerned environments can take appropriate decisions and actions against those sources which produce a higher pollution. In nutshell, the use of this data can assist in the future urban planning.

4.4. Throughput of Proposed Architecture

Moreover, the throughput of our proposed architecture is calculated as depicted in Figure 18. It is noticed that with the increase in size of data, the speed of processing is decreased. The proposed system efficiency is considerably higher as compared to existing classical MR-based solution. In addition, the throughput comparison with classical MR is also given in Figure 19.

5. Conclusion and Future Work

In this paper, a Hadoop-based smart urban data management is proposed to deal with the problems in Big Data analytics. The projected solution particularly deals with Big Data loading into Hadoop, cluster management and computation. The proposed scheme comprised of Big Data loading and storage in Hadoop file system and Big Data computation and processing. The first part is responsible for transferring and storing the Big Data in Hadoop. The data loading performance and efficiency is tested using our proposed methodology, based on a variety of experiments, to load the Big Data to a distributed and processing platform, i.e., Hadoop. In addition, data loading is performed and compared with different decisions repeatedly and influenced features are examined. The second part of the research deals with data computation and processing. Unlike traditional MapReduce architecture, YARN-based cluster resource management solution is utilized in this research to manage the cluster resources and process the data using MapReduce algorithm separately. YARN is customized with dynamic scheduling. Using Hadoop framework, the proposed architecture is tested with reliable datasets to verify and reveals that the proposed solution offers precious impending into the society development structures to obtain better architectures. In addition, the proposed solution will be used for OFFLINE application as Hadoop only provides offline processing. Moreover, the effectiveness of our proposed scheme with regard to throughput is also highlighted in this paper.

References

[1] Judith Hurwitz, Alan Nugent, Fern Halper, and Marcia Kaufman. *Big data for dummies.* John Wiley & Sons, 2013.

[2] Chris Snijders, Uwe Matzat, and Ulf-Dietrich Reips. " big data": big gaps of knowledge in the field of internet science. *International Journal of Internet Science*, 7(1):1–5, 2012.

[3] Amy Nordrum. Popular internet of things forecast of 50 billion devices by 2020 is outdated. *IEEE Spectrum*, 18, 2016.

[4] Paul Zikopoulos, Chris Eaton, et al. *Understanding big data: Analytics for enterprise class hadoop and streaming data.* McGraw-Hill Osborne Media, 2011.

[5] Muhammad Babar and Fahim Arif. Smart urban planning using big data analytics to contend with the interoperability in internet of things. *Future Generation Computer Systems*, 77:65–76, 2017.

[6] Muhammad Babar, Ataur Rahman, Fahim Arif, and Gwanggil Jeon. Energy-harvesting based on internet of things and big data analytics for smart health monitoring. *Sustainable Computing: Informatics and Systems*, 2017.

[7] Ying Zhou, Quansen Sun, and Jixin Liu. Robust optimisation algorithm for the measurement matrix in com-

pressed sensing. *CAAI Transactions on Intelligence Technology*, 3(3):133–139, 2018.

[8] Muhammad Babar and Fahim Arif. Smart urban planning using big data analytics based internet of things. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 397–402. ACM, 2017.

[9] Houbing Song, Ravi Srinivasan, Tamim Sookoor, and Sabina Jeschke. *Smart Cities: Foundations, Principles, and Applications*. John Wiley & Sons, 2017.

[10] Eric Woods and Noah Goldstein. Navigant research leaderboard report: Smart city suppliers. In *Assessment of strategy and execution for 15 smart city suppliers*. 2014.

[11] Big Data. A new world of opportunities. *NESSI White Paper*, 2012.

[12] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[13] Alexandros Labrinidis and Hosagrahar V Jagadish. Challenges and opportunities with big data. *Proceedings of the VLDB Endowment*, 5(12):2032–2033, 2012.

[14] Richard L Villars, Carl W Olofson, and Matthew Eastwood. Big data: What it is and why you should care. *White Paper, IDC*, 14, 2011.

[15] Zhanyu Liu. Research on the internet of things and the development of smart city industry based on big data. *Cluster Computing*, pages 1–7, 2017.

[16] Luis MA Bettencourt. The uses of big data in cities. *Big Data*, 2(1):12–22, 2014.

[17] Saint John Walker. Big data: A revolution that will transform how we live, work, and think, 2014.

[18] Michael Batty. Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279, 2013.

[19] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[20] Boris Lublinsky, Kevin T Smith, and Alexey Yakubovich. *Professional hadoop solutions*. John Wiley & Sons, 2013.

[21] Tom White. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.

[22] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*, pages 1–10. Ieee, 2010.

[23] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. *The Google file system*, volume 37. ACM, 2003.

[24] Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, and Ernoe Kovacs. Building a big data platform for smart cities: Experience and lessons from santander. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 592–599. IEEE, 2015.

[25] Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, et al. Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238, 2014.

[26] Rong Wenge, Xiong Zhang, Cooper Dave, Li Chao, and Sheng Hao. Smart city architecture: A technology guide for implementation and design challenges. *China Communications*, 11(3):56–69, 2014.

[27] Vinod Kumar Vavilapalli, Arun C Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, et al. Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing*, page 5. ACM, 2013.

[28] Jia-Chun Lin, Ingrid Chieh Yu, Einar Broch Johnsen, and Ming-Chang Lee. Abs-yarn: A formal framework for modeling hadoop yarn clusters. In *International Conference on Fundamental Approaches to Software Engineering*, pages 49–65. Springer, 2016.

[29] Amogh Pramod Kulkarni and Mahesh Khandewal. Survey on hadoop and introduction to yarn. *International Journal of Emerging Technology and Advanced Engineering*, 4(5):82–87, 2014.

[30] Gaizhen Yang. The application of mapreduce in the cloud computing. In *Intelligence Information Processing and Trusted Computing (IPTC), 2011 2nd International Symposium on*, pages 154–156. IEEE, 2011.

[31] Ivanilton Polato, Reginaldo Ré, Alfredo Goldman, and Fabio Kon. A comprehensive view of hadoop researcha systematic literature review. *Journal of Network and Computer Applications*, 46:1–25, 2014.

[32] Rong Gu, Xiaoliang Yang, Jinshuang Yan, Yuanhao Sun, Bing Wang, Chunfeng Yuan, and Yihua Huang. Shadoop: Improving mapreduce performance by optimizing job execution mechanism in hadoop clusters. *Journal of parallel and distributed computing*, 74(3):2166–2179, 2014.

[33] Satyanarayana V Nandury and Beneyaz A Begum. Smart wsn-based ubiquitous architecture for smart cities. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 2366–2373. IEEE, 2015.

[34] Sandesh Uppoor, Oscar Trullols-Cruces, Marco Fiore, and Jose M Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075, 2014.

[35] Huansheng Ning and Ziou Wang. Future internet of things architecture: like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4):461–463, 2011.

[36] Susanne Schatzinger and Chyi Yng Rose Lim. Taxi of the future: Big data analysis as a framework for future urban fleets in smart cities. In *International conference on Smart and Sustainable Planning for Cities and Regions*, pages 83–98. Springer, 2015.

[37] Trinh Hoang Nguyen, Vimala Nunavath, and Andreas Prinz. Big data metadata management in smart grids. In *Big data and internet of things: A Roadmap for Smart Environments*, pages 189–214. Springer, 2014.

[38] Muhammad Mazhar Ullah Rathore, Anand Paul, Awais Ahmad, Bo-Wei Chen, Bormin Huang, and Wen Ji. Real-time big data analytical architecture for remote sensing application. *IEEE journal of selected topics in applied earth observations and remote sensing*, 8(10):4610–4621, 2015.

[39] Tein-Yaw Chung, Ibrahim Mashal, Osama Alsaryrah, Chih-Hsiang Chang, Tsung-Hsuan Hsu, Pei-Shan Li, and Wen-Hsing Kuo. Mul-swot: a social web of things platform for internet of things application development. In *Internet of Things (iThings), 2014 IEEE Interna-*

tional Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE, pages 296–299. IEEE, 2014.

[40] Dominique Guinard, Vlad Trifa, Thomas Pham, and Olivier Liechti. Towards physical mashups in the web of things. In *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, pages 1–4. IEEE, 2009.

[41] Xuan Hung Le, Sungyoung Lee, Phan Tran Ho Truc, Asad Masood Khattak, Manhyung Han, Dang Viet Hung, Mohammad M Hassan, Miso Kim, Kyo-Ho Koo, Young-Koo Lee, et al. Secured wsn-integrated cloud computing for u-life care. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–2. IEEE, 2010.

[42] Qingchen Zhang, Laurence Tianruo Yang, Zhikui Chen, Peng Li, and Fanyu Bu. An adaptive droupout deep computation model for industrial iot big data learning with crowdsourcing to cloud computing. *IEEE Transactions on Industrial Informatics*, 2018.

[43] Shuce Zhang, Hiromu Iwashita, and Kazushi Sanada. Thermal performance difference of ideal gas model and van der waals gas model in gas-loaded accumulator. *International Journal of Hydromechatronics*, 1(3):293–307, 2018.

[44] Jack L Johnson. Design of experiments and progressively sequenced regression are combined to achieve minimum data sample size. *International Journal of Hydromechatronics*, 1(3):308–331, 2018.

[45] Peng Li, Zhikui Chen, Laurence Tianruo Yang, Jing Gao, Qingchen Zhang, and Jamal Deen. An incremental deep convolutional computation model for feature learning on industrial big data. *IEEE Transactions on Industrial Informatics*, 2018.

[46] Palaiahnakote Shivakumara, Dongqi Tang, Maryam Asadzadehkaljahi, Tong Lu, Umapada Pal, and Mohammad Hossein Anisi. Cnn-rnn based method for license plate recognition. *Caai Transactions on Intelligence Technology*, 3(3):169–175, 2018.

[47] P. Dataset. Dataset collection. `http://iot.ee.surrey.ac.uk:8080/datasets.html#pollution`, Last accessed on April 7, 2018.

[48] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit P Sheth, Alessandra Mileo, and Payam Barnaghi. Semantic modelling of smart city data. 2014.

[49] Ralf Tönjes, P Barnaghi, M Ali, A Mileo, M Hauswirth, F Ganz, S Ganea, B Kjærgaard, D Kuemper, Septimiu Nechifor, et al. Real time iot stream processing and large-scale data analytics for smart city applications. In *poster session, European Conference on Networks and Communications*, 2014.

[50] Sefki Kolozali, Maria Bermudez-Edo, Daniel Puschmann, Frieder Ganz, and Payam Barnaghi. A knowledge-based approach for real-time iot data stream annotation and processing. In *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE*, pages 215–222. IEEE, 2014.

Highlights

- Big Data Management Scheme is proposed for Smart Urban Planning
- Hadoop default architecture is customized for efficient data analysis
- Efficient data loading is attained using parallel algorithm
- Resourceful data processing is achieved using efficient scheduling
- Authentic and reliable dataset is tested to verify the performance

Muhammad Babar is pursuing his Ph.D degree in Computer Software Engineering from National University Sciences and Technology (NUST), Islamabad, Pakistan. He did his Masters of Sciences in Computer Software Engineering from National University Sciences and Technology (NUST), Islamabad, Pakistan in 2012. He receives his Bachelors in Computer Sciences with distinction from University of Peshawar, Pakistan in 2008. His research area includes but not limited to Big Data Analytics, Internet of Things (IoT), Smart City Design and Planning, and Social Web of Things (SWOT). He has published his research work in various IEEE and ACM/Springer International conferences and journals.

Dr. Fahim Arif receives his Bachelors in Telecommunication from College of Telecommunication Engineering (UET Lahore) in 1995 and Master in Sciences in Computer Software Engineering from National University Science and Technology, Islamabad in 2003. He has won NUST Endowment fund scheme scholarship for NUST in 2003 and International Research Support Initiative Program Fund from HEC in 2007. He worked as international research scholar in System and Computer Engineering Department, Carleton University, Ottawa, Canada in 2007 and participated in numerous research and academic activities. He is principal investigator (PI) for a project funded by NUST. Recently, his biography has been published by South Asian Publication Who's Who in the World 2008 Edition and awarded with Star Laureate 2008 in recognition to his contributions to knowledge and research.

Dr. Mian Ahmad Jan completed his PhD at the Faculty of Engineering and Information Technology (FEIT) of the University of Technology, Sydney (UTS). His research interests are on Wireless Sensor Network Clustering Protocols, Congestion Control, Internet and Web of Things and efficient Network Intrusion detection techniques. He has published his research in top ranked Elsevier, IEEE Transactions and Conference in these areas of research. He has been recipient of various research awards such as UTS International Research Scholarship and CSIRO Top-up scholarship. Besides, he was awarded the best researcher award for the year 2014 at the University of Technology, Sydney Australia. After completing PhD, he has joined Abdul Wali Khan University Mardan, Pakistan as Assistant Professor in March 2016.

Dr. Zhiyuan Tan is a Lecturer at the School of Computing at the Edinburgh Napier University. He was awarded his PhD degree by University of Technology Sydney, Australia in 2014. Prior to joining Edinburgh Napier University, he had been a postdoctoral researcher at the University of Twente since2014. Zhiyuan has a research background in network security and pattern recognition. Zhiyuan has published 28 quality scholarly articles. His most recent research achievements have been published in several highly-cited IEEE Transactions and Elsevier journals including IEEE Transactions on Computers (TC), IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE Transactions on Cloud Computing (TCC), Future Generation Computer Systems (FGCS), and Computer Networks (CN). His research contribution on network security is internationally recognised. According to Google Scholar, he has an H-Index of 10 and his research articles have received over 370 citations since 2010. Moreover, he has earned various research awards, including a Best Paper Award and a Kaspersky Lab's Annual Student Cyber Security Conference Finalist Award, over the past years.

Mr. Fazlullah Khan is an Assistant Professor of Computer Science at Abdul Wali Khan University Mardan, Pakistan. He is currently pursuing his Ph.D, and has studied MS from Nagaoka University of Technology Japan and another MS from NUST, Pakistan. He is gold medalist from Peshawar University Pakistan, and recipient of prestigious scholarships. He has published his research work in various IEEE and ACM/Springer International conferences and journals. He has been an active reviewer for IEEE Transaction on ITS, IEEE Communication Magazine, Springer MONET, IET Wireless Sensor Systems etc.

Mr. Muhammad Babar	
Dr. Fahim Arif	
Dr. Mian Ahmad Jan	
Dr. Zhiyuan Tan	
Mr. Fazlullah Khan	

ACCEPTED MANUSCRIPT