A Scalable Big Data Framework for Real-Time Traffic Monitoring System

Wilfried Yves Hamilton Adoni (adoniwilfried@gmail.com)
International University of Casablanca
Tarik Nahhal
University of Hassan II Casablanca
Najib Ben Aoun
Al Baha University
Moez Krichen
Al Baha University
Mohammed Alzahrani
Al Baha University

Research Article

Keywords: Road sensor, GPS sensor, Intelligent transportation system, Big Data, Real-time analysis, Traffic monitoring, Roads Sensor, Urban mobility, Hadoop, IBM InfoSphere

Posted Date: January 3rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1200646/v1

RESEARCH

A Scalable Big Data Framework for Real-Time Traffic Monitoring System

Wilfried Yves Hamilton Adoni[1,2*], Tarek Nahhal[3], Najib Ben Aoun[4,5], Moez Krichen[4,6] and Mohammed Y. Alzahrani[4]

*Correspondence: adoniwilfried@gmail.com
[1]Engineering School, International University of Casablanca, Casablanca, Morocco
[2]University of Hassan II, Casablanca, Morocco
Full list of author information is available at the end of the article

Abstract

In this paper, we present a scalable and real-time intelligent transportation system based on a big data framework. The proposed system allows for the use of existing data from road sensors to better understand traffic flow, traveler behavior, and increase road network performance. Our transportation system is designed to process large-scale stream data to analyze traffic events such as incidents, crashes and congestion. The experiments performed on the public transportation modes of the city of Casablanca in Morocco reveal that the proposed system achieves a significant gain of time, gathers large-scale data from many road sensors and is not expensive in terms of hardware resource consumption.

Keywords: Road sensor; GPS sensor; Intelligent transportation system; Big Data; Real-time analysis; Traffic monitoring; Roads Sensor; Urban mobility; Hadoop; IBM InfoSphere

1 Introduction

Nowadays, the new technologies bring many benefits and the transport sector does not escape this rule [1, 2, 3]. Thanks to the mass of data generated, new strategies can be developed to anticipate certain unforeseen events, making the systems even more efficient and reliable.

The exploitation of big data technology enables transportation systems to analyze significant amount of stream data in order to prevent traffic jams, incidents, delays or track given vehicles in real-time. Big data technology has also had an impact on users' lifestyles. Indeed, we can cross the whole data of the traffic coming from several sources of information for an effective decision-making aid with regard to the mobility of the travelers.

Several projects and research works have been done in the big data technologies, this new technology allows to cross the boundaries of traditional database management system. This allows to face the challenges of massive data processing and encompasses the new data management innovations for better performance and productivity growth of enterprises [4]. Zhend, Liu and Hsieh [5] have done research on the evolution of air quality in overpopulated cities. By using big data real-time analytic tools, they provide information about air quality. By combining historical and real-time air data from existing monitor stations and weather stations, they provide derived information on real-time air pollution in the cities of Beijing and Shanghai. This work had a great impact on the environment by reducing the emission of greenhouse gases.

Our research was motivated by [6, 7, 8, 9, 10, 11, 12] and made feasible by access to data from Casablanca's buses, tramways, and GIS data of Morocco. The data originates from a variety of sources, including GPS (Global Positioning System) sensors, traffic control devices, and car embedded systems. We provide helpful information to end-users in this area by integrating static data with mobility data. The volume and variety of traffic data, the veracity and the velocity with which all traffic events are dealt in this area.

In this paper, we present the main architecture of our distributed transportation system. The proposed system is based on big data framework and allows the processing of historical and real-time traffic data. This method incorporates a wide range of technologies that may be used to handle a number of transportation issues, such as traffic prediction, regularization difficulties, the traveling salesman problem (TSP), the shortest path problem, and so on. We chose IBM big data technology as a solution, which includes InfoSphere BigInsights, InfoSphere Streams, and InfoSphere Warehouse, among other platforms.

The rest of this paper is organized as follows. Section 2 introduces related works, while Section 3 describes the conceptual model and architecture of the intelligent transport system. We introduce our experiments and results in Section 4. Finally, in Section 5, we wrap up the work and suggest some potential future expansions.

2 Related work

Big data concept attracts most interests for transport because the quantity of information managed is too large and more complex [13, 12, 14]. Current intelligent transportation systems provide multi-modal services adapted for the community. Nevertheless, it becomes a perilous task when it is a request taking into account several mobility constraints. In work conducted in [15, 16], the authors used big data techniques to cross and analyze data from traffic flow and call detail records in order to predict the density traffic flow on road network.

In the case of road traffic, all data generated by passengers is progressively augmented by additional data collected in real-time by an expanding number of traffic sensors. Dodge and Kitchin [17] have developed a transportation system that analyzes the movement of cars on the road network in real time. They evaluate vehicle speeds and automatically apply fines for speeding violations by integrating data from road sensors. The police utilize this information to respond quickly to incidents and create a log file collecting all vehicle tracking records.

The notion of big data and machine learning were combined by Lécué Freddy and all [18] in the forecast of traffic incidents in Dublin, Ireland. Tests performed prove the capability of the system to collect and manage large amount of historical sensor data and react in real-time to all captured traffic events. Other works [10, 9] proposed an intelligent framework based on big data stream flow. The proposed system collects large volume of data from many GPS sensors and combines them road network data. Similarly, Bouillet and Ranganathan [11] use the same method to generate a picture of traffic conditions using data from GPS sensors.

Recently, Adoni et al.[6] have introduced a MapReduce-based approach to analyze traffic stream data in order to detect abnormal traffic events. Their approach consists of partitioning the log file of traffic events into a set of sub-events then

analyzing traffic events on each event block in parallel way. Their proposed technique is very inspiring and achieves significant gain in term of runtime complexity. They used the similarly technique to propose parallel and distributed version of A* pathfinding algorithm called MRA* (MapReduce-A*) [7, 8] which consists to combine the A* algorithm with MapReduce paradigm to compute the shortest path on large-scale road network.

3 Proposed Framework

We present our transportation system's conceptual model. Our system is built to handle massive amounts of data from traffic flow. We used a method based on the **4V** (**V**olume **V**elocity **V**ariety **V**eracity) properties of big data. In this context we use big data solutions because it is an integrated production platform and includes all Apache Hadoop ecosystem (MapReduce, Hive, Pig, Impala, Mahout, HBase, Sqoop, etc.). It helps to take on the challenges and the complexity of MapReduce program allowing users with limited SQL knowledge to be able to manipulate lot of unstructured data. The proposed system allows to collect and analyze petabytes of structured and unstructured traffic data, to build real-time predictive models of traffic flow in order to detect and react quickly to any traffic events.

3.1 Architecture overview

Figure 1 shows an architecture overview our system and how it works in practice. The proposed architecture is based on Hadoop IBM BigInsights [1]. The first layer is a stream layer, it contains all Stream Process Application (SPA) that allow to manage data in motion. The processing of traffic stream is composed of three phases. The first phase entails real-time traffic data processing. This involves getting data from embedded devices in automobiles and processing it. The data is consolidated and combined with data from road sensors in the second phase, allowing it to react quickly to incidents that disrupt traffic conditions. Vehicle location, incident detection, congestion or traffic jam detection, and vehicle delay are all activities conducted during this phase. Finally, the distributed file system is used to store the streaming data.

The second layer concerns the data storage, which incorporates all Hadoop components that allow you to work with information stored in a distributed file system to create ad hoc analysis requests and prediction models across time intervals using analytic tools.

The third layer is based on a standard analytical model similar to an n-tier architecture. The analytical and predictive treatments are performed through BigInsights analytical tools (SPSS and RStudio). In the first step of this treatment, we acquire real-time data and combine them with static data. This step consists to extract, transform and filter traffic dataset (pre-processing). The second step concerns the statistical analysis to predict traffic conditions. Statistical models combined with the stream analytic process allow the real-time prediction of traffic. The final step provides visualization of KPI (Key Performance Indicator), dashboard and map view of analytical results. The end-user layer ensures presentation of related traffic

[1]http://www-03.ibm.com/software/products/en/ibm-biginsights-for-apache-hadoop

information. It is the end-users layer that allows to see traffic information via web console and smartphone equipped with GPS sensor [19] .

Figure 1 Achitecture of our intelligent transportation system

3.2 Stream layer

Large data streams created in real-time are called data in motion, this provides for fast response to congested roads events as well as continuous communication with end-users. We've created streams programs that evaluate streaming data in real-time. Several data sources feed information flows; in order to improve the design and execution of the streams process, we categorize the data sources into two groups: 1) road sensors, and 2) tramway checkpoints, as well as data from embedding GPS in buses. These smart objects also provide information on the mobility of passengers such as entrances/exits and peak hours.

The general framework of the traffic stream graph is depicted in Figure 2, it consists of a collection of data sources linked to operators. Large amounts of unstructured data are generated by a variety of sensors are supported by IBM Stream.

A common set of communication functions is used to access data from the sensors (socket). Many adapters are available in InfoSphere Streams for gathering information via data transfer protocols. The first step is to figure out which communication protocol various sensors employ. Other types of exchange protocols can also be added if the platform does not support them. The second stage is to conduct real-time traffic data analysis. To do this, InfoSphere Streams includes a variety of operators such as transformations, correlate, filter, annotate, aggregate, and so on. Streams link all of the operators, each operator connects with the others immediately through their input and output ports. The stream operator ends with the detection of abnormal events from all traffic data from the cluster as a whole.

Our stream layer is fully compatible and can be used for the design of stream process on the Apache Flume NG[2]. The operating mode of this platform is based on stream agents (Source, Memory Channel and Sink).

Two types of events are captured from traffic data: 1) Incident detection: detect all vehicles with abnormal behavior (vehicle with speeding and accident between two or more vehicles); 2) Traffic congestion: characterized by slow traffic speeds and increased queue of vehicles. The operators used in the stream layer of our transportation system are:

1. **FileSource**: The FileSource operator is used to read data from file and generates tuples as a result.
2. **TCPSource**: The TCPSource operator is used for reading data from a TCP socket and generates tuples as a result.
3. **UDPSource**: The UDPSource operator is used for reading data from a UDP socket and generates tuples as a result.
4. **FileSink**: The FileSink operator is used for writing output of tuples into local file or the distributed file system.

[2]https://flume.apache.org/

5 **Filter**: The Filter operator is used for filtering tuples based on specific criterias.
6 **Custom**: The Custom operator is used to define specific functions and logic clauses.

Figures

Figure 2 Overall view of traffic stream pipeline

Figure 3 shows the stream pipeline of fraud and accident detections. It consists of FileSource and UDP/TCP Source, FileSink, Filter and Custom operators. The filter operator filters all vehicle streams based on their speed while the custom operator implements **Incident_Detection** function. The principle of fraud detection consists of detecting all vehicles whose speed exceeds the speed limit. In case of collision between vehicles, we assume that the vehicles are stopped on the roadway. Based on this assumption, we identify all vehicles with zero speed. The final step is to determine the crash location, it is made possible by getting all GPS positions (latitude and longitude) of the damaged vehicles.

Figure 4 shows the stream pipeline of congestion detection. It consists of FileSource, FileSink, Aggregate and Custom operators. The Aggregate operator allows to merge and count all vehicles by roadway. The custom operator implements a **Congestion_Detection** function. The custom function takes as input the density, critical density and jam density. The density represents the number of vehicles on roadway length at a given time.

Figures

Figure 3 Incident detection stream pipeline

The critical density is the highest density that can be supported under free flow. Jam density refers to extreme traffic density when traffic flow stops completely. In case of congestion, the density is between the critical density and jam density while in the case of traffic jam, the density is greater than the jam density.

Figures

Figure 4 Congestion detection stream pipeline

3.3 Storage layer

The traffic data flow gathered in real-time is stored in the storage layer, and it includes road weather conditions, real-time weather information, vehicle tracking, passenger behavior, and associated occurrences. Traditional data storage methods and relational databases aren't designed to handle massive amounts of unstructured data and can't guarantee data scalability.

We used Hadoop framework [3] [20, 21] to manage enormous amounts of scalable and distributed traffic data. It allows our transportation system to handle petabytes of unstructured data distributed over thousands of nodes. Each node is equipped with a set of core processors and disks [22, 21]. The cluster is made up of a network of racks that are connected by a master/slaves topology and interact using a Hadoop-specific block protocol [23]. Each rack contains multiple nodes, see Figure 5(a). The master node plays the role of NameNode and JobTracker, it manages the HDFS and coordinates the execution of MapReduce programs on the slave nodes. The slave nodes play the role of DataNode and TaskTracker, they only store the datasets and execute MapReduce programs.

Our system is built to withstand hardware failures and is built around two major components:

1 The Hadoop Distributed File System (HDFS), which stores traffic data.

2 The MapReduce engine, which processes data across the whole cluster.

By duplicating data throughout the whole cluster, HDFS achieves dependability by replicating data across several nodes and assuming nodes may fail. [21, 24], see Figure 5 (b). To avoid hardware failures, we split traffic data into block files of 128MB and each file is stored on a DataNode. To prevent the system from failing, we configure three replicas of each block file by default. Hadoop transmits the MapReduce algorithm to each node to process the traffic data that it can access in the HDFS. This enables the cluster to process data more quickly and effectively than a traditional supercomputer design based on a distributed file system with data computations dispersed over high-speed networks using Gigabit Ethernet connections [25].

Figures

Figure 5 Multi-nodes cluster of our transportation system

4 Experiments

The findings of experimental tests conducted on the city of Casablanca are presented in this section. This work was made possible through dataset produced by embedded systems in buses, tramways and utility vehicles. We evaluate the performance of our transportation system on its ability to widely cover information from GPS sensors. The objective is to show that our system is able to instantly process data from 944 GPS access points (866 GPS access points for the buses and 78 GPS access points for the tramways).

4.1 Dataset

The traffic data are varied, unstructured and come in different formats. We have collected two categories of data: static and stream data. We used OpenStreetMap data from Casablanca, public transportation lines (buses and tramways), GPS devices placed in public transportation vehicles, historical vehicle tour schedules, passenger mobility, and real-time data from traffic control station.

[3] http://hadoop.apache.org/

Bus data is centralized in four zones, with 70 lines spanning 1250 kilometers and many recordings of the behavior of 21000 children transported daily. The tramway's data includes a 31-kilometer fork-shaped line with 50 stations and several records detailing the daily behavior of 120 000 people. We used streams data from road construction and maintenance activities, as well as planned events with a small or large audience and vehicle tour schedules; this activity is usually completed weekly.

4.2 Test Environment

For experimental tests, the stream application was deployed on a 5-nodes cluster consisting of 1 master and 4 slaves. Each node is equipped with a processor Intel(R) Core i5-2410M CPU 2.30GHz (4 CPUs) with 8GB RAM running on Red Hat Enterprise Linux6 64 OS.

4.3 Urban mobility analysis

Every day, about 10 million transportation moves are made in Casablanca city. The average waiting time at peak hours are increasingly longer, Figure 6. The road network configuration of the city is the major source congestion in some districts, the traffic utility vehicles may suffer significant downturn. Figure 7 (a) shows the daily traffic loads by district based on vehicle counting by using inductive loop. We are witnessing a significant flow of vehicles in the southwest, east and downtown. As a result, this causes main points of congestion and traffic jams. Figure 7(b) shows the average traffic speed by district at peak hours. We remark that, areas with high mobility are more affected by downturns creating long time of traffic congestion.

Figures

Figure 6 Average passenger waiting time at peak hours

Figures

Figure 7 Traffic flow of Casablanca city: (a) Daily traffic loads and mobility flow of different districts; (b) Average speed by district at peak hours

4.4 Analysis of number of stream tuples

All embedded systems communicate directly with our system via TCP/IP and UDP/IP protocols. Each embedded system sends information about vehicle locations, speed, stop and entries-exit of passengers. Figure 8 shows the average number of GPS access point processed per second. Indeed, the process streams are distributed across all nodes and run simultaneously, this is called parallel computing. In our test the distribution of stream process is automatically managed by the Stream Scheduler of InfoSphere Streams. Each node can manage a set of stream data from multiple GPS access points. Each GPS access point correspond to a tuple and it is managed by the source operators. Note also that the throughput processed by the system is influenced by node configurations (CPU and RAM memory).

Figure 8 Number of GPS access points processed per second in the multi-nodes cluster

Figures

4.5 Analysis of traffic events

The system presents better performance, it consumes few resources of ram and every second, 2 million GPS access points are processed. The experimental values of frauds and incident detections from streaming data are shown in Table 1. We report all captured events in the table with the total number of frauds, number of detected frauds, total number of collision and number of detected collisions. We report the relative gap between frauds and captured frauds and the relative gap between collisions and captured collisions.

Based on the results provided by column 4, it is possible to deduce that the proposed fraud detection technique provides the best results and captures all fraud events (% fraud = 100%). In case of collision events (column 7), we remark that the quality of results decreases with the increasing number of vehicle stream. This is caused by a large number of vehicles with abnormal behavior which stop temporarily along the roadsides. In this case, it is difficult to deduce if it is an accident or a temporary stop. But on a highway, we can conclude that it is an accident or a breakdown.

5 Conclusion and further work

In this paper, we proposed an innovative architecture for an intelligent transportation system based on the Hadoop big data framework. We proved the system's ability to handle enormous amounts of traffic data while maintaining fault tolerance and providing real-time traffic flow monitoring. It was made possible by using InfoSphere BigInsights and our system is well-equipped to meet scalability and adaptation difficulties. We also showed how the system can gather and manage traffic data from an increasing number of GPS access points with different transfer protocols. We used the stream process for real-time detection of frauds, accidents and congestion events from traffic flow. Our system satisfies the 4V criteria of big data and face the challenges of storing large size and diverse variety of traffic data and support the scalability of traffic data. In term of velocity and veracity, the response time and relevance of information made available to end-users are instantly with a high quality of precision. For further work, we will: (1) provide a path planning algorithm based on A* search [26] and combining different variants such users preferences, weather and traffic conditions; (2) analyze traffic flow and suggest efficient solutions to urban transportation planning agencies and (3) purpose a stream process application which can relate all vehicle incidents to emergency services.

6 Declarations

Acknowledgements
Not applicable

Funding
Not applicable

Table 1 Experimental results of event detections.

Stream	Fraud event			Collision event		
#Tuples	Total	Detected	Ratio	Total	Detected	Ratio
$10x10^3$	63	63	100%	12	12	100%
$50x10^3$	96	96	100%	29	29	100%
$10x10^4$	188	188	100%	34	34	100%
$50x10^4$	277	277	100%	45	48	106.6%
$1x10^6$	427	427	100%	53	58	109.4%
$5x10^6$	524	524	100%	74	80	108.1%
$10x10^6$	748	748	100%	86	92	106.9%
$25x10^6$	1478	1478	100%	128	136	106.2%
$50x10^6$	3472	3472	100%	222	253	113.9%
$75x10^6$	4128	4128	100%	289	312	107.95%
$100x10^6$	6174	6174	100%	417	448	107.43%

Abbreviations

GPS: Global Positioning System; TSP: Traveling Salesman Problem; MRA: MapReduce-A*; 4V: Volume,Velocity,Variety and Veracity; SPM: Stream Process Application; KPI: Key Performance Indicator; HDFS:Hadoop Distributed File System;

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Ethics approval and consent to participate

Not applicable

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

Dr Adoni, Dr Tarik and Dr Najib have designed and developed the proposed system. In addition, Dr Adoni has conducted the literature review, Dr. Tarik has collected the data, and Dr Najib has prepared the figures. Dr. Moez and Dr. Mohammed have analyzed the results and wrote the main manuscript text. All authors have reviewed the manuscript and approved the final version.

Authors' information

Wilfried Yves Hamilton Adoni obtained his PhD in Computer Science and Applied Mathematics in 2020. He is currently an Assistant Professor at the International University of Casablanca and a member of the Research Lab. on Computer Science and Modeling of Complex Systems - LIMSAD (Morocco). He is the Vice-President of Ivorian Society of Operations Research and scientific partner at the African Institute of Mathematical Sciences. He is reviewer in many scientific journals. He has many publications related to big data and distributed systems. His main Research Interest are Applied Mathematics, Big Data, Artificial Intelligence, Operations Research & Graph Theory, Formal Verification, Parallel & Distributed Systems. Moreover, he works on applying his research work to several Trend Technologies like Smart Agriculture, Smart Village, Smart Cities, Computer Vision, Healthcare, Blockchain, etc.

Tarik Nahhal obtained his HDR (Ability to Conduct Researches) in Computer Science from the University of Hassan II University of Casablanca (Morocco) in 2014. He obtained his PhD in Computer Science in 2007. He is currently a Full Professor at the Hassan II University of Casablanca and a member of the Research Lab. on Computer Science and Modeling of Complex Systems - LIMSAD (Morocco). He has many publications in the field of Computer Science. His main Research Interest includes Big Data, Artificial Intelligence, Parallelism & Distributed Systems. Moreover, he works on applying Formal Machine Learning & Deep Learning algorithms to improve Model-Based Testing Methodologies.

Najib Ben Aoun obtained the MSc and the Ph.D degrees in Computer Systems Engineering from the National Engineering School of Sfax (ENIS), Tunisia, in 2008 and 2014, respectively. He is currently an assistant professor at the College of Computer Science and Information Technology (CCS&IT) of Al-Baha University, Saudi Arabia. He is a member of the REsearch Groups in Intelligent Machines (REGIM-Lab). His main research interests include computer vision and data science, in particular to image/objects classification, video event/action recognition and detection, pattern recognition, biometrics, graph-based modeling and deep learning.

Moez Krichen obtained his HDR (Ability to Conduct Researches) in Computer Science from the University of Sfax (Sfax, Tunisia) in 2018. He obtained his PhD in Computer Science in 2007. He is currently an Associate Professor at the University of Al-Baha (KSA) and a member of the Research Laboratory on Development and Control of Distributed Applications - REDCAD (Tunisia). His main Research Interest is Model-Based Testing Methodologies for Real-Time & Distributed Systems. Moreover, he works on applying Formal Methods to several Modern Technologies like Smart Cities, Smart Vehicles, Healthcare, etc.

Mohammed Y. Alzahrani has received his B.Sc. In Computer Engineering from Albaha Private College of Science and M.Sc. in Information Technology from Heriot Watt University, Edinburgh, UK. After this he obtained Ph. D.

degree in Computer Science from Heriot Watt University, Edinburgh, UK. Currently he is the dean of College of Computer Science and Information Technology at Albaha University, Al Baha, Saudi Arabia. His research interest includes Model Checking, Data Mining and Cyber Security.

Author details
[1]Engineering School, International University of Casablanca, Casablanca, Morocco. [2]University of Hassan II, Casablanca, Morocco. [3]FDMS Research unit, Hassan II University of Casablanca, Casablanca, Morocco. [4]Al Baha University, Al Baha, Saudi Arabia. [5] REGIM-Lab: Research Groups in Intelligent Machines, University of Sfax, Sfax, Tunisia. [6]ReDCAD Laboratory, University of Sfax, Sfax, Tunisia.

References

1. Jabbar, R., Fetais, N., Kharbeche, M., Krichen, M., Barkaoui, K., Shinoy, M.: Blockchain for the internet of vehicles: How to use blockchain to secure vehicle-to-everything (v2x) communication and payment? IEEE Sensors Journal (2021)
2. Jabbar, R., Kharbeche, M., Al-Khalifa, K., Krichen, M., Barkaoui, K.: Blockchain for the internet of vehicles: A decentralized iot solution for vehicles communication using ethereum. Sensors **20**(14), 3928 (2020)
3. Abbas, A., Krichen, M., Alroobaea, R., Malebary, S., Tariq, U., Piran, M.J.: An opportunistic data dissemination for autonomous vehicles communication. Soft Computing, 1–14 (2021)
4. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.: Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute
5. Zheng, Y., Liu, F., Hsieh, H.-P.: U-air: When urban air quality inference meets big data. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '13, pp. 1436–1444. ACM, New York, NY, USA (2013). doi:10.1145/2487575.2488188
6. Adoni, Y.H. Wilfried, Nahhal, T., Aghezzaf, B., Elbyed, A.: The mapreduce-based approach to improve vehicle controls on big traffic events. In: 2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), pp. 1–6 (2017)
7. Adoni, Y.H. Wilfried, Nahhal, T., Aghezzaf, B., Elbyed, A.: Mra*: Parallel and distributed path in large-scale graph using mapreduce-a* based approach. In: Sabir Essaid, G.M. García Armada Ana, Debbah, M. (eds.) Ubiquitous Networking. Springer, Cham (2017)
8. Adoni, Y.H. Wilfried, Nahhal, T., Aghezzaf, B., Elbyed, A.: The mapreduce-based approach to improve the shortest path computation in large-scale road networks: the case of a* algorithm. Journal of Big Data **5** (2018). doi:110.1186/s40537-018-0125-8
9. Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., Koutsopoulos, H., Moran, C.: Ibm infosphere streams for scalable, real-time, intelligent transportation services. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. SIGMOD '10, pp. 1093–1104. ACM, New York, NY, USA (2010). doi:10.1145/1807167.1807291
10. Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., Koutsopoulos, H.N., Rahmani, M., Güç, B.: Real-time traffic information management using stream computing. IEEE Data Eng. Bull. **33**(2), 64–68 (2010). doi:10.1.1.165.2473
11. Bouillet, E., Ranganathan, A.: Scalable, real-time map-matching using ibm's system s. In: 2010 Eleventh International Conference on Mobile Data Management (MDM), pp. 249–257 (2010). doi:10.1109/MDM.2010.36
12. Kitchin, R.: The real-time city? big data and smart urbanism. GeoJournal **79**(1), 1–14 (2014). doi:10.1007/s10708-013-9516-8
13. Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., Li, Y.: Robust causal dependence mining in big data network and its application to traffic flow predictions. Transportation Research Part C: Emerging Technologies **58**, 292–307. doi:10.1016/j.trc.2015.03.003
14. Hao, J., Zhu, J., Zhong, R.: The rise of big data on urban studies and planning practices in china: Review and open research issues. Journal of Urban Management **4**(2), 92–124 (2015). doi:10.1016/j.jum.2015.11.002
15. Toole, J.L., Colak, S., Sturt, B., Alexander, L.P., Evsukoff, A., González, M.C.: The path most traveled: Travel demand estimation using big data resources. Transportation Research Part C: Emerging Technologies **58**, 162–177. doi:10.1016/j.trc.2015.04.022
16. Dong, H., Wu, M., Ding, X., Chu, L., Jia, L., Qin, Y., Zhou, X.: Traffic zone division based on big data from mobile phone base stations. Transportation Research Part C: Emerging Technologies **58**, 278–291. doi:10.1016/j.trc.2015.06.007
17. Dodge, M., Kitchin, R.: The automatic management of drivers and driving spaces. Geoforum **38**(2), 264–275 (2007). doi:10.1016/j.geoforum.2006.08.004
18. Lécué, F., Tallevi-Diotallevi, S., Hayes, J., Tucker, R., Bicer, V., Sbodio, M., Tommasi, P.: Smart traffic analytics in the semantic web with star-city: scenarios, system and lessons learned in dublin city. Web Semantics: Science, Services and Agents on the World Wide Web **27–28**, 26–33 (2014). doi:10.1016/j.websem.2014.07.002
19. Krichen, M.: Anomalies detection through smartphone sensors: A review. IEEE Sensors Journal (2021)
20. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters **51**(1), 107–113. doi:10.1145/1327452.1327492
21. Ghemawat, S., Gobioff, H., Leung, S.-T.: The google file system. In: ACM SIGOPS Operating Systems Review, vol. 37, pp. 29–43. ACM, New York, NY, USA. doi:10.1145/1165389.945450
22. Arpaci-Dusseau, R.H., Anderson, E., Treuhaft, N., Culler, D.E., Hellerstein, J.M., Patterson, D., Yelick, K.: Cluster i/o with river: Making the fast case common. In: Proceedings of the Sixth Workshop on I/O in Parallel and Distributed Systems. IOPADS '99, pp. 10–22. ACM. doi:10.1145/301816.301823
23. Fox, A., Gribble, S.D., Chawathe, Y., Brewer, E.A., Gauthier, P.: Cluster-based scalable network services. In: Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles. SOSP '97, pp. 78–91. ACM. doi:10.1145/268998.266662

24. Liskov, B., Ghemawat, S., Gruber, R., Johnson, P., Shrira, L., Williams, M.: Replication in the harp file system. In: Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles. SOSP '91, pp. 226–238. ACM. doi:10.1145/121132.121169
25. Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., West, M.J.: Scale and performance in a distributed file system **6**(1), 51–81. doi:10.1145/35037.35059
26. Bell, M.G.H.: Hyperstar: A multi-path astar algorithm for risk averse vehicle navigation. Transportation Research Part B: Methodological **43**(1), 97–107. doi:10.1016/j.trb.2008.05.010

Figures

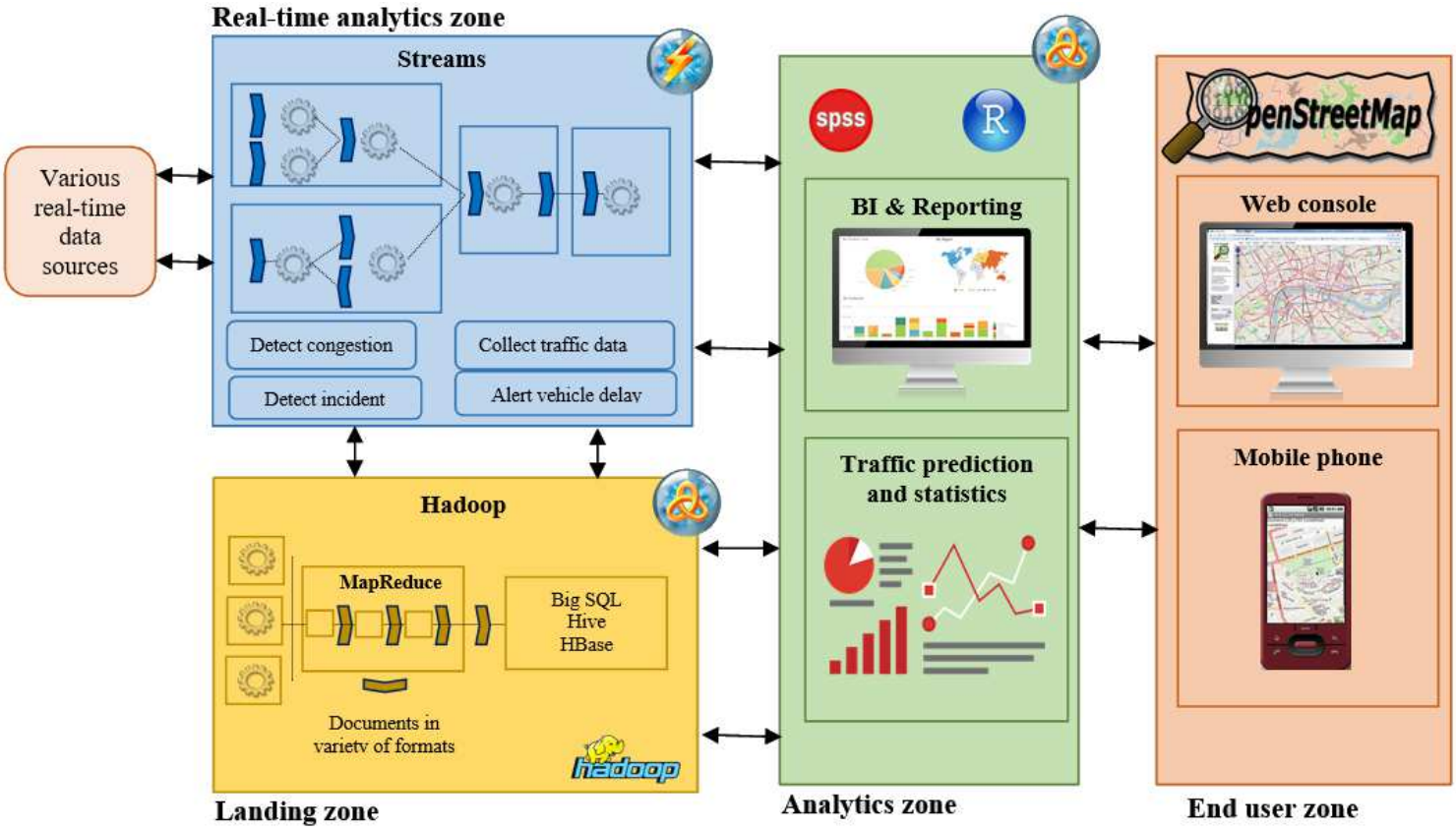

Figure 1

Achitecture of our intelligent transportation system

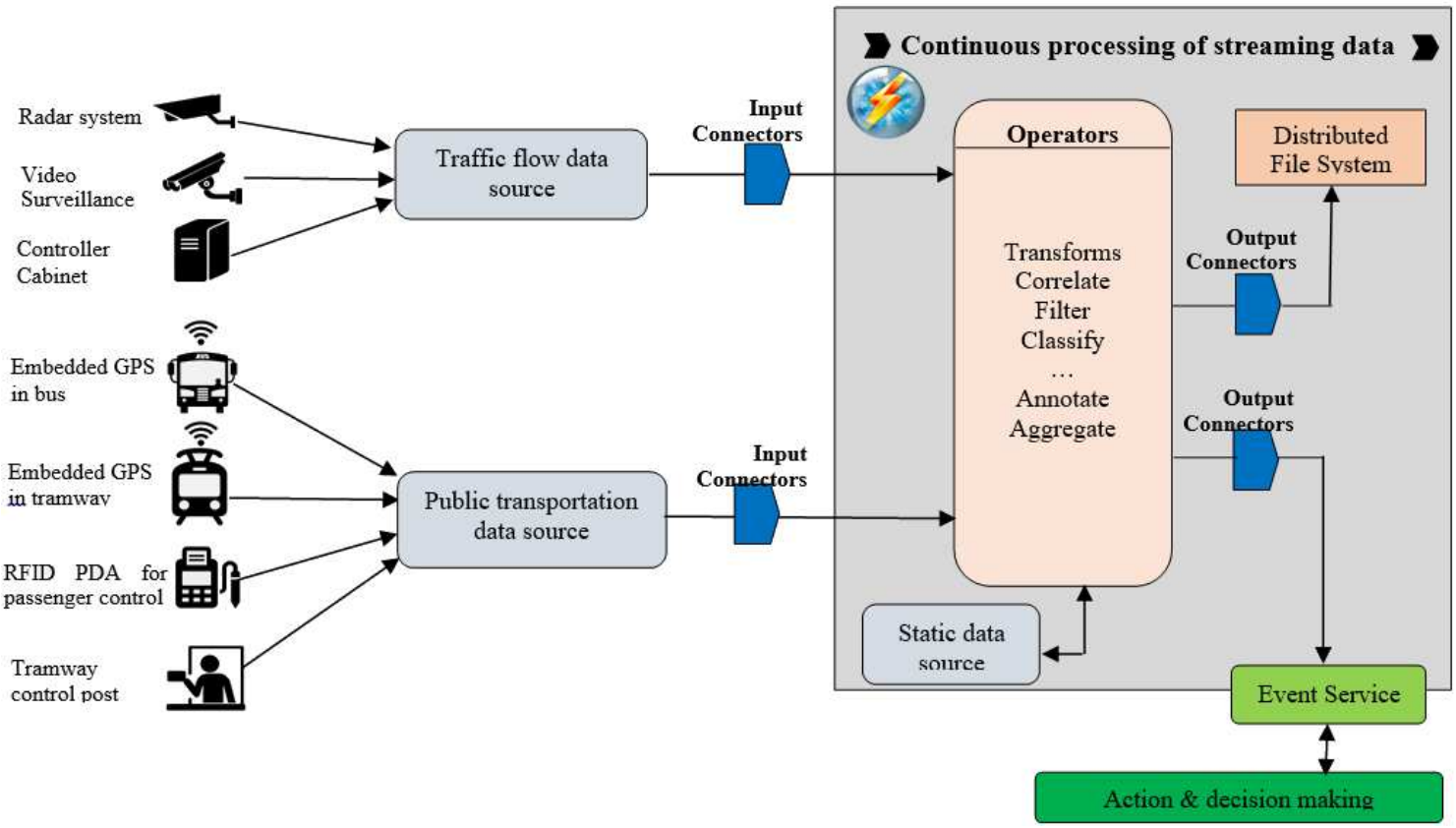

Figure 2

Overall view of traffic stream pipeline

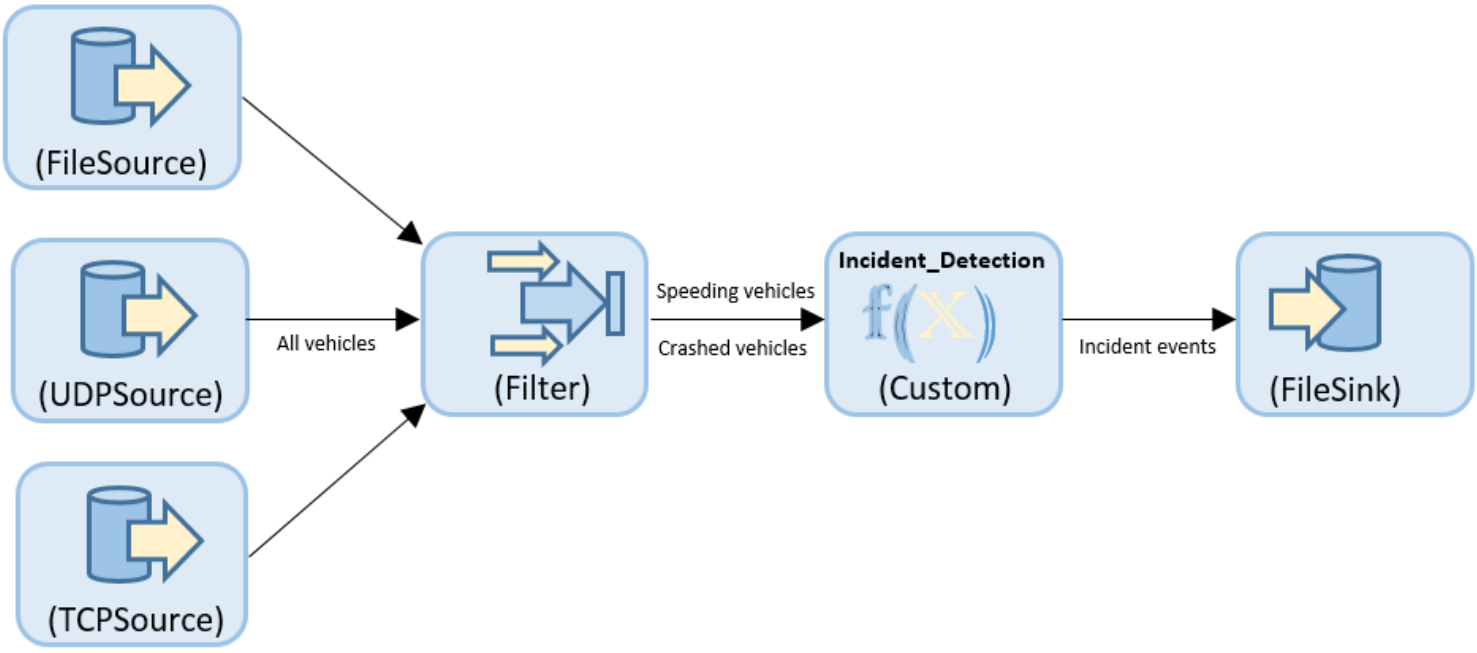

Figure 3

Incident detection stream pipeline

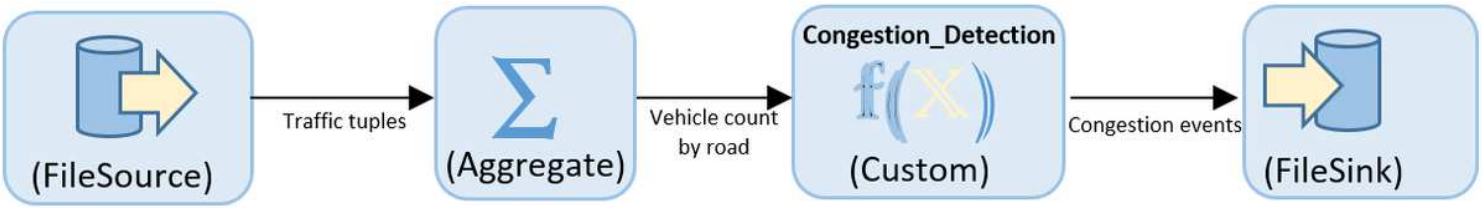

Figure 4

Congestion detection stream pipeline

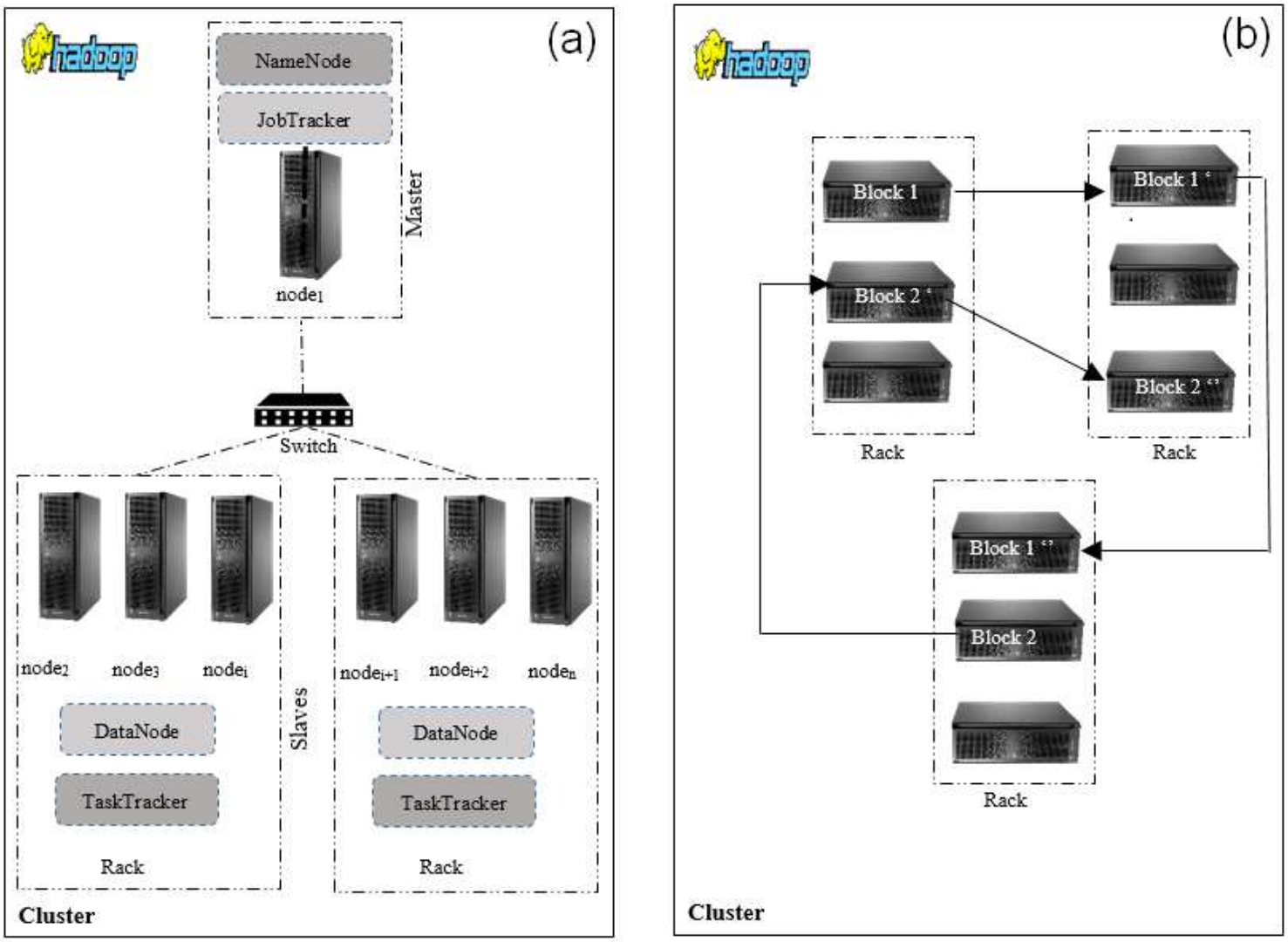

Figure 5

Multi-nodes cluster of our transportation system

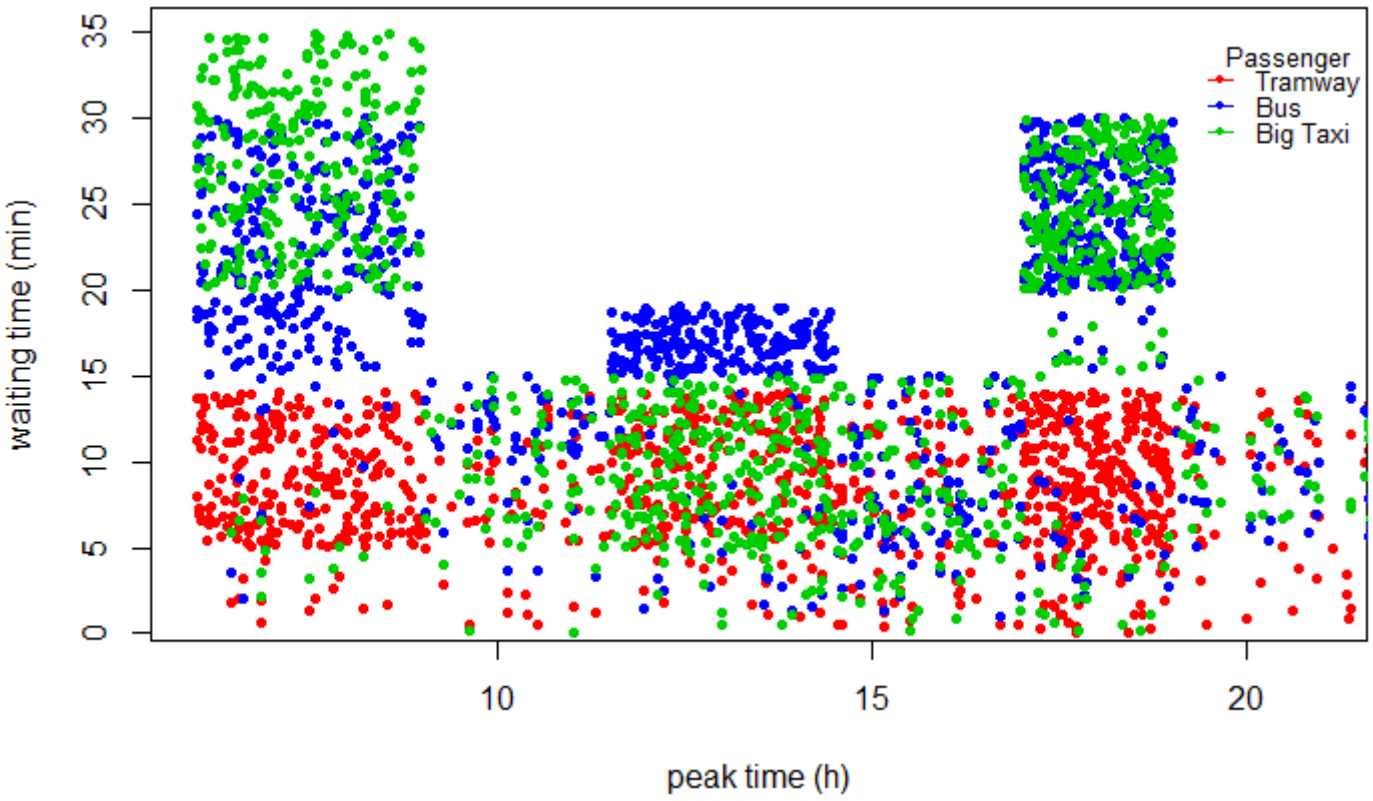

Figure 6

Average passenger waiting time at peak hours

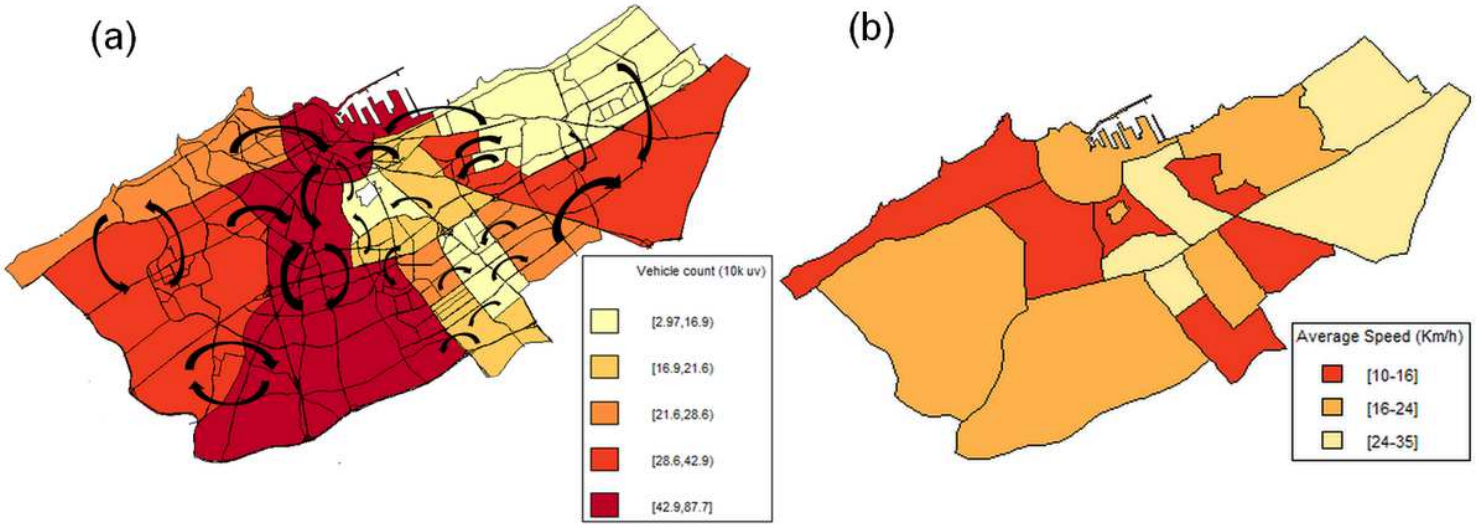

Figure 7

Traffic flow of Casablanca city: (a) Daily Traffic loads and mobility flow of different districts; (b) Average speed by district at peak hours

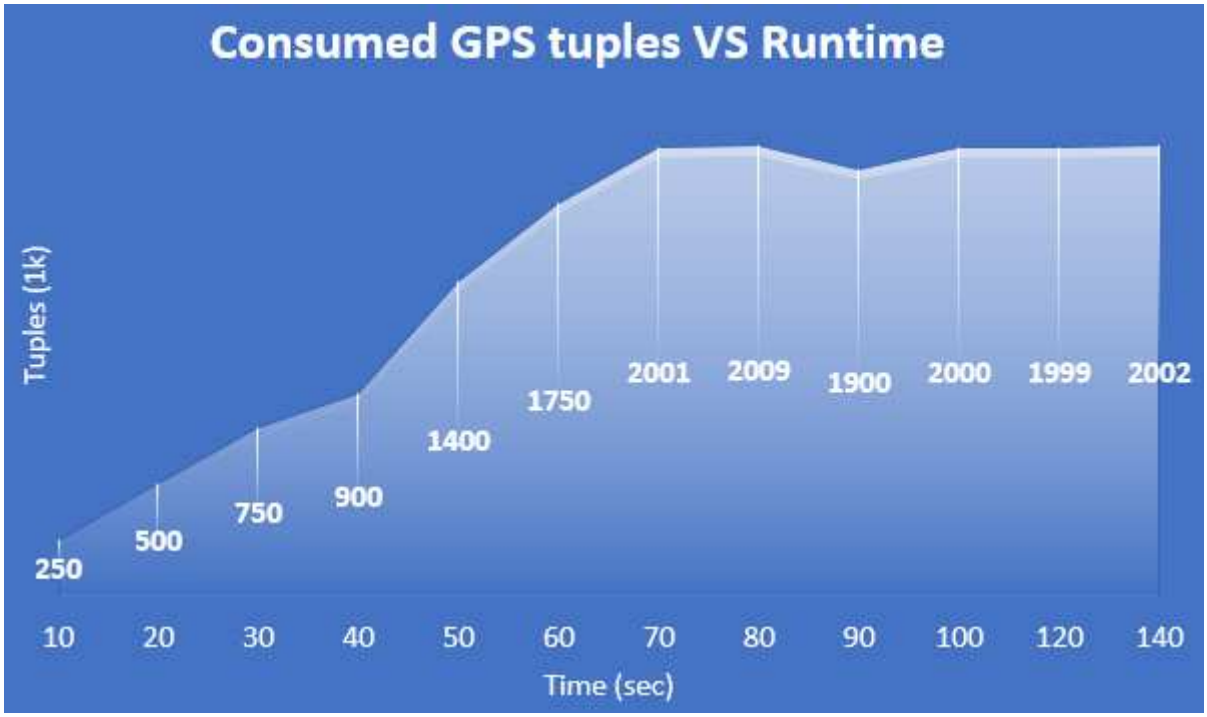

Figure 8

Number of GPS access points processed per second in the multi-nodes cluster