TOWARDS A CLOUD BASED SMART TRAFFIC MANAGEMENT FRAMEWORK

M. M. Rahimi [a], F. Hakimpour [b]*

[a] MSc student of GIS, Department of Surveying and Geomatics Engineering, College of Engineering, University of Tehran
rahimi.masoud@ut.ac.ir
[b] Assistant Professor, Department of Surveying and Geomatics Engineering, College of Engineering, University of Tehran
fhakimpour@ut.ac.ir

KEY WORDS: Traffic Management, Big Data, Cloud Computing, Hadoop

ABSTRACT:

Traffic big data has brought many opportunities for traffic management applications. However several challenges like heterogeneity, storage, management, processing and analysis of traffic big data may hinder their efficient and real-time applications. All these challenges call for well-adapted distributed framework for smart traffic management that can efficiently handle big traffic data integration, indexing, query processing, mining and analysis. In this paper, we present a novel, distributed, scalable and efficient framework for traffic management applications. The proposed cloud computing based framework can answer technical challenges for efficient and real-time storage, management, process and analyse of traffic big data. For evaluation of the framework, we have used OpenStreetMap (OSM) real trajectories and road network on a distributed environment. Our evaluation results indicate that speed of data importing to this framework exceeds 8000 records per second when the size of datasets is near to 5 million. We also evaluate performance of data retrieval in our proposed framework. The data retrieval speed exceeds 15000 records per second when the size of datasets is near to 5 million. We have also evaluated scalability and performance of our proposed framework using parallelisation of a critical pre-analysis in transportation applications. The results show that proposed framework achieves considerable performance and efficiency in traffic management applications.

1. INTRODUCTION

Nowadays, one of the most important challenges in transportation systems is traffic congestion. According to recent statistics, transportation has the second place in greenhouse gas emission factors ranking of USA (STATISTICS, 2015). In 2014, traffic congestion costs 6.9 billion hours of citizenry and 3.1 billion gallons of fuel, 160 billion dollars loss to US economy (STATISTICS, 2015). Infrastructure improvement is an expensive solution to traffic congestion challenge.

A Traffic Management System (TMS) as one of the most important components of Intelligent Transportation System (ITS) offers capabilities that can potentially be used to reduce road traffic congestion, improve response time to incidents and ensure better travel experience for commuters.

Some of the most important services of TMS are vehicle routing to shorten commuter journey, traffic prediction that enables early detection of bottlenecks, parking management that ensure optimal usage of parking spots and interact with routing and prediction services for improved control of traffic flow and finally infotainment services that provide useful information for both drivers and passengers (Djahel et al., 2015).

In recent years, researchers have shown great interest in using advances in wireless sensing and communication technologies along with novel techniques and methodologies in TMSs to make them more efficient. With the emerging of new sources of traffic data like Wireless Sensor Networks (WSNs), Machine to Machine communication (M2M), Floating Car Data (FCD), mobile sensing and social media new opportunities for different transportation applications has been created. These big data can be used for real-time road management and decision-making, data mining and knowledge extraction. In order to exploit this big data potential in TMS applications, traditional frameworks face some technical challenges like:

- Data heterogeneity: these data are from different heterogeneous sources in different structures.
- Data management and storage: when a dataset outgrows the storage capacity of a single physical machine, it is necessary to partition it across multiple separate machines.
- Data processing and analysis: with this tremendous valuable big data, real-time query processing, analysis and data mining in traditional frameworks is a time-consuming inefficient task.

All these problems call for well-adapted distributed framework for TMS that can efficiently handle big traffic data integration, indexing, query processing, mining and analysis.

Cloud computing is a useful, scalable and cost-effective solution to answer traffic big data challenges in TMS. Hadoop (Apache) as a popular cloud computing framework on commodity hardware provides high availability and scalability along with fault tolerance for real-time traffic management applications.

In this paper, we attempt to facilitate storage, management, processing and analysis of traffic heterogeneous big data by using a state-of-the-art distributed framework for smart traffic management system. The main requirements of the proposed framework are:

- Low-latency data storage and access: the framework should contain an efficient and flexible tool for data gathering and management for massive traffic data with high performance and low latency.

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W4-447-2017 | © Authors 2017. CC BY 4.0 License.

- Fast data processing, analysis and data mining: the framework should contain fast powerful calculation engines to support different type of query processing, traffic modeling, analysis and data mining.
- Easy Implementation for different applications: the programming paradigm for implementation of different calculation models should be easy for implementation and development.
- Fault Tolerance, Elasticity and High Availability: consistency of increasing size of traffic data, adapting to unpredictable changes and high uptime to support real-time applications along with fault tolerance is some of the most important features in a distributed system.

This research is a major step to development of an efficient and real-time traffic management system. The rest of the paper is organized as follows: First we review major related works to our framework, then we will propose our framework architecture and finally, we will evaluate the framework performance and scalability.

2. RELATED WORKS

Hadoop is a popular open-source software framework for cloud computing which consists of two main modules. Hadoop Distributed File System (HDFS) and a parallel processing framework named MapReduce. HDFS is a distributed file system. In HDFS, every single file partitions across several separate machines including a master node and some slave nodes. The master node should store file system metadata while the data stores on slave nodes. MapReduce is a parallel programming model for large-scale data processing. MapReduce can be written in various programming languages like Java, Ruby and Python. Based on HDFS, HBase (Apache) is developed as a No-SQL distributed database. In HBase real-time and random big data read and write is done in a fast and easy manner. Hive (Hadoop; Thusoo et al., 2009) is a data warehouse software which facilitates reading, writing, and managing large datasets residing in distributed storage using SQL commands. Pig (Hadoop) is a platform for large-scale datasets analysis programs which contains a high level programming language “PigLatin” along an infrastructure for evaluating these programs. Mahout (Hadoop) is a distributed machine learning and data mining tool. The aim of mahout is build an environment for quickly creating scalable performant machine learning applications. Oozie (Hortonworks) is a workflow scheduler. Oozie is a web-based java program that uses for scheduling apache Hadoop jobs. Flume is distributed, reliable and available service for efficiently collecting, aggregating, and moving large amounts of streaming event data. Sqoop (Apache) is an open source tool used for integration between Hadoop framework and structured relational databases (e.g. Oracle).

Some of the most critical challenges of using distributed systems in spatial applications are storing, management and processing of spatial big data. Hadoop-GIS (Aji et al., 2013) is high-performance scalable system for spatial big data storage on HDFS which supports spatiotemporal queries using SQL like commands. Similarly, SpatialHadoop (Eldawy and Mokbel, 2015) is developed as an efficient framework for spatial queries which employs a two level spatial index structure and some basic spatial functionalities.

In recent years, several efforts have been made on development of an efficient traffic management system in both academic and commercial research societies. In commercial society, *Trafficware* in Houston government and *TransCore* in Washington D.C. are trying to provide an advanced system for enabling the Traffic management personnel to manage the congestion more effectively. Besides, as a public traffic management platform, *Waze* is an android-based application that uses crowd sourcing data and methods. The users have the app running in the background while travelling to their destination, thereby passively contributing traffic data and other incident data. In academic society, there are some studies on traffic management in literature. Xiao et al. (Xiao et al., 2015) proposed a spatial data oriented platform “DriveNet” for freeways performance analysis. The aim of DriveNet is traffic data integration, sharing, analysis and visualization. In this platform a central web server processes user’s requests using HTTP(s) protocol. Then the server obtained data from different sources and process and analyse them. This analysis is done using internal and external modules like R servers. The main drawback of this platform is using central server and lack of fault tolerance, elasticity and high availability. With the aim of finding patterns in traffic data, (Khazaei et al., 2015) developed a cloud based big data analytic platform. (Xiong et al., 2016) discussed different aspects in the design of ITS including future trends and current ITS development considerations. (Sekar et al., 2017) has developed a data mining framework for traffic congestion prediction and route planning using hybrid clustering techniques.

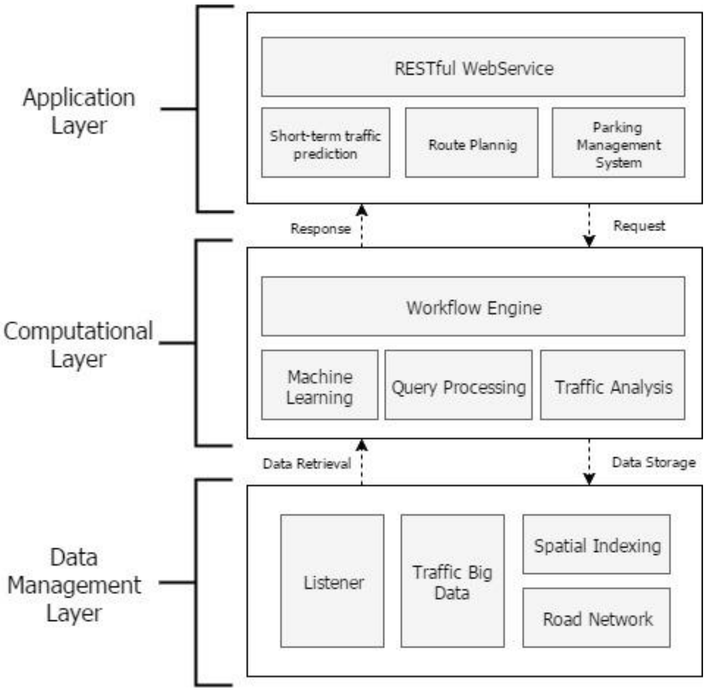
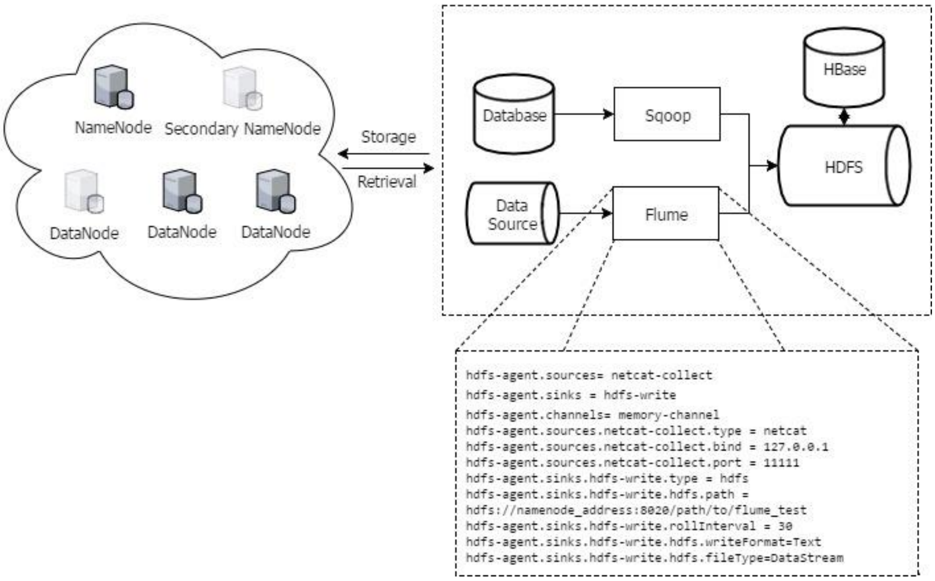


Figure 1. Proposed framework architecture

(Yu et al., 2013) has developed a traffic big data mining system based on Software as a Service (SaaS) architecture. It is observed that most of the efforts on development of an efficient smart traffic management system suffer from the lack of comprehensiveness.



```
hdfs-agent.sources= netcat-collect
hdfs-agent.sinks = hdfs-write
hdfs-agent.channels= memory-channel
hdfs-agent.sources.netcat-collect.type = netcat
hdfs-agent.sources.netcat-collect.bind = 127.0.0.1
hdfs-agent.sources.netcat-collect.port = 11111
hdfs-agent.sinks.hdfs-write.type = hdfs
hdfs-agent.sinks.hdfs-write.hdfs.path =
hdfs://namenode_address:8020/path/to/flume_test
hdfs-agent.sinks.hdfs-write.rollInterval = 30
hdfs-agent.sinks.hdfs-write.hdfs.writeFormat=Text
hdfs-agent.sinks.hdfs-write.hdfs.fileType=DataStream
```

Figure 2. Data management layer

An efficient smart traffic management system requires data integration, management, storage and analysis as long as data mining information and knowledge discovery techniques. Thus, the proposed framework is a new idea to fill in the gaps we had found these systems to make a comprehensive framework. In this paper, we focus more on data integration, data analysis in traffic management applications.

3. THE FRAMEWORK ARCHITECTURE

In this section, we have proposed a framework for traffic management system. The main goal of the framework is answering to traditional TMS drawbacks that has discussed earlier. The proposed framework is using Hadoop cloud computing technologies including HDFS, MapReduce paradigm and some of the other members of Hadoop ecosystem. As shown in Figure 1, the proposed framework consists of three layers architecture. Data from different sources imports to data management layer which low-latency data storage and access. Then the computational layer is used to provide different computation paradigms to ensure TMS applications requirements. The upper layer is application layer that provides system interaction with end users using web services or graphical interfaces.

3.1 Data Management Layer

Traffic Management needs applicable data about current traffic status and traffic flow characteristics. These heterogeneous data should be collected from different sources in different formats in TMS for further applications. As the result, the cornerstone of traffic management is data management layer. The data management layer aims to integrate all kinds of data into a distributed file system to fully exploit their potential. Data management layer should also interact with different databases to take advantage of their capabilities. Figure 2 represents data management layer architecture.

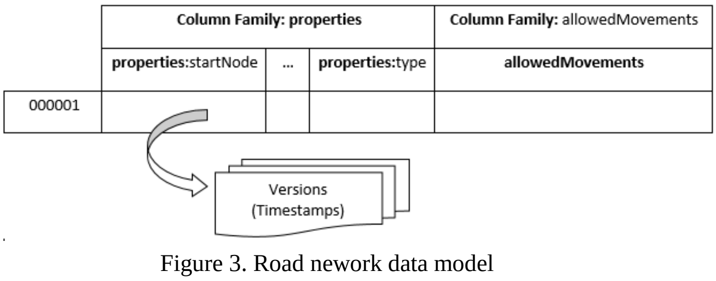
Generally, there are two kind of data source in the proposed framework. One of them are road networks which represents system functional area. Besides road networks, continuous stream of real-time traffic data for further processing is also another kind of available data in TMS which are collected from different sources and in different formats. Table 1 represents different traffic data types and their characteristics in TMS.

The generated Input data of TMS from different sources sends to a central server. Server by flume agent collects the data and stores them in HDFS. In the data manager layer Sqoop tool is used for transferring data between HDFS and structured relational databases that is a critical requirement in development of an adequate TMS.

One of the most important spatial data in TMS is roads network. A roads network is a directed graph G(V,E,M) where V is a set of vertices which are main points or intersection of roads, E is a set of edges which are vectors connecting two vertices and M is a set of allowed movements.

In a distributed framework, storage and indexing of roads network are important challenges. To answer these challenges and due to static characteristic of roads network, in the proposed framework we have used HBase database. HBase is a distributed, column-based, No-SQL database that uses HDFS as its underlying storage. HBase provides read/write access of individual rows and batch operations for reading and writing data in bulk. The storage unit in HBase is cell which defined uniquely by row-key, column family name, column name and version. By default, a cell version is a timestamp auto-assigned by HBase at the time of cell insertion. On a physical level, every column family data continuously writes on the disk and this data will be sorted by its row key, column name and version.

In this framework, we have stored roads network in two HBase tables.



	Column Family: properties			**Column Family:** allowedMovements
	properties:startNode	...	**properties**:type	**allowedMovements**
000001				

Figure 3. Road nework data model

In the first table, vertices and their properties are stored using “nodeID” as row key. In the second table, edges and properties are stored in different column families and an "edgeID" is used as row key. For each edge, a set of allowed movements is pre-computed and their IDs are stored separated by semicolons. Figure 3 represents road network data model.

Data Source Class	Data Source	Data Structure	Descriptions
Infrastructural Sensors	Pneumatic Road Tube	Structured	Volume/count, speed and vehicle classification
	Magnetic sensors	Structured	Volume/count, speed and vehicle classification
	Inductive Loops	Structured	Volume/count, speed, vehicle classification and occupancy
	Active infrared	Structured	Volume/count, speed and vehicle classification
	Passive infrared	Structured	Volume/count, speed, vehicle classification and occupancy
	Microwave radar	Structured	Volume/count, speed, vehicle classification and occupancy
	Ultrasonic	Structured	Volume/count
	Acoustic sensor	Structured	Volume/count, speed, vehicle classification and occupancy
	Bluetooth scanning	Structured	Travel time prediction
	V2X (V2V or V2I)	Structured	Travel time prediction
Floating Car Data	Vehicle-based	Structured	Vehicle location, speed and heading
	Mobile-based	Structured	Mobile location, speed and heading
Web and social networks	Web and social networks	Unstructured	Collective information from the public
Video surveillance	CCTV	Unstructured	BLOB videos
Media	Media	Semi-structured	Audio / Textual Reports

Table 1. Different traffic data and their properties in TMSs

In HBase, data is stored by key and is randomly accessible based on key. In addition, HBase does not have capabilities for secondary indexing. Therefore, with the aim of fast and efficient storage and retrieval of spatial road network, a secondary spatial index is needed. At present several spatial index algorithm has been proposed in literature (Finkel and Bentley, 1974; Guttman, 1984). However, integration between this methods and Hadoop environment faces several challenges. While Hadoop is using functional programming, all of these methods need procedural approach. Besides, in Hadoop once a file is written on HDFS, it cannot be modified later.

Considering this limitations, in this paper we adopt regular grid spatial indexing method for road network. We have divided the geographical region of map in k different regions. For each rectangular region, we find any edges in data table that are either connected to intersect the rectangle. We build another table in HBase as shown in Figure 4 that has a column family for edges. Each row in this table represents a region in the index. All "edgeID"s are separated by semicolons.

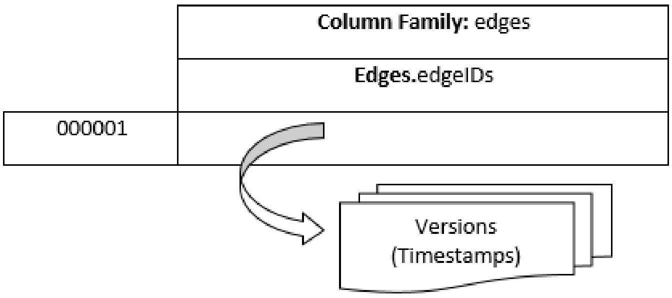


Figure 4. Spatial index data model

3.2 Computational Layer

Storage and management of various data in TMS is an important challenge in development of an adequate framework. However, this data should be used for extraction of valuable information and traffic parameters. An adequate TMS should have capable of running data analysis or mining the data using statistical approaches or machine learning methods. By using of big data technologies, in the proposed computational layer we have developed a computational engine for every traffic analysis needed for TMS services (Figure 5).

As mentioned before, one of the basic requirements of an adequate TMS is easy implementation for different applications. Therefore, in the central core of the computational layer, we have used Hadoop MapReduce engine. MapReduce paradigm due to easy programming, automatic load balancing and scalability is a suitable approach for using in a TMS framework. Generally, in a MapReduce program, map function is used for creating intermediate data. Then the intermediate data aggregated using reduce function to generate the final result. By this simple architecture, parallel functions can be automatically managed and load balance significantly reduces compared to traditional platforms.

There are some other modules in the computational layer. Hive and Pig are two open-source data warehousing and query processing tools. Hive prepares an easy read and write interface for distributed storage by using SQL-like language named HiveQL. In Hive, a command-line tool and a JDBC driver for users interaction is included. Hive converts HiveQL queries to MapReduce programs and then runs them on Hadoop. It is clear that MapReduce and Hive final results are the same but MapReduce coding and debugging is more complicated than simple SQL-like commands. Generally, Hive provides higher level of abstraction compared to MapReduce. However like every other high level abstraction, using hive needs additional computation which leads to lower performance of the system (Thusoo et al., 2009). Similarly, Pig also provides higher level of abstraction compared to MapReduce. Pig is an analytical platform for big datasets that contains a high level programming language "PigLatin" for data analysis programming along an infrastructure for evaluating this programs. PigLatin is a simple, expandable and optimize programming language.

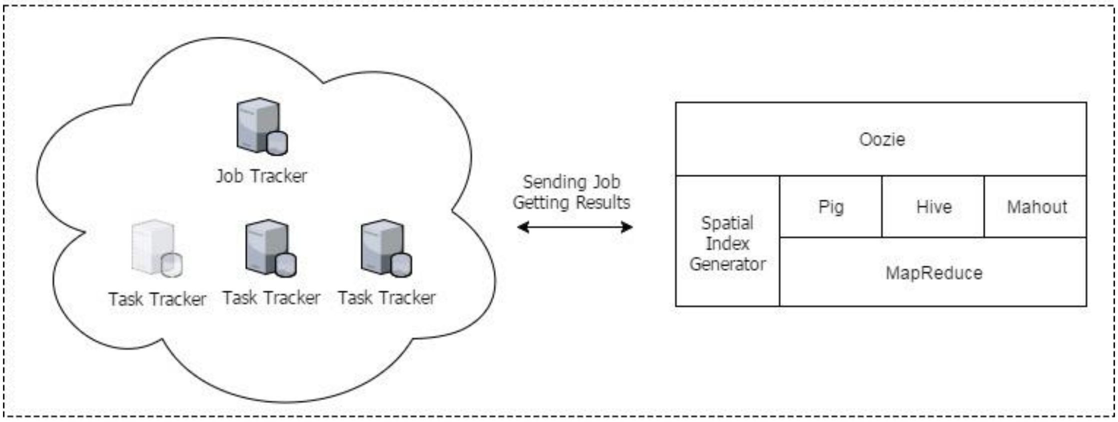


Figure 5. Computational layer

As mentioned earlier, in some complicated traffic management analysis data mining and machine learning tools are needed. One of the most important open-source projects in Hadoop ecosystem is Mahout. Mahout is a distributed framework for machine learning. Mahout supports various algorithms like classification, clustering and filtering in a parallel manner. The main goal of Mahout is development of a scalable machine-learning library. In this work, we have used Oozie as the workflow engine.

3.3 Application Layer

Some of the most important services of TMS are vehicle routing to shorten commuter journey, traffic prediction that enables early detection of bottlenecks, parking management that ensure optimal usage of parking spots and interact with routing and prediction services for improved control of traffic flow and finally infotainment services that provide useful information for both drivers and passengers (Djahel et al., 2015). In order to make an interaction between user and system, a set of RESTful web services are designed. Various functions for simple and flexible interaction of user with this web services has been considered.

4. FREAMEWORK EVALUATION

We have setup our framework on a Grid5000’s Nancy cluster using 3 nodes. The master node is responsible for data processing along with cluster management. The other slave nodes are only responsible for data processing. The commodity environment used for this evaluation is shown in Table 2. Since our work is still under development, we have only implemented a prototype of our proposed framework and parts of data management and computational layer components have been used in this evaluation.

We have evaluated data management layer performance by importing a real OpenStreetMap[1] (OSM) trajectory datasets as a sample traffic data to our framework. Several datasets with different sizes are selected to evaluate the performance. As shown in Fig.6 (a), the speed of records import to data management layer exceeds 8000 records per second when the size of datasets is near to 5 million. We also evaluate performance of data retrieval in our proposed framework. As shown in Fig.6 (b) the retrieval performance is faster than the import process. The data retrieval speed exceeds 15000 records per second when the size of datasets is near to 5 million.

Hadoop Version	2.7.1
HBase Version	1.1.2
Flume Version	1.5.2
Cluster Model	Dell PowerEdge R730
Nodes Processors	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz
Memory	64GB RAM
Network	10 Gigabit Ethernet DA/SFP+

Table 2. Evaluation environment

For evaluation of computational layer, we have parallelized a critical pre-analysis in traffic management application. In this use case, we have evaluated scalability and performance of our proposed framework capabilities for traffic management applications. FCD is one of the most important data sources in traffic management applications that represents Spatio-temporal trajectory of a vehicle. Due to FCD nature, there are a lot of noises in this data. Therefore, FCD has limited accuracy and cannot be matched on the road network by itself. To face this issue, an important traffic analysis named map matching has been developed. In this study we have used (Liu et al., 2012) method.

Our analysis is conducted on the OSM real trajectories and road network that has been used for data management and computational layers evaluation. Fig.7 (a) represents scalability evaluation of our map matching implementation. As shown in Fig.7 (a), with the growth of computation nodes, processing time has decreased significantly. We also evaluate map matching speed using different processing nodes. As shown in Fig.7 (b), by the increase of computational nodes, matching speed shows a significant growth.

[1] www.openstreetmap.org

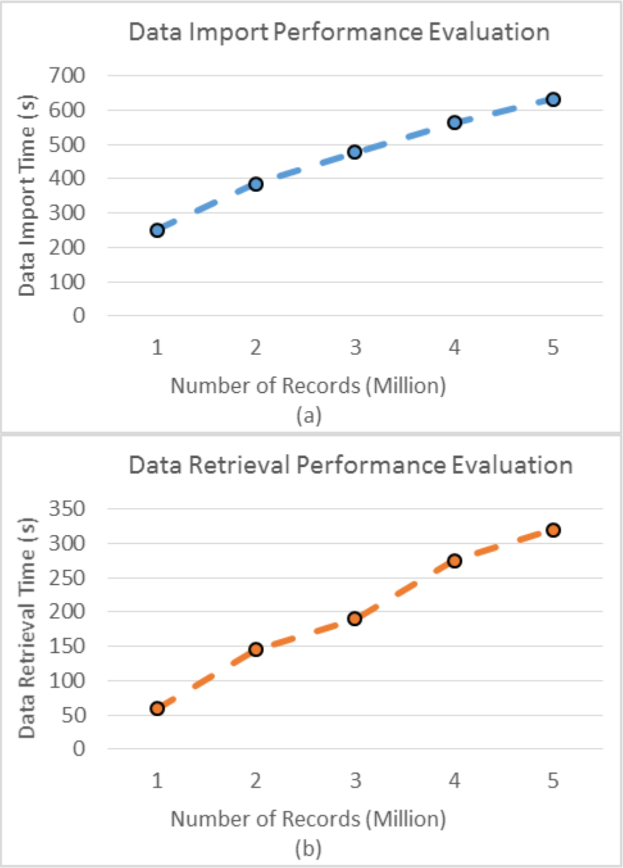


Figure 6. Data management layer evaluation

5. CONCLUSION AND FUTURE WORK

Traffic Big data has initiated great opportunities for traffic management applications. Beside such opportunity, several technical challenges have appeared. This paper takes a major step towards development of an efficient and real-time traffic management system using cloud computing technologies. The proposed distributed three layer framework contains several open-source components and libraries for supporting traffic management applications. Our evaluation results show that the proposed framework has efficiently stored, managed and analysed traffic big data. Using Hadoop ecosystem and MapReduce paradigm, proposed approach provides high availability and scalability along with fault tolerance for real-time traffic management applications. In our future work, we plan to improve our framework by considering data mining and knowledge discovery techniques, improving our spatial indexing method and concentrate more on our application layer to support more application functionalities.

ACKNOWLEDGEMENTS

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations[2].

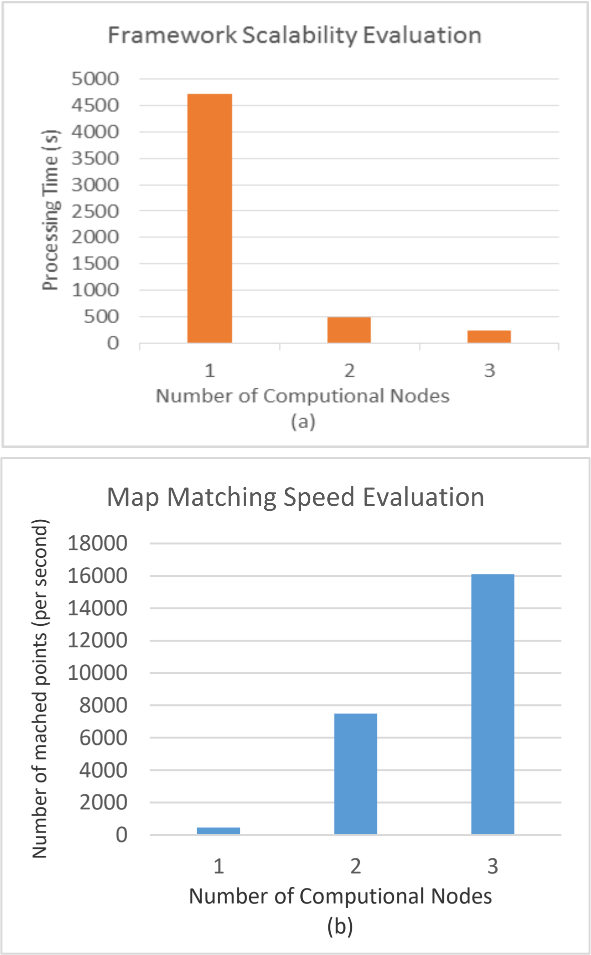


Figure 7. Computational layer evaluation

REFERENCES

Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., Saltz, J., 2013. Hadoop GIS: a high performance spatial data warehousing system over mapreduce. Proceedings of the VLDB Endowment 6, 1009-1020.

Apache, Apache Sqoop.

Apache, Welcome to Apache Flume.

Apache, Welcome to Apache Hadoop!

Apache, Welcome to Apache HBase™.

Djahel, S., Doolan, R., Muntean, G.-M., Murphy, J., 2015. A communications-oriented perspective on traffic management systems for smart cities: challenges and innovative approaches. IEEE Communications Surveys & Tutorials 17, 125-151.

Eldawy, A., Mokbel, M.F., 2015. Spatialhadoop: A mapreduce framework for spatial data, 2015 IEEE 31st International Conference on Data Engineering. IEEE, pp. 1352-1363.

Finkel, R.A., Bentley, J.L., 1974. Quad trees a data structure for retrieval on composite keys. Acta informatica 4, 1-9.

[2] https://www.grid5000.fr

Guttman, A., 1984. R-trees: a dynamic index structure for spatial searching. ACM.

Hadoop, APACHE HIVE TM.

Hadoop, Welcome to Apache Pig!

Hadoop, What is Apache Mahout? Apache.

Hortonworks, Apache Oozie.

Khazaei, H., Zareian, S., Veleda, R., Litoiu, M., 2015. Sipresk: A Big Data Analytic Platform for Smart Transportation, EAI International Conference on Big Data and Analytics for Smart Cities.

Liu, S., Liu, C., Luo, Q., Ni, L.M., Krishnan, R., 2012. Calibrating large scale vehicle trajectory data, 2012 IEEE 13th International Conference on Mobile Data Management. IEEE, pp. 222-231.

Sekar, E.V., Anuradha, J., Arya, A., Balusamy, B., Chang, V., 2017. A framework for smart traffic management using hybrid clustering techniques. Cluster Computing, 1-16.

STATISTICS, B.O.T., 2015. Transportation Statistics Annual Report, in: Transportation, U.S.D.o. (Ed.).

Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R., 2009. Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment 2, 1626-1629.

Xiao, S., Liu, X.C., Wang, Y., 2015. Data-Driven Geospatial-Enabled Transportation Platform for Freeway Performance Analysis. IEEE Intelligent Transportation Systems Magazine 7, 10-21.

Xiong, G., Zhu, F., Dong, X., Fan, H., Hu, B., Kong, Q., Kang, W., Teng, T., 2016. A kind of novel ITS based on space-air-ground big-data. IEEE Intelligent Transportation Systems Magazine 8, 10-22.

Yu, J., Jiang, F., Zhu, T., 2013. Rtic-c: A big data system for massive traffic information mining, Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on. IEEE, pp. 395-402.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W4-447-2017 | © Authors 2017. CC BY 4.0 License.