# An Architecture for Big Data Processing on Intelligent Transportation Systems

*An application scenario on highway traffic flows*

Guilherme Guerreiro, Paulo Figueiras, Ricardo Silva, Ruben Costa, Ricardo Jardim-Goncalves
CTS, UNINOVA, Dep. de Eng.ª Eletrotécnica,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
2829-516 Caparica, Portugal
g.guerreiro@campus.fct.unl.pt, paf@uninova.pt, p110358@campus.fct.unl.pt, rddc@uninova.pt, rg@uninova.pt

*Abstract*—**The transportation sector, and in particularly intelligent transportation systems, generate large volumes of real-time data that needs to be managed, communicated, interpreted, aggregated, and analyzed. To this end, innovative big data processing and mining as well as optimization techniques, need to be developed and applied in order to support real-time decision-making capabilities. Towards this end, this paper presents an ETL (extract, transform and load) architecture for intelligent transportation systems, addressing an application scenario on dynamic toll charging for highways. The ETL approach presented here, is responsible for preparing the data to be used by traffic prediction services, which will dynamically affect toll prices within different contexts. The proposed architecture relies on the adoption of "big data" technologies, to process and store large volumes of data from heterogeneous sources, provided by different highway operators. The proposed architecture is capable of handling real-time and historical data using big data technologies such as Spark on Hadoop and MongoDB. The DATEX-II data model is adopted, in order to harmonize traffic data provided by the highway operators. The work presented here, is still part of ongoing work currently addressed under the EU H2020 OPTIMUM project. Preliminary results achieved so far do not address the final conclusions of the project, but enabled us to demonstrate considerable gains in performance, when compared to other traditional ETL approaches, and also form the basis for pointing out and discuss future work directions and opportunities in the area of the development of big data processing and mining methods under the ITS domain.**

***Keywords— ITS, Tolling Systems, Big Data Processing, Data Mining, ETL***

## I. Introduction

In the last few decades, technology changed the way people live, interact and work. The revolution made by smartphones, internet and sensors, lead to the collection of large volumes of data on a daily basis. Intelligent transportation systems (ITS) have to deal with the acquisition and processing of data coming from road sensors, sensing mobile devices, cameras, radio-frequency identification readers, microphones, social media feeds and other sources, in order to help daily commuters and transportation companies in the decision making process. The efficient processing and storage of transportation data, can help mitigating many of the transportation challenges, such as excessive CO2 emissions, traffic congestions, increased accident risks, and reduced quality of life.

All actors in ITS behave as data providers and consumers, leading to large volumes of available data needed to be processed. The growth in data production is being driven by: individuals and their increased use of media (social networks); novel types of sensors and communication capabilities in vehicle and the traffic infrastructure; application of modern information and communication technologies (ICT) (Cloud computing, Internet of Things (IoT), etc.) [1]. Data sets grow in size because they are being gathered increasingly by ubiquitous information-sensing mobile devices, aerial sensory technologies (remote sensing), software logs, cameras, microphones, radio-frequency identification readers, and wireless sensor networks. Thus, there is an emerging Big Data challenge in the ITS domain.

The big challenge in ITS is not on how to collect, but how to process and model large volumes of unstructured data for posterior analytics, which cannot be handled effectively by traditional approaches. There is a need for developing innovative services and applications capable of process and infer information in real-time for better support the decision making, but also to predict complex traffic related situations before they occur and take proactive actions. Other aspects that must be considered are the challenges driven by a big data context, such as inconsistencies on the data itself (for example, radio-frequency highway counters are susceptible to several anomalies), outdated data, the bandwidth of the connection. Such inconsistencies lead to the lack of quality on the data sets, adding to that, the challenge in fusing and harmonizing large volumes of data from many sources at the same time (volume to variety ratio). In the case of the application scenario presented here, toll areas of the same highway are concessioned to different operators, resulting in different reporting schemes of highway traffic data.

In order to address the challenges stated previously, data needs to be gathered, cleaned, transformed and stored efficiently. Only then, is possible to extract value of a large amount of data. The ETL concept, stands for Extract-Transform-Load and represents the process in which data is

978-1-5090-1354-8/16/$31.00 ©2016 IEEE

loaded from a source to a unified data repository. ETL software houses have been extending their solutions to provide big data extraction, transformation and loading between big data platforms and traditional data management platforms, describing ETL now has “Big ETL” [2].

This work proposes and evaluates an ETL-based approach, able to process and model large volumes of raw traffic data efficiently. The proposed architecture should be able to: (i) take into account the data quality; (ii) the ability to cope with already existing data standards under the ITS domain, such as DATEX-II, to guarantee harmonization among the data; (iii) provide a robust and scalable storage system. The architecture will prepare the data so it can be used by additional ITS services, able to predict traffic flows in real-time or detect traffic related events. For that purpose, the proposed architecture adopts “big data technologies” as Apache Spark and SparkSQL for data processing, and MongoDB for data storage. A more detailed overview of the proposed architecture also with an evaluation of its performance, is presented in the next sections. This approach follows the principles of parallel, in-memory processing by Big Data technologies with all the advantages showed in studies [3], [4].

The approach presented here, is being developed under the H2020 R&D OPTIMUM project [5]. The OPTIMUM project operates in an environment of ubiquitous connectivity throughout the transportation system and its surroundings that continuously provides data on the present state and emerging situations. Examples include traffic and in-vehicle sensors, positioning information (e.g. GNSS data), occupancy of public transportation, crowd data sourcing through social networks and availability of modalities such as shared bicycles and cars. Within this context one of the application scenarios of OPTIMUM project involves the development of a dynamic (toll) charging model, aiming to induce behavioral changes in drivers by transferring heavy traffic from the urban and national roads into highways.

This paper is structured as follows: Section 2 presents the related work. The methods used for data fusion and harmonization are briefly explained in section 3. Section 4 highlight the application scenario used for adopting the proposed architecture. Section 5 describes the technical architecture. The assessment of the work is described in section 6. Finally, section 4 presents our preliminary conclusions.

## II. Related work

In order to cope with challenges and objectives previously identified, the work proposed here aims at developing a big-data platform for fusing and harmonize heterogeneous, dynamic streams of transportation data, provided by public authorities, transport operators as well as social media for a better management of transportation networks. Data interoperability is seen here as an important challenge to be tackled, since the proposed architecture needs to be compliant with a large diversity of transportation related data/services developed in different formats. The adoption of common standards, the strong contribution to existing standards as well as a vendor-independent philosophy are expected to lower this barrier, which nevertheless cannot be underestimated.

The term big data is used for extremely large and complex data sets, which cannot be handled properly by traditional approaches and tools. Big data represents the assets characterized by high volume, velocity and variety requiring specific technology and analytical methods for its transformation into value [6]. As the definition behind, the three aspects most considered to define big data in the literature are: (i) Volume, from the ever augmenting data collected; (ii) Velocity, from the growth on data acquisition; (iii) Variety, from heterogeneity of data formats and protocols used. Other aspects that must be considered are the issues in transporting data in a big data context, such as the many inconsistencies of the data itself (from gathering values from broken road sensors, for example), outdated data, the speed variation of the internet connection, or others. Leading to a low veracity ratio, since the quality of data is constantly changing, even from the same source, and big complexity from the difficulties to fuse large amounts of data from many sources at the same time (Volume to Variety ratio).

Hadoop [7] from Apache, the most well-known framework for big data processes, it was developed to process large data sets in a distributed manner, typically used for batch processing. Hadoop was designed for scalable applications and offers also his own type of storage.

Apache Spark is a high level and complete framework for BDPM, offering Spark SQL for working with structured data, Spark Streaming for stream processing, MLlib for machine learning libraries and GraphX for graphs and graph-parallel computation. It runs on Hadoop but uses a different kind of working data sets, with Resilient Distributed Datasets (RDD), which are distributed through the cluster nodes memory when jobs are running. RDDs gives efficient recovery after failure. Another great advantage of Spark is that runs in- memory, being more efficient in some operations such as iteration work.

Apache Storm is a free and open source distributed real-time computation system. Storm is used for real-time analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Storm is fast: a benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate.

For storage, regular SQL relational databases are not built to support today´s Big Data, so, the adoption of a NoSQL (Not Only SQL) technology is considered more appropriate. In the field of Big Data storage, there are technologies such as Hive, Cloudera, Cassandra and MongoDB. While the first two are based on Hadoop, the second two are based in NoSQL.

Proposing and ETL architecture that addresses the challenges of the big data presents several challenges, nevertheless the literature shows us some relevant works around the Big ETL concept, and possible approaches using big data technologies applied to the ITS domain. The authors in [3], present a technical architecture consisting of core Hadoop services and libraries for data ingest, processing and analytics, operating on an automotive domain processing a dataset of multiple terabytes and billions of rows. In comparison to traditional data warehouses, SQL on Hadoop provides greater agility supporting relational data, unstructured data and schema-on-read data organization. The authors in [4] propose a design scheme based on distributed architecture, mainly using Kafka, Storm and Spark clusters, used to process smart city data as: Map and POI data, GPS data, traffic data, video surveillance data, environment data, social activity data. Some results indicate that Spark performs better than other approaches.

In [8], [9], [4] and [3] it is perceived the many advantages of distributed processing in ETL and data analytics tasks, regarding structured and non-structured data, as well as done in [3] and [4] a performance comparison between some technologies that implement the distributed approach. In [9], the authors enhance an existing mobility analytics framework to perform mobility analytics of mobile IoT gateway nodes (along with IoT end-devices), using Spark technology. The authors argue that Spark proved to be a suitable technology for infrastructure as well as ad-hoc modes.

In order to manage with several heterogeneous data sources, this work adopts an ITS standard based data model. DATEX-II [10] standard, was first published in the end of 2006 and acknowledged in 2011 by the European Technical Specification Institute (ETSI) [11], for modelling and exchanging ITS related information, being a European standard for ITS since then. From the beginning it has been developed to provide a way to standardize information covering the communication between traffic centers, service providers, traffic operators or media partners.

Since the first release many aspects have been improved (current version is 2.3) and now offers many other features. Is at this time developed and maintained by the EasyWay project [12] and supported by the European Commission. Some of the main uses are: (i) Routing/ rerouting using traffic management; (ii) Linking traffic management and traffic information systems; (iii) multi-modal information systems; (iv) information exchange between cars or between cars and traffic infrastructure systems.

## III. Methodology

The methodology adopted to implement the proposed ETL architecture was the CRISP-DM (Cross Industry Standard Process for Data Mining). It is considered a well stablished methodology, providing a uniform framework and guideline for data miners. Although it was first published in 1999, CRISP- DM has been refined over the year, and with his six steps implementation assures that in the end of the process the knowledge and results are the expected for deployment. Summarily the six steps consist in:

- Business Understanding: This step uncovers important factors including success criteria, business and data mining objectives and requirements as well as business terminologies and technical terms. In this step, business requirements were elicitated from dynamic tool charging. Several interviews were conducted with highway road operators.
- Data Understanding: This step focuses on data collection, checking quality and exploring of data to get insight of data to form hypotheses for hidden information. Reports about data quality and data availability were developed in order to assess the quality level of the data sources.
- Data Preparation: selection and preparation of final the data set. This phase may include many tasks, such as records, tables and attributes selection as well as cleaning and transformation of data. This step is where the work addressed in this paper is concerned. Data preparation deals also with the development of the ETL architecture.
- Modeling: Selection and application of various modeling techniques. Different parameters are set and different models are built for same data mining problem. The objective here, is the adoption of data forecasting algorithms (for short, medium-term predictions), which will be used as input to the dynamic tool charging model.
- Evaluation: Evaluation of obtained models and decision on how to use the results. Interpretation of the model depends upon the algorithm and models can be evaluated to review whether it achieves the objectives properly or not. This steps deals with the validation of the dynamic tool charging model, from both technical and business perspective.
- Deployment: Determining possible uses for obtained knowledge and results. This phase also focuses on organizing, reporting and presenting the discovered knowledge when needed.

CRISP-DM is considered one of most widely methodologies for data mining projects, together with SEMMA and KDD [13] [14]. CRISP-DM was decided to be adopted here, mainly because is considered more complete than SEMMA and more practical to apply in real case scenarios with defined objectives. While KDD can be more useful in generic research.

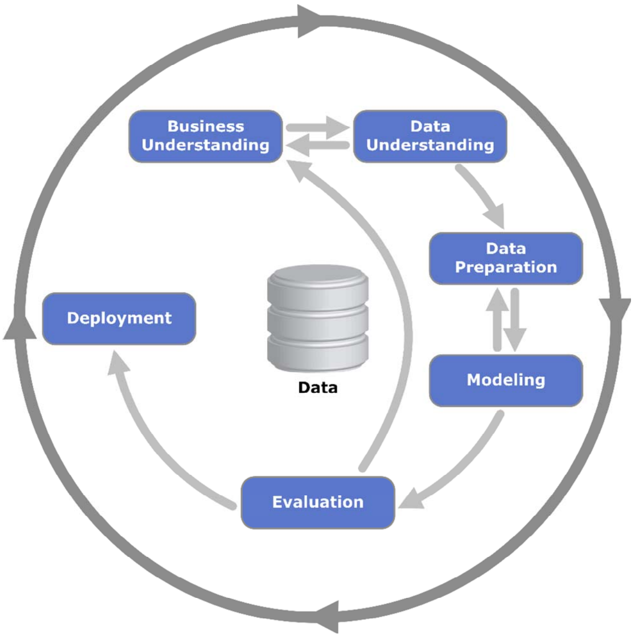


Figure 1. CRISP-DM Methodology [13]

## IV. Application Scenario

The design and development of a dynamic toll pricing model for highways, means that, the design of a pricing model have to take into account traffic flow in real time (including all types of events, maintenance, accidents, weather related situations) and traffic flow prediction (from historical data), resulting also in quality of service prediction for the highway and national roads. This dynamic pricing has the objective of attracting or discouraging the use of specific motorways according to the quality of service prediction in those motorways and/or adjacent complementary roads.

The objective of the proposed approach, is to encourage drivers in using highways by affecting the tolling prices during peak hours on heavy used national roads. We believe that an approach based on a congestion toll pricing model, will additionally improve the financial ratios in infrastructures' management, and allowing an increased quality of life for users and population living near the urban/national roads.

The proposed dynamic toll pricing model changes tolls' prices, depending on several factors, such as real-time conditions of road networks, quality of service, road safety, environmental data, cost maintenance, toll revenues, congestion, traffic events and weather conditions. In order to support the model, there is the need to accurately predict the status of road networks for real-time, short and medium term horizons, by using machine learning algorithms. Such algorithms will be used to feed the dynamic toll pricing model, reflecting the present and future traffic situations on the network. Since traffic data quantity and quality are crucial to the prediction of road networks' statuses, real-time and predictive analytics methods will use a panoply of data sources. Traffic data is collected from road sensors nodes, placed at several sections of the road network, collecting data at different volumes and at different levels of granularity, which presents a challenge in terms of data homogeneity and reliability. Therefore, the work to be presented, addresses the development of a platform based on Big Data technologies, able to capture, process and store traffic data from sensors, such as road-counters, telematics data, traffic incidents, GPS tracks and social media.

The highway operator has provided with vehicle-counting sensors throughout their road network. For the purpose of the application scenario, the area covers portion of Portugal's highway network, focusing on the northern part of Portugal, with 270 active sensors (135 in each road direction) from which to extract vehicle passages, road occupancy and average speeds.

## V. Proposed Architecture

For the descried application scenario, it is proposed an architecture to extract, transform and store efficiently (ETL stage) all the data previously described. The proposed architecture addresses the following technical requirements: (i) able to deal with raw data in many formats and sizes; (ii) assure data quality; (iii) efficient big data transformation and storage; (iv) being able to address interoperability at the data level, enabling the development of additional value added services for highways users; (v) and a robust and efficient distributed storage system, that is scalable in order to process data from additional traffic sensors.

To address the previous technical requirements, the proposed architecture was developed under Apache Spark used for large-scale data processing, which includes the tasks of data cleaning and transformation. Spark is one of the parallel processing technologies identified in the literature to deal with Big Data ITS information, some authors state that in some cases it outperforms Hadoop [4] [3] regarding processing time. It also provides a framework for data management, such as SparkSQL in an integrated environment.

MongoDB was used, as a NoSQL approach, for storing and manage the traffic data. Mongodb is a NoSQL database, related literature considers as a flexible distributed database technology with low latency, adopting a document oriented approach. Because it is scalable and easily integrated with Spark, it fits the requirements of the proposed scenario.

Fig.2, depicts the conceptual architecture, where the data sources may arrive in different means (local or server documents, web services using SOAP or REST), and in different formats (txt, CSV, XLS, XML, JSON) and with all kind of sizes. Spark is responsible for the data cleaning and harmonization step.

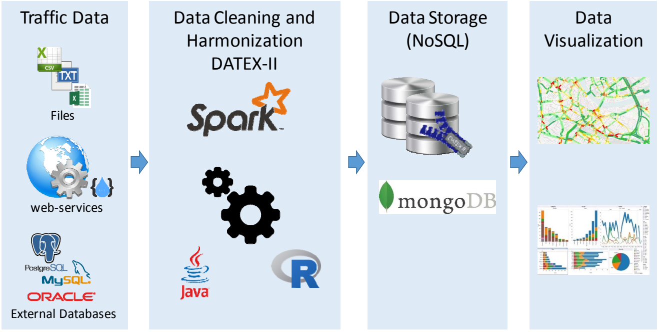


Figure 2. Data Harmonization Approach

In this architecture, MongoDB was configured by having two different collections of documents, one for collecting real time data, such as traffic sensor data and traffic events (accidents, road blocks, road works), and another collection for historical data. This fulfill some advantages discussed in the related work, a unified mobility framework for data cleaning, including a historic and real -time data, enabling the coupling of machine learning methods, such as traffic prediction.

With respect to the traffic data being processed, those were grouped into two main categories: tolling data and vehicle counting data. Leading into two different data sources presenting different attributes and different semantic meanings.

Tooling data, describes the actual number of vehicles that were tolled within a 5-minute interval by each class for a certain highway segment, it is available through csv files containing the attributes described in Table I.

TABLE I. TOLLING DATA

| | |
|---|---|
| **Concession Id** | Integer value, identifying the region concessioned |
| **Toll Gantry Id** | Integer value identifying the toll |
| **Date** | Date and time (YYYMMDDHHmm) |
| **Class1** | Total number of class 1 vehicles (height of 1st axis <1,10m) |
| **Class2** | Total number of class 2 vehicles (height of 1st axis >=1,10m) |
| **Class3** | Total number of class 3 vehicles (3 axis) |
| **Class4** | Total number of class 4 vehicles (more than 4 axis) |
| **Class5** | Total number of class 5 vehicles |

The classes of vehicles are identified in the figure below. To refer that class 5, refers to motorcycles with onboard electronic device.

Vehicle counting data, provides the total number of vehicles counted in a certain point of the highway within a 5-minute interval, the data is available through csv files containing the attributes described in Table II.

TABLE II. VEHICLE COUNTING DATA

| | |
|---|---|
| **Concession Id** | Integer value, identifying the region concessioned |
| **Road** | Road identifier |
| **Road Segment** | Identification of the segment |
| **Bearing** | Northbound our Southbound |
| **Equipment type** | Sensor type |
| **Date Time** | Date and time (YYYY-MM-DD hh:mm:ss) |
| **Total Vehicles** | Total number of vehicles |
| **Light** | Total number of light vehicles |
| **Heavy** | Total number of heavy vehicles |
| **Category A** | Total number of Motorcycles |
| **Category B** | Total number of Light vehicles |
| **Category C** | Total number of Heavy vehicles of goods |
| **Category D** | Total number of Heavy vehicles of passengers |

As mention previously, in terms of data harmonization, the DATEX-II standard was chosen for harmonizing traffic related data. DATEX-II is a complete standard for ITS related information, most generally used today and very adaptable, being able to assemble the main required types of data. All information is mapped to XML, so it is easy to transport and retrieve, regardless of the technologies used. Table III, highlights some transformations performed at the data level, in order to be DATEX-II compliant.

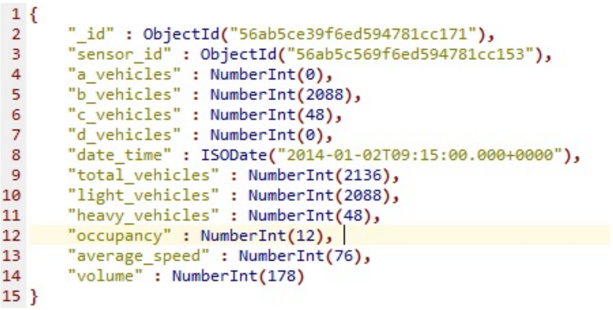
TABLE III. HETEROGENEOUS SOURCES TO DATEX-II TRANSFORMATIONS

| CSV Fields | | DATEX-II |
|---|---|---|
| sensor_id | → | **datex:**SiteMeasurements(measurementSiteReference) |
| date_time | → | **datex:**DateTimeValue(dateTime) |
| total_vehicles | → | **datex:**TrafficFlow(vehicleFlowValue) |
| occupancy | → | **datex:**TrafficConcentration(occupancy) |
| average_speed | → | **datex:**TrafficSpeed(averageVehicleSpeed) |
| volume | → | **datex:**TrafficFlow(vehicleFlowValue) |

## VI. VALIDATION AND RESULTS

For validation and testing, it was used CSV files as input, containing raw traffic flow data. The approach proposed here grows upon a typical ETL approach, by adopting big data technologies, therefore the metrics used for validation will measure the performance of the proposed approach versus a typical ETL approach without big data technologies. The tests were performed using historical traffic flows from 2010 till 2016 in several highways, in a total of 36 CSV, resulting in 30 million records to be processed. Each record in the files are composed by concession id, toll id, date and each car category flow. The data is cleaned, transform to DATEX-II and store directly in the MongoDB historic collection, due to the nature of the data.

The transformation rules were written in pure java code, and are executed within the ETL process. Figure 3, illustrates an example of a query response to the MongoDB data repository.



```
{
    "_id" : ObjectId("56ab5ce39f6ed594781cc171"),
    "sensor_id" : ObjectId("56ab5c569f6ed594781cc153"),
    "a_vehicles" : NumberInt(0),
    "b_vehicles" : NumberInt(2088),
    "c_vehicles" : NumberInt(48),
    "d_vehicles" : NumberInt(0),
    "date_time" : ISODate("2014-01-02T09:15:00.000+0000"),
    "total_vehicles" : NumberInt(2136),
    "light_vehicles" : NumberInt(2088),
    "heavy_vehicles" : NumberInt(48),
    "occupancy" : NumberInt(12),
    "average_speed" : NumberInt(76),
    "volume" : NumberInt(178)
}
```

Figure 3. MongoDB tolling data example

Fig. 5, highlights the performances resulting using a traditional approach without any big data technology, an approach using Spark configured to run locally with 2 threads, and using Spark configured to run locally with 4 threads.

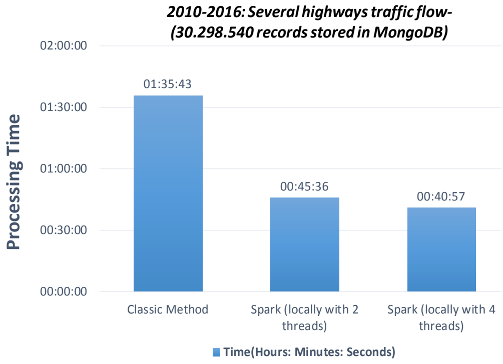


Figure 4. Processing time “classical method” vs “spark running localy”

In Fig. 6, it is depicted the performances of the proposed architecture, using the classical method, the method running Spark locally with 4 threads and using Spark on a distributed cluster environment (in standalone mode and 1 additional worker).

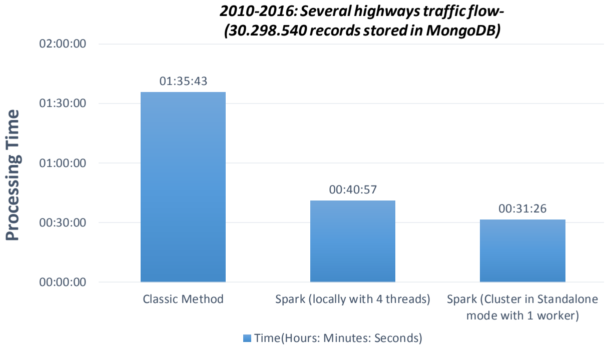


Figure 5. Processing time “classical method” vs “spark running localy” vs “spark running on a distributed cluster enviroment”

Spark currently supports three cluster managers: (i) Standalone; (ii) Apache Mesos, a cluster manager that runs Hadoop MapReduce and service applications; and (iii) Hadoop YARN, resource manager in Hadoop. For the current validation process, it was decided to adopt Spark in standalone mode, due to its simplicity in setting the cluster and because it is already included in Spark. It is worth to mention that Spark was running locally on an i5-4200U, with 8gb ram machine, and the worker node was working on a i7-4790, with 8gb ram machine.

The results indicate that a “big data” approach, as the one presented here, is more suitable in terms of performance with respect to traditional approaches for ETL tasks. In Fig. 4, the classic method performance was compared with a local installation of Spark running with a 2 threads and a 4 threads mode. There is a clearly increase of performance when using Spark technology, around 50% of decrease in the processing time. When comparing the 2 thread and 4 thread mode, the improvement is not so obvious. For that reason, it was decided to test it on a cluster environment running spark in a standalone mode with an additional remote worker machine, where the results showed a 25% gain when compared to the local mode with 4 threads.

Other Spark configurations were also tested, namely increasing the number of local threads and also increasing the number of remote workers. Such configurations did not clearly show substantial gains in terms of performance. By reading other related works, we have found out that Spark presents some issues related with garbage collection increase and file I/O time. Therefore, as future work, it is expected to test the approach to a larger data set.

## VII. Conclusions and Future Work

This paper presents an ETL architecture for ITS, addressing an application scenario on dynamic toll charging for highways. We strongly believe that the approach proposed here, offers a novel big data-driven solution, which will lead to seamless transportation services for the movement of people and goods. The fusion of transportation data and their transformation into actionable information, meets the mobility needs of passengers and ensures a wider choice of transportation services. Having access to proper transportation related information allows European citizens to make better use of the existing infrastructure when travelling, whereas through information personalization and persuasive mechanisms a shift to more environmentally friendly modes of transport can be achieved. Although the approach presented here is still on an early stage, we are greatly encouraged by the results of our preliminary explorations.

The work also highlights, the impact of running Spark cluster as opposed to other traditional approaches. The effectiveness of Spark cluster in a standalone mode, is observed by comparing the outcomes of our tests. Our preliminary results clearly show Spark cluster is more effective in terms of performance.

The next phase, includes the integration of additional traffic data from other traffic sensors in order to cover a bigger geographical area. One of the objectives of ETL is to prepare and model data, to be used/consumed by high level services, which includes traffic forecasting and complex event processing (CEP). The development of traffic forecasting services based on regression techniques are required in order to feed the dynamic toll charging model, on the other hand, CEP mechanisms are also required in order to identify relevant events (traffic accidents, traffic jams) in real-time which can heavily influence the status of the transportation network.

## Acknowledgment

The authors acknowledge the European Commission for its support and partial funding and the partners of the research project: H2020–644715 OPTIMUM

## References

[1] J. Fiosina, M. Fiosins and J. Müller, "Big Data Processing and Mining for Next Generation Intelligent Transportation Systems," *Jurnal Teknologi,* vol. 63, no. 3, pp. 21-38, 2013.

[2] J. Caserta and E. Cordo, "Big ETL: The Next 'Big' Thing," 9 February 2015. [Online]. Available: http://data-informed.com/big-etl-next-big-thing/.

[3] A. Luckow, K. Kennedy, F. Manhardt, E. Djerekarov, B. Vorster and A. Apon, "Automotive big data: Applications, workloads and infrastructures," in *IEEE International Conference on Big Data*, Santa Clara, CA, 2015.

[4] S. Ma and Z. Liang, "Design and Implementation of Smart City Big Data Processing Platform Based on Distributed Architecture," in *10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Taipei, 2015.

[5] OPTIMUM consortium, "OPTIMUM Project," 1 October 2015. [Online]. Available: http://optimumproject.eu/. [Accessed 20 April 2016].

[6] A. De Mauro, M. Greco and M. Grimaldi, "What is big data? A consensual definition and a review of key research topics," in *proceedings of the 4th International Conference on Integrated Information*, Madrid, 2015.

[7] The Apache Software Foundation., "Hadoop," 2014. [Online]. [Accessed 2016].

[8] O. Andersen, B. B. Krogh and K. Torp, "An open-source based ITS platform," *Proceedings - IEEE International Conference on Mobile Data Management,* vol. 2, pp. 27-32, 2013.

[9] M. Taneja, "A Mobility Analytics Framework for Internet of Things," pp. 113-118, 2015.

[10] EIP/EIP+ Project, "Datex Easyway," 2014. [Online]. Available: http://www.datex2.eu/.

[11] European Telecommunications Standards Institute, 2015. [Online]. Available: http://www.etsi.org/.

[12] Easyway, "DATEX II –The standard for ITS on European Roads," 2011. [Online]. Available: http://www.datex2.eu/.

[13] A. Azevedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: a parallel overview," *IADIS European Conference Data Mining,* pp. 182-185, 2008.

[14] U. Shafique and H. Qaiser, "A Comparative Study of Data Mining Process Models ( KDD , CRISP-DM and SEMMA )," *International Journal of Innovation and Scientific Research,* vol. 12, no. 1, pp. 217-222, 2014.