Service Oriented Big Data Management for Transport

Gavin Kemp, Genoveva Vargas-Solar, Catarina Ferreira da Silva, Parisa Ghodous, Christine Collet

This a final pre-published version. The published version can be found at http://link.springer.com/chapter/10.1007/978-3-319-27753-0_15
DOI 10.1007/978-3-319-27753-0_15

Service oriented big data management for transport

Gavin Kemp[1], Genoveva Vargas-Solar[2,3], Catarina Ferreira Da Silva[1], Parisa Ghodous[1] and Christine Collet[2]

[1]Université Lyon 1, LIRIS, CNRS, UMR5202, bd du 11 novembre 1918, Villeurbanne, F69621, France
[2]Grenoble Institute of Technology, LIG, 681 rue de la Passerelle, Saint Martin d'Hères, France
[3]LIG-LAFMIA, CNRS 681 rue de la Passerelle, Saint Martin d'Hères, France

Keywords: ITS, Big Data, Cloud Services, NoSQL, Service Oriented architecture

Abstract. The increasing power of computer hardware and the sophistication of computer software have brought many new possibilities to information world. On one side the possibility to analyze massive data sets has brought new insight, knowledge and information. On the other, it has enabled to massively distribute computing and has opened to a new programming paradigm called Service Oriented Computing particularly well adapted to cloud computing. Applying these new technologies to the transport industry can bring new understanding to town transport infrastructures. The objective of our work is to manage and aggregate cloud services for managing big data and assist decision making for transport systems. Thus this paper presents our approach to propose a service oriented architecture for big data analytics for transport systems based on the cloud. Proposing big data management strategies for data produced by transport infrastructures, whilst maintaining cost effective systems deployed on the cloud, is a promising approach. We present the advancement for developing the Data acquisition service and Information extraction and cleaning service as well as the analysis for choosing a sharding strategy.

1 Introduction

During the last five years, the problem of providing intelligent real time data management using cloud computing technologies has attracted more and more attention from both academic researchers, e.g. P. Valduriez team in France [1], Freddy Lecue's work at Ireland IBM Research Lab [2], Big Data Initiative CSAIL Laboratory in MIT, USA, Cyrus Shahabi's team University of Southern California in USA [3] and industrial practitioners like Google Big Query, IBM, Thales. They mostly concentrate on modelling stream traffic flow, yet they barely combine different data flows with other big data to provide new intelligent transport services (ITS). ITS apply technology for integrating computers, electronics, satellites and sensors for making every transport mode (road, rail, air, water) more efficient, safe, and energy saving. ITS effectiveness relies on the prompt processing of the acquired transport-related information for react-

ing to congestion, dangerous situations, and, in general, optimizing the circulation of people and goods. Integration, storage and analysis of huge data collections must be adapted to support ITS for providing solutions that can improve citizens' lifestyle and safety.

In order to address these challenges it is important to consider that big data introduce aspects to consider according to its properties described by the 5V's model [4]: Volume, Velocity, Variety, Veracity, Value.

Volume and velocity (i.e., continuous production of new data) have an important impact in the way data is collected, archived and continuously processed. Transport data are generated at high speed by arrays of sensors or multiple events produced by devices and transport media (buses, cars, bikes, trains, etc.). This data need to be processed in real-time, near real-time or in batch, or as streams. Important decisions must be made in order to use distributed storage support that can maintain these data collections and apply on them analysis cycles. Collected data, involved in transport scenarios, can be very heterogeneous in terms of formats and models (unstructured, semi-structured and structured) and content. Data variety imposes new requirements to data storage and database design that should dynamically adapt to the data format, in particular scaling up and down. ITS and associated applications aim at adding value to collected data. Adding value to big data depends on the events they represent and the type of processing operations applied for extracting such value (i.e., stochastic, probabilistic, regular or random). Adding value to data, given the degree of volume and variety, can require important computing, storage and memory resources. Value can be related to quality of big data (veracity) concerning (1) data consistency related to its associated statistical reliability; (2) data provenance and trust defined by data origin, collection and processing methods, including trusted infrastructure and facility.

Processing and managing big data, given the volume and veracity and given the greedy algorithms that are sometimes applied to it, for example, giving value and making it useful for applications, requires enabling infrastructures. Cloud architectures provide unlimited resources that can support big data management and exploitation. The essential characteristics of the cloud lie in on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services [5]. These characteristics make it possible to design and implement services to deal with big data management and exploitation using cloud resources to support applications such as ITS.

The objective of our work is to manage and aggregate cloud services for managing big data and assist decision making for transport systems. Thus this paper presents our approach for developing data storage, data cleaning and data integration services to make an efficient decision support system. Our services will implement algorithms and strategies that consume storage and computing resources of the cloud. For this reason, appropriate consumption models will guide their use.

The remainder of the paper is organized as follows. Section 2 describes work related to ours. Section 3 introduces our approach for managing transport big data on the cloud for supporting intelligent transport systems applications. Section 4 presents a

case study of the application that validates our approach. Finally, Section 5 concludes the paper and discusses future work.

2 Related work

2.1 Big data transport systems

This section focuses on big data transport projects, namely to optimize taxi usage, and on big data infrastructures and applications for transport data events.

Transdec [3] is a project to create a big data infrastructure adapted to transport. It is built on three tiers comparable to the MVC (Model, View, Controller) model for transport data. The presentation tier, based on GoogleTM Map, provides an interface to express queries and expose the result, the query interface provides standard queries for the presentation tier and a data tier is spatiotemporal database built with sensor data and traffic data. This work provides an interesting query system taking into account the dynamic nature of town data and providing time relevant results in real-time.

Urban insight [6] is a project studying European town planning. In Dublin they are working event detection through big data, in particular on an accident detection system using video stream for CCTV (Closed Circuit Television) and crowdsourcing. Using data analysis they detect anomalies in the traffic and identify if it is an accident or not. When there is an ambiguity they rely on crowdsourcing to get further information. The project RITA [7] in the United States is trying to identify new sources of data provided by connected infrastructure and connected vehicles. They work to propose more data sources usable for transport analysis. L. Jian and co [8] propose a service-oriented model to encompass the data heterogeneity of several Chinese towns. Each town maintains its data and a service that allows other towns to understand their data. These services are aggregated to provide a global data sharing service. These papers propose methodologies to acknowledge data veracity and integrate heterogeneous data into one query system. An interesting line to work on would be to produce predictions based on this data to build decision support systems.

N. J. Yuan and co [9], Y. Ge and co [10] and D. H. Lee and co [11] worked a transport project to help taxi companies optimize their taxi usage. They work on optimizing the odds of a client needing a taxi to meet an empty taxi, optimizing travel time from taxi to clients, based on historical data collected from running taxis. Using knowledge from experienced taxi drivers, they built a mapping of the odds of passenger presence at collection points and direct the taxis based on that map. These research works do not use real-time data thus making it complicated to make accurate predictions and react to unexpected events. They also use data limited to GPS and taxi usage, whereas other data sources could be accessed and used.

D. Talia [12] presents the strengths of using the cloud for big data analytics in particular from a scalability stand point. They propose the development of infrastructures, platforms and service dedicated to data analytics. J. Yu and co [13] propose a service oriented data mining infrastructure for big traffic data. They propose a full infrastructure with services such accident detection. For this purpose they produce a

large database with the collected data by individual companies. Individual services would have to duplicate the data to be able to use it. This makes for highly redundant data as the same data is stored by the centralized database, the application and probably the data producers. What is more, companies could be reluctant to giving away their data with no control for its use.

The state of the art reveals a limited use of predictions from big data analytics for transport-oriented systems. The heavy storage and processing infrastructures needed for big data and the current available data-oriented cloud services make possible the continuous access and processing of real time events to gain constant awareness, produce big data-based decision support systems, which can help take immediate informed actions. Cloud based big data infrastructure often concentrate around the massive scalability but don't propose a cheap method to simply aggregate big data services.

2.2 Big data analysis

H. V. Jagadish and co [4] propose a big data infrastructure based on five steps: data acquisition, data cleaning and information extraction, data integration and aggregation, big data analysis and data interpretation. X. Chen and co [14] use Hadoop-gis to get information on demographic composition and health from spatial data. J. lin and D. Ryaboy [15] present their experience on twitter to extract information from log information. They concluded that an efficient big data infrastructure is a balancing speed of development, ease of analysis, flexibility and scalability. Proposing a big data infrastructure on the cloud will make developing big data infrastructures more accessible to small businesses for several reasons: little initial investment, ease of development through Service-Oriented Architecture (SOA) and using services developed by specialist of each service.

Satish Narayana Srirama and co [16] demonstrated their cloud infrastructure for scientific analysis. Using Hadoop mapreduce, they classified the scientific algorithms according to how easy they could be adapted to mapreduce. Thus class 1 is when an algorithm can be executed with one mapreduce, class 2 is when the algorithm needs sequential mapreduce, class 3 is when each iteration of an algorithm executes one map reduce and class 4 is when each iteration needs multiple mapreduce.

Kurt Thearling [17] has put online a document introducing to the main families and technics for data mining. Whilst he claims the statistical technics are not data mining under the strictest of definitions, he included them since they are very used. He classified into two main families. The classical technics include statistical models very good for making predictions, nearest neighbour, clustering and generally technics visualizing data as space with as many dimensions as variable. The second is the Next Generation Techniques that include decision trees, neural networks and rules induction, they view data analysis as a series of test. There are also more advanced methods [18].

And finally, Ricardo [19] is a tool which proposes to integrate the R scripting language and Hadoop. The objective of this tool is to provide data analyst easy tools to use mapreduce. Ricardo provides an Application Programing Interface to R that con-

nects to a Hadoop cluster. It can convert R object into JaQL [20] queries to analyses the data. Whilst this technique has been proven successful with analytical technics like Latent-Factor Model or principal component analysis it showed less efficient than a straightforward mapreduce, on the other hand this tools greatly reduce the time of development.

2.3 Service oriented big data

Domenico Talia [12] proposes three levels of big data analytical service to the image of the three levels of services in cloud. The SaaS provides data mining algorithms and knowledge discovery tools. The PaaS provides a platform for the development of new data analytical services. The IaaS provides the low level tools to data mining. In the same way Zibin Zheng and co [21] have proposed a similar vision applied to analyzing logs.

H. Demirkan and D. Delen [22] proposes a service oriented decision support system using big data and the cloud. They do this by combining data from multiples databases into a single database then duplicate it to services.

Eric E. Schadt and co [23] demonstrate the efficiency that cloud computing could have for big data analytics, showing that analysis of 1 peta Byte of data in 350 minutes for 2040 dollars.

Zhenlong Li and co [24] proposed a service oriented architecture for geoscience data were they separate the modelling service for geoscience, the data services, processing service and the cloud infrastructure.

Several articles have demonstrated the strength of cloud and big data in particular for instancing large quantities of computing power [25], [26], [27].

2.4 Conclusion of the state of the art

These papers have shown that using big data for transport can provide very interesting applications. Big data analytics is a domain combining both old methods and new technology, that the data expert hasn't necessarily mastered. The use of the cloud for big data analytics has shown great results in both analytical speed but also cost and more importantly provides great elasticity.

On the other hand these papers have shown that big data analytics is viewed as a single service and not as a family of services responsible for the individual steps in the data management and analysis. Also data experts being general expert in their area, providing tools to ease the use of the new technology is important. By proposing a service oriented architecture for big data analysis, we hope to propose easy to develops tools for transport.

3 Big data on the cloud

In cloud computing everything is viewed as a service (XaaS). As a consequence cloud software (SaaS) is built as an aggregate of services exploiting services available on

the cloud infrastructure (IaaS). In this spirit, we build a big data architecture where individual services manage the treatment level of big data. This also means that the companies wanting a big data infrastructure will be able to simply build it from an aggregation of services proposed by specialized companies.

Following the 5 step in big data proposed by H. V. Jagadish and co[4], we will design 5 types services (**Fig. 1**) for both historical data and real time data. These data services are: data acquisition services, data cleaning and extraction service, data integration and aggregation, data analytical services, and decision support services. The next paragraph will go into more detail for each service.

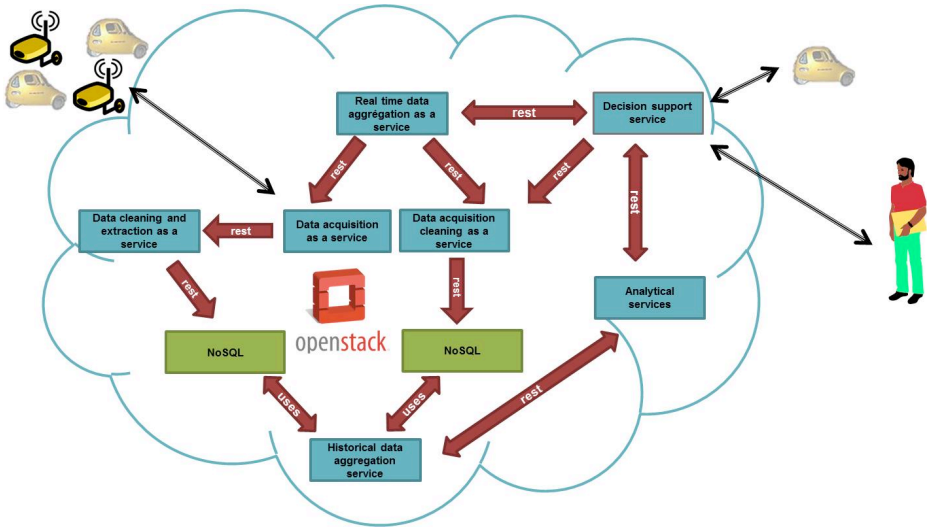

Fig. 1. Big Data architecture

3.1 Data acquisition service

The first step of a big data infrastructure is well collecting the big data. This is basically hardware and infrastructure services that transfer, to NoSQL data stores adapted to the format of the data, the data acquired by the vehicles, users, and sensors deployed in cities (e.g. roads, streets, public spaces). This is done by companies and entities such as town or companies managing certain public spaces, who have data collecting facilities. These companies propose and sell their data on the cloud in our case the university Openstack infrastructure [28]. Using NOSQL storage like MongoDB [29], these companies will have a highly scalable and sharable data store. Also the sharding capability of these data stores offers high horizontal scalability but also faster analysis through MapReduce and data availability.

3.2 Information extraction and cleaning service

The next step is cleaning and data extraction. This consists of both extracting the information from unstructured data and cleaning the data. This could be done by the company producing the data or an independent company depending on the level of structuration of the data. Highly structured data would likely be cleaned by the company producing the data as they understand best its production and thus know how best to clean it up. For highly unstructured data like sound or video data, highly specialized expert would be needed to extract the information.

This would be used to pot outliers in the data. Using MapReduce, the company acquiring the data or the company contracted to do it would perform statistical analysis to identify for example outliers in the data. This is important as, for example, a malfunction in a sensor loop could either ignore passing traffic or register non-existing traffic. Cleaning these events is important since inaccurate data produced by a dodgy sensor can break a model.

3.3 Integration and aggregation services

The objective of big data analytics is to use the large volume of data to extract new knowledge by searching, for example, for patterns in the data. This often has a consequence of data coming from a wide variety of sources. This means the data has to be aggregated into a usable format for the analytics tools to use. This service proposes services for real-time data aggregation and historical data aggregation.

The real-time data aggregation service gets the data from the individual data stores real-time data services and proposes a formatted file with the data from all the data acquiring service simply by fusing together the data provided by the real-time data acquisition services. Thus we aggregate data from the city, state of recharging stations, having location of people based on the time stamp or the GPS location.

The historical data aggregation will have to find a way to do similar action but with the data stores. The problem is that having data on several separate data stores is not a usable format. Importing all the data into a new huge data store would be redundant on already existing resources making this service potentially excessively expensive and as for temporary stores would be long to build when having to import terabytes of data as well as being expensive on network cost as well as time consuming. To solve this problem, this service will propose a query interface for simple querying and processing service to process the data mass by converting a form simple programing language into UNQL queries [30] to collect and pre-process the data before being integrated into a model.

3.4 Big data analytical and decision support services

The whole point of big data is to identify and extract information from the mass of data. Predictive tools can be developed to anticipate the future. The role of this service is to provide a computer model of the historical data. It also provides the algorithm applied to the individual pieces of data. Thus using the model provided by the

analytical service and the algorithm applied to the real-time data we can approach similar situations and act accordingly.

The decision support service composes several services. On the strategic level and using the model and the algorithm proposed by the big data analytical services, the decision support service provides an interface exposing the data situation in real-time but also predictions on events. For example, regularly observing an increase in the population in one place and traffic jams 30 minutes later we can deduct cause and effect and intervene in future situations so the taxis avoid and evacuate that area.

This service also generates data on the decision taken by the strategists to build more elaborate model including the consequence of this decision to then provide better decision support. On the vehicle level, services will provide advice to the vehicle for optimal economic driving based on the driving conditions. It also provides a database were the information on the dangers of the road is stored.

4 Managing transport big data in smart cities

Consider the scenario where a taxi company needs to embed decision support in electric vehicles, to help their global optimal management. The company uses electric vehicles that implement a decision cycle to reach their destination while ensuring optimal recharging, through mobile recharging units. The decision making cycle aims at ensuring vehicles availability both temporally and spatially; and service continuity by avoiding congestion areas, accidents and other exceptional events. The taxis and mobile devices of users are equipped with video camera and location trackers that can emit the location of the taxis and people. For this purpose, we need data on the position of the vehicles and their energies levels, have a mechanism to communicate unexpected events and have usage and location of the mobile recharging station.

Fig. 2 presents the general architecture of the transport data store as a service that we propose. It adopts a polyglot persistence [31] approach that combines several NoSQL systems for providing a storage support. Profiting from the cluster-oriented architecture of these systems our store uses a multi-cloud cluster based storage layer.

Our service extends an UnQL [30] layer with data processing operators including joins and filters for storing and retrieving data in an homogeneous way.

Furthermore, it exploits the sharding strategies of the NoSQL [32] systems for distributing and duplicating data, and ensuring availability. Shards are organized according to ranges of values of given attributes, or to hash functions and tags related to geographic zones. This induces request balancing and ensures better performance when data must be inserted and retrieved.

The service exploits also the persistence supports of clients (disk and cache) installed in mobile devices in order to distribute data processing and ensure data availability. For example, in our transport scenario, the service uses storage provided by devices used by taxis and users to process and manage data necessary for ITS services described in the previous section. In this way it avoids data transfer that can be penalizing in terms of response time and economic cost for accessing 3G or 4G networks.

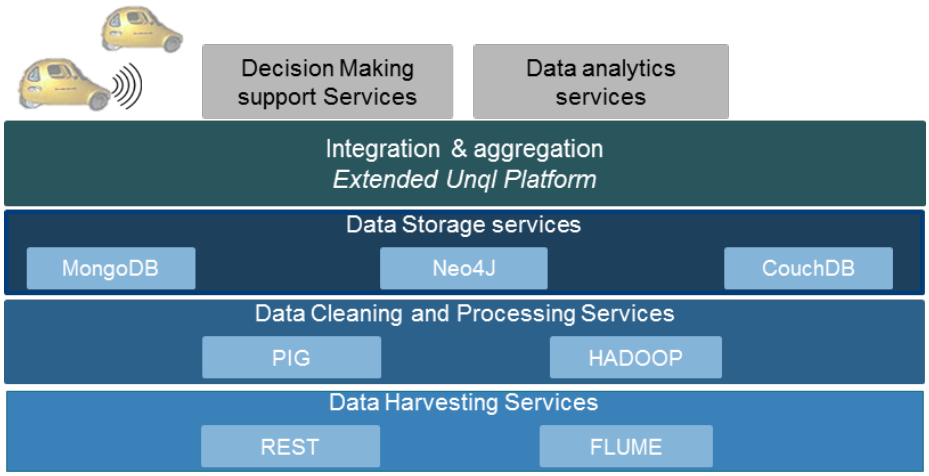

Fig. 2. Big data services

Our data store service provides a global access to clusters providing NoSQL and relational support, and enables applications designers to configure their resources provision and non-functional properties according to given requirements and cloud subscriptions. An application defines the data structures that must persist in the UML eclipse plugin and then the tool Model2Roo (https://code.google.com/p/model2roo/) generates the necessary bindings to interact with different NoSQL stores. The application designer according to a profiling phase executed using the QDB benchmark (https://github.com/qdb-io) chooses the NoSQL stores.

4.1 Designing transport data collections

The data collections “Evènement routier temps reel”, “Etat du trafic temps reel”, “Borne Criter”, “Tronçon Web Criter”, “Trafic historique”, “plan Lyon”, “Aménagement cyclable”, “Caméra Web Criter” and “Station Velo'v”, provided by the project Grand Lyon [33], are sought and stored by our service in order to be able to correlate collected data with data describing the city and its infrastructures (parks, roads, commercial zones, river). This data is highly heterogeneous in format, information and update rates. There are images in JPG, JSON, XML, and PDF formats. The data is also updated at varying rates going from yearly updates to real-time data passing by daily and minutely updates. GPS and location data in devices and vehicles are seen by our service as continuous data that can be correlated to other collected data useful for performing some decision making requests, such as which is the closest taxi (considering distance and time) to a client?, according to traffic and taxi-energy level, which are the possible destinations it can accept? Data are sharded by our service to perform this type of requests that require computing resources. Our service uses a MongoDB cluster to store these data.

Data stemming from social networks particularly Twitter and Waze of taxi users are collected and stored in NeO4J. This collection provides a real-time view of the traffic, road and zones status and events. Data are sharded thanks to our storage service locally on mobile devices and on NeO4J instances deployed in the cloud.

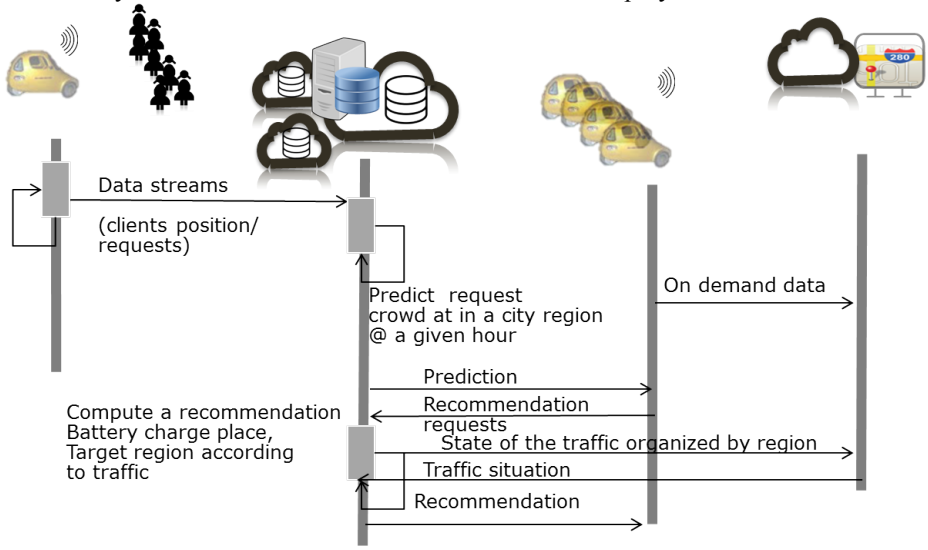

Fig. 3. UML sequence diagram of the decision making process.

4.2 Making global transport decisions

We conducted an experimental validation of our transport data store as a service for the scenario we described in Section 3. The experiment implements a polyglot multi-database that contains data collected from the French city Lyon. These data are retrieved by applications and infrastructure integrated by the project Grand Lyon. We then implement some important operations of the decision making cycle of the scenario (**Fig. 3**). The decision making cycle consists in:

Collecting data streams from taxis and users that are mobile data providers evolving in Lyon and feeding the data store service.

We focus particularly in three operations that use the transport data storage as a service approach, which are dissemination of events, optimization of energy recharging and scaling taxi provision of exceptional situations. We describe this use cases hereafter.

Disseminating events

The applications deployed in taxis and users can be used for disseminating exceptional situation events, for example, unexpected dangers (**Fig. 4Erreur ! Source du renvoi introuvable.**). In our scenario a pedestrian is about to cross the road. “Vehicle A” is arriving in the same place but has no line of sight. “Vehicle B” in the area

"sees" the pedestrian. The data sent from "Vehicle B" is then sent to data collection services and stored a NoSQL database. As the vehicle comes in the area, the vehicle computer will make HTTP query to a cloud which will access the data in the NoSQL database.

Depending on the nature of the danger the data store will make decisions on how long to keep that information and during which period it will re-execute the dissemination to taxis getting close to the zone.

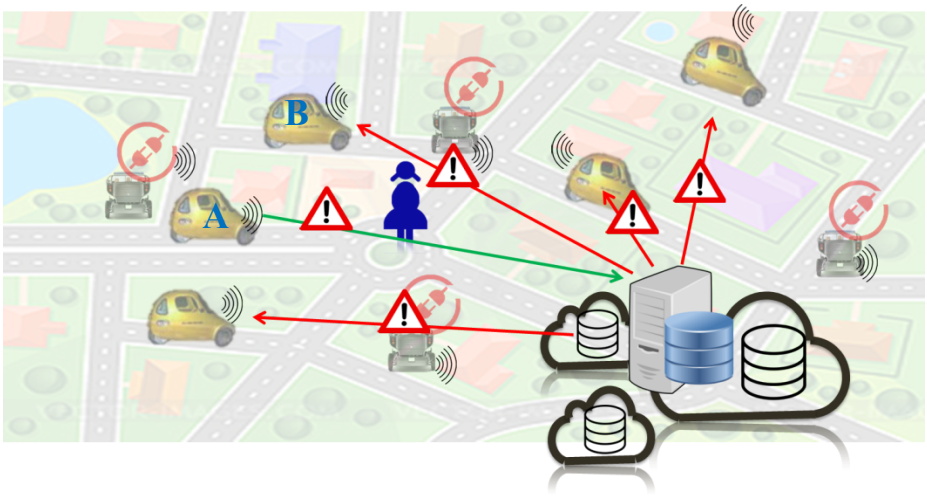

Fig. 4. Disseminating exceptional events

Optimizing battery recharging

Part of the objective of taxi companies is use only electric vehicles. Unfortunately the lack of data makes it complicated to make good strategic solutions on the locations of the recharging stations that are also mobile (**Fig. 5**).

Using UnQL queries from the data integration service, the historical data stored in the NoSQL databases is periodically analyzed to extract information to build classification models or regression models for the real time data. Using this model and real-time data the system will make predictions on the location of taxi users and the traffic. As decision makers take decisions, this information is feed into the model to help the decision maker optimize the number of operational taxi, the location of these taxis and the location of the recharging stations by exposing the consequence previous similar decisions had.

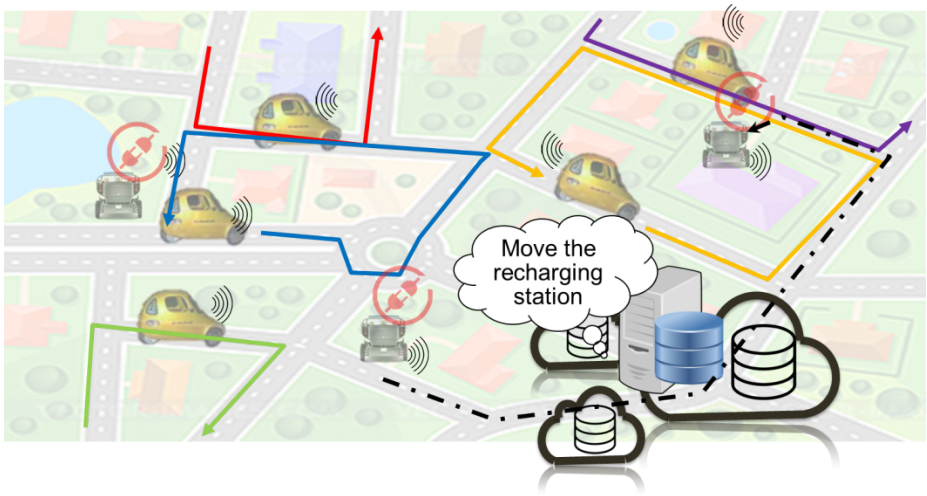

Fig. 5. Optimizing battery recharging

The next section will present the state of development of the data collection service and the data storage services.

5 Implementation and testing

In this section we present the data acquisition service and the information extraction and cleaning service. We also looked for the ideal sharding strategy for the MongoDB.

5.1 Data acquisition service

We have implemented and tested the data acquisition service. This services uses NodeJS module to acquire the city data from the Grand Lyon [33] but also from Twitter and from Bing search engine using REST requests. Still using REST requests these services will post the data onto a Mongodb database container to store as historical data. The service provides functions to access data via REST either with the key to the data store when wanting to query or analyses the historical data or the latest file acquired when using the real-time data service. The data is stored under XML, JSON or the original image file.

5.2 Information extraction and cleaning service

So far 43 649 Kb of data has been stored into individual mongoDB database per data acquisition service built on 1 config server, 1 router, and 3 replicating shards to insure data persistence.

A comparison between a hashing distribution and a ranged distribution has been performed. It reveals significant differences with faster inserting and requesting data ranged over hashed for the two size of data set (**Fig. 6**).

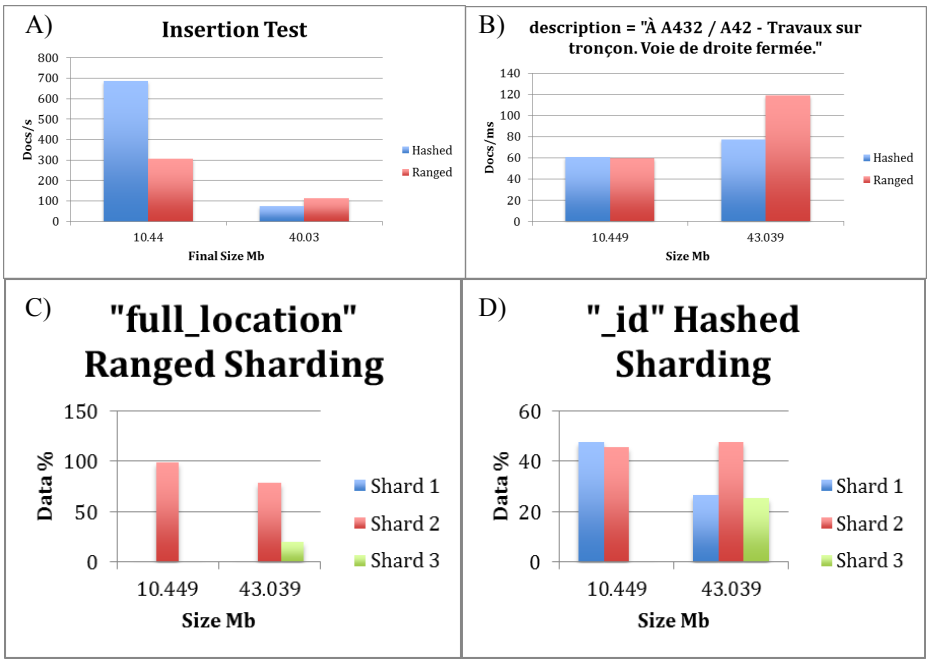

Fig. 6. Strategy comparison

We observer generally the data distribution between the shards is better for hashed IDs over ranged IDs (**Fig. 6** C and D). This induces better performances (**Fig. 6** A and B) for the hashed data since it allows a better distributed computing, where each individual shard has less data to analyses. The ranged show improved results when the data set becomes large, largely because, provided the query is related to the ordered variable, one does not have to analyses all the data longer.

Thus using ranged ids based on the coordinates of the data will be at least able to optimize location based queries.

6 Conclusion and future work

This paper proposes a transport data store as a service that implements a distributed storage approach. Our approach uses NoSQL systems deployed in a multi-cloud setting and makes sharding decisions for ensuring data availability.

The transport data store service is validated in a scalable and adaptable ITS for electric vehicles using big data analytics on the cloud. This provides a global view of

current status of town transport, helps making accurate strategic decisions, and insures maximum security to the vehicles and their occupants.

For the time being our storage service concentrates in improving design issues with respect to NoSQL support. We are currently measuring performance with respect to different sizes of data collections. We have noticed that NoSQL provides reasonable response times once an indexing phase has been completed. We are willing to study the use of indexing criteria and provide strategies for dealing with continuous data. We will also be developing the other services for our big data architecture. These issues concern our future work.

7 Acknowledgement

We thank the Région Rhône-Alpes who finances the thesis work of Gavin Kemp by means of the ARC 7 programme (http://www.arc7-territoires-mobilites.rhonealpes.fr/), as well as the competitiveness cluster LUTB Transport & Mobility Systems, in particularly Mr. Pascal Nief, Mr. Timothée David and Mr. Philippe Gache for putting us in contact with local companies and projects to gather use case scenarios for our work.

8 References

1. V. Gulisano, R. Jiménez-Peris, M. Patiño-Marťnez, C. Soriente, and P. Valduriez, "StreamCloud: An elastic and scalable data streaming system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, pp. 2351–2365, 2012.
2. F. Lecue, S. Tallevi-Diotallevi, J. Hayes, R. Tucker, V. Bicer, M. L. Sbodio, and P. Tommasi, "STAR-CITY," in *Proceedings of the 19th international conference on Intelligent User Interfaces - IUI '14*, 2014, pp. 179–188.
3. U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, "TransDec:A Spatiotemporal Query Processing Framework for Transportation Systems," *IEEE*, pp. 1197–1200, 2010.
4. H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, *Big Data and Its Technical Challenges*, vol. 57, no. 7. 2014.
5. P. M. and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," 2008.
6. A. Artikis, M. Weidlich, A. Gal, V. Kalogeraki, and D. Gunopulos, "Self-Adaptive Event Recognition for Intelligent Transport Management," pp. 319–325, 2013.
7. D. Thompson, G. McHale, and R. Butler, "RITA," 2014. [Online]. Available: http://www.its.dot.gov/data_capture/data_capture.htm.
8. L. Jian, J. Yuanhua, S. Zhiqiang, and Z. Xiaodong, "Improved Design of Communication Platform of Distributed Traffic Information Systems Based on SOA," in *2008 International Symposium on Information Science and Engineering*, 2008, vol. 2, pp. 124–128.
9. N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, pp. 2390–2403, 2013.

10. Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, “An energy-efficient mobile recommender system,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’10*, 2010, p. 899.
11. D.-H. Lee, H. Wang, R. Cheu, and S. Teo, “Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions,” *Transp. Res. Rec.*, vol. 1882, pp. 193–200, 2004.
12. D. Talia, “Clouds for scalable big data analytics,” *Computer (Long. Beach. Calif).*, vol. 46, no. 5, pp. 98–101, 2013.
13. J. Yu, F. Jiang, and T. Zhu, “RTIC-C: A Big Data System for Massive Traffic Information Mining,” in *2013 International Conference on Cloud Computing and Big Data*, 2013, pp. 395–402.
14. X. Chen, H. Vo, A. Aji, and F. Wang, “High performance integrated spatial big data analytics,” in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial ’14*, 2014, pp. 11–14.
15. J. Lin and D. Ryaboy, “Scaling big data mining infrastructure : The twitter Experience,” *ACM SIGKDD Explor. Newsl.*, vol. 14, no. 2, p. 6, Apr. 2013.
16. S. Tavakoli and A. Mousavi, “Adopting user interacted mobile node data to the Flexible Data Input Layer Architecture,” in *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2008, pp. 533–538.
17. A. Berson, S. Smith, and K. Thearling, “An Overview of Data Mining Techniques,” ... *Data Min. Appl. CRM*, pp. 1–49, 2004.
18. W. Yan, U. Brahmakshatriya, Y. Xue, M. Gilder, and B. Wise, “p-PIC: Parallel power iteration clustering for big data,” *J. Parallel Distrib. Comput.*, vol. 73, no. 3, pp. 352–359, Mar. 2013.
19. S. Das, P. J. Haas, and K. S. Beyer, “Ricardo : Integrating R and Hadoop Categories and Subject Descriptors,” pp. 987–998, 2000.
20. S. Lim, ““Scalable SQL and NoSQL Data Stores,” *Statistics (Ber).*, 2008.
21. Z. Zheng, J. Zhu, and M. R. Lyu, “Service-generated big data and big data-as-a-service: An overview,” *Proc. - 2013 IEEE Int. Congr. Big Data, BigData 2013*, pp. 403–410, 2013.
22. H. Demirkan and D. Delen, “Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud,” *Decis. Support Syst.*, vol. 55, no. 1, pp. 412–421, 2013.
23. E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P. Nolan, “Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology.,” *Nat. Rev. Genet.*, vol. 12, no. 3, p. 224, Mar. 2011.
24. Z. Li, C. Yang, B. Jin, M. Yu, K. Liu, M. Sun, and M. Zhan, “Enabling big geoscience data analytics with a cloud-based, MapReduce-enabled and service-oriented workflow framework.,” *PLoS One*, vol. 10, no. 3, p. e0116781, Jan. 2015.
25. V. Abramova and J. Bernardino, “NoSQL databases: a step to database scalability in web environment,” *Proc. Int. C* Conf. Comput. Sci. Softw. Eng. - C3S2E ’13*, no. July, pp. 14–22, 2013.
26. S. Hipgrave, “Smarter fraud investigations with big data analytics,” *Netw. Secur.*, vol. 2013, no. 12, pp. 7–9, Dec. 2013.
27. B. K. Tannahill and M. Jamshidi, “System of Systems and Big Data analytics – Bridging the gap,” *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 2–15, Jan. 2014.
28. Open, “Openstack,” 2015. [Online]. Available: http://www.openstack.org/.
29. P. J. Sadalage and M. Fowler, *NoSQL Distilled*. 2012.

30. P. Buneman, M. Fernandez, and D. Suciu, “UnQL: a query language and algebra for semistructured data based on structural recursion,” *VLDB J.*, vol. 9, no. 1, p. 76, Mar. 2000.
31. C. Nance, T. Losser, R. Iype, and G. Harmon, “NoSQL vs RDBMS - Why there is room for both.,” *Proc. South. Assoc. Inf. Syst. Conf.*, pp. 111–116, 2013.
32. R. Cattell, “Scalable SQL and NoSQL data stores,” *ACM SIGMOD Rec.*, vol. 39, no. 4, p. 12, May 2011.
33. GrandLyon, “Smart Data,” 2015. [Online]. Available: http://data.grandlyon.com/.