Cloud big data application for transport

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

Cloud big data application for transport

Gavin Kemp*

LIRIS, CNRS,
Université Lyon 1,
UMR5202, Bd du 11 Novembre 1918,
Villeurbanne, F69621, France
Email: gavin.kemp@liris.cnrs.fr
*Corresponding author

Genoveva Vargas-Solar

Grenoble Institute of Technology,
CNRS, LIG-LAFMIA,
681 rue de la Passerelle,
Saint Martin d'Hères, F38401, France
Email: genoveva.vargas@imag.fr

Catarina Ferreira Da Silva and
Parisa Ghodous

LIRIS, CNRS,
Université Lyon 1,
UMR5202, Bd du 11 Novembre 1918,
Villeurbanne, F69621, France
Email: catarina.ferreira@univ-lyon1.fr
Email: parisa.ghodous@univ-lyon1.fr

Christine Collet

Grenoble Institute of Technology,
CNRS, LIG,
681 rue de la Passerelle,
Saint Martin d'Hères, F38401, France
Email: Christine.Collet@grenoble-inp.fr

Pedro Pablo Lopez Amalya

LIRIS, CNRS,
Université Lyon 1,
UMR5202, Bd du 11 Novembre 1918,
Villeurbanne, F69621, France
Email: pg.lopezamaya@gmail.com

Abstract: This paper presents a cloud service oriented approach for managing and analysing big data required by transport applications. Big data analytics brings new insights and useful correlations of large data collections providing undiscovered knowledge. Applying it to transport systems brings better understanding to the transport networks revealing unexpected choking points in cities. This facility is still largely inaccessible to small companies due to their limited access to computational resources. A cloud-oriented architecture opens new perspectives for providing efficient and personalised big data management and analytics services to (small) companies.

Keywords: intelligent transport system; big data; cloud services; NoSQL.

Reference to this paper should be made as follows: Kemp, G., Vargas-Solar, G., Da Silva, C.F., Ghodous, P., Collet, C. and Amalya, P.P.L. (xxxx) 'Cloud big data application for transport', *Int. J. Agile Systems and Management*, Vol. X, No. Y, pp.000–000.

Biographical notes: Gavin Kemp is PhD student in the Computer Science Department of the Claude Bernard Lyon 1 University, France, and joined the Service Oriented Computing team of the LIRIS Lab in 2014. His current research interests include cloud computing services, big data and intelligent transport systems.

Genoveva Vargas-Solar is a Senior Scientist of the French Council of Scientific Research (CNRS) and Deputy Director of the Franco-Mexican Laboratory of Informatics and Automatic Control (LAFMIA, UMI 3175). She is also a member of the Informatics Laboratory of Grenoble, France and invited research fellow of the Data and Knowledge Management Group at Universidad de las Américas Puebla. Her research contributes to the construction of service-based database management systems. Her objective is to design data management services guided by Service Level Agreements (SLA). She proposes methodologies, algorithms and tools for integrating, deploying and executing a service composition for programming data management functions. The results of her research are validated in the context of grids, embedded systems and clouds.

Catarina Ferreira da Silva is an Associate Professor in the Computer Science Department at the University Institute of Technology of the Claude Bernard Lyon 1 University, France, and joined the Service Oriented Computing team of the LIRIS lab in 2012. Previously, she worked at the Centre for Informatics and Systems of the University of Coimbra, Portugal. She obtained her PhD thesis in computer science (2007) from the University of Lyon 1. Her current research interests include cloud computing services, linked data and semantic web.

Parisa Ghodous is currently a Full Professor in Computer Science Department at University of Lyon I. She is the Head of Cloud Computing Theme of LIRIS UMR 5205 (Laboratory of Computer Graphics, Images and Information Systems). Her research expertise is in the following areas: cloud computing, interoperability, web semantic, web services, collaborative modelling, product data exchange and modelling and standards. She is an editorial boards of *CERA*, *ICAE* and *IJAM* journals and in the committees of many relevant international associations such as concurrent engineering, ISPE, interoperability.

Christine Collet is currently a Full Professor of Computer Science at the Grenoble Institute of Technology, Grenoble, France. She is the Head of the Database Management Group (HADAS) of the Grenoble Informatics Laboratory – UMR 5217. Her research domain concerns databases and their evolution in terms of data models, languages and architectures. More precisely, she contributed or still contributes to research activities on: object and active database management systems; multi-scale distributed and heterogeneous data(base) management systems; adaptive optimisation and evaluation of hybrid queries (on large datasets and streams); distributed composite event management ; data (service) mediation and integration. She is the author or co-author of around 100 publications: books, chapters and articles in national and international journals and conferences (see the DBLP and Google Scholar servers). She directed more than 20 PhD theses. She is currently the President of the EDBT association.

Pedro Pablo Lopez Amaya was an interne to the LIRIS in 2015 to finish up his MsC. He worked with us on the development of a prototype

This paper is a revised and expanded version of a paper entitled 'Towards cloud big data services for intelligent transport systems' presented at [name of conference], Delft University of Technology, Netherland, 20–22 July 2015.

Comment [t1]: Author: Please complete where highlighted.

1 Introduction

In later years, we have been observing an explosion in available data (http://www.csc.com/insights/flxwd/78931-big_data_universe_beginning_to_explode), due to the accumulation over the years of data and to the continuous production of streams by different kinds of providers like sensors, smart devices and smart cities infrastructure. Also whilst computing power has also increased, it has not increased at the same rate as data. Even if Moore's law seems to be showing its limit (Schaller, 1997) to process very large data collections, cloud architectures promise to deliver unlimited resources dealing with challenges introduced by those large data collections. This has opened up for a new research to manage to analyse large data collections (i.e., big data). In order to address big data storage and analytics it is important to consider Big Data properties described by the 5V's model (Jagadish et al., 2014): 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.). These data need to be processed in real-time, recurrent 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. Intelligent transport systems (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 can be challenging, given volume and veracity and the greedy algorithms that are sometimes applied to it. Furthermore, giving value to data 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 computing lie in on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services (Mell and Grance, 2008). 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.

On the other hand, clouds provide a ready to use execution environments with the required physical resources and platforms to be used for big data management, and elasticity management mechanisms to adapt the provision of resources at runtime. Thus, this paper introduces the implementation of data collection and storage strategies provided as services to manage big data collections with high degree of variety and velocity. These strategies are implemented by a multi-holder service oriented big data infrastructure used by ITS.

The contribution of this work is of showing the effects of the various strategies involved in sharding. We show the effect of hashed sharding on various keys and ranged sharding. The remainder of the paper is organised as follows. Section 2 describes work related to transport big data, big data analytics and service oriented big data. Section 3 presents the architecture and the individual services for this service oriented architecture. Section 4 presents the data collection and storage services needed in the proposed architecture. 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 optimise taxi usage, and on big data infrastructures and applications for transport data events.

TransDec (Demiryurek et al., 2010) is a project to create a big data infrastructure adapted to transport. It is built on three tiers 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 that 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 (Artikis et al., 2013) 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 Closed Circuit Television (CCTV) 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 (Thompson et al., 2014) in the USA 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. Jian et al. (2008) 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.

Yuan et al. (2013), Ge et al. (2010) and Lee et al. (2004) worked a transport project to help taxi companies optimise their taxi usage. They work on optimising the odds of a client needing a taxi to meet an empty taxi, optimising 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.

Talia (2013) 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. Yu et al. (2013) propose a service oriented data mining infrastructure for big traffic data. They propose a full infrastructure with services helping 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 centralised 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 enable 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 architectures often concentrate around the massive scalability but do not propose a method to simply aggregate big data services.

2.2 Big data analytics

Jagadish et al. (2014) 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. Chen et al. (2014) use Hadoop-GIS to get information on demographic composition and health from spatial data. Lin and Ryaboy

(2013) present their experience on Twitter to extract information from logs. They concluded that an efficient big data infrastructure is a balancing speed of development, ease of analysis, flexibility and scalability.

Tavakoli and Mousavi (2008) demonstrated their cloud infrastructure for scientific analysis. Using Hadoop map-reduce, they classified the scientific algorithms according to how easy they could be adapted to map-reduce. Thus, class 1 is when an algorithm can be executed with one map-reduce; class 2 is when the algorithm needs sequential map-reduce; class 3 is when each iteration of an algorithm executes one map reduce; and class 4 is when each iteration needs multiple map-reduce.

Thearling (Berson et al., 2004) has put online a document introducing to the main families and techniques for data mining. Whilst he claims the statistical techniques are not data mining under the strictest of definitions, he included them since they are widely used. He classified them into two main families: classical and next generation. The classical techniques include statistical models very good for making predictions, computing the nearest neighbour, clustering; and general techniques for visualising data within n-dimensional spaces with as many dimensions as variables. The Next Generation Techniques include decision trees, neural networks and rules induction, they view data analysis as a series of tests. There are also advanced methods proposed in Yan et al. (2013).

And finally, Ricardo (Das et al., 2000) is a tool which proposes to integrate the R scripting language and Hadoop. The objective of this tool is to provide the data analyst easy tools that use map-reduce. Ricardo provides an Application Programming Interface (API) to R that connects to a Hadoop cluster. It can convert R objects into JaQL (Lim, 2008) queries to analyse the data. Whilst this technique has been proven successful with analytical techniques like latent-factor model or principal component analysis it showed less efficient than a straight forward map-reduce, on the other hand this tools greatly reduce the time of development.

2.3 Service oriented big data

Talia (2013) proposes to perform analysis tasks on cloud and big data. They propose a three level of big data analytical service to the image of the three service levels in the 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, Zheng et al. (2013) have proposed a similar vision applied to analysing logs.

Demirkan and Delen (2013) propose 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 produce duplicates for the services using the data.

Schadt et al. (2011) demonstrate the efficiency that cloud computing could have for big data analytics, showing that analysis of 1 Petabyte of data in 350 minutes for 2040 dollars.

Li et al. (2015) propose a service-oriented architecture for geoscience data where they separate the modelling service for geoscience, the data services, the 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 (Abramova and Bernardino, 2013; Hipgrave, 2013; Tannahill and Jamshidi, 2014).

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 classic well tested methods and new technology that the data expert does not necessarily master. The use of the cloud for big data analytics has shown great results in both analytical speed but also in monetary cost and more importantly it provides great elasticity and unlimited resources necessary for costly analytic algorithms.

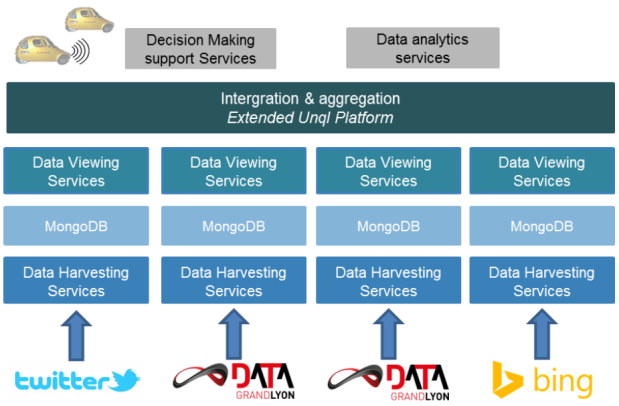
On the other hand, these papers have shown that big data analytics is viewed as a single service and not as a family of services responsible for the individual steps in the data management and analysis. Also data experts being general expert in their area, providing tools to ease the use of the new technology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

3 Managing transport big data in smart cities

Consider the scenario where a taxi company needs to embed decision support in vehicles, to help their global optimal management. The company uses vehicles that implement a decision-making 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. 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. More details about this scenario can be found in Kemp et al. (2015).

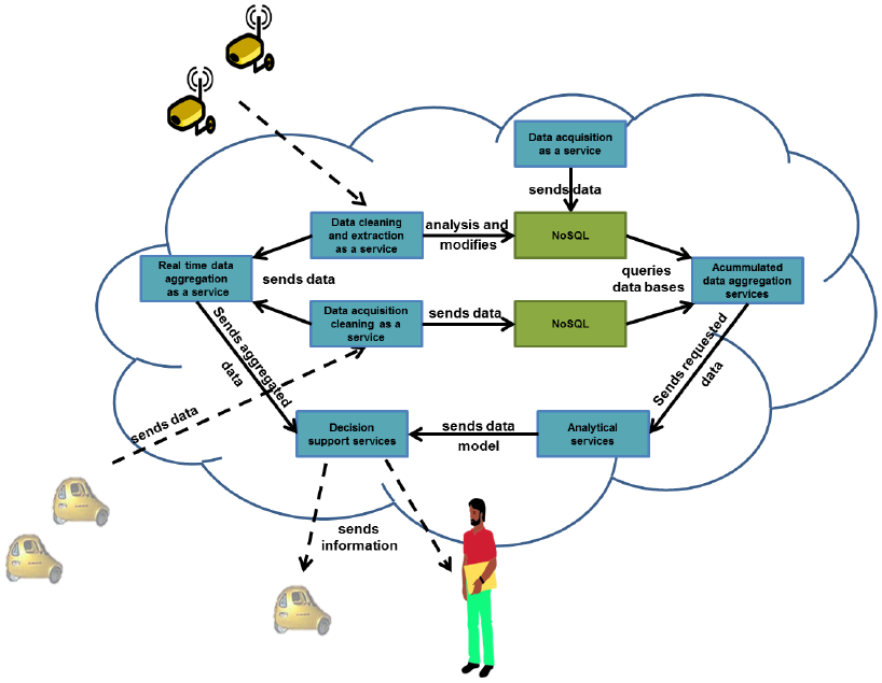
Figure 1 shows the services that these applications rely on. They concern data acquisition, cleaning and information extraction; and big data analysis, integration and aggregation, and decision-making support services.

Figure 1 Big data services (see online version for colours)



In cloud computing, everything is viewed as a service (XaaS). In this spirit, we are building a big data infrastructure (Figure 2), specialised in three types of complex operations: collection, storage and analytics. Services implementing different algorithms and strategies for implementing these operations can be used for the personalisation of an ITS application. Thus, companies using our big data services will be able to simply build their applications by aggregating services according to their requirements.

Figure 2 Our big data architecture (see online version for colours)



3.1 *Data acquisition services*

The first step of a big data infrastructure is collecting the data. This is basically hardware and infrastructure services that produce data consumed by services and archive them in different NoSQL data stores according to their characteristics. Data are acquired by the vehicles, users, and sensors deployed in cities (e.g., roads, streets, public spaces) according to different strategies such as collective explicit and implicit crowdsourcing, push continuous data production performed by sensors. This is done by companies and entities such as town or companies managing certain public spaces, who have data collecting facilities.

3.2 Information extraction services

Raw data from sensors cannot be used directly to perform analytics processes, since it can be incomplete, it can contain noise, and there is little knowledge about its structure and the conditions in which it has been produced. To be able to exploit the data, the analyst needs information about the data structure (variable, graph or table …) and about its content like the distribution of values of every attribute, possible dependencies among attributes, possible missing values or erroneous ones, provenance. Information extraction and cleaning services are tools to extract comprehensive views of the data. These tools include mostly statistical exploratory methods that are very good to provide a comprehensive view of the data such principal component of analytics (Tannahill and Jamshidi, 2014). Views produced by these information extraction services can be used by data scientists to determine how to clean them and produce collections that can be used to perform analytics.

3.3 Integration and aggregation services

The objective of big data analytics is to extract new knowledge by searching for example for patterns within proper and representative data collections. This means heterogeneous data stemming from different providers has to be integrated into a usable format for the analytics tools to use. Integration and aggregation services propose algorithms for real-time data aggregation and historical data aggregation.

The real-time data aggregation service gets the data produced by the real-time data acquisition services and generates a database that integrates data from all the data acquisition services. Thus, an integrated database could aggregate data from the city, states of recharging stations having, location of people, for example, based on their time stamp.

The historical data aggregation service has to find a way to do this action. Importing all the data into a new huge data store would be redundant on already existing resources making this service 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.

3.4 Big data analytics and decision support services

The whole point of big data is to identify and extract meaningful information. Predictive tools can be developed to anticipate the future. The role of the big data analytics and decision support services in our infrastructure is to provide data analytics solutions. These solutions can be used for predicting events or for decision-making tasks by composing several services. For example, regularly observing an increase in the population in one place and traffic jams 30 minutes later we can deduce cause and effect situations and intervene in future situations so the taxis avoid and evacuate that area. Data on decisions made by strategists are stored to be used in future tasks. For example, provide advice to the vehicle for optimal economic driving based on the driving conditions and on previous recommendations given in similar situations.

These services will expose data under the form of view as defined as follows: view (Genoveva and Alexandrov, 2016) is a document that provides a description of every family of attributes of a raw document collection. The objective of these views is to provide information to the developer on the information in a database under the form of metadata coming from basic statistics, legal information and practical information. These views can be generated from raw data and be analysed data by other service to generate other views.

Based on the literature, we identified two major types of big data analytics algorithms: descriptive analytics used to build a comprehensive view of the data and predictive analytics used to predict a value or a class for a sample. These services will exploit data for the NoSQL database services and data from other views. The initial algorithms focus on the descriptive analytics for two reasons: one because these algorithms give an understanding of the data available and two because predictive algorithms can often be stacked on top of a descriptive analysis for easy development.

The next section presents the implementation and the architecture of the data collection services and the data storage strategies used for storing data.

4 Implementation and results

This section presents implementation issues of the collection services used for developing an experimental testbed of our approach. We programmed REST (https://dev.twitter.com/rest/public) services that offer streaming functions to return public statuses that match one or more filters about traffic. The filters used for our application are ‘track’ and ‘location’. Collected data are archived in NoSQL stores using different data sharding (i.e., horizontal fragmentation) techniques that ensure data availability.

4.1 Data collection service

The general architecture of our data collection service is divided into three layers (see Figure 3): the `Software Development Kit (SDK)`, `RetrievingStoring API` and `Collector` and `Database Proxy`. The SDK layer communicates with other layers using the HTTP requests and responses. The server is deployed on the layer exporting the interface `RetrievingStoring API`. The data collected and returned by the `Collector` are transformed according to the NoSQL data model. To store the documents into the NoSQL database we use the `Database Proxy`. In this module, we implemented all the Create, Read, Update and Delete (CRUD) operations. Based on this architecture we implemented the Twitter REST service (Figure 4) for collecting posts about traffic. We built it as a RESTful Web service following the parameters shown in Table 1. We used MongoDB NoSQL database to store the retrieved data according to different sharding strategies for organising data and ensuring availability.

Figure 3 Data collection, three layer architecture (see online version for colours)

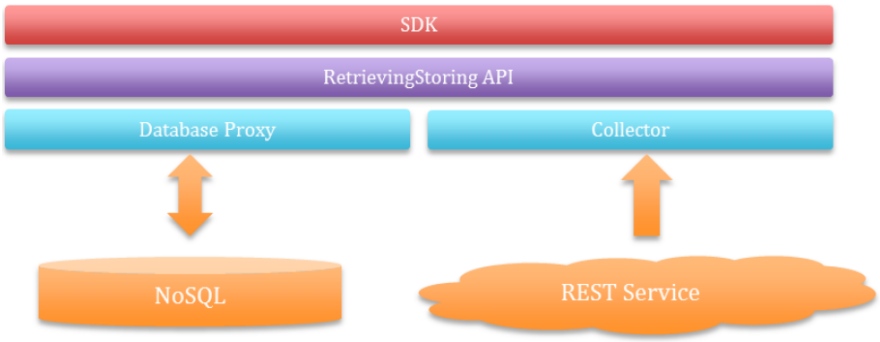


Figure 4 Collecting and storing using Twitter REST service (see online version for colours)

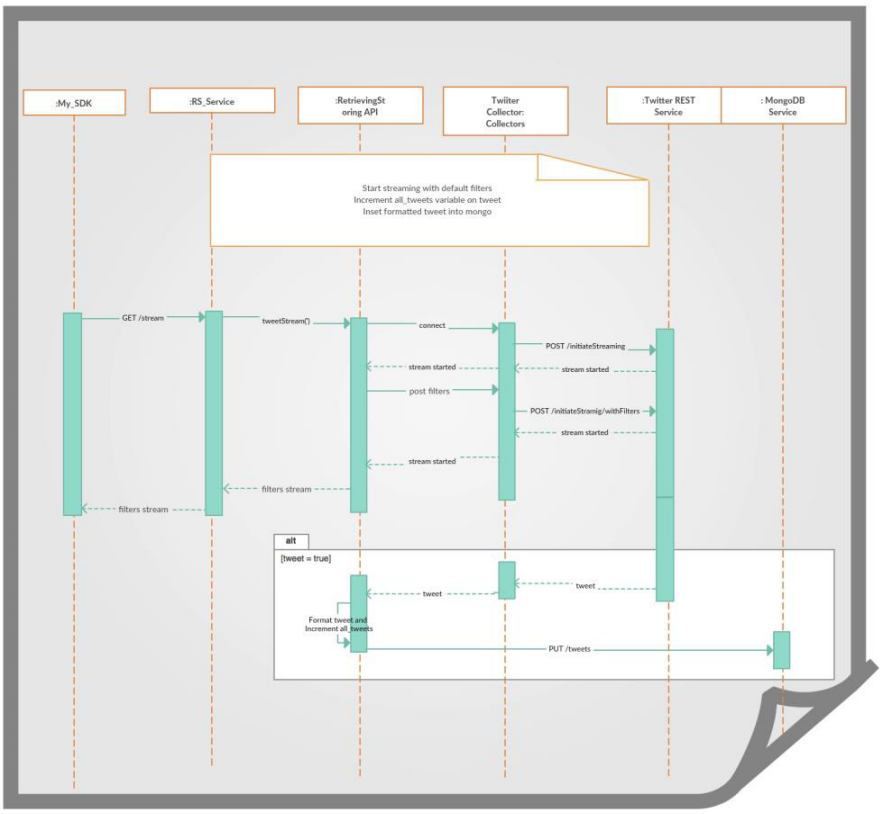


Table 1 Twitter service parameters

Parameters	*Description*
URI/stream	Start streaming for Twitter. Optional parameters location and tracking cannot be combined. Default parameters are the parameter's example.
Location optional	Location for tweet streaming. Bounding Box. (SW lon/lat NE lon/lat) Example: Lyon 4.771760,45.707432,4.898380,45.808289
Tracking optional	Keyword for streaming. Only one key word or phrase key word per request. Example: Trafic
Collection optional	Collection name to store the received tweets. Example: tweets
URI/stream/status	Information about tweets received since the beginning of streaming
URI/stream/stop	Stop streaming. Response will contain all the tweets received since the beginning of stream.
URI/stream/filters	Filters on stream
URI/stream/remove filters	Both parameters have to be sent if any of them is used.
Type optional	Filter type to remove Example: Location or tracking
Filter optional	Keyword or location to identify the filter. Example: 4.771760,45.707432,4.898380,45.808289or Trafic

Figure 4 shows a sequence diagram specifying the collect and store functions. The URI represented in the UML sequence diagram is a simplification from the real URI. As shown in the diagram, the streaming of the service is triggered by a GET request sent to the server, a sequence of actions is followed until data is delivered to the `Collector`. The `Collector` sends a POST request to connect to the streaming by the authorisation framework and a second POST request with the filters. The NodeJs module `node-tweet-stream` uses functions to connect to the Twitter REST API and to send the default filters. Received tweets are transformed and stored in MongoDB.

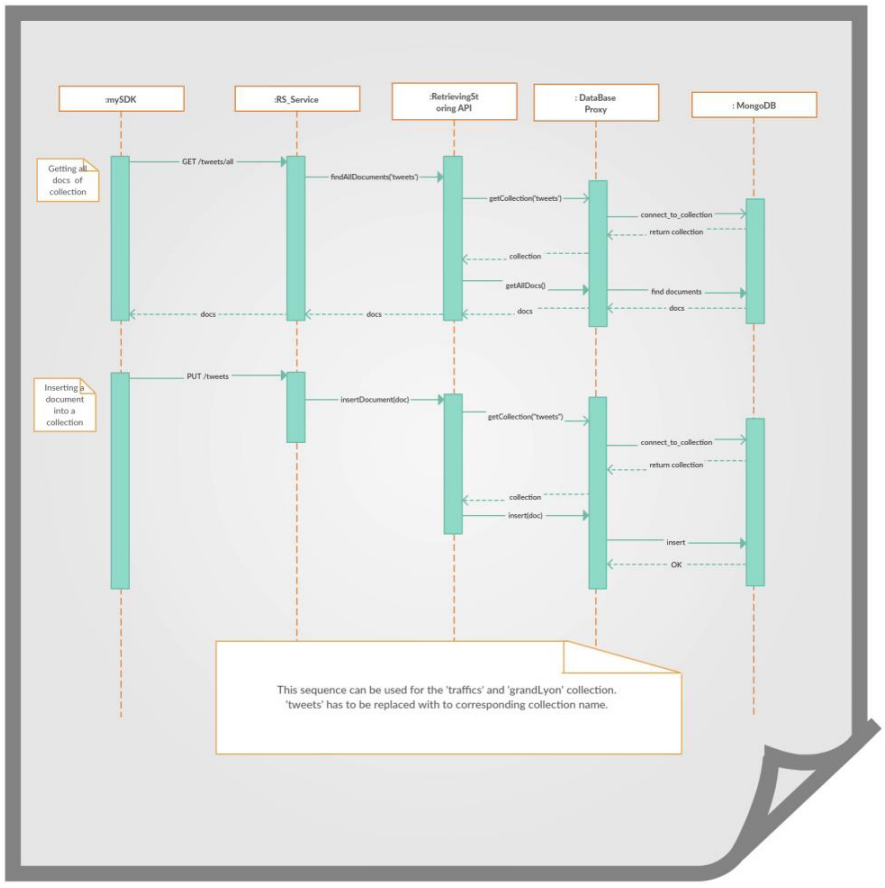
4.2 Storage service

Our preliminary tests collected a total of 10 GB based on the implemented services. The UML class diagram in Figure 5 is a simplified version of the elements inside each different type of documents.

The retrieved documents are stored into a MongoDB store deployed in the Openstack Cloud infrastructure, at the Lyon 1 University (Open, 2015). The MongoDB infrastructure runs with three configuration servers responsible to manage configuration of each machine in the database cluster, four routers (one for each service) acting as an interface to database users and which redirect data and queries to the relevant shards and nine replica sets shared in between three shards to store and analyse the data. Each shard is comprised of a primary replica set which receives the data and performs the queries and two secondary replicas set acting as backups. Each of the 16 machines run on a

m1.large instance of Openstack corresponding to 2 GHz quadcore instances with 8 GB of ram.

Figure 5 MongoDB implemented service (see online version for colours)



The principle of sharding (M. Inc., 2015) is to create data collection fragments of reasonable size in order to organise data into 'small' databases and thereby balance the number of requests to process, if the sharding is cleverly done. Each shard is an independent database, and collectively, the shards make up a single logical database with the possibility of data duplication. We have worked on two sharding strategies: hash and interval. A hashed fragmentation encourages the even distribution of data over the three shards (i.e., data stores) each one managed by a MongoDB instance. The ranged strategy fragments data according to intervals defined with respect to the range of values of a specific attribute.

The three types of sharding strategies were tested into the target collection (Hashed and Ranged sharding). For the 'hashed' strategy we used as shard key the attribute `_id` from the documents of the collection because this is the object which by definition is

unique, this makes it easier for MongoDB to create chunks of even size. For the 'ranged' strategy, we chose as shard key the attribute `user.location` (element communicated by the user on their location). And for a final comparison we used a 'hashed' strategy on the `user.location` field. We chose this shard key because we know French regions and we assume that they do not overlap thus, the intervals could lead to a balanced distribution of data. The data was distributed amongst the shards with $\texttt{shard}_1$ containing the document for `user.location` from `MinKey` to '`Lyon,FRANCE`', $\texttt{shard}_2$ from '`Lyon,FRANCE`' to '`Lyon, Rhône-Alpes`', and $\texttt{shard}_3$ from '`Lyon, Rhône-Alpes`' to `MaxKey`. The values '`Lyon,FRANCE`' and '`Lyon, Rhône-Alpes`' were found to be the ones which divided into thirds the database in unicode order.

Figure 6, Figure 7, Figure 8, present the data distribution between each shard for this database.

Figure 6 Chunk distribution (see online version for colours)

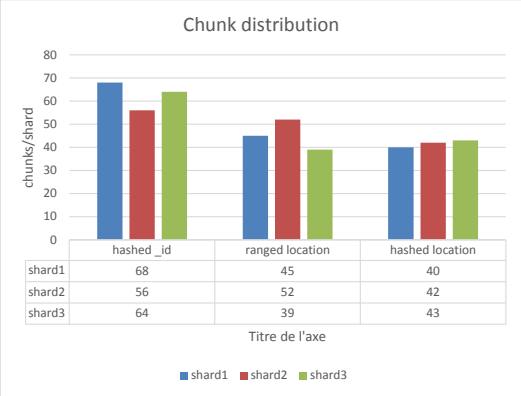


Figure 7 Document distribution (see online version for colours)

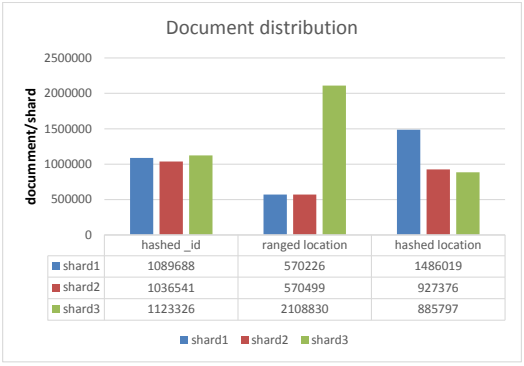
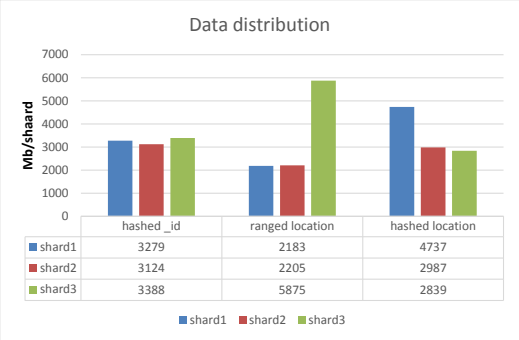


Figure 8 Data distribution (see online version for colours)



It reveals that whilst the chunk distribution between each shard are comparable for the hashed collection, the shared collection and location hashed collection (with a standard deviation of 10%, 14% and 4% respectively), data and document distribution are substantially less effective in the case of a sharded collection (standard deviation of 62% and 82% respectively compared to the 4% for the hashed collections). This has consequences on the performance of the collection as seen in the following test. The hashed location provides a middle ground at 30% standard deviation for both the document and data distribution.

To test the performance of the strategies we ran some tests using a query into the 'tweets' collections. We used the 6 following queries to analyse the reaction of the two sharding strategies. Each query was performed 10 times and outliers were removed.

The first two queries where done to observe the effect of using existing values amongst the shard keys:

- `{user.location: «Lyon»}` searches for the document with the user.location field equal to `Lyon`.
- `{user.location:null}` searches for the document with the user.location field equal to `null`.

The next two queries are defined to observe the effect of not using strict equality. In the following examples we therefore used regular expressions:

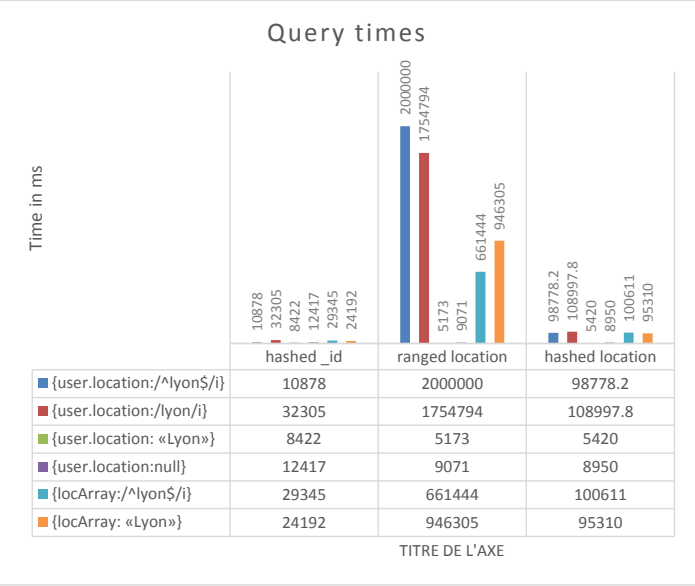
- `{user.location:/^lyon$/i}` searches for the document with the `user.location` equal to Lyon ignoring case.
- `{user.location:/lyon/i}` searches for the document with the `user.location` field containing `Lyon` ignoring case.

The last two queries where done to observe the effect of sharding when querying unrelated field. The `locArray` field corresponds to an array of word from `user.location`.

- `{locArray:/^lyon$/i}` searches for the document with the `locArray` equal to `Lyon` ignoring case.
- `{locArray: «Lyon»}` searches for the document with the `locArray` equal to `Lyon`.

Based on these queries we can observe that queries related to specific values of the sharding key are approximately 25% more efficient than the collection which uses a different key (Figure 9). On the other hand, when using regular expressions or fields not corresponding to the shard key the ranged strategy is substantially less efficient partially due to the document distribution in the collections. We observe partially the same effect with the hashed location strategy but to a much lesser degree.

Figure 9 Query performance tests (see online version for colours)



What we can learn from our experiments is that query time mostly depends on the distribution of the data through out the shards. Ranged sharding relies on choosing a very good shard key attribute, which can guarantee an even distribution of the data. This requires a good knowledge of the data and associated semantics. Also ranged sharding is particularly interesting if used for specialised queries that will rely on the semantic context of the application. For instance, in our example, expected queries will reason about French geographic organisation cities, departments, regions.

5 Conclusions

Cloud computing and big data have grown to become major contributors for ITS. Big data and cloud are a good combination since big data need a substantial computer power and cloud computing provides cheap and fast instantiation of computer power through the use of virtualised on-demand resources. What is more, cloud computing encourages the use of service oriented computing providing further agility to application development. In this spirit, we propose a service-oriented architecture for a big data management, hosting several individual services for each step of big data analytics. These services can then be composed to produce a decision support service.

This paper focuses on the cycle collection-storage of data streams. Using our architecture, we built services to collect data from REST services and store them in a MongoDB database by means of a storage service. The storage service enables the use of different sharding strategies to organise data and increase read/write performances. We used collected data to run experiments using different keys and sharding techniques. For our database collections the ranged strategy seems more efficient than a hashed strategy.

The contribution of this work is demonstrating the effect of the various strategies involved in sharding. Based on our observations the ranged strategy is the least interesting since potentially having impractical query times. On the other hand we believe this could be greatly improved provided better data distribution. The choice of a sharding key that is globally efficient is an uphill task as sharded collections provide the fastest query times but also the longest due to the difficulty in distributing the data evenly and providing an efficient index. This relies on very deep understanding of the data collected. Also as the collection grows the databases will have to call onto extra shards to store the data. At that point, the sharding strategy used will fall apart as it was configured for a smaller number of shards. Future work could include finding efficient and automated strategies to identify significant sharding key or dynamic methods to redistribute data when the previous distribution loses efficiency. We also presented a concept for using what we called data views to synthesise the information in that data. Future work consists of more accurately defining these views and finding solutions to perform operations between views.

Acknowledgements

The authors would like to thank the region Rhône-Alpes who finances the thesis work of Gavin Kemp by means of the ARC seven programs (http://www.arc7-territoires-mobilites.rhonealpes.fr/).

References

Abramova, V. and Bernardino, J. (2013) 'NoSQL databases: a step to database scalability in web environment', *Proc. Int. C* Conf. Comput. Sci. Softw. Eng. – C3S2E'13*, July, pp.14–22.

Artikis, A., Weidlich, M., Gal, A., Kalogeraki, V. and Gunopulos, D. (2013) 'Self-adaptive event recognition for intelligent transport management', *2013 IEEE International Conference on Big Data*, IEEE, pp.319–325 [online] http://doi.org/10.1109/BigData.2013.6691590.

Berson, A., Smith, S. and Thearling, K. (2004) 'An overview of data mining techniques', *Data Data Mining Application for CRM*, pp.1–49 [online] http://www.stat.ucla.edu/~hqxu/stat19/dm-techniques.pdf.

Chen, X., Vo, H., Aji, A. and Wang, F. (2014) 'High performance integrated spatial big data analytics', *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data – BigSpatial'14*, pp.11–14.

CSC, *Data Universe Explosion & the Growth of Big Data* [online] http://www.csc.com/insights/flxwd/78931-big_data_universe_beginning_to_explode (accessed 29 October 2015).

Das, S., Haas, P.J. and Beyer, K.S. (2000) 'Ricardo: integrating R and Hadoop categories and subject descriptors', *Proceedings of the 2010 International Conference on Management of Data – SIGMOD '10*, ACM Press, New York, New York, USA, p.987 [online] http://doi.org/10.1145/1807167.1807275.

Demirkan, H. and Delen, D. (2013) '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.

Demiryurek, U., Banaei-Kashani, F. and Shahabi, C. (2010) 'TransDec: a spatiotemporal query processing framework for transportation systems', *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pp.1197–1200, IEEE [online] http://doi.org/10.1109/ICDE.2010.5447745.

Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M. and Pazzani, M. (2010) 'An energy-efficient mobile recommender system', *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD'10*, p.899.

Genoveva, J.A.E. and Alexandrov, V.V. (2016) 'Comparing electoral campaigns by analysing online data', *ICCS*.

Hipgrave, S. (2013) 'Smarter fraud investigations with big data analytics', *Network Security*, No. 12, pp.7–9 [online] http://doi.org/10.1016/S1353-4858(13)70135-1.

Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R. and Shahabi, C. (2014) 'Big data and its technical challenges', *Communication of the ACM*, Vol. 57, No. 7, pp.86–94.

Jian, L., Yuanhua, J., Zhiqiang, S. and Xiaodong, Z. (2008) 'Improved design of communication platform of distributed traffic information systems based on SOA', *2008 International Symposium on Information Science and Engineering*, Vol. 2, pp.124–128.

Kemp, G., Vargas-Solar, G., Da Silva, C.F. and Ghodous, P. (2015) 'Aggregating and managing big realtime data in the cloud: application to intelligent transport for smart cities', *VEHITS 2015*.

Lee, D-H., Wang, H., Cheu, R. and Teo, S. (2004) 'Taxi dispatch system based on current demands and real-time traffic conditions', *Transportation Research Record*, Vol. 1882, pp.193–200 [online] http://doi.org/10.3141/1882-23.

Li, Z., Yang, C., Jin, B., Yu, M., Liu, K., Sun, M. and Zhan, M. (2015) 'Enabling big geoscience data analytics with a cloud-based, MapReduce-enabled and service-oriented workflow framework', *PLoS One*, January, Vol. 10, No. 3, p.e0116781.

Lim, S. (2008) 'Scalable SQL and NoSQL data stores', *Statistics (Ber)*.

Lin, J. and Ryaboy, D. (1013) 'Scaling big data mining infrastructure: the Twitter experience', *ACM SIGKDD Explor. Newsl.*, April, Vol. 14, No. 2, p.6.

M. Inc. (2015) 'Sharding and MongoDB', pp.1–80.

Mell, P. and Grance, T. (2008) *The NIST Definition of Cloud Computing*, Recommendations of the National Institute of Standards and Technology.

Open (2015) *Openstack* [online] http://www.openstack.org/ (accessed 16 October 2014).

Schadt, E.E., Linderman, M.D., Sorenson, J., Lee, L. and Nolan, G.P. (2011) 'Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology', *Nat. Rev. Genet.*, March, Vol. 12, No. 3, p.224.

Schaller, R.R. (1997) 'Moore's law: past, present and future', *IEEE Spectr.*, June, Vol. 34, No. 6, pp.52–59.

Talia, D. (2013) 'Clouds for scalable big data analytics', *Computer*, Vol. 46, No. 5, pp.98–101, Long Beach, California.

Tannahill, B.K. and Jamshidi, M. (2014) 'System of systems and big data analytics – bridging the gap', *Comput. Electr. Eng.*, January, Vol. 40, No. 1, pp.2–15.

Tavakoli, S. and Mousavi, A. (2008) 'Adopting user interacted mobile node data to the Flexible Data Input Layer Architecture', *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp.533–538.

Thompson, D., McHale, G. and Butler, R. (2014) RITA [online] http://www.its.dot.gov/data_capture/data_capture.htm (accessed 9 November 2014).

Yan, W., Brahmakshatriya, U., Xue, Y., Gilder, M. and Wise, B. (2013) 'p-PIC: parallel power iteration clustering for big data', *J. Parallel Distrib. Comput.*, March, Vol. 73, No. 3, pp.352–359.

Yu, J., Jiang, F. and Zhu, T. (2013) 'RTIC-C: a big data system for massive traffic information mining', *2013 International Conference on Cloud Computing and Big Data*, pp.395–402.

Yuan, N.J., Zheng, Y., Zhang, L. and Xie, X. (2013) 'T-finder: a recommender system for finding passengers and vacant taxis', *IEEE Trans. Knowl. Data Eng.*, Vol. 25, No. 10, pp.2390–2403 [online] http://doi.org/10.1109/TKDE.2012.153.

Zheng, Z., Zhu, J. and Lyu, M.R. (2013) 'Service-generated big data and big data-as-a-service: an overview', *Proc. – 2013 IEEE Int. Congr. Big Data, 2013*, pp.403–410.