Oktober 31, 2016

MASTER THESIS

TOWARDS A SCALABLE ENVIRONMENT FOR TRAFFIC INFORMATION SERVICES

Joanne ter Maat

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Programmes of Business Information Technology & Computer Science

Exam committee:
dr.ir. M. J. van Sinderen
prof.dr. M. E. Iacob
dr. M. Daneva
dr.ir. N. Besseling

UNIVERSITY OF TWENTE.

Abstract

Providers of traffic information services face the challenge of an ever growing volume of traffic data. To use this traffic data to discover useful information about trends in traffic congestion an approach is required that can handle this volume. We have explored the enterprise architecture of such a provider of traffic information services to identify areas of improvement. These improvements lead to a more scalable environment where the traffic data can be analysed. To demonstrate the outcome of these improvements, a new information service is created in the form of an information dashboard, where behaviour of traffic congestion can be studied.

Contents

Chapter 1

Introduction

For the Netherlands, with its high population density and a high fraction of the population level possessing a car, the intensity of traffic is also high. At the same time, transportation plays an important role. In 2013, 8.5% of the gross domestic product of the Netherlands was generated by the transportation sector, with transportation both of goods and individuals [9]. Traffic jams are a major inconvenience for the transportation sector. The two primary reasons for traffic jams are high intensity and accidents [37]. As much as 67% of traffic jams in the Netherlands are caused by high intensity, while the number of kilometres driven on the national road network is still rising. Accidents cause 19% of traffic jams. Traffic jams in turn cause arrival delays. Individuals and goods will be late and road transport is less efficient, because delivery times are not met. The financial cost of traffic jams for road transport over 2013 in the Netherlands is estimated to amount to somewhere between €613M and €797M [48]. Therefore, it would be beneficial to make use of the road network more efficiently.

Accurate and up-to-date information about the road network and traffic situation could help with better traffic management. Having information about the location of traffic jams and accidents available can help road transport companies to reroute their trucks or change their estimated delivery times. This is helpful for both sender and receiver. The transportation company is aware of trucks and drivers being unavailable for a longer period, while the receiver of goods knows it should plan receiving and unpacking of goods on a different time.

Historical information can be used to recognize patterns and changes in traffic. The patterns in turn can be useful for predicting traffic conditions like congestion. Several different types of users could make use of these predictions, changes and patterns. Logistic parties can use the information about patterns in their planning, to ensure the most optimal route is planned, avoiding possible traffic congestion. Urban planners can also benefit from information about traffic patterns when deciding, for example, where to place hospitals or other buildings that need to be easily accessible. Historical information can also be used for analysing how changes to the road network, like closing a road or building a roundabout have affected traffic conditions. This might be relevant for several institutions that are involved in road management.

1.1 Problem statement

The problems statement is provided by OVSoftware, an organization that, through its subsidiary Simacan, offers its customers services that can, for example, do route planning, fleet management or provide insight into traffic conditions. Simacan's core business is in real-time traffic information, meaning the services they offer are also focused on real-time or recent traffic information, however, they are interested in being able to provide analytics on long term data. We use Simacan throughout this thesis as a typical example of a provider of traffic information services.

There are several different suppliers that provide traffic data to support these services, offering

different types of traffic data that can tell the driving speed at a certain road, but also the intensity of traffic, or the specific location of construction works or incidents. The amount of data that arrives and needs to be processed at such providers of traffic information services is growing. The traffic data is often updated each minute, and as the road coverage of the traffic data increases, so does the amount of data that is received, resulting in a larger volume of data available for queries and analytics.

One example of traffic data is the data about the driving speed at the major roads. This data, like many other types of traffic data, is updated each minute. At Simacan this data is stored in a relational database that can be queried for the driving speed at a specified time and location. New traffic data about the driving speeds is received each minute, with the volume of data from one supplier of this data estimated to amount to 118MB per hour. Although this is by no means a large amount of data for today's standards, inserting and deleting data every minute in the traditional relational database has proven to result in a degraded IO performance of the database. The organizations offering traffic information services have to innovate to keep up with the growing amount of available data. It is the question whether this current environment for traffic information services is capable of efficiently storing and analysing historical traffic data.

One property of traffic data is that it always relates to a specific time and location for which the data is valid. To indicate for which specific part of the road network the traffic information is valid, several different location referencing methods exist. These methods can go a little further than specifying some coordinates, but in short this means that traffic information always has both a spatial and a temporal dimension.

In this thesis we aim to achieve two goals. First, we want to *achieve scalability in a traffic information services environment*. We aim to achieve this goal by proposing an architecture, based on ”big data” techniques, for storing and analysing the traffic data from a historical period.

Second, the scalable environment for traffic information offers new possibilities. New knowledge can be gained from the data, that goes further than reviewing current and very recent traffic conditions. This provides opportunities for *creating new information services*, that can offer insight in, for example, common patterns of traffic congestion. We want to explore how the new, scalable environment can be leveraged by the service provider, in our case Simacan, to establish new information services. These information services will provide knowledge to the service user through a dashboard.

1.2 Research questions

Based on the problem and goal described above, there are two main research questions, both with several sub questions.

1 **How can scalability be achieved in a traffic information environment?**

- 1.1 What is the current architecture?
- 1.2 What big data solutions exist for traffic data?
- 1.3 How can scalability of a traffic information environment be measured?
- 1.4 Is the traffic information environment scalable in terms of analysing large data volumes?
- 1.5 Is the traffic information environment scalable when ingesting data?

2 **How can a scalable traffic information system provide added value?**

- 2.1 Which new information services are enabled by the scalable architecture that are of value to the service consumer?
- 2.2 How should a dashboard be designed to visualize the new information service?

1.3 Research process

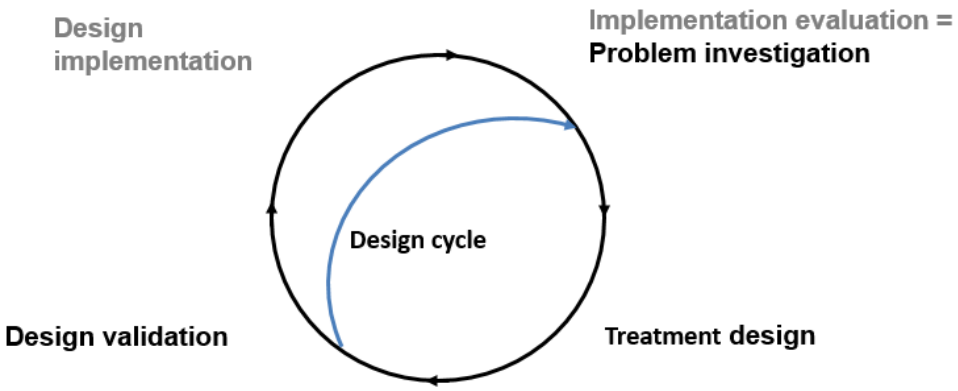


Figure 1.1: Design and engineering cycle, taken from [54]

Our goal is to improve the scalability of traffic information services by applying big data techniques such that a larger amount of traffic data can be analysed in order to get a better insight in the trends in traffic congestion.

According to the definition by Wieringa, this is a design problem [54]. Wieringa proposes a design cycle to solve a design problem. This cycle can also be placed in a larger cycle, the engineering cycle, where the proposed design is implemented and evaluated. For our design problem, we will follow this design cycle. The design cycle consists of 3 phases, problem investigation, treatment design and validation of the design. After this, the design can be implemented and evaluated. The design cycle with these 2 additional steps is called the engineering cycle, the complete cycle is depicted in figure 1.1.

Problem investigation: The problem investigation aims to learn more about the problem. The first exploration of the problem is already introduced in this chapter. Furthermore, research question 1.1 aims to further investigate the problem context.

Treatment design: In this step requirements are specified for an artifact, based on the problem investigation. Possible existing solutions can be considered, or, if necessary, new ones can be designed. The answer for research question 1.2 can be used to come to a design for the artifact for research question 1, a scalable traffic information environment. For research question 2, the artifact to be designed is a new information service, based on question 2.1

Design validation: The third step in the design cycle consists of testing the designed artifacts. For the artifact of question 1, the scalable environment, a prototype will be implemented that will be validated, answering questions 1.4 and 1.5. The answer to question 1.3 will help in creating the validation of the artifact.

Design implementation & implementation evaluation: The engineering cycle consists of these 2 additional steps, where the designed artifact, in our case a scalable system for traffic information services, is inserted into the actual problem context and its use evaluated. The evaluation asks very similar questions to the problem investigation step, but this time, the goal is to evaluate if the problem was solved, instead of eliciting the problem. These steps will not be conducted in this thesis.

1.4 Research methods

Question	Method	Output
1.1	Interviews	Data model & architectural model
1.2	Literature study	Overview of possible solutions
1.3	Literature study	Scalability metrics
1.4	Experiments	Scalability results
1.5	Experiments	Scalability results
1	**Answers to sub questions for 1**	**Scalable architecture**
2.1	Literature & interviews	A design for an information service
2.2	Questionnaire	Evaluation results
2	**Answers to sub questions for 2**	**Traffic information dashboard**

Table 1.1: Methods and output for the research questions

In order to answer the research questions, several methods are employed to go through the process described in section 1.3. For question 1.1, interviewing employees at Simacan will result in a model of the current architecture. To determine what solutions already exist for big traffic data, what scalability is and answer questions 1.2 and 1.3, a literature study will be conducted. Experiments will be done in order to assess the scalability of the architecture in terms of data volume that can be analysed and ingested, answering questions 1.4 and 1.5. The questions together will help to answer the first main question, resulting in a proposal for a scalable architecture.

For assessing how the scalable architecture can enable new and useful information services for answering question 2.1, we will look at the literature to find out how traffic information is being used and interview employees at Simacan to find out what wishes there are concerning offering traffic information services. We will use this information to design a new information service in the form of a dashboard. By letting end users evaluate the dashboard we answer question 2.2. The result of the two questions will be a dashboard with traffic information.

1.5 Structure

In this chapter we have introduced the research we will conduct. The rest of this thesis will be structured as follows. First, the relevant literature will be discussed in chapter 2. In chapter 3 we will further explore the problem context. Here, we describe the available traffic data, introduce the use case and show the current enterprise architecture of a provider of traffic information services. In chapter 4 the proposed improvements to the enterprise architecture will be introduced. This is also where we discuss the implementation of our prototype. Chapter 5 will evaluate this prototype on its scalability and usability. The thesis will be concluded in chapter 6, where we will have the answers to our research questions, discuss limitations of our research and propose future work.

Chapter 2

Literature Study

As a preparation for this thesis a systematic literature study, following the method of Kitchenham[25], was done to explore the current state of scientific literature on big data solutions for spatial and traffic data [46]. In this chapter we will discuss the findings of [46] in section 2.1 and 2.2 and expand upon it with other literature relevant to our scalable system for traffic information services.

2.1 Big spatial data

Traffic data contains some spatial information because it relates to a location, so the initial focus of our literature study was on big spatial data [46]. We categorized the work into four categories, those relating to indexing, spatial queries, complete systems and other papers.

2.1.1 Spatial indexing

Only 4 papers that were included in the literature review focused solely on spatial indexing, however, it was a subject also touched upon in most of the other found papers. The most frequently occurring indices in the literature about big spatial data were k-d trees, quad trees, r-trees and grid indices.

K-d tree

A k-d tree partitions data according to k dimensions [7]. Each node in the tree divides the space along one of the k dimensions into two sub trees. Where the border lies, depends on the distribution of the data. Which dimension is split is alternated on each level of the tree, so the root might split the data along the first dimension, the first level of sub trees splits the data along the second dimension etc. After k dimensions this repeats. An example of data in a two-dimensional space being split according to this method can be seen in figure 2.1.

Quad tree

A quad tree is a tree index where each node has four children [19]. The non-leaf nodes of the tree divide a two-dimensional space into four sub regions, with each of its child nodes representing one of the 4 regions. If no points are in a sub region, the sub region doesn't have to be divided further. Otherwise, a region is split into four regions again, until the number of data points in each sub region is below a certain threshold. An example of how the same data from figure 2.1 would be indexed according to a quad tree, can be found in figure 2.2. The difference with the k-d tree is that the quad tree divides the space into 4 equally sized boxes each time, instead of splitting it in 2 halves based on the data distribution.

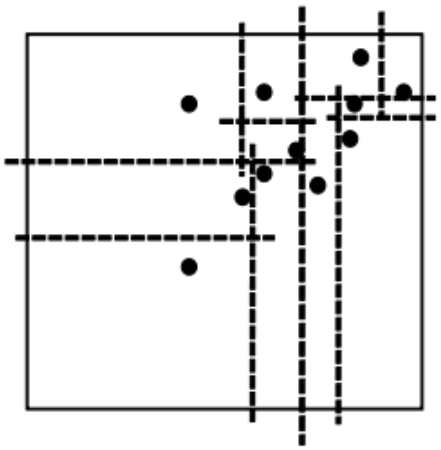


Figure 2.1: Example of a k-d tree, taken from [39]

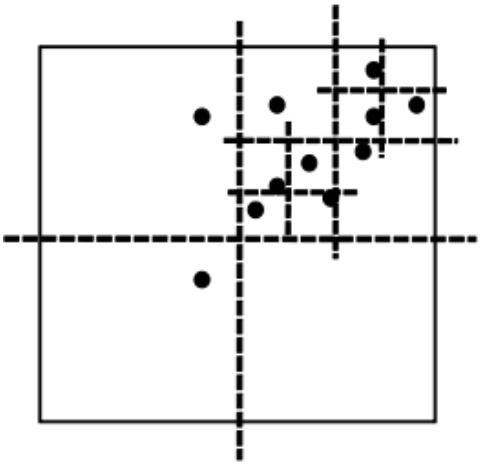


Figure 2.2: Example of a quad tree, taken from [39]

R tree

In an r-tree multidimensional data objects are placed in a bounding box [20]. Bounding boxes are grouped together in larger boxes. The index points to the larger boxes, which in turn hold pointers to the smaller bounding boxes, containing the data objects. In figure 2.3 the red boxes are the minimum bounding boxes for the data objects, they are grouped together into the larger blue and black boxes. In figure 2.3, R3 and R4 overlap, and R1 and R2 cover quite some empty space below R5. Several different variants of the r-tree exist that try to optimize the tree by minimizing the amount of overlap of the boxes and the covering of empty space.

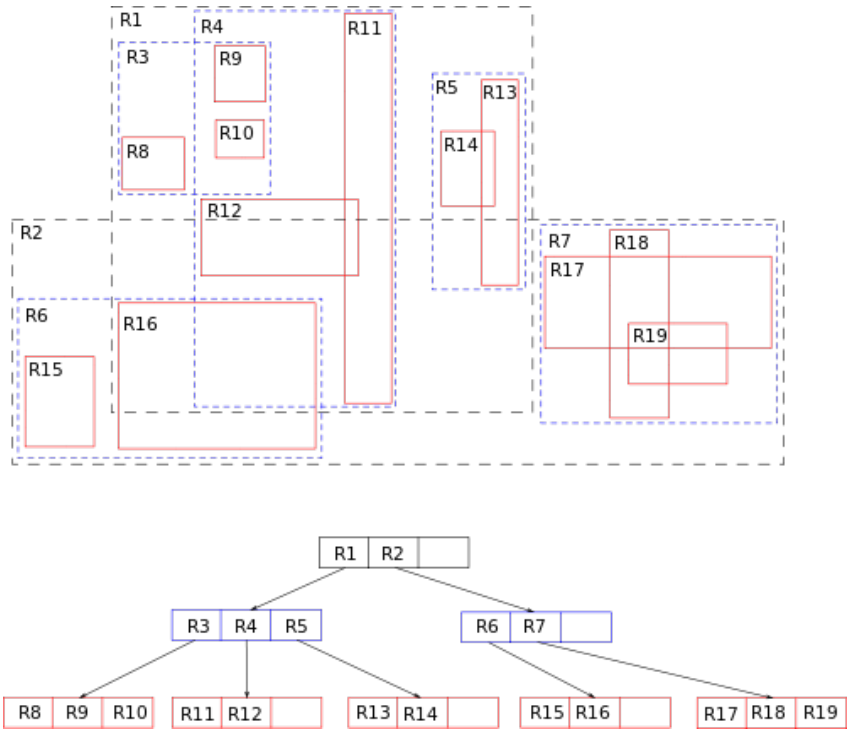


Figure 2.3: Example of an R-tree [55]

Grid index

A grid index divides the multidimensional space into a grid of equally sized boxes. Data objects can be looked up based on the box they reside in. The index contains an identifier for each box, with pointers to the data objects located in that box.

2.1.2 Spatial queries

In the literature, algorithms for several types of spatial queries can be found. They are usually supported by a spatial index. Spatial joins and intersections combine data from two different sources based on location. For traffic data, this could be relevant, for example, when trying to see what factors affect traffic, by combining the traffic data with data about the weather or events.

Another spatial query is the nearest neighbour query. Here, the query is given a location point, and the x nearest data objects are searched for in the data set. This might not be the most useful query for traffic data, the time dimension of traffic data ensures there are many different data objects with a different time but the same location. This would result in the query giving back many data objects about the same location, which defeats the purpose of this type of query. Spatial range queries work in a similar way, with a location and a radius, or some other way to delimit an area, these might be more suitable for traffic data, for example, for analysing congestion in a specified area.

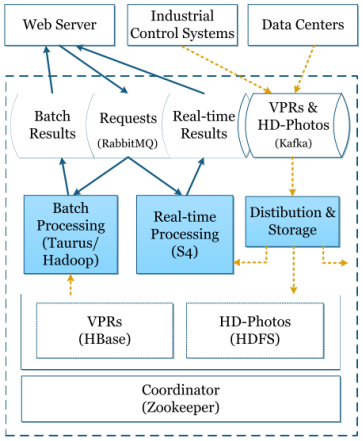
The skyline query is a query that was originally meant for 2-dimensional, not necessary spatial, data. It is used to find the most optimum points if there are criteria for 2 dimensions of the data. When looking at spatial points with 2 coordinates, the skyline query can be used to define a polygon that encompasses all *points* in a data set, it is not very suitable for spatial data types other than points.

2.1.3 Systems for big spatial data

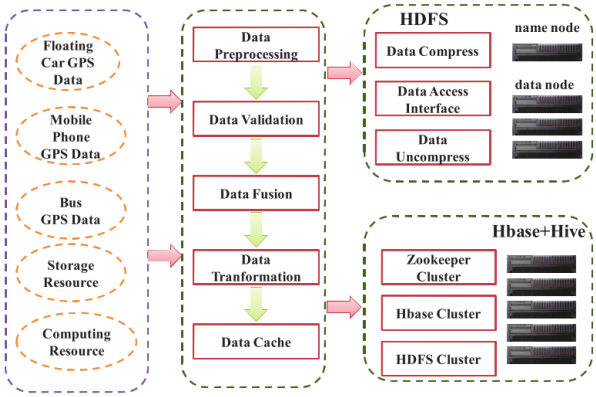
A part of the papers found in the preliminary literature study focused on proposing a complete system for spatial data analysis with storage and query capabilities. In one case, a completely new database system, AsterixDB, was proposed [4]. One work did nothing more than sketch a proposed architecture [26]. Most works, however, used an existing system and extended it to support spatial data. Both SpatialHadoop and Hadoop-GIS are solutions based on Hadoop's MapReduce, extended with spatial capabilities. SpatialHadoop was in one example used to process data about political conflicts that have a location [31]. Hadoop-GIS also was used to process spatial data with MapReduce [3, 12]. Impala is query processing system that uses data stored on Hadoop's file system(HDFS), but does not make use of MapReduce. One work modified Impala to accept spatial queries and spatial data [56]. HBase is a database on top of HDFS that was in one work used to store spatial data [59]. The spatial part of the data was stored in Well Known Binary (WKB) format. One table served as a grid index, where the row key represents the coordinates of a tile in the grid. The values in the index table are the row keys for another table with the actual data. In another paper, MongoDB was used for storing and querying satellite imagery data. [60]. Metadata was stored in a separate collection, which, claimed the authors, was useful for spatial data. VegaGiStore is a system that uses HDFS in combination with its own file format to store and query spatial data [61]. It uses a global quad tree and local index for the distributed data storage and leverages MapReduce for processing queries. One work introduced Pigeon as an extension of Pig [16]. Pig is a platform for analysing data stored on HDFS with a simple querying language that converts queries to MapReduce tasks. Pigeon adds spatial capabilities by creating user defined functions for Pig. This enables users to perform, for example, intersections on spatial data.

2.1.4 Other results

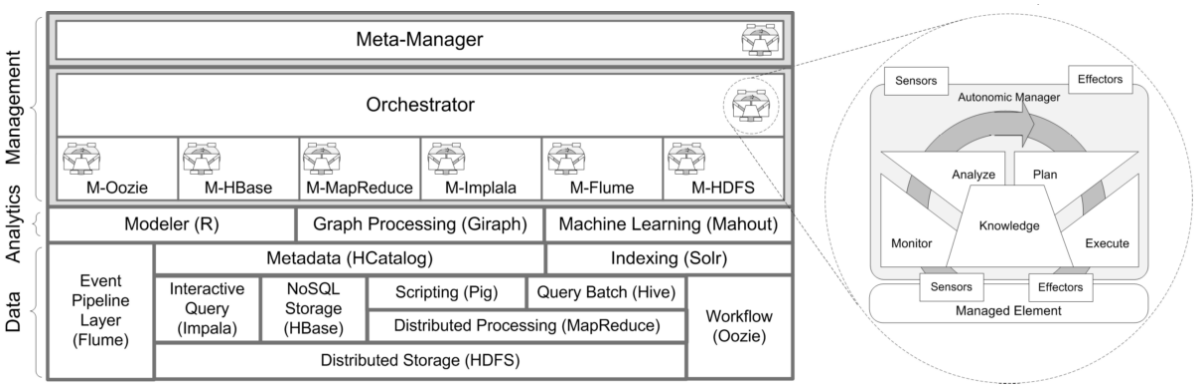
A few of the results of the preliminary study didn't fit any of the above categories. One was a short article from a trade journal for geospatial professionals, which briefly introduces a toolkit for spatial data processing on Hadoop, Esri's GIS tools for Hadoop [22]. One work gave an overview of some current work on systems for big spatial data [17]. It mentions some systems already found in our literature study, together with the following new ones. MD-HBase builds upon HBase with a k-d tree and quad index. Parallel Secondo combines the Secondo database for moving objects with Hadoop. GeoMesa is a system for spatiotemporal data, based on Apache Accumulo and using GeoTools, a java library



(a) Grandland architecture from [13]



(b) RTIC-C architecture from [57]



(c) Godzilla architecture from [42]

Figure 2.4: Proposed big traffic data architectures

for spatial data. SciDB's data model is based on arrays, making it suitable for raster data like satellite images. GeoTrellis, like SciDB, has a focus on imagery and raster data. Finally, SpatialSpark is a library for Spark offering some basic spatial functionality. Spark is a solution for large scale, in-memory data processing.

2.1.5 Conclusion

The literature on big spatial data gave insight into possible indices, different types of spatial queries and existing technologies that can be used to analyse spatial data. None of the literature found so far actually applied their contribution to traffic data, although in one case, moving object data was used. Applications in the found papers used data in either point or polygon form in almost all cases where this was specified, while the traffic data we are working with is in the form of links or linestrings. The question is whether any of the spatial indices found could actually be useful for traffic data.

2.2 Big traffic data

Since the preliminary literature study on big spatial data didn't find any literature that applied to traffic data, it was extended to include literature on traffic data and intelligent transportation systems in the context of big data. Some of this literature proposed a general (unimplemented) architecture for processing and analysing traffic data, while other literature focused on a specific goal for analysing the data.

Three of the works proposed an architecture for a system for storing and analysing big traffic data. They can be found in figure 2.4. The architecture in figure 2.4a is based on the processing of data from

traffic cameras, with images and Vehicle Passing Records(VPR) extracted from the camera footage [13]. The architecture in figure 2.4b, intended for detecting traffic jams and accidents using GPS data, has two different data stores, for two different types of analysis that can be done [57]. When all data needs to be handled for the analysis, the data on HDFS can be used. When the analysis requires the data to be filtered, the data in HBase can be used with Hive to find the data relevant for the analysis. The architecture proposed in [42] consists of three layers, the data, analysis and management layer, shown in figure 2.4c. With their proposed system, Shtern et al try not only to encompass the data storage and analysis, but also the management of resources that can be used by the different components in the architecture.

The proposed architectures have some components in common. They need to *ingest* and, if necessary, *preprocess* the incoming data before storing it. Kafka, Flume and S4 were suggested as systems for ingesting and processing the incoming data. Another common component is the *data storage*. HDFS was the basis for data storage in each of these papers. HBase was also included in all three works as a database for storing the data. Finally, the architecture needs to provide the possibility to *analyse the data*. MapReduce was suggested for batch processing of the data once. Hive was suggested in two out of three works as a data warehouse for analysis of the data. Pig and Impala were also included in one architecture as ways of querying the data.

Another work takes a different approach by proposing to store the data in an ontology database [51]. MapReduce is used to process the incoming data and produce the most recent traffic conditions.

Other works focused on specific applications for which traffic data was analysed. These applications can be roughly divided into two categories. Analysing current or historical data for detecting patterns in congestion or traffic activity and analysing data for predicting future traffic conditions. In most of the works the analysis was done using MapReduce. Once the analysis was done using a combination of Hive and Mahout, which is a library for machine learning. In another case RHive was used, which integrates the data warehouse Hive with the statistical programming language R. Two works used the in-memory database Hana for data storage and analytics.

When information was provided about the data used for the analysis, this was often data belonging to links or road segments. In a few cases, data was based on points, referring to sensor locations. Only one work used a spatial index [14]. This happened to be a paper that worked with data based on points, here, a k-d tree was used. Other works on big traffic data did not specify if and how they indexed the data. We can conclude that the use of a spatial index on traffic data is not common in literature. Either it hasn't been investigated yet or it has proven not to be practical.

2.3 Location referencing methods

Location referencing methods serve to indicate which precise part of the road some piece of traffic data corresponds to. When communicating information about an accident or traffic jam, for example, simply mentioning the road name is not detailed enough. It needs to be specified which driving direction and lane the information applies to. For a more complicated highway junction, with several ramps, it is even more important to have a good location referencing method, or it would be impossible to communicate correctly where exactly a traffic jam is occurring. Ideally, different parties involved in the communication should be able to use different maps, however, not all methods for describing locations support this. Methods that support this are the so-called *dynamic* location referencing methods. Other methods use pre-coded location references.

2.3.1 Pre-coded Location Referencing

The most straightforward way of referring to a location is to give a location a code and store these codes along with the locations in a database. A widely used location referencing method using pre-

Loc. Code	Loc. Type Code	Loc. Type Description	Road number	First name	Second name	Lin. Ref.	Negative offset	Positive offset
3332	L1.1	Road	A5	Haarlem	Amsterdam			
3330	L3.0	Ord 1 segm	A5	Haarlem	Halfweg	3332		3331
3331	L3.0	Ord 1 segm	A5	Halfweg	Amsterdam	3332	3330	
8340	P1.3	Motorway Junction/Exit		Haarlem		3330		8341
8341	P1.2	Motorway Triangle		Rottepolderplein	A9	3330	8340	8342
8342	P1.3	Motorway Junction/Exit		Halfweg	N5	3330	8341	8343
8343	P1.3	Motorway Junction/Exit		Ijmuiden	N202	3331	8342	8344
8344	P1.1	Motorway Junction/Exit		Slotermeer	N206	3331	8343	8345
8345	P1.1	Motorway Cross		Amsterdam	A10	3331	8344	

Figure 2.5: Example data in the Alert-C protocol, taken from [47]

coded locations is the Alert-C protocol. Alert-C is used by the traffic message channel(TMC). TMC is the service that broadcasts traffic information through a radio signal in many countries around the world. Alert-C makes use of a database in which the locations are stored [47]. Three types of locations can be stored, areas, lines and points. Each location in the database has a unique location code, but also a type code and a location name. Figure 2.5 shows an example of what locations described in the Alert-C protocol look like. All locations in the examples are either lines or points, indicated by the first character of the location type code. Another thing to note in figure 2.5 is the lack of spatial information. No actual spatial details are stored.

Although this way of referring to locations is fairly simple, it also has some disadvantages. When communicating about traffic data, all parties involved in the communication need to have the table with locations, in order to look up where a location code refers to. Furthermore, all parties also need the same map, so that the location codes are sure to be linked to the same locations on the map. Another limitation of this method is the limited amount of locations that can be referred to. If a location does not have a code in the table, that location cannot be communicated about.

2.3.2 Dynamic Location Referencing

To avoid the limitations of the pre-coded locations, dynamic location referencing methods were invented. Using dynamic methods ensures that locations can be encoded and decoded on different maps and any location can be referenced. When investigating these methods, 2 methods predominantly show up in the search results, AGORA-C and OpenLR. TPEG-ULR is a third method for dynamic referencing. Although OpenLR and TPEG-ULR are open, AGORA-C is commercially licensed with very little information openly available.

AGORA-C

One work shortly describes AGORA-C as background information[52]. It states that AGORA-C makes use of location points, intersection points and reference points, with location points indicating the start and end of a location and reference points as additional points along the line to make clear how the location runs. Intersection points indicate there is an intersection at that point along a location. Points

are defined by their longitude and latitude. Some additional information can also be included in the description of points, e.g. bearing or functional road class(FRC). In another report AGORA-C is described as complex to implement [45]. It states that additional agreements between sender and receiver of data are required, otherwise certain road elements can be interpreted differently between different parties.

OpenLR

Another method for dynamically referring to locations on the road is through OpenLR [49]. OpenLR is able to refer to lines and several types of points and areas. Points can be simple points, or points along lines or with an access point. Areas can be polygons, but also circles, rectangles or grids. Any type of location is described with one or more location reference points(LRP). An LRP consists of a coordinate pair and some other attributes, depending on the type of location being described. For a line, each LRP along the line should, beside coordinates, have a bearing, FRC and form of way(FOW). Bearing says something about direction, while FRC and FOW say something about the type of road. In a line, all but the last LRP should also have a distance to the next LRP and a lowest FRC on the path to the next LRP. Specifying the lowest FRC limits the number of roads that should be taken into account when decoding at the receiver. Any location can be encoded to be sent as binary, or in an xml format when bandwidth is less important.

TPEG-ULR

TPEG-ULR, like Alert-C is able to refer to locations of the type point, line and area [40]. A location reference always specifies which type of location is described together with some other obligatory elements. For a point these are coordinates. For an area the elements are a set of points to form a polygon, or for a simpler area one point with an expansion. Lines are described with points for the from, to and via locations. All but the last point have a direction and distance to the next point.

2.4 Architecture

In section 2.2 a few architectures for traffic data were introduced, with some similar elements in each. This section will look at reference architectures for big data systems, to establish what such an environment should look like.

Pääkkönen and Pakkala base a reference architecture on case studies of companies working with large amounts of data [35]. Data sources in this architecture can vary along 2 dimensions. It can be streaming data or data that is already present. It can also be either structured or unstructured. Several functions are incorporated in the architecture, data extraction, loading & preprocessing, processing, analysis and transformation. Each of these functionalities can have a separate (temporary) data store for storing the result of that specific function. An additional function without its own data store is the interfacing and visualization, where user applications are located. A separate part of the architecture is devoted to the scheduling of jobs and extracting models from the data.

In one master thesis a reference architecture for big data was developed based on a literature study [29]. The literature study was conducted to explore characteristics of big data and identify what requirements there are for a big data application. This information was used to come to a reference architecture. Functionalities supported in this architecture partially overlap with those from Pääkkönen and Pakkala [35]. The architecture consists of components for data extraction and stream processing, information extraction, management of data quality, data integration, data analysis and data distribution. These functionalities are supported by components for storage, metadata management, data lifecycle management and privacy.

2.4.1 NIST Reference Architecture

NIST, the American standards organisation, also released a reference architecture in their 'Big Data interoperability framework' [32]. The architecture they propose was drafted based on a survey where experts could send in their proposed architecture. The reference architecture NIST drafted based on the results of this survey can be found in figure 2.6.

Several relevant roles were identified, that resulted in 5 functional components for the reference architecture. The five components are data provider, data consumer, system orchestrator, big data application provider and big data framework provider. The components are placed on two "fabrics", a Management fabric and a Security & Privacy fabric. The architecture is organized horizontally from raw data to information and vertically from available computing resources and infrastructure to using and managing these resources.

System Orchestrator

The system orchestrator is responsible for managing the activities of the other components. The system orchestrator is the business owner of the system, being aware of business goals. The system orchestrator also manages policies for the data life cycle and requirements for data analytics and system architecture. Next to defining these policies and requirements, the system orchestrator is also responsible for monitoring the different system components to ensure the requirements are met. There can be multiple actors involved in fulfilling the role of system orchestrator, these actors can be humans or software components or both.

Data Provider

The data provider is the component responsible for making available new data to the system. The data provider can belong to a different organization than the other system components. The data provider performs several different activities, including the capturing of data, scrubbing it from privacy sensitive information and distributing the data. Data can be captured from a wide variety of sources, for example sensors or mobile devices. The data provider is also responsible for, among other things, managing access rights, providing some form of distribution mechanism for the data, annotating the data and creating metadata, such that other system components can correctly interpret and use the data.

Big Data Application Provider

The big data application provider is responsible for the collecting, preparing, analysing and visualizing of the data, while also offering an interface to data consumers to access the data. The application provider can be one application, or a combination of smaller applications that together provide the functionality of the application provider.

The collection activity of the application provider interacts with the data provider and extracts the data from the delivery mechanism of the data provider. Collection should also keep the data until it can be stored by the framework provider. This usually happens after preparation.

The preparation activity transforms the incoming data. The data might be validated, cleaned, or standardized, for example.

Analytics ensures that new knowledge is derived from the data. It can make use of batch or stream processing frameworks from the framework provider. Result of the analysis should be communicated to the data consumer through the visualization activity. Analytic tasks in a big data context can be broken down into sub tasks that can be distributed across a cluster.

Visualization is responsible for formatting the results in a meaningful way for the data consumer. Formatting the results can produce a graphical image, but a textual representation of the results is also possible.

The access activity provides the communication between the analytics and visualization activities of the application provider and the data consumer. It should be able to handle request from the data consumer and retrieve the required response from the analytic and visualization activities.

Big Data Framework Provider

The big data framework provider is the component which specifies the utilities on which the data is stored and processed. It consists of infrastructure, data platform and processing frameworks. The infrastructure framework defines the networking, computing and storage resources, along with additional necessities like the power supply.

The data platform framework specifies three approaches for data organization. The first approach, in-memory storage, exploit RAM performance, resulting in faster processing, but requiring additional steps for achieving data persistence. A second approach centres around file systems, which can be central or distributed. Data is kept in text, binary or delimited files. A final approach is indexed storage, where the indexing approach is categorized based on the complexity of the data that is stored. There are five types of storage that are distinguished, key-value, relational, column, document and graph storage.

The final part of the framework provider component is the processing framework. The processing frameworks can go from batch processing to stream processing. The processing framework can be divided into three phases, ingestion, analysis and dissemination. Batch processing frameworks focus mostly on the analysis phase and have high latency, while stream processing frameworks might cover all three phases and are intended to process and respond to events in near real time. Not all frameworks fall strictly under batch or streaming, they might have properties of both.

Next to the infrastructure, data and processing platforms the framework provider should also facilitate messaging and communication to transfer data between the different activities of the application provider and the other system components. Furthermore, the framework provider should also manage the resources for the framework. The demand for computing resources changes when the workload of the system changes, and this should be managed.

Data Consumer

The data consumer is an end user or some other application that consumes the data analysed by the big data application provider. The consumption of data might be inter-active, meaning the data consumer requests some form of information from the application, or it might be in the form of a subscription, where the consumer automatically receives whatever information the application outputs.

2.5 Scalability

Scalability refers to the ability of a system to keep its performance similar when the workload increases and the available computing resources are also increased proportionally. Two different types of scalability can be found in literature, scaling up and scaling out [6, 30], also called vertical and horizontal scaling. The first, scaling up or vertical scalability, refers to scalability that is achieved on a single machine by adding more computing resources to that machine. Scaling out refers to the scalability that emerges when adding new nodes to a cluster, each with its own resources. On the other hand, with an unscalable system "*the additional cost of coping with a given increase in traffic or size is excessive*", according to Bondi [8]. Bondi defines scalability as "*the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement*". Scalability is a broad term that is hard to define. When measuring the scalability, we'll have to formulate what it is exactly that we mean with scalability. We will adopt a definition for scalability that is in line with the following statement: For a given analysis (or query), if the data size grows, and

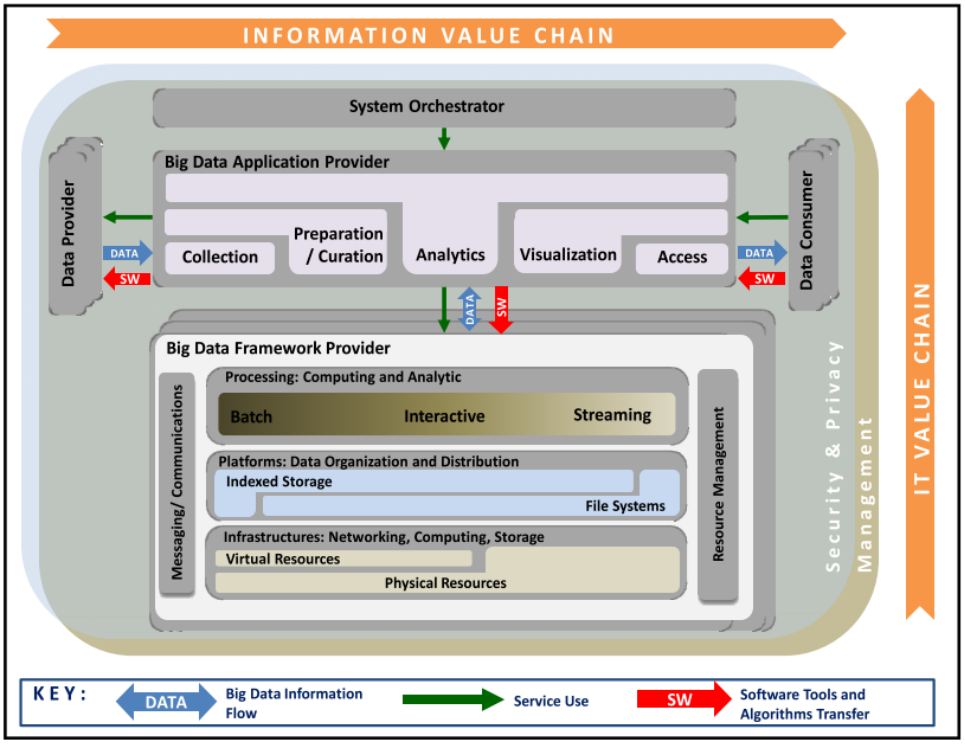


Figure 2.6: NIST reference architecture from [32]

computing resources are added in proportion with the growth of the data size, the response time for the analysis should remain equal.

2.6 Traffic Data Analysis

Analysis of traffic data often focuses on the occurrence of patterns in traffic speeds. In one work, commuter routes were analysed to identify which routes are more congested [15]. The data used consisted of travel times on road segments, with a group of segments making up a route. Clustering the data resulted in 3 groups of days, one consisting of weekends and holidays, one cluster with Mondays and Fridays and the final cluster with other weekdays. Analysis of slowest segments on the routes showed that the slowest part of the route can be caused by traffic lights, merging of lanes or crossing of the commuter routes. Start and end times of congested periods for each route could also be identified.

Ibrahim and Far proposed a transportation data mining system consisting of 3 phases, the offline, real-time and decision support phase [23]. The focus of their work is on the offline phase. In this phase, first, preprocessing of the historical data takes place. Then, patterns in the data about traffic accidents can be discovered. Decision tree classification is used to identify accident severity based on a set of conditions. By dividing the data in a training and validation set the classification method can be evaluated. In the real-time phase, the classification model can be used to predict traffic status and accidents, based on the current traffic conditions. These predictions can be used in the decision support phase by decision makers in the transportation sector.

Jo et al. also analyse historical traffic data [24]. First, they analyse the traffic speed. They don't use clustering methods on the data, but define their own grouping. Data is divided into 8 clusters, one for each day of the week and a separate one for holidays. Mean traffic speeds in 10 minute intervals for each day are calculated to get an indication of what normal traffic speed is. A metric called mutual information is introduced, that indicates how much the speed of traffic is dependent on a road feature, e.g. number of lanes. Using features with a high score of mutual information results in better predictions of traffic speed.

One work proposed its own clustering method to identify traffic patterns, where data is first divided

in standard and special days [44]. For standard days, hourly and daily patterns are made. Weekly averages are used to identify seasonal patterns. Special days are compared to existing patterns for standard days. If a match can be found, the special day is categorized in an existing pattern, otherwise, a new pattern is added. The patterns are a representation of the trends in traffic volume.

Weijermars and van Berkum use Wards clustering method to identify patterns in traffic volume on the Dutch highway A50 [53]. They want to look at travel demand, measured in traffic flow. They reason that heavy congestion is not representative of travel demand and exclude days with congestion from the dataset. Wards clustering method was used in two different situations. First, it was used on the data, resulting in 3 clusters, one with weekdays, one with only Sundays and one with special days and weekends. Then the same clustering method was applied, but this time the data was classified beforehand in working days and non-working days. This resulted in a larger number of more accurate patterns, without this pre-classification step the variation within the clusters is large.

2.7 Traffic Information Dashboard

Visualization is an important way to better understand data, supporting the ability to interpret the data and see patterns. Han, Wang and Shaw stress this point by stating "*it is often easier to detect a pattern from a picture than from a numeric output*" [21]. They explain how some very basic charts can be used to visualize data about traffic volume in different ways.

Another work looked at trajectory data, where a position at a moment in time is recorded [11]. Before visualizing the data, 4 steps are required for the *pre-processing of the data* before visualization, namely *cleaning*, *matching*, *organizing* and *aggregating*. Data cleaning involves the identifying of errors in the data and fixing them. Matching identifies to which position on the road the data refers. Data should then be organized in a data warehouse or database with appropriate indices. Aggregation serves to reduce the data size and facilitates the analysis. For traffic data, 4 dimensions for aggregation were given, temporal, spatial, directional and attribute-related. Several manners for visualizing traffic data were given. For the temporal visualization line charts were proposed. If some periodic cycle for traffic data needs to be shown, a radial layout can also be used, e.g. to show how traffic changes over a week. Spatial visualization, both of points and lines, can be done on a map. With sizes of points and thickness of lines, along with use of colours, several options exist to express the value of a traffic statistic at a certain location.

Chacon and Kornhauser used line and bar charts to show the amount of traffic delay on Los Angeles freeways for certain measurement stations. They also visualized the stations on a map. By clicking a station on the map a pop up appeared to show the chart [10].

Shekhar et al. visualized patterns in historical traffic data [41]. They stated that 4 groups of users would benefit from visualized traffic data, namely transportation managers, traffic engineers, travellers and researchers. Transportation managers can use it to identify if traffic behaviour deviated on a specific day. Traffic engineers can use the visual patterns to see if changes in the road network need to be made. Travellers can use this information to avoid areas that are usually congested.

A Traffic Cube is used in one work [43] to represent traffic counts in 3 dimensions, space, date and time. Each of the 3 dimensions requires an aggregation hierarchy. The traffic count measurements come from counter stations, and aggregating the spatial dimension is done by grouping the stations by highway route. This solution was created using MatLab for the user interface and visualization.

Loop detector data, providing speed and volume of traffic, was used for visualizing both real-time and historical information [28]. Visualizing the real-time data was used for detecting incidents. Using the historical data, a visualization of the impact of an incident can be made, showing how the speed for road links near the incident, or an entire highway were influenced by the incident. This was done using contour plots, where time and location are dimensions of the plot, and colour indicates the driving speed.

Contour and colour maps where used in another work to indicate how busy traffic was [58]. With traffic count data from different sensor locations, interpolation was used to colour a map. This gave a quick overview of the areas where traffic volume was high for a specified time interval and thus where congestion might occur.

One work proposed visualization using 3 UI components, map, chart and calendar [36]. The data that was visualized was the count of traffic volume for intersections per minute. Each UI component was created using a different library. The chart component was done with Highstock, and showed the total traffic volume in a line chart over time. The data for the chart can be aggregated per day, week or using all data. The Map UI used Google's Map API to display the map with coloured circles at the intersections, indicating the average volume of traffic at an intersection for a specified time frame. The calendar component was done using D3.js and also uses colour to show the mean traffic volume per day.

Liu et al. identified 3 challenges for visualizing taxi GPS data, namely the cleaning of data, the query efficiency and how to visualize data from a large number of different taxi's [27]. Cleaning is done by first discarding data from taxi's whose GPS unit send data with very large time gaps. Then the locations need to be matched to segments of the road network. The problem of query efficiency is tackled by using indices. First, the road segments are indexed with a grid index. A second index is created that maps the road segments to taxi ids and time. For the visualization, a method is developed called fingerprinting visualization, with two different views, density and speed. Each fingerprint on the map is a radial chart that uses colours to indicate the speed or density for different times and days.

ViaRodos (viarodos.cz) is a system that already can be visited by the public and shows several metrics about the current status of Czech traffic, together with weather information [18]. This was not done on a map, but a linear layout was chosen. A part of the road network is laid out horizontally, with road numbers, visualized like signposts, indicating positions along the road. Beneath, statistics like speed, delay and traffic intensity for the different sections of the road are shown. Organizing this information linearly was done to improve readability, it would be unpractical to display lots of different statistics or details on a map.

2.8 Databases

As stated in the introduction and at the beginning of this chapter, a preliminary study was done on big data solutions for spatial and traffic data. This section will serve as an overview of used databases found in this preliminary study. Obviously, this is no complete list of currently existing spatial databases, but a list of solutions that were successfully used in the scientific literature for spatial or traffic data, complemented with some products that came from expert interviews and from researching already found products. An attempt to identify advantages and disadvantages has been made, however, these are difficult to identify without having any experience with the database.

name	type	Support for spatial data	Remarks
HBase	Column family database	No	
Hana (SAP)	In-memory column database	Support for spatial datatypes, but not spatial indices	In-memory database gives fast performance, but higher cost
MongoDB	Document database	Supports GeoJSON objects and can index them	Good documentation and support for multiple languages

AsterixDB	Data Warehouse	Supports spatial data through its own defined datatypes, spatial data can be indexed with an R tree	Documentation not very elaborate
Vertica (HP)	Column database	Yes, with Vertica Place spatial data can be stored and polygons can be indexed	Free community edition available
Redshift (Amazon)	Data Warehouse (column store)	No	Offered as a service
Cassandra	Column family database	No	
DynamoDB (Amazon)	Key-value store	Yes, through GeoLibrary points can be indexed with a GeoHash	Offered as a service
GeoMesa	Database building on several existing data stores	Yes, spatial datatypes can be indexed	Works with Accumulo, HBase, Cassandra & Kafka
MariaDB	Relational database	Supports spatial data (WKT & WKB) and indexing spatial data (R-tree)	
CouchDB	Document database	Through an extension called GeoCouch GeoJSON data is supported and can be queried by using bounding boxes	
Hive	Data Warehouse	Can work with spatial data in several formats through ESRI's GIS tools for Hadoop	

Table 2.1: Databases

The results, that can be found in table 2.1, consist of the following databases. HBase is a column family database, that works with the Hadoop environment for distributed processing of data, where data is stored on a distributed file storage called the Hadoop File System(HDFS) and processing of data can be done using MapReduce. HBase makes use of HDFS for storing its data. In one work both HBase and HDFS were used and the conclusion was that data is better stored directly on HDFS when a query needs to analyse all the data, while HBase should be used if a selection of data needs to be analysed [57]. HBase is fairly well supported with plenty of documentation to be found, however, it has no native support for spatial data. Another well-known column family database is Cassandra.

Hana is an in-memory database using a column store. It keeps the dataset in memory, for the best performance. It does support spatial datatypes and spatial queries, but not the indexing of spatial data. Creating secondary indices on non-spatial data is possible, but in most cases it does not offer benefits [38].

MongoDB is a document-oriented database. The database schema is dynamic instead of the more traditional table structure. MongoDB offers support for GeoJSON, meaning it can store, index and query various types of spatial data. In one work HBase was adapted for spatial data into a system called HBaseSpatial. Its performance was then compared to MongoDB. The performance of HBaseSpatial was better for querying line data, but when the query was done on point data, the performance of HBaseSpatial was similar to MongoDB [59].

Another document-oriented database is CouchDB. When used in combination with an extension called GeoCouch, this database can also store, index and query spatial data.

Another database that was found in the literature is AsterixDB [4], according to its creators a Big Data Management System (BDMS). The goal behind AsterixDB is to create a system for ingesting, indexing and querying large quantities of data. Although it is not the main focus, the BDMS does offer support for spatial data in the form of some primitive spatial datatypes, spatial functions and one spatial index (r-tree).

Vertica is a column database offered by HP. An additional package called Vertica Place offers support for storing, indexing and querying spatial data.

Amazon offers Redshift, a data warehouse. It uses a column store as the underlying data storage and is offered as a service.

DynamoDB is another database offered as a service by Amazon. It is a key value database that offers support for spatial point data through an additional library, GeoLibrary [5]. It has up to 25 GB of free storage.

GeoMesa is a database for spatiotemporal data that builds on several other technologies. It makes use of Accumulo, which in turn requires Hadoop and Zookeeper. It supports spatial data and indexes the data with a three-dimensional space filling curve, the third dimension, next to latitude and longitude being time.

MariaDB is a relational database offering native support for spatial data, indices and queries. It is an open source database based on MySQL.

Hive is a data warehouse for data stored on Hadoop. Queries can be formulated in SQL and are converted to MapReduce jobs that are then executed. There is no support for spatial data, but through Esris GIS tools for Hadoop storing and querying spatial data is made possible.

This set of results contains several types of databases with different properties. Relational databases are databases that organize data in tables. Data objects of the same type and with the same attributes end up in the same table. Each column indicates an attribute of the data, while each row represents one data object with its values for each attribute. Relational databases store data in rows.

Column databases are able to represent the same data as relational databases, but instead of storing data organized in rows, the data is stored organized in columns. In column databases, read operations only need to access the relevant columns, making read performance better as opposed to relational databases. On the other hand, database transactions are slower, inserts need to be split into the different attributes and written to the different corresponding columns [1]. Column family databases organize the data both around columns and rows, enabling the storage of sparse data, the columns that are present can differ per row. They offer the same benefit of good read performance for analytics.

A key-value storage does not specify any schema or model for the data, this means the stored value can be anything. Data can be retrieved based on the key, but no additional constraints can be placed on the query [2].

A document database is a specific kind of key-value storage, but here, the value is not just any type of data, but a document with known attributes. Even though the attributes are known, this differs from a relational database, because not every data object, or in this case, document, has to have the same attributes. A document can even contain a nested document.

A few of the found databases can be categorized as data warehouses. A data warehouse is a specific type of database that is suitable for data analytics. [33]. The focus is more on analytics of historical data than on presenting the latest, real-time data. It contains data from all departments of an enterprise, offering an integrated view to support decision making.

Most databases store their data on disks, however, a subgroup of databases rely on main memory to store data. Using the main memory for storing the data eliminates the time it takes to search for and read data from a disk. The result is better performance, however, to achieve data persistence, storage on disk is still needed. This can be achieved by writing snapshots of the data to disk in time intervals and keeping a log of transactions after the most recent snapshot. So while in memory databases provide a

better performance, recovering from failure is more complicated and the operating costs are higher due to the needed memory.

Chapter 3

Problem context

–This chapter has been removed for confidentiality reasons–

Chapter 4

Design

–This chapter has been removed for confidentiality reasons–

Chapter 5

Evaluation

The aim of the new architecture is to provide a scalable environment for analysing traffic data. We use a prototype consisting of Cassandra and Spark to measure the scalability of the proposed architecture. We are interested in the scalability of volume and velocity. The scalability of volume ensures data can be stored and analysed for a long period. The scalability of velocity ensures the system can ingest large and varying amounts of data each minute. The dashboard with visualized data created in Tableau will be evaluated for its usefulness and ease of use.

5.1 Scalability of volume

We want to find out if our design is scalable in terms of the data volume that can be analysed.

We can define the scalability of volume in several different ways:

Definition 5.1.1. When increasing the amount of data, the time for the analysis should increase approximately linearly

Definition 5.1.2. When increasing the ”computing resources” the time for the analysis should decrease approximately linearly

Definition 5.1.3. When increasing the amount of data, the increase of computing resources necessary to keep the time of analysis approximately the same should be proportional.

A small preliminary experiment with our prototype was done for the second type of scalability. The amount of data was kept small and constant (1 day), while the amount of memory and CPU cores was variable. This showed that memory had only a little influence in speeding up the analysis, while increasing the number of cores did result in faster analytics. The fact that CPU influences the performance of Spark is also corroborated by Ousterhout et al, who find that CPU is a bottleneck [34]. In the next sections we try to see how a larger amount of data affects the performance of the analysis and if the amount of cores still has a positive impact on the performance when more data is added. We will look at the scalability in terms of the first 2 definitions given above.

5.1.1 First experiment

Goal The goal of the first experiment is to see how our prototype performs under different workloads and with different amount of computing resources available. This information can be used to help us understand if the design is scalable.

Setup We will measure the time it takes to load all data from the three tables in the database into memory and perform the calculation of the delay on 5 predefined routes. In other words, we measure the time it takes to perform the following 6 steps:

1. Load three tables from Cassandra into Spark (traffic data, decoded location information and routes). Sparks datastores are lazy, meaning an action needs to be done before data is actually read, so after loading the three tables, a count on all three is done to ensure all data is actually loaded from Cassandra into Spark.
2. Calculation of the minimum, average and maximum delay each hour for the route a27 from Utrecht (knooppunt Rijnsweerd) towards Almere (knooppunt Almere)
3. Calculation of the minimum, average and maximum delay each hour for the route a20 from Hoek van Holland (Maasdijk) towards Gouda (knooppunt Gouwe)
4. Calculation of the minimum, average and maximum delay each hour for the route a12 from Utrecht (knooppunt Lunetten) towards Arnhem (knooppunt Velperbroek)
5. Calculation of the minimum, average and maximum delay each hour for the route a13 from the Hague (knooppunt Ypenburg) towards Rotterdam (knooppunt Kleinpolderplein)
6. Calculation of the minimum, average and maximum delay each hour for the route a27 (a27b) from Utrecht (knooppunt Rijnsweerd) towards Breda (Breda Noord)

To assess the scalability, we will perform these steps under several different conditions. We would like to see how different volumes of data affects the performance, so we'll repeat the experiment for 1, 2, 4 and 7 days of traffic data.

We also want to see how the available resources affect the performance. Our VM will have 8GB of memory, with 4GB allocated to the application submitted in Spark. This should be plenty to keep the traffic data in memory. We'll repeat the experiment for the different volumes of data with different amounts of CPU available. Within our VM we are able to to use either 2, 3 or 4 cores. Our prototype does not execute with only 1 core available.

To ensure reliability of the results, we have repeated the 6 steps for each condition a 100 times, taking the average times.

Data

The data can be found in appendix A and the visualized results in figure 5.1 to 5.6

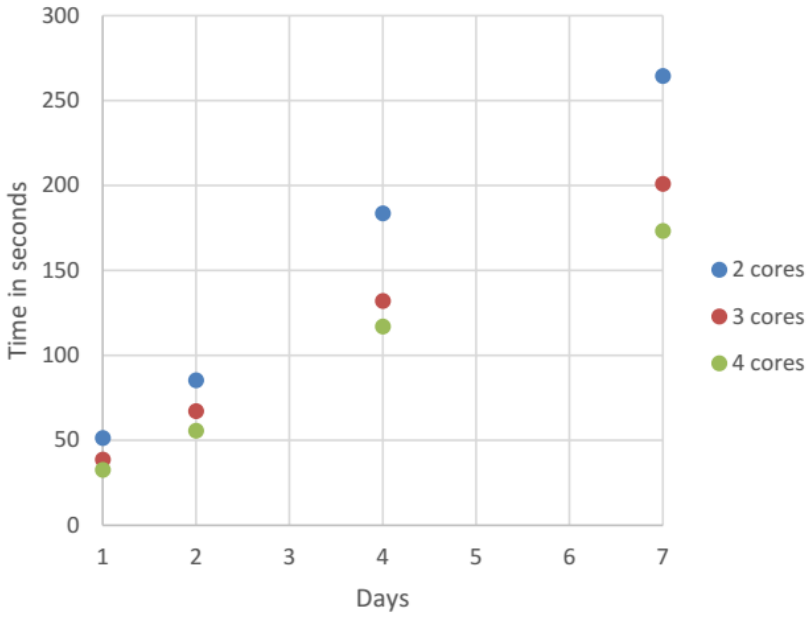


Figure 5.1: Loading data from database into memory

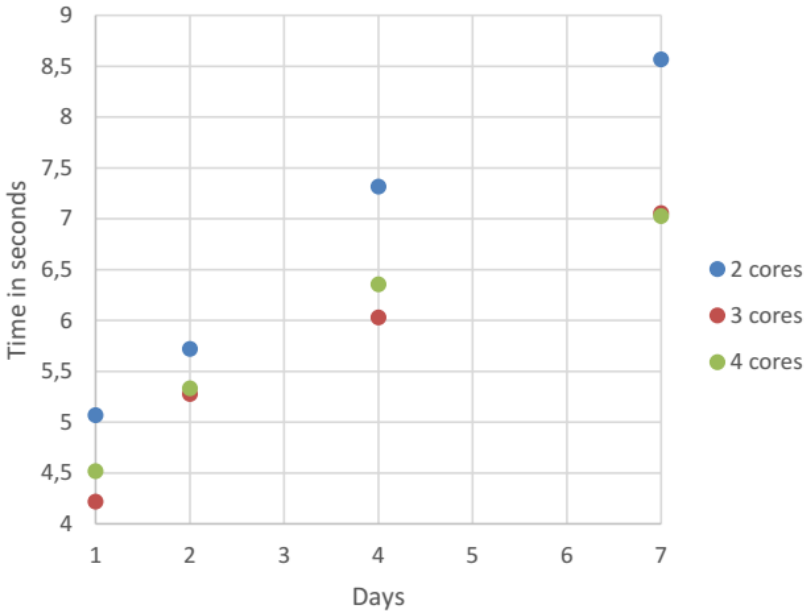


Figure 5.2: Calculating delays for a27

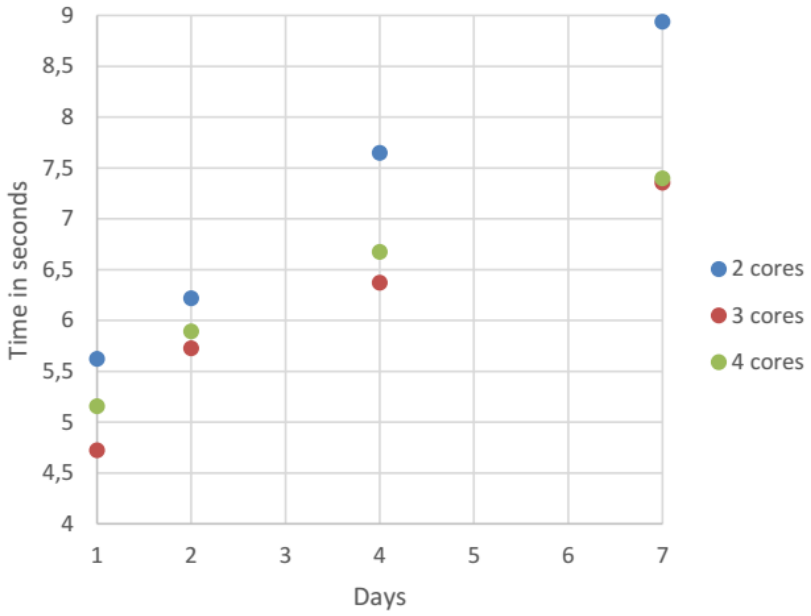


Figure 5.3: Calculating delays for a27b

Results

Tables A.1 to A.7 show the result of the first set of experiments. In tables A.1 to A.3 the effect of changing the data volume can be seen, relating to definition 5.1.1. According to this definition, the system is very scalable when it comes to the analysis of the data, but not when looking at the loading of the data. This task of reading the data from database into memory is the costliest task, that also scales the least. In some cases, this shows a more than linear increase in time. The expectation was that the system would scale linearly, however, this only happens for loading the data. For the analysis of the routes the system performs much better in terms of scalability.

Tables A.4 to A.7 show the effect of varying the amount of computing power while keeping the amount of data similar, relating to definition 5.1.2. We used the number of cpu cores as an indicator of computing power. The expectation is that adding more computing power will lower the time for an analysis significantly. The ratio should be low to indicate a better scalability. The system is not very scalable when looking at this definition. The use of more computing power does in some cases speed

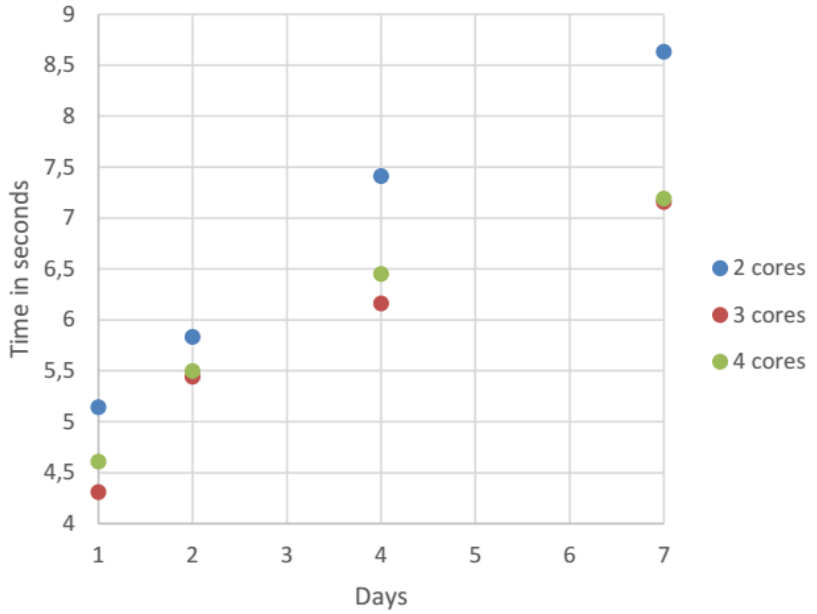


Figure 5.4: Calculating delays for a12

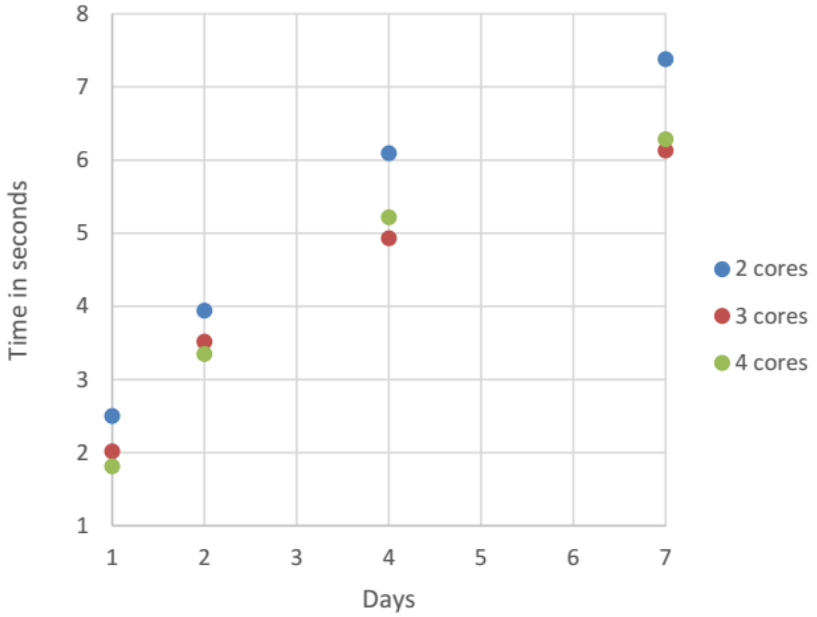


Figure 5.5: Calculating delays for a13

up the analysis, but never by a great amount. When using 4 cores, the analysis doesn't speed up at all, in some cases it even slows down.

Figures 5.1 to 5.6 show the results under different circumstances for each of the measured steps described in the setup, so loading the data and calculating delays on 5 different routes. The figures clearly show that analysing more days of data requires more time, but that the increase in time is less than proportional with the increase in data. The figures also show how using 2 cores has by far the worst performance in all cases, but using 3 or 4 are very similar in performance, with 3 cores even outperforming 4 in the analysis.

Explanation

So far the experiments have indicated that the system is scalable when adopting the first definition of scalability presented at the beginning of the chapter. Doubling the amount of data does not cause the time for the analysis to double. However, taking the second definition the system appears to be less

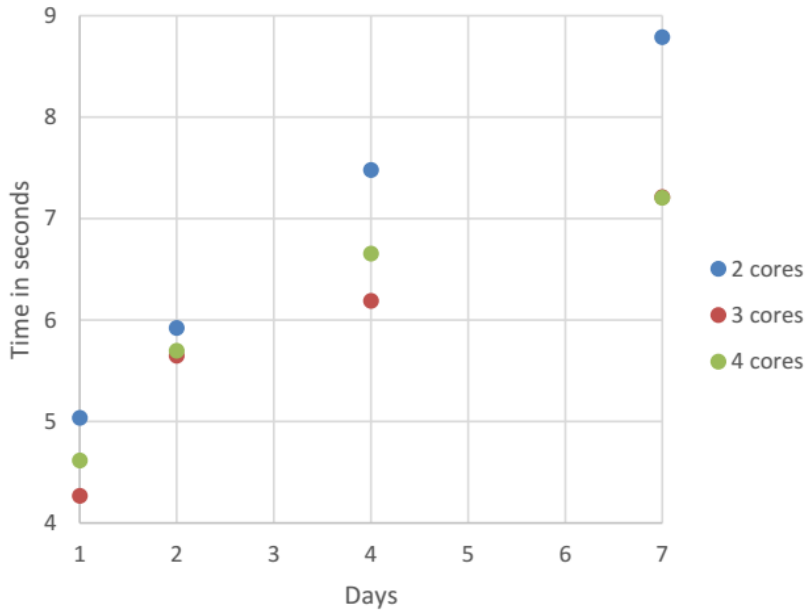


Figure 5.6: Calculating delays for a20

scalable, although this is hard to quantify, what would be considered a doubling of resources? The experiments so far have been done with a small volume of data, considering the chosen data platform and processing framework are intended for much larger volumes, the results might be affected by the overhead of the chosen processing framework. Using additional computing resources is not always beneficial during the analysis. Perhaps there is a trade-off where the amount of data becomes so large that using 4 cores will actually become more beneficial than using 3.

5.1.2 Second experiment

Goal

In the first experiment, we chose to first load the entire dataset into memory before doing the analysis. The rationale behind this was that since Spark is a framework for in-memory cluster computing and in general, memory performs faster than disk, this would be beneficial for the speed of the analysis. However, the first experiment indicated that loading the data in memory is a time consuming task and also the task that scales the least well. Therefore, in our second experiment, we repeat the same analysis of the 5 routes as in the first experiment, but without the preliminary step of first loading the entire dataset into memory. Instead, the analysis is slightly adjusted to only retrieve the data relevant for the specific route from the database and include this in the measurement per route. With this experiment we want to find out if this new manner of doing the analysis has a different result in terms of scalability. Moreover, we want to do the analysis over a larger amount of data, to see what happens to the scalability with more than 7 days of data.

Setup

The setup for the second experiment will be very similar to the first one. We will measure the time it takes for 5 routes to retrieve the data relevant to that route and calculate the delays per hour on that route. The routes will be the same ones as those used in the first experiment.

Again, we measure the time under different circumstances. We will evaluate the same amounts of data as in the first experiment, but also add more days of data. Our VM will still have 8GB of memory, with 4GB allocated to the application submitted in Spark. We'll repeat the experiment for the different volumes of data with different amounts of CPU available. Within our VM we are able to use either 2, 3

or 4 cores. To ensure reliability of the results, we have repeated the 5 calculations for each condition a 100 times, taking the average times.

Data

The data can be found in appendix B and the visualized results in figure 5.7 - 5.11.

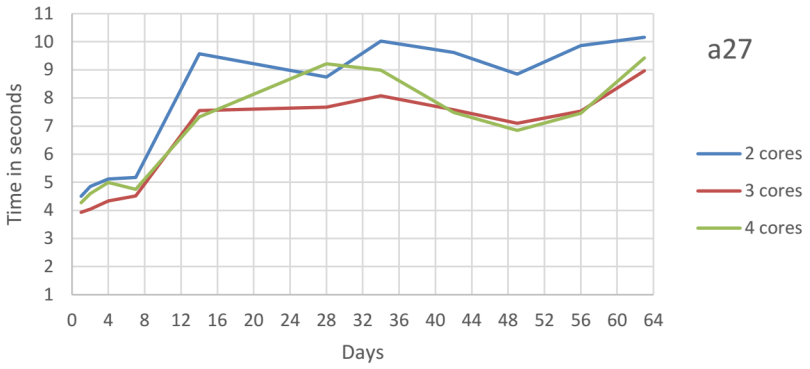


Figure 5.7: Calculating delays for a27

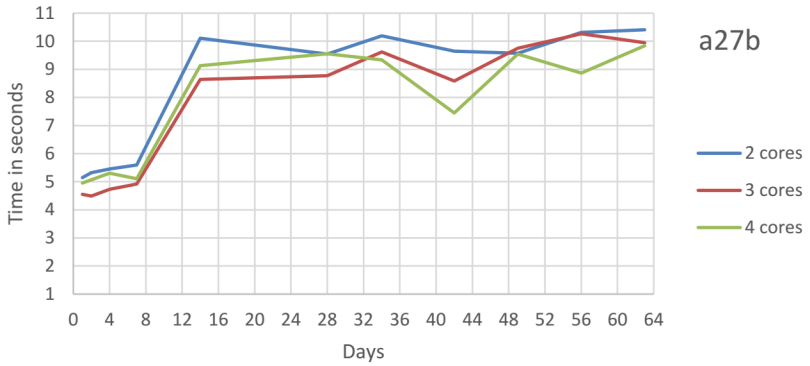


Figure 5.8: Calculating delays for a27b

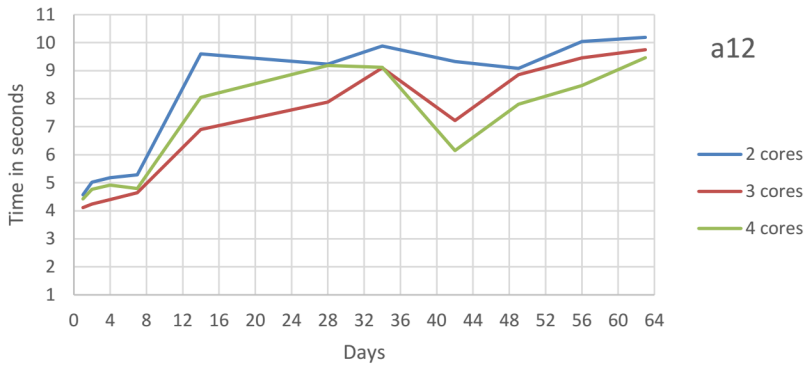


Figure 5.9: Calculating delays for a12

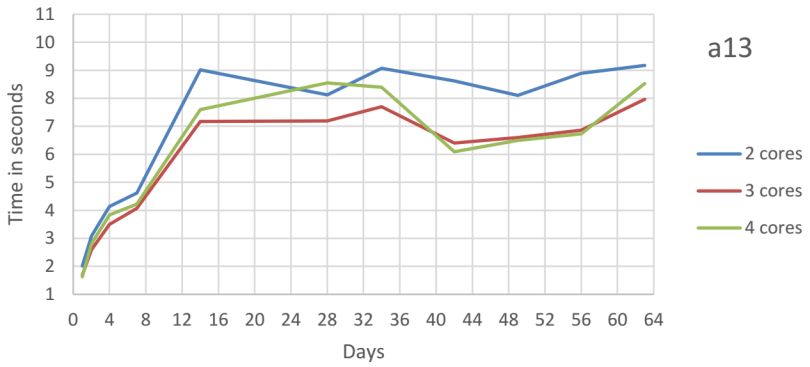


Figure 5.10: Calculating delays for a13

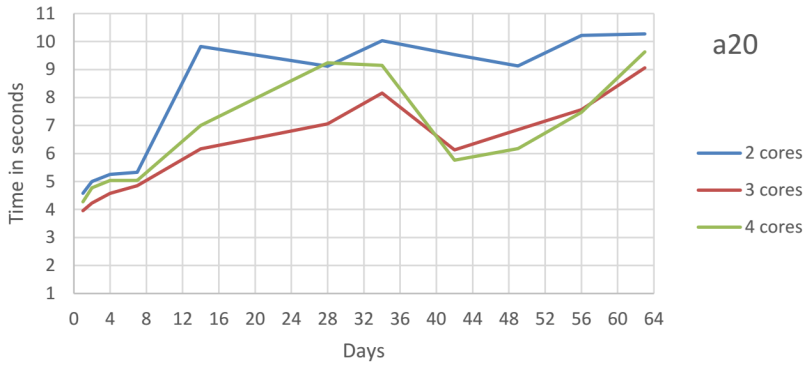


Figure 5.11: Calculating delays for a20

Results

The second round of experiments shows the prototype is not perfectly scalable (figure 5.7 - 5.11). The system scales when adopting definition 5.1.1, but not according to definition 5.1.2. When looking at the times for analysing up to 7 days, comparing them with the results from section 5.1.1, this second way of conducting the experiments is actually faster than the first round of experiments.

The expectation, based on the first round of experiments, would be for each of the routes to show a slight increase in the amount of time when adding more data. Although you could interpret the graphs as showing an increasing trend, the increase proceeds very irregular. Moreover, it is expected that with a larger amount of data the difference in performance between different amounts of computing resources would become more pronounced. This is not the case.

After 7 days the prototype shows a leap where a larger amount of data requires more additional time. In no case does the time required actually double, but the sudden increase is remarkable compared with the other data points. Another thing that stands out is that in some cases the time required for the analysis decreases with more data, most notably when analysing 42 days or 6 weeks.

Explanation

The fact that omitting the loading of all data in memory offers a faster analysis compared to the first experiment makes sense. Loading all the data in memory results in a large dataset in memory. Retrieving only the relevant data, based on time and location, from the database during the analysis can be done faster, since only a small fraction from the data is required. So instead of having to search through a large dataset in memory that has no index for the needed data, the small part of actually relevant data

can be retrieved from the database that has location and time as primary key in its schema. Why the provided computing power doesn't always have a clear influence or why the analysis speeds up with 6 weeks of data is hard to assess without further information. Both Spark and Cassandra run in the same virtual machine. Although the Spark Master is responsible for managing the available resources for Spark, it is not entirely clear if and how the Cassandra Spark connector is responsible for dividing the resources between the two. Inspecting the physical query plans of Spark for different amounts of data all show the same plan.

Other directions that might be investigated are the amount of memory and the garbage collecting strategy. It might be possible that to optimally utilize additional computing resources in the form of cpu cores more memory is required. Garbage collecting(GC) is an activity that might have influence on the performance of both Spark and Cassandra. During our experiments we used default settings, however, a short trial with a different GC strategy on one route analysing 4 and 6 weeks showed different results from the ones in figure 5.7 to 5.11, with 4 cores actually offering faster performance. This should be further investigated.

5.2 Scalability of velocity

Goal

To get a picture of how our system can deal with the velocity of new data, we conduct the following experiment. We want to see if the system is able to ingest different volumes of data, so we will measure the time it takes to insert different amounts of data. In particular, we want to find out how many data the system can store each minute.

Setup

We measure the time it takes to insert 10000, 100000 and 1000000 data points into a table in our Cassandra database. Our data is provided in csv files. Therefore, we will insert it by having Spark read the csv files, take the specified amount of data points and write these to Cassandra. We measure the time it takes to write the data to Cassandra in our virtual machine with 4 cores and 8GB of memory. One data point consists of a combination of one string, 3 integer values and a timestamp, as specified by the table in listing *4.1*. For each amount of data we repeat this 10 times consecutively.

Data

	10000 data points	100000 data points	1000000 data points
	1,993	9,427	117,39
	0,962	6,659	94,843
	0,925	7,722	152,137
	0,734	8,451	161,240
	0,611	9,029	156,468
	0,878	12,578	151,068
	0,609	12,56	152,127
	0,962	14,294	156,942
	0,818	13,135	146,613
	0,909	14,082	147,011
Avg time (seconds)	0,9401	10,7937	143,584

Table 5.1: Time for inserting data

Results

All measurements from this experiment can be found in table 5.1, with the averages at the bottom. It shows the prototype should be able to keep up easily with 10000 or 100000 new data points to write to the database each minute, but for 1 million data points the prototype will fall behind. The averages are plotted in figure 5.12, showing the insert time has an almost linear behaviour. Looking at this figure, the estimated amount of the traffic data we use that can be stored each minute is slightly above 400.000 data points consisting of a string, timestamp and 3 integer values.

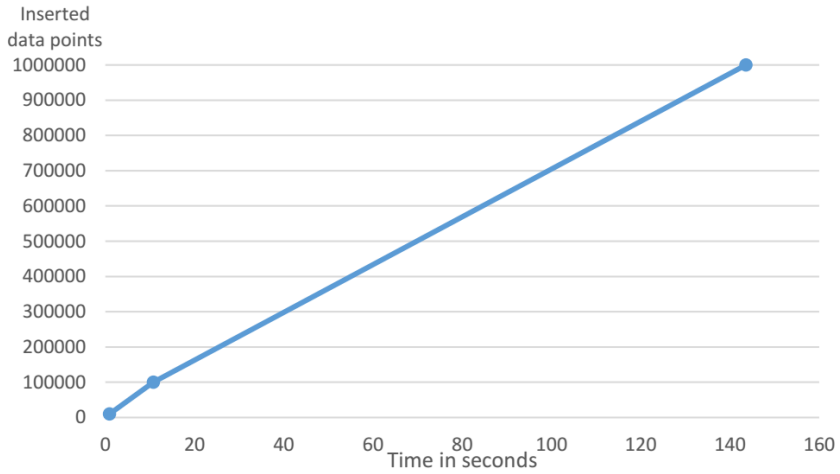


Figure 5.12: Average time for writing data

Explanation

In section *4.3* we already discussed some literature, where some works stated Cassandra was scalable for reads and some for writes. The result of the velocity experiment is meeting the expectations.

5.3 Evaluate the new information service

The new information service that was created and presented in the form of a dashboard needs to be evaluated. We use the extended Technology Acceptance Model (TAM) [50] as a guideline to perform this evaluation. Although the goal of this model is to predict user acceptance, it does so based on a user's perception of the usefulness and ease of use of a system, or, in our case, a dashboard. The model identifies social influence processes and cognitive instrumental processes that affect a user's willingness to adopt a technology. These are tested based on several statements, or Likert items, that a user can rate.

5.3.1 Questionnaire

We are mainly interested in the cognitive constructs (job relevance, output quality, result demonstrability and perceived ease of use). Since the dashboard is not in use the social constructs (subjective norm, voluntariness and image) cannot yet be assessed. We have used the suggested statements for the cognitive constructs from TAM [50] as a basis for our questionnaire, which can be found in appendix C. Our questionnaire consists of a set of statements that a user has to rate on a 7 point Likert scale, where a user can indicate how strongly he agrees or disagrees with the statement, with 1 indicating strong disagreement and 7 indicating strong agreement. The questionnaire is finished with 2 open questions to leave some room for additional comments.

5.3.2 Evaluation method

For the evaluation of the dashboard, we use the opinion of domain experts on the statements in our questionnaire. We asked domain experts since end users were not available for our questionnaire and domain experts are considered to have knowledge on the processes in which end users would use the dashboard. We were able to get the opinion of 3 people who have the function of product manager within Simacan, they have frequent contact with customers and are aware of their needs and wishes. The dashboard was also evaluated by 2 people who were involved in developing a similar dashboard, the one described in section *4.7*. They can also be considered domain experts. The evaluation was done by each of these 5 people independently. They were given access to the dashboard and the questionnaire without instructions on how to use the dashboard. They were asked to explore the dashboard and then fill in the questionnaire.

We are not necessarily interested in the cohesion between the different items on the questionnaire, as was studied in TAM, but in the actual responses on each item, because we want to get an idea of how the dashboard is perceived. Therefore, we study the results of the individual Likert items in each construct.

5.3.3 Evaluation results

The individual responses to the questionnaire can be found in appendix D. The answers to the open questions are translated from Dutch and slightly shortened, because they contained some contact information and other remarks not relevant to the evaluation. The questions relate to the cognitive constructs mentioned in section 5.3.1. A summary of the results can be found in table 5.2. We give the average and mode. The average score on the 7 point Likert scale is based on the 5 responses. To get a better image of the responses we also include the mode, the most chosen answer. We'll discuss the results of the questionnaire based on the item scores within each of the different constructs.

Intention to use

The intention to use the dashboard is measured with the first two questions of the questionnaire. The responses show that there definitely is the intention to use the dashboard. The question on using the dashboard regularly is met with somewhat more neutral responses. When looking at the function of the dashboard, as a supporting tool to gain insight in behaviour of traffic, this makes sense.

Perceived ease of use

The third to fifth question are referring to the perceived ease of use. One respondent did find it a bit unclear how to use the dashboard, but in general the attitude towards the ease of use is positive. The dashboard is considered easy to work with.

Job relevance

Job relevance is assessed with questions 6 and 7. Here the respondents have a relatively neutral opinion, most slightly agreeing, some slightly disagreeing. Although the evaluation shows there is a slightly positive attitude to job relevance, the dashboard might be improved with other relevant functions and additional information to gain a better score.

Output quality

The quality of the information presented was considered sufficient in question 8.

Result demonstrability

Questions 9 to 11 assess the extent to which a user perceives the results of using the system as tangible. Respondents would be able to explain the output of the dashboard easily and can also tell others how this output can be used. Some respondents would find it difficult to explain why this output would or would not be beneficial, however, most results in this category indicate there is a positive attitude towards the output of the dashboard offering tangible results.

Perceived usefulness

Question 12 and 13 gauge the perceived usefulness. On average, the dashboard is perceived as a somewhat useful tool by the respondents. The improvements to get a better score for job relevance will increase the perceived usefulness.

Other feedback

The last two questions were open questions, intended to offer respondents the ability to provide feedback and suggestions for improvement, since the Likert-style questionnaire doesn't leave room for such responses. This resulted in suggestions to improve the dashboard in various areas.

The main improvement that was mentioned multiple times would be to show other information besides averages. In particular, using a median or the distribution would be a better way to show the fluctuations in the delay. Using the average also hides the existence of any outliers. Other useful additions relating to functionality and usefulness would be the use of more sources of data, being able to add routes or comparing the data with the real-time traffic situation. With these suggestions, the dashboard could be improved to achieve a better score on job relevance and usefulness. The ability to zoom in charts about a larger time period or to manually set the time granularity could make some of the charts easier to interpret. Finally, some suggestions related to improving the ease of use. Having date and period selection boxes in more than one place makes the dashboard unclear, so this should be managed in one central place. The buttons to navigate between the two pages of the dashboard don't stand out enough, they should be more notable.

Statement	Average	Mode
Intention to use		
If I would have access to the dashboard, I intend to use it	4.8	6
I would make regular use of the dashboard	3.6	4
Perceived ease of use		
It is clear to me how I should use the dashboard	5.2	6
Using the dashboard doesn't require a lot of effort	5.6	6
I find the dashboard easy to use	5.2	5 & 6
Job relevance		
In my area of work the use of the dashboard is important	4.4	5
In my area of work the use of the dashboard is relevant	4.6	5
Output quality		
I find the presented information of good quality	5.0	5
Result demonstrability		
I could easily explain the information offered on the dashboard to others	5.8	5 & 6
I could explain to others how the information on the dashboard can be used	5.2	5 & 6
I would find it hard to explain to others why the dashboard would or would not be beneficial	4	3 & 5
Perceived usefulness		
The dashboard is useful in my area of work	4.8	5
I think the dashboard could help me in my job	4.4	5

Table 5.2: Results of the questionnaire on a 7 point Likert scale, with 1 indicating strong disagreement, 7 indicating strong agreement

Chapter 6

Conclusion

In this thesis, we set out to achieve scalability in a traffic information services environment, to accommodate the ever growing amount of traffic data available and be able to use this data to create new information services. Information services based on this large volume of data can help understand the patterns and changes in congestion of traffic, which in turn can be used by parties involved in road- or traffic management. In this final chapter we provide an answer to the research questions. We also discuss the limitations of our work and the contribution. We propose some future work and give recommendations based on our findings.

6.1 Answers to research questions

6.1.1 How can scalability be achieved in a traffic information environment?

This question was answered through the following sub questions.

1.1 What is the current architecture?

We studied the architecture of providers of traffic information services based on interviews with employees of a Dutch provider of such services. The results can be found in chapter 3. We found there are numerous sources and types of traffic data. Traffic data can be about intensity, locations of traffic jams or driving speeds, for example. We studied a part of the enterprise architecture, based on the use case of a customer interested in changing traffic patterns. We found the architecture consists of several separate applications, operating independent of each other. Within the architecture, data is stored only for a short period of time, at most 2 weeks, before it is moved to an archive. NIST offers a big data reference architecture, which offers guidelines for creating an architecture suitable for handling big data. We have used this reference architecture to identify areas of improvement to facilitate the analysis of larger volumes of traffic data than currently possible, in order to support new traffic information services. This resulted in two framework components from the reference architecture that can be used to improve the enterprise architecture for offering traffic information services, namely the data platform and the analytic framework. A suitable data platform in the enterprise architecture should enable the storing and accessing of larger volumes of data, while the analytic framework supports the querying and analysis of this larger data volume. The applications in the current architecture don't have the ability to analyse long term data, so the architecture could be improved by implementing analytics that leverage the analytic framework and creating corresponding visualizations.

1.2 What big data solutions exist for traffic data?

We explored this question in chapter 2. We initially focused on spatial data, but found no work regarding traffic data. Looking at big data solutions for traffic data, several architectures are proposed, with only

one evaluated in terms of performance. Specific analytics are often built upon MapReduce, a batch processing framework. The type of analysis done on traffic data often uses clustering to group the data, finding regular patterns for working days, weekends and holidays. This can then be used to gain insight in traffic behaviour and to predict future traffic on a specific type of day. Visualisation of traffic data is often done through line charts and contour plots, although some more ingenious designs can also be found in literature.

1.3 How can scalability of a traffic information environment be measured?

Before we can measure the scalability of a traffic information environment two steps need to be taken. First, we need to define what scalability is. We explored the concept of scalability in the literature in section 2.5. We used this knowledge to formulate our own definition of scalability in section 5.1. In particular, we adopt definition 5.1.1 and 5.1.2, where scalability in the form of definition 5.1.3, which we consider the most complete definition of scalability, can be derived from the first two definitions.

Next to a definition of scalability, we also need to define how are going to measure the scalability. We are interested in finding out the suitability of our design for the specific purpose of analysing traffic information. We therefore measured the scalability of volume by measuring the time it takes to analyse the delays on some defined routes for different volumes of data. The scalability of velocity was measured by inserting different amounts of traffic data in the used database.

1.4 Is the traffic information environment scalable in terms of analyzing large data volumes?

We made a prototype consisting of Cassandra for storing the data and Spark for the analytics. We inserted different amounts of data and then performed an analysis of delay times on specified routes. We used this setup and analysis to say something about the scalability in terms of definition 5.1.1 and 5.1.2. When looking at definition 5.1.1, the architecture is definitely scalable. Although the performance is not as stable as expected, the time does not increase drastically when analysing more data. When looking at definition 5.1.2, the architecture is not scalable and shows unexpected results, where adding more computing resources makes the system go slower in several cases. No proper explanation for this phenomenon could be identified, requiring further investigation.

1.5 Is the traffic information environment scalable when ingesting data?

The velocity scalability was studied by measuring the time it takes to insert data into a table like the one in listing *4.1*. Only the scalability according to definition 5.1.1 was assessed by inserting different amounts of data and measuring the time to write the data to the database. The result showed that the database performance is almost perfectly linear for inserting data. The estimated amount of data that can be stored each minute is 400.000 rows of traffic data.

6.1.2 How can a scalable traffic information system provide added value?

This question was answered through the following sub questions.

2.1 Which new information services are enabled by the scalable architecture that are of value to the service consumer?

The scalable traffic information environment enables the access to a large amount of data for analytics and mining. With the traffic data available to us, we can provide information like length and duration of traffic jams, speeds, delays and travel times. Because the scalable architecture offers the possibility to analyse longer periods of data, this means we can now show the fluctuations in these values, daily and weekly patterns and compare different periods with each other. For our prototype we have focused

on visualizing the delay in minutes on routes, however, it is possible to provide other kinds of traffic information in this architecture, for example driving speeds or duration of traffic jams.

2.2 How should a dashboard be designed to visualize the new information service?

The dashboard is created from a need to get insight in changes, patterns and statistics of traffic. We used the output of the analysis for measuring the scalability to create visualizations. In other words, we used information about the average delay in minutes on a route for displaying on a dashboard. A prototype dashboard was made in Tableau, with figures *4.3* to *4.6* showing the design of the dashboard. The prototype served to demonstrate that the scalable architecture can be used to provide new information services and to evaluate the usefulness of this new information. An evaluation was done by domain experts with knowledge of the wishes and needs of service consumers. The evaluation showed that in general the dashboard was perceived as useful and easy to use, although it could still be improved, which is as expected, since it concerns a prototype. In particular, the usefulness could be improved with more functionality and additional information about the delays, like the normal distribution.

6.2 Limitations

There are several limitations to our work. The enterprise architecture was studied from the point of view of one customer, for the sake of scope, resulting in a limited overview instead of the complete enterprise architecture. The improvements were done based on a specific use case for storing and analysing a larger volume of traffic data. This resulted in suggested improvements for this specific use case within the part of the enterprise architecture that was studied, however, this does not necessarily mean this is the best improvement when looking at the whole enterprise architecture, nor does it mean that there are no other improvements that could be done.

When looking at the evaluation of the suggested improvements, there are some areas of concern. The scalability of volume was measured using data for different numbers of days, however, not each day had the same amount of data, so doubling the number of days' worth of data does not necessarily double the actual volume of data. We chose to do this since this we wanted to represent the reality as close as possible, however, it makes it harder to interpret the results and draw conclusions from them. Furthermore, we only measured the scalability based on one type of traffic data, with one type of analysis. Ideally, we would also like to know how different types of traffic data and analysis influence the scalability. Another limitation is that only the scaling up has been measured, not the scaling out. We did the analysis on a single machine, however, it would be interesting to know if the architecture can also scale out.

The scalability of velocity was only measured with a small range of different values. We only measured the time it takes to insert 3 different amounts of data. We also did not take into account any preprocessing that might have to be done on the data when it is received, only measuring the time it takes to write a certain amount of data to the database.

6.3 Contribution

6.3.1 Contribution to theory

In this thesis, we investigated the applicability of big data solutions to traffic data analytics. We have shown the usefulness of the NIST reference architecture in improving the enterprise architecture to support long term data storage and analysis. We have used this reference architecture to propose an architecture that is relevant to any provider of traffic information services. We found that literature on analytics of traffic data is dominated by MapReduce, however, we have shown that Spark, a more recent

processing framework than MapReduce, is also very much suitable for analysis in the area of traffic data. Finally, we have measured the scalability of Cassandra and Spark for the purpose of analysing traffic data. Although several works already exist that discuss the scalability of both, their applicability to a traffic information environment hasn't been studied.

6.3.2 Contribution to practice

We elicited the need to analyse traffic data to identify patterns and shifting trends in congestion of traffic, in order for a provider of traffic information services to offer new services. To facilitate this, we studied parts of the enterprise architecture and identified elements that can be added to acquire scalability. These are a more scalable data platform suitable for storing large volumes of data that need to be analysed and a processing framework capable of performing these analytics. We created a prototype with these elements to evaluate its scalability and to support the development of a new information service in the form of a dashboard. A prototype dashboard was also made, to demonstrate the possibilities the scalable environment offers. Although the prototype only had limited visualizations, it confirmed that our proposed scalable architecture can provide the wanted information services that show patterns and trends. These services can be used by the customers of Simacan and other providers of traffic information. They can be used by traffic news outlets, providing news about growth or decline in the amount of congestion, for example. Information about the regular patterns of congestion can be useful for any logistics party involved with the planning of transportation. The information about changes in congestion can be used by any party in the area of road management, to support decision making concerning changes to the road network.

6.4 Future research

There are still some areas that could use further research. First of all, it should be investigated where the limit lies of scalability on a single machine, which is easily capable of analysing 2 months of data. In our case, the storage space within the virtual machine became the limiting factor during the end of our research. Then, it should also be investigated if the system can scale out. Considering the data platform and processing framework used, we assume this is possible, however, it should still be evaluated. It should also be investigated how the system handles different analytics on the traffic data or when it has to store types of traffic data. Some future work also lies in finding an explanation for the phenomena where the analytics go faster when adding more data and where the analytics don't go faster when adding computing resources.

The scalability of velocity could use some further study. In particular, it should be evaluated how much data can be ingested when the received data needs to be processed first and then stored and also if the performance degrades if large amounts of data are inserted constantly for a longer period of time.

6.5 Recommendations

The recommendations to Simacan can be done on two different levels. On the one hand there are the recommendations on the architectural level, concerning the improvements based on NIST's reference architecture. On the other hand, there are the recommendations on the level of our specific implementation of the architecture, regarding the use of Cassandra, Spark, Zeppelin and the dashboard.

Parallel to our work, a tool to view statistics was also developed at Simacan, but with a different approach from ours, more in line with their current architecture. This means that our proposed improvements based on the analysis of the architecture are not implemented yet at Simacan and are still applicable. Based on the exploration of the enterprise architecture and the comparison with the

reference architecture of NIST, we recommend the use of a processing framework to analyse large volumes of data. If the limitations of the currently used relational data platform is reached, we would also recommend the use of a column-oriented database for storing the data that needs to be analysed.

Concerning our specific implementation of the architecture our recommendations are somewhat more restrained. We would recommend to carry out the proposed future research to ensure the performance of our specific implementation of the architecture is as expected. In particular, it should be tested if the prototype also scales out and the scalability of velocity should be tested more thoroughly.

Appendix A

Results of first experiment

	1 day	2 days	4 days	7 days	deceleration 1 vs 2 days	deceleration 2 vs 4 days	deceleration 4 vs 7 days
load	51.440	85.352	183.643	264.430	1.66	2.15	1.44
a27b	5.622	6.218	7.648	8.938	1.11	1.23	1.17
a13	2.499	3.940	6.093	7.379	1.58	1.55	1.21
a27	5.068	5.720	7.316	8.568	1.13	1.28	1.17
a20	5.038	5.923	7.480	8.789	1.18	1.26	1.18
a12	5.143	5.834	7.412	8.633	1.13	1.27	1.16

Table A.1: Varying amounts of data with 2 cores

	1 day	2 days	4 days	7 days	deceleration 1 vs 2 days	deceleration 2 vs 4 days	deceleration 4 vs 7 days
load	38.635	67.133	132.147	201.031	1.74	1.97	1.52
a27b	4.725	5.728	6.371	7.356	1.21	1.11	1.15
a13	2.019	3.519	4.930	6.131	1.74	1.40	1.24
a27	4.218	5.276	6.030	7.056	1.25	1.14	1.17
a20	4.268	5.650	6.190	7.211	1.32	1.10	1.16
a12	4.310	5.441	6.162	7.157	1.26	1.13	1.16

Table A.2: Varying amounts of data with 3 cores

	1 day	2 days	4 days	7 days	deceleration 1 vs 2 days	deceleration 2 vs 4 days	deceleration 4 vs 7 days
load	32.564	55.693	116.987	173.316	1.71	2.10	1.48
a27b	5.156	5.894	6.674	7.399	1.14	1.13	1.11
a13	1.812	3.348	5.219	6.288	1.85	1.56	1.20
a12	4.610	5.497	6.452	7.192	1.19	1.17	1.11
a27	4.518	5.334	6.354	7.025	1.18	1.19	1.11
a20	4.611	5.698	6.656	7.204	1.24	1.17	1.08

Table A.3: Varying amounts of data with 4 cores

	2 cores	3 cores	4 cores	ratio 2 vs 3 cores	ratio 3 vs 4 cores
load	51.440	38.635	32.564	0.75	0.84
a27b	5.622	4.725	5.156	0.84	1.09
a13	2.499	2.019	1.812	0.81	0.90
a27	5.068	4.218	4.518	0.83	1.07
a20	5.038	4.268	4.615	0.85	1.08
a12	5.143	4.310	4.610	0.84	1.07

Table A.4: One day of data with various resources

	2 cores	3 cores	4 cores	ratio 2 vs 3 cores	ratio 3 vs 4 cores
load	85.352	67.133	55.693	0.79	0.83
a27b	6.218	5.728	5.894	0.92	1.03
a13	3.940	3.519	3.348	0.89	0.95
a27	5.720	5.276	5.334	0.92	1.01
a20	5.923	5.650	5.698	0.95	1.01
a12	5.834	5.441	5.497	0.93	1.01

Table A.5: Two days of data with various resources

	2 cores	3 cores	4 cores	ratio 2 vs 3 cores	ratio 3 vs 4 cores
load	183.643	132.147	116.987	0.72	0.89
a27b	7.648	6.371	6.674	0.83	1.05
a13	6.093	4.930	5.219	0.81	1.06
a27	7.316	6.030	6.354	0.82	1.05
a20	7.480	6.190	6.656	0.83	1.08
a12	7.412	6.162	6.452	0.83	1.05

Table A.6: Four days of data with various resources

	2 cores	3 cores	4 cores	ratio 2 vs 3 cores	ratio 3 vs 4 cores
load	264.430	201.031	173.316	0.76	0.86
a27b	8.938	7.356	7.399	0.82	1.01
a13	7.379	6.131	6.288	0.83	1.03
a27	8.568	7.056	7.025	0.82	1.00
a20	8.789	7.211	7.204	0.82	1.00
a12	8.633	7.157	7.192	0.83	1.00

Table A.7: Seven days of data with various settings

Appendix B

Results of second experiment

	1 day	2 days	4 days	7 days	14 days	28 days	35 days	42 days	49 days	56 days	63 days
a27b	5.146	5.320	5.451	5.596	10.109	9.542	10.196	9.651	9.571	10.319	10.409
a13	2.012	3.084	4.133	4.613	9.017	8.121	9.076	8.626	8.108	8.898	9.174
a27	4.503	4.854	5.118	5.169	9.572	8.745	10.023	9.621	8.850	9.866	10.159
a20	4.580	4.999	5.252	5.330	9.821	9.122	10.030	9.532	9.123	10.218	10.273
a12	4.570	5.017	5.181	5.283	9.603	9.229	9.877	9.331	9.084	10.038	10.190

Table B.1: Varying amounts of data with 2 cores

	1 day	2 days	4 days	7 days	14 days	28 days	35 days	42 days	49 days	56 days	63 days
a27b	4.553	4.490	4.732	4.915	8.639	8.775	9.619	8.581	9.746	10.266	9.944
a12	4.113	4.236	4.402	4.646	6.902	7.882	9.100	7.222	8.854	9.459	9.749
a13	1.688	2.597	3.497	4.072	7.175	7.191	7.700	6.406	6.598	6.868	7.965
a27	3.931	4.042	4.338	4.510	7.548	7.674	8.076	7.578	7.098	7.534	8.966
a20	3.958	4.227	4.574	4.846	6.166	7.055	8.156	6.125	6.856	7.563	9.060

Table B.2: Varying amounts of data with 3 cores

	1 day	2 days	4 days	7 days	14 days	28 days	35 days	42 days	49 days	56 days	63 days
a27b	4.952	5.065	5.306	5.110	9.134	9.549	9.333	7.451	9.541	8.867	9.840
a13	1.625	2.812	3.835	4.218	7.593	8.550	8.395	6.095	6.500	6.736	8.531
a12	4.423	4.769	4.919	4.792	8.051	9.183	9.116	6.151	7.800	8.474	9.464
a27	4.274	4.601	4.991	4.746	7.321	9.216	8.985	7.483	6.845	7.461	9.423
a20	4.275	4.775	5.042	5.041	7.044	9.239	9.147	5.763	6.180	7.477	9.630

Table B.3: Varying amounts of data with 4 cores

Appendix C

Questionnaire dashboard evaluation

1. Als ik toegang zou hebben tot het dashboard ben ik van plan hier gebruik van te maken.
   If I would have access to the dashboard, I intend to use it.
2. Ik zou geregeld van het dashboard gebruik maken.
   I would make regular use of the dashboard.
3. Het is duidelijk hoe ik het dashboard moet gebruiken.
   It is clear to me how I should use the dashboard.
4. Het gebruik van het dashboard kost me niet veel inspanning.
   Using the dashboard doesn't require a lot of effort.
5. Ik vind het dashboard makkelijk te gebruiken.
   I find the dashboard easy to use.
6. In mijn werkgebied is het gebruik van het dashboard belangrijk.
   In my area of work the use of the dashboard is important.
7. In mijn werkgebied is het gebruik van het dashboard relevant.
   In my area of work the use of the dashboard is relevant.
8. Ik vind de gepresenteerde informatie van goede kwaliteit.
   I find the presented information of good quality.
9. Ik zou de informatie op het dashboard makkelijk aan anderen uit kunnen leggen.
   I could easily explain the information offered on the dashboard to others.
10. Ik zou aan anderen uit kunnen leggen hoe de informatie op het dashboard gebruikt kan worden.
    I could explain to others how the information on the dashboard can be used.
11. Ik vind het moeilijk om uit te leggen waarom het dashboard wel of geen voordeel biedt.
    I would find it hard to explain to others why the dashboard would or would not be beneficial.
12. Het dashboard is nuttig binnen mijn werkgebied.
    The dashboard is useful in my area of work.
13. Ik denk dat het dashboard mij kan helpen bij mijn werk.
    I think the dashboard could help me in my job.

Open questions

14. Het dashboard zou nog verbeterd kunnen worden door:
    The dashboard can be improved by:

15. Ruimte voor eventuele opmerkingen.
    Other remarks.

Appendix D

Responses to questionnaire

Statement *(1 = strongly disagree - 7 = strongly agree)*	1	2	3	4	5	6	7
1. If I would have access to the dashboard, I intend to use it						x	
2. I would make regular use of the dashboard				x			
3. It is clear to me how I should use the dashboard						x	
4. Using the dashboard doesn't require a lot of effort						x	
5. I find the dashboard easy to use						x	
6. In my area of work the use of the dashboard is important			x				
7. In my area of work the use of the dashboard is relevant			x				
8. I find the presented information of good quality					x		
9. I could easily explain the information offered on the dashboard to others							x
10. I could explain to others how the information on the dashboard can be used					x		
11. I would find it hard to explain to others why the dashboard would or would not be beneficial					x		
12. The dashboard is useful in my area of work			x				
13. I think the dashboard could help me in my job			x				

14. The dashboard can be improved by	Use semi-transparency on the upper right chart on the first screen. Using a median would be better than using the average for showing the normal behaviour. Initially I didn't see the back button on the screen.
15. Other remarks	-

Table D.1: Response one

Statement	1	2	3	4	5	6	7
1. If I would have access to the dashboard, I intend to use it				x			
2. I would make regular use of the dashboard			x				
3. It is clear to me how I should use the dashboard					x		
4. Using the dashboard doesn't require a lot of effort						x	
5. I find the dashboard easy to use					x		
6. In my area of work the use of the dashboard is important					x		
7. In my area of work the use of the dashboard is relevant					x		
8. I find the presented information of good quality					x		
9. I could easily explain the information offered on the dashboard to others						x	
10. I could explain to others how the information on the dashboard can be used						x	
11. I would find it hard to explain to others why the dashboard would or would not be beneficial					x		
12. The dashboard is useful in my area of work					x		
13. I think the dashboard could help me in my job					x		

14. The dashboard can be improved by	Add new routes through a map. Zoom in on a chart to a finer time granularity. Show distribution together with averages. Manually tune the aggregation level to the desired time granularity.
15. Other remarks	Points of improvement are based on ideal image. Make the dashboard more interactive.

Table D.2: Response two

Statement	1	2	3	4	5	6	7
1. If I would have access to the dashboard, I intend to use it						x	
2. I would make regular use of the dashboard				x			
3. It is clear to me how I should use the dashboard			x				
4. Using the dashboard doesn't require a lot of effort				x			
5. I find the dashboard easy to use				x			
6. In my area of work the use of the dashboard is important				x			
7. In my area of work the use of the dashboard is relevant					x		
8. I find the presented information of good quality				x			
9. I could easily explain the information offered on the dashboard to others					x		
10. I could explain to others how the information on the dashboard can be used				x			
11. I would find it hard to explain to others why the dashboard would or would not be beneficial			x				
12. The dashboard is useful in my area of work					x		
13. I think the dashboard could help me in my job					x		

14. The dashboard can be improved by	Selecting dates and periods on multiple places makes the overview less clear. Averages don't say much about expected distribution, the average might be affected by an outlier, but this is impossible to distinguish.
15. Other remarks	-

Table D.3: Response three

Statement	1	2	3	4	5	6	7
1. If I would have access to the dashboard, I intend to use it			x				
2. I would make regular use of the dashboard		x					
3. It is clear to me how I should use the dashboard						x	
4. Using the dashboard doesn't require a lot of effort						x	
5. I find the dashboard easy to use						x	
6. In my area of work the use of the dashboard is important					x		
7. In my area of work the use of the dashboard is relevant					x		
8. I find the presented information of good quality						x	
9. I could easily explain the information offered on the dashboard to others						x	
10. I could explain to others how the information on the dashboard can be used						x	
11. I would find it hard to explain to others why the dashboard would or would not be beneficial			x				
12. The dashboard is useful in my area of work						x	
13. I think the dashboard could help me in my job					x		

14. The dashboard can be improved by	Use more different sources of data. Have an option to add other routes.
15. Other remarks	The dashboard is easy to use and useful. I wouldn't use the dashboard often within my work, but I can imagine traffic control centers can use this more often if it is easy to add new routes.

Table D.4: Response four

Statement	1	2	3	4	5	6	7
1. If I would have access to the dashboard, I intend to use it					x		
2. I would make regular use of the dashboard					x		
3. It is clear to me how I should use the dashboard						x	
4. Using the dashboard doesn't require a lot of effort						x	
5. I find the dashboard easy to use					x		
6. In my area of work the use of the dashboard is important					x		
7. In my area of work the use of the dashboard is relevant					x		
8. I find the presented information of good quality					x		
9. I could easily explain the information offered on the dashboard to others					x		
10. I could explain to others how the information on the dashboard can be used					x		
11. I would find it hard to explain to others why the dashboard would or would not be beneficial				x			
12. The dashboard is useful in my area of work					x		
13. I think the dashboard could help me in my job				x			

14. The dashboard can be improved by	Use percentiles (for example in a boxplot). Visualize the routes on a map. Show comparison with the real-time traffic congestion.
15. Other remarks	

Table D.5: Response five

References

[1] Abadi, D. J., Boncz, P. A., and Harizopoulos, S. Column-oriented Database Systems. In *2009 VLDB Endowment* (2009), pp. 1664–1665.

[2] Aerospike. What is a key-value store? http://www.aerospike.com/what-is-a-key-value-store/.

[3] Aji, A., Teodoro, G., and Wang, F. Haggis: Turbocharge a MapReduce based Spatial Data Warehousing System with GPU Engine. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial '14* (New York, New York, USA, 2014), ACM Press, pp. 15–20.

[4] Alsubaiee, S., Altowim, Y., Altwaijry, H., Behm, A., Borkar, V., Bu, Y., Carey, M., Cetindil, I., Cheelangi, M., Faraaz, K., Gabrielova, E., Grover, R., Heilbron, Z., Kim, Y.-S., Li, C., Li, G., Ok, J. M., Onose, N., Pirzadeh, P., Tsotras, V., Vernica, R., Wen, J., and Westmann, T. AsterixDB: A Scalable, Open Source BDMS. *Proceedings of the VLDB Endowment 7*, 14 (2014), 1905–1916.

[5] Amazon.com. Announcing geospatial indexing library for Amazon DynamoDB, 2013. https://aws.amazon.com/about-aws/whats-new/2013/09/05/announcing-amazon-dynamodb-geospatial-indexing/.

[6] Appuswamy, R., Gkantsidis, C., Narayanan, D., Hodson, O., and Rowstron, A. Scale-up vs Scale-out for Hadoop: Time to rethink? In *Proceedings of the 4th annual Symposium on Cloud Computing (SoCC '13)* (2013), p. Article 20.

[7] Bentley, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM 18*, 9 (1975), 509–517.

[8] Bondi, A. Characteristics of Scalability and Their Impact on Performance. In *Proceedings of the 2nd international workshop on Software and performance* (2000), pp. 195–203.

[9] CBS. Transport en mobiliteit 2015. Tech. rep., CBS, 2015.

[10] Chacon, S. H., and Kornhauser, A. Analysis, characterization, and visualization of freeway traffic data in los angeles. Tech. rep., 2009.

[11] Chen, W., Guo, F., and Wang, F.-y. A Survey of Traffic Data Visualization. *IEEE Transactions on Intelligent Transportation Systems 16*, 6 (2015), 2970–2984.

[12] Chen, X., Vo, H., Aji, A., Wang, F., Informatics, B., and Wang, F. High performance integrated spatial big data analytics. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial '14* (New York, New York, USA, 2014), ACM Press, pp. 11–14.

[13] Cui, X., Dong, Z., Lin, L., Song, R., and Yu, X. GrandLand Traffic Data Processing Platform. In *2014 IEEE International Congress on Big Data* (jun 2014), IEEE, pp. 766–767.

[14] DA SILVA, T. L. C., NETO, A. C. A., MAGALHAES, R. P., DE FARIAS, V. A., DE MACÊDO, J. A., AND MACHADO, J. C. Towards an Efficient and Distributed DBSCAN Algorithm Using MapReduce. In *16th International Conference on Enterprise Information Systems* (2014), pp. 163–170.

[15] DANGEL, U., MCDONAGH, P., AND MURPHY, L. Micro Analysis of Urban Vehicular Data for Enhanced Information Services for Commuters. In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)* (may 2014), IEEE, pp. 1–7.

[16] ELDAWY, A., AND MOKBEL, M. F. Pigeon: A spatial MapReduce language. *Proceedings - International Conference on Data Engineering* (mar 2014), 1242–1245.

[17] ELDAWY, A., AND MOKBEL, M. F. The Era of Big Spatial Data. In *2015 31st IEEE International Conference on Data Engineering Workshops* (apr 2015), IEEE, pp. 42–49.

[18] FEDORČÁK, D., KOCYAN, T., HÁJEK, M., SZTURCOVÁ, D., MARTINOVIČ, J., SZTURCOV, D., AND MARTINOVI, J. viaRODOS: Monitoring and visualisation of current traffic situation on highways. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2014), vol. 8838, Springer Verlag, pp. 290–300.

[19] FINKEL, R. A., AND BENTLEY, J. L. Quad trees a data structure for retrieval on composite keys. *Acta Informatica 4*, 1 (1974), 1–9.

[20] GUTTMAN, A. R-Trees: A Dynamic Index Structure For Spatial Searching. *84 Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (1984), 47–57.

[21] HAN, W., WANG, J., AND SHAW, S. L. Visual exploratory data analysis of traffic volume. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2006), vol. 4293 LNAI, Springer Verlag, pp. 695–703.

[22] HENDERSON, C. Big spatial data analytics, 2014.

[23] IBRAHIM, H., AND FAR, B. H. Data-Oriented Intelligent Transportation Systems. In *IEEE IRI 2014 15th International Conference on Information Reuse and Integration* (aug 2014), IEEE, pp. 322–329.

[24] JO, H., LEE, B., NA, Y.-C., LEE, H., OH, B., YUN, C., YANG, J., LEE, M., AND KIM, M. Prioritized Traffic Information Delivery Based on Historical Data Analysis. In *2007 IEEE Intelligent Transportation Systems Conference* (sep 2007), IEEE, pp. 568–573.

[25] KITCHENHAM, B. Procedures for Performing Systematic Reviews. Tech. rep., Keele University, Keele, 2004.

[26] LEE, J.-G., AND KANG, M. Geospatial Big Data: Challenges and Opportunities. *Big Data Research 2*, 2 (feb 2015), 74–81.

[27] LIU, S., PU, J., LUO, Q., QU, H., NI, L. M., AND KRISHNAN, R. VAIT: A visual analytics system for metropolitan transportation. *IEEE Transactions on Intelligent Transportation Systems 14*, 4 (2013), 1586–1596.

[28] LU, C.-T., BOEDIHARDJO, A. P., DAI, J., AND CHEN, F. HOMES: Highway Operation Monitoring and Evaluation System. In *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems* (2008), pp. 529–530.

[29] MAIER, M. *Towards a Big Data Reference Architecture*. PhD thesis, Eindhoven University of Technology, 2013.

[30] MICHAEL, M., MOREIRA, J. E., SHILOACH, D., AND WISNIEWSKI, R. W. Scale-up x scale-out: A case study using nutch/Lucene. In *Proceedings - 21st International Parallel and Distributed Processing Symposium, IPDPS 2007* (2007), pp. 1–8.

[31] NAAMI, K. M. A., SEKER, S., AND KHAN, L. GISQF: An Efficient Spatial Query Processing System. *2014 IEEE 7th International Conference on Cloud Computing* (2014), 681–688.

[32] NIST. NIST Big Data Interoperability Framework : Volume 6 , Reference Architecture. Tech. rep., NIST, 2015.

[33] ORACLE. Data Warehousing Concepts. https://docs.oracle.com/cd/B10500_01/server.920/a96520/concept.htm.

[34] OUSTERHOUT, K., RASTI, R., RATNASAMY, S., SHENKER, S., AND CHUN, B.-G. Making Sense of Performance in Data Analytics Frameworks. *NSDI* (2015), 293–307.

[35] PÄÄKKÖNEN, P., AND PAKKALA, D. Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems. *Big Data Research 2*, 4 (2015), 166–186.

[36] PICOZZI, M., VERDEZOTO, N., POUKE, M., VATJUS-ANTTILA, J., AND QUIGLEY, A. Traffic Visualization - Applying Information Visualization Techniques to Enhance Traffic Planning. In *International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications* (2013), pp. 554–557.

[37] RIJKSWATERSTAAT. Publieksrapportage Rijkswegennet. Tech. Rep. 2e periode 2015, Rijskwaterstaat, 2015.

[38] SAP. Golden rules for new SAP HANA developers. http://scn.sap.com/community/hana-in-memory/blog/2013/12/29/6-golden-rules-for-new-sap-hana-developers.

[39] SAYAR, A., EKEN, S., AND ÖZTÜRK, O. Kd-tree and quad-tree decompositions for declustering of 2D range queries over uncertain space. *Frontiers of Information Technology & Electronic Engineering 16*, 2 (feb 2015), 98–108.

[40] SCHRAMM, A., KWELLA, B., SCHMIDT, M., PIETH, N., AND ERNST, T. Universal Location Referencing: A New Approach for Dynamic Location Referencing in. Tech. rep., Fraunhofer FIRST, 2012.

[41] SHEKHAR, S., LU, C., LIU, R., AND ZHOU, C. CubeView: a system for traffic data visualization. *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, September (2002), 674–678.

[42] SHTERN, M., MIAN, R., LITOIU, M., ZAREIAN, S., ABDELGAWAD, H., AND TIZGHADAM, A. Towards a Multi-cluster Analytical Engine for Transportation Data. In *2014 International Conference on Cloud and Autonomic Computing* (sep 2014), IEEE, pp. 249–257.

[43] SONG, Y., AND MILLER, H. J. Exploring traffic flow databases using space-time plots and data cubes. *Transportation 39*, 2 (2012), 215–234.

[44] SORIGUERA, F. Deriving Traffic Flow Patterns from Historical Data. *Journal of Transportation Engineering 138*, 12 (2012), 1430–1441.

[45] SVENSK, P., AND WIKSTRÖM, L. Georeferencing Methods - A study on how to deal with georeferencing in the ROSATTE implementation platform. Tech. rep., Triona AB, 2012.

[46] TER MAAT, J. Investigating Possibilities of Big Traffic Data. Tech. rep., University of Twente, Enschede, 2016.

[47] TMC COMPENDIUM. Alert-C Coding Handbook Version F02.1. Tech. rep., 1999.

[48] TNO. Rapportage TNO Economische wegwijzer. Tech. rep., TNO, 2014.

[49] TOMTOM INTERNATIONAL B.V. OpenLR White Paper, 2012. http://www.openlr.org/data/docs/OpenLR-Whitepaper_v1.5.pdf.

[50] VENKATESH, V., AND DAVIS, F. D. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science 46*, 2 (2000), 186–204.

[51] WANG, W. Q., ZHANG, X., ZHANG, J., AND LIM, H. B. Smart Traffic Cloud: An Infrastructure for Traffic Applications. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems* (dec 2012), IEEE, pp. 822–827.

[52] WEI-FENG, L., XI, D., AND TONG-YU, Z. Research of Dynamic Location Referencing Method Based on Intersection and Link Partition. *International Journal of Computer, Electrical, Automation, Control and Information Engineering 2*, 10 (2008), 3296–3300.

[53] WEIJERMARS, W., AND VAN BERKUM, E. Analyzing highway flow patterns using cluster analysis. In *IEEE Intelligent Transportation Systems Conference* (2005), pp. 831–836.

[54] WIERINGA, R. *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014.

[55] WIKIMEDIA. Wikimedia Commons. commons.wikimedia.org.

[56] YOU, S., ZHANG, J., AND GRUENWALD, L. Scalable and efficient spatial data management on multi-core CPU and GPU clusters: A preliminary implementation based on Impala. In *2015 31st IEEE International Conference on Data Engineering Workshops* (apr 2015), vol. 2015-June, IEEE, pp. 143–148.

[57] YU, J., JIANG, F., AND ZHU, T. RTIC-C: A Big Data System for Massive Traffic Information Mining. In *2013 International Conference on Cloud Computing and Big Data* (dec 2013), IEEE, pp. 395–402.

[58] ZHANG, H.-S., ZHANG, Y., LI, Z.-H., AND HU, D.-C. Spatial Temporal Traffic Data Analysis Based on Global Data Management Using MAS. *IEEE Transactions on Intelligent Transportation Systems 5*, 4 (2004), 267–275.

[59] ZHANG, N., ZHENG, G., CHEN, H., CHEN, J., AND CHEN, X. HBaseSpatial: A Scalable Spatial Data Storage Based on HBase. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications* (sep 2014), IEEE, pp. 644–651.

[60] ZHAO, D., GU, Y., AND HUANG, Z. A new data-intensive parallel processing framework for spatial data. In *Lecture Notes in Electrical Engineering* (2014), vol. 277 LNEE, Springer Verlag, pp. 375–383.

[61] ZHONG, Y., HAN, J., ZHANG, T., LI, Z., FANG, J., AND CHEN, G. Towards Parallel Spatial Query Processing for Big Spatial Data. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum* (may 2012), IEEE, pp. 2085–2094.