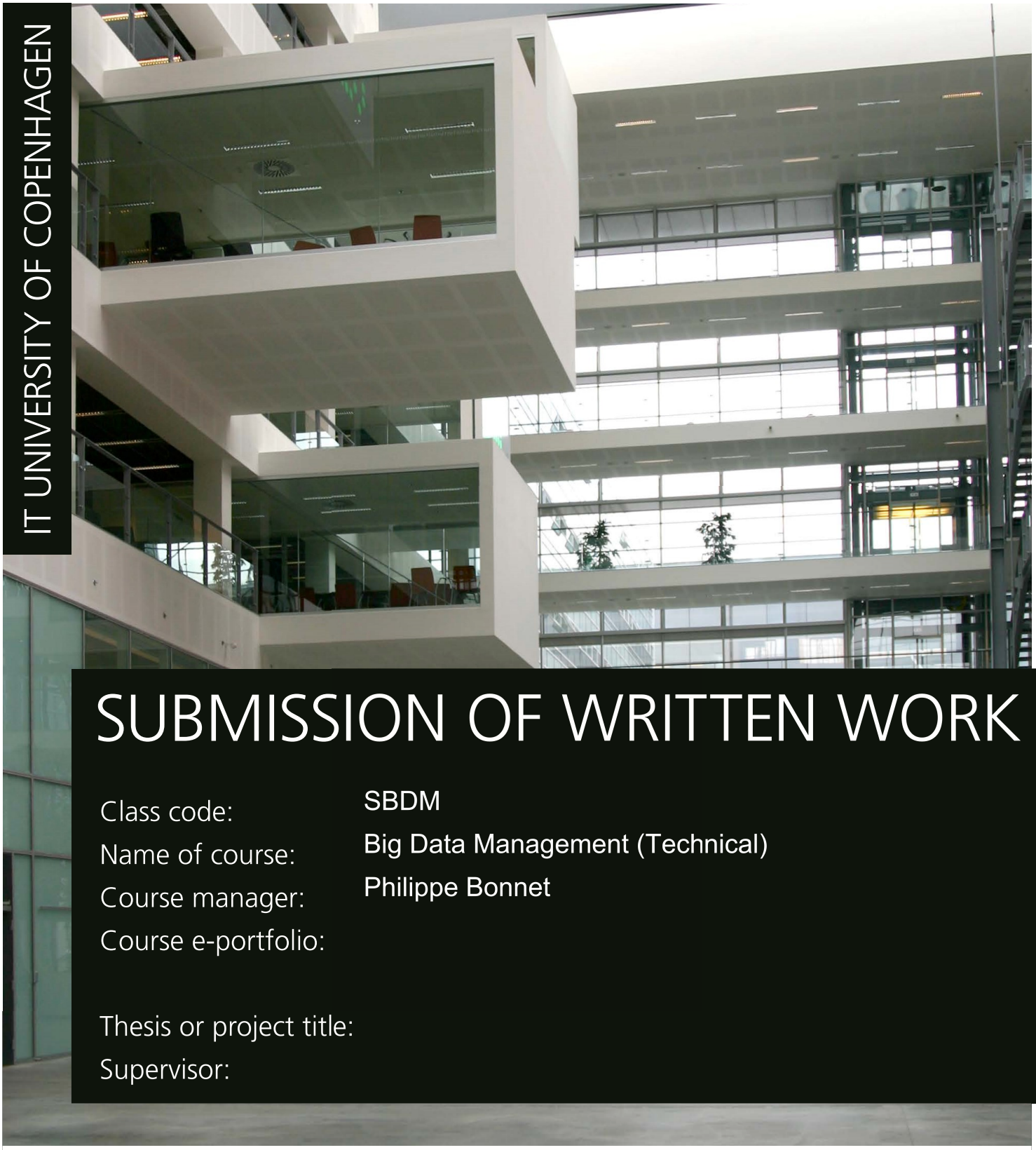
IT UNIVERSITY OF COPENHAGEN



SUBMISSION OF WRITTEN WORK

Class code: SBDM

Name of course: Big Data Management (Technical)

Course manager: Philippe Bonnet

Course e-portfolio:

Thesis or project title:

Supervisor:

	Full Name:	Birthdate (dd/mm-yyyy):	E-mail:
1.	Anders Wind Steffensen	10/02-1993	awis@itu.dk
2.			@itu.dk
3.			@itu.dk
4.			@itu.dk
5.			@itu.dk
6.			@itu.dk
7.			@itu.dk

Big Data Exam - Autumn 2016

Anders Wind Steffensen

December 13th 2016

Contents

1 Question 1

1.1 A)

> *"Consider Apache Flink:* `https://flink.apache.org`*. You should characterize this system, describe how it can be used in the context of the Lambda architecture and compare it with systems you have used during your projects."*

Apache Flink[Flink] is a streaming dataflow engine. It works in a distributed setting and makes analysis of data in motion, and data at rest easier. It incorporates multiple other systems, for machine learning, graph-analysis, and more. To further characterize Flink I will use the characterization model presented in the course.

Datamodel: Flink works on event-based streams of data. The specific format of the events are Java and Scala embedded objects. These streams can either be infinite such as a sensor which continuously sends data, or finite such as a file. Kapfka which is a stream gathering and storing framework based on HDFS, is used by Flink to get its stream of events[3]. If events come out of order in real-time, Flink is able to sort them based on logical time instead, which can greatly simplify for example windowed computations.

Partition Management: To be able to scale, Flink partitions the computations on multiple nodes, which can be placed on the same server or distributed on multiple machine on a network. This approach is different from working with data in rest, where the partitioning is based on the data itself. Flink works with streams and as such, partitions the computations instead and the data/events flow between those computation nodes. Flink automatically tries to optimize the placement of the operations nodes, such that the overhead of sending the events through the network is minimized[2].

Failure handling: Flink supports replaying of a stream to be able to recover from failures, that is if a failure occurs the stream is replayed from the last checkpoint.

The checkpoint mechanism is different from what most other big data systems use as a fall-back mechanism. The state of the nodes is periodically persisted on HDFS or in memory, such that in case of a failure replaying from that checkpoint is possible. The algorithm behind the checkpoint barriers is based on the snapshot algorithm by Chandy and Lamport[1], such that the checkpoint is serially consistent across the distributed nodes, and it is not necessary to duplicate information. Once all data has flown through the barrier/checkpoint the computation is done and the computation between checkpoints either succeeds or fails atomically as a whole. The flow of the events are never held back or stopped by the checkpoint mechanism and as such,

most nodes will always be in use and bottlenecks on some nodes should not hinder other nodes to d their calculations. The flow of the events happen in a directed acyclic graph structure which also means that the consistency is strong since any parallel process will always see the data in the same order[1].

The checkpoint technique also separates the responsibility of calculation and failure handling, since changing the frequency of checkpoints does not alter the results of the stream.

Batch and Stream Processing: Flink provides two APIs, one for batch analysis and one for stream analysis. Since Flink only works on streams of data, batch processing has just been implemented as *finite* stream processing. This makes Flink a framework which is based on the concept of queries at rest, instead of data at rest. An advandtagde with this approach is also that the processing code can in a lot of cases be reused for both streaming and batch views. The two API's can be used from Java or Scala, and they provide an interface much like the Java 8's Stream library[1], where it is easy to do SQL-like commands, such as `where`, `groupby`, `sum` and so on. Furthermore Flink supports streaming windows over both time or counts which allows for more sophisticated analysis. As explained in the partitioning section, the different operations are distributed over a network of nodes, which each have a specific responsibility in the computation of the view.

Throughput: Flink prides itself with being low latency by having a powerful API to do stream processing, and by offloading some of the batch processing to the stream processing. What the streaming analysis does for low latency the ability to send information quickly forth to batch analysis, as well as the ability to scale horizontally allows for high throughput.

On data-artisan.com[2] a graph of the throughput of different big data system is made. On figure 1 the graph can be seen. The strengths of Flink become very apparent, and as we can see in this case the throughput is many times higher than for example storm, which is also known for its high throughput. Flink even has a lower latency, than any of the other measured frameworks.

[1] `https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html`
[2] `http://data-artisans.com/wp-content/uploads/2015/08/grep_throughput.png`

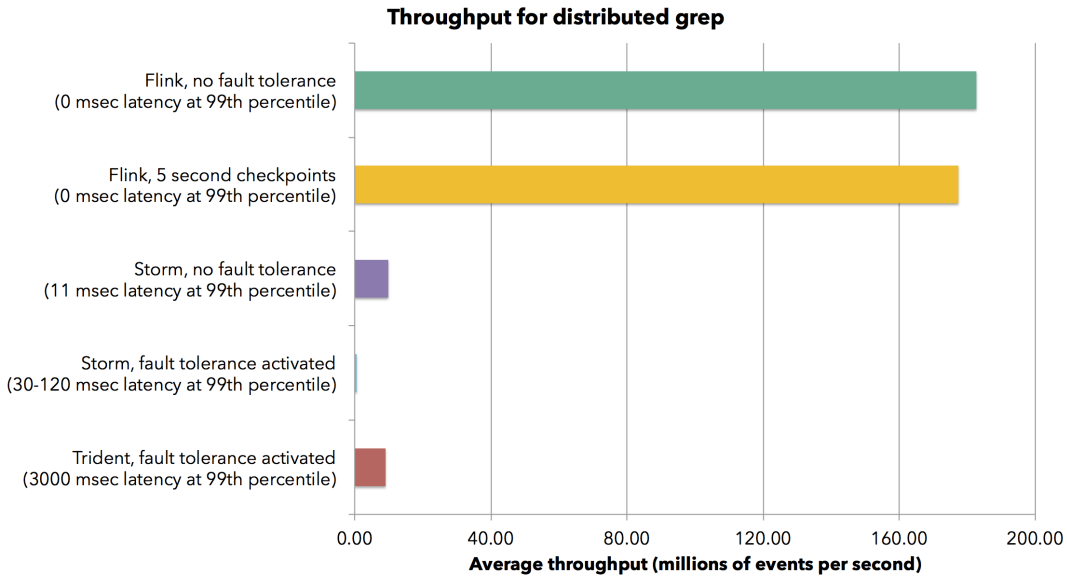


Figure 1: A graph showing the performance of different big data processing frameworks.

Interestingly enough Flink compares very well to the Lambda architecture. The Lambda architecture as introduced in the course, is separated into the following parts; the data sources, the master dataset, the batch layer, the serving layer, the speed layer and the queries. At the batch layer, batch views are computed, and similarly at the speed layer streaming analysis is done, and therefore it becomes quite clear that Flink naturally fits the lambda architecture by being able to be the main framework for each of these layers.

One point where the two are less alike are that Flink only works in terms of queries at rest, even in the batch layer, whereas the batch layer is in the lambda architecture described as data in rest with the processing *moving* over the data.

For project 2 and 3, we could have made both batch jobs and streaming jobs, but decided to only develop batch jobs. Most of our batch jobs are made out of logic which could be expressed as `-where`, `-join` and `-groupby` statements which are available with Flink. Therefore it would be possible to have written the batch views with the Flink APIs, and have gained higher parallelism, and furthermore had the ability to with ease introduce a speed layer which could do similar calculations, but allow for lower latency from data to queries.

1.2 B)

"You are asked to store a master data set of 80 GB given to you as an XML file. Why is the XML data format problematic when working with Map-Reduce? Would a

format transformation from XML to JSON be helpful? Would a transformation from XML to CSV be helpful? How would you store this master data set? Explain your answers"

The XML format is a tree structure format, and might not be easily be partitioned into smaller parts, which can be distributed among the data nodes of the storage system that Map-Reduce works on (HDFS). Furthermore XML is also a very verbose format and therefore data will take up more space which will make the transfer of data somewhat slower and require more space on the server. Even though Hadoop of course handles this quite fine, using less space is always to be preferred when the information is virtually the same.

Transforming the data to JSON would mostly help with the amount of data, since JSON is less verbose than XML. JSON also allows for tree-structure objects and therefore partitioning can still be difficult. Since JSON objects do not specify a start and an end tag it can actually be more difficult to split up than XML.

CSV on the other hand is a flat data structure and therefore is easily partitioned per line and split across multiple machines. Appending new data is also easy since each row is self-contained, it can be added to any data node, and therefore it is easy to do load balancing. Furthermore CSV has the advantage that it is not very verbose and it is easy to extend the input with more columns if needed.

Another possibility that we have used in Project 2 is to use a binary format. We used the serialization framework Avro[3] for this. By choosing to use a binary format, you can represent numbers as actual numbers and not text, therefore cutting down on the needed space. Furthermore if done properly it is still possible to make the format flat and therefore making partitioning easier. This approach of course requires more work to be done, and also enforces a scheme on the data, which makes it more difficult to extend the system later on.

For project 3 we converted the data to a CSV format and stored it using Hive since it was well integrated. Hive stores the CSV data as distributed files on HDFS but uses an abstraction layer which makes the access like a regular database access. Hive allowed us to make all the views we needed and therefore it seemed like a good choice to store the data.

It is important to notice that as soon as one transforms data, it per definition becomes derived data. Therefore, the master data set will be derived if stored in another format than it originally was, which asks the question on what to do with the primary data. I will argue that a transformation can be done from XML to CSV, which allows one to go through a similar process which converts the resulting CSV back into XML data. Even though the resulting data

[3] `https://avro.apache.org/`

is not the same, it is equal to the primary data, and therefore just keeping the CSV and not the primary data is enough.

1.3 C)

> *"Describe pros and cons of using the Hadoop ecosystem, based on the lessons you learnt from project 2 and project 3."*

The Hadoop ecosystem, has changed the industry, and how it looks at data in general. By going from a restricted view, the big data movement tries to break the boundaries but it is still very much in its youth. The relational databases go back to the 1970s and have had many years to polish its rough edges and making it easily available to developers. The big data movement is still trying to do this and most frameworks in the Hadoop data system exactly tries to sell themselves as easy to use, but in my experience most of these systems still have a high learning curve. Setting up a server with a Hadoop ecosystem requires a lot of time. Systems like Horton tries to make this process easier by creating a single entry point for organizing and managing the large amount of the different Hadoop frameworks which are often required to have a complete big data system with on par functionality as a relational database system.

Another con is that integrating different frameworks is often difficult. Since standards often times does not exist, frameworks are made to integrate well with other specific frameworks, but if it is desired to integrate with another system then the developers are often left to figure how to do that themselves if it is even possible.

For small systems, or embedded systems where the amount of is know or will never surpass a certain low limit, the overhead of using big data technologies is also often not worth it. A lot of the frameworks also have a lot of overhead on what they do, even though they scale better. Therefore in certain systems going with a SQLite database or a system specific Relational Database would be the best answer.

That said, the Hadoop ecosystem really shines when it comes to large amounts of data. A lot of businesses saw a huge rise in the amount of data they stored through the 2000s with the rising popularity of the internet, and now that processors were nearing their clock speed limit, being able to scale systems horizontally were very important. The Hadoop ecosystem is build around concepts of being able to abstract the distributiveness of the data away and allow developers to write code which automatically would scale to an arbitrary amount of machines.

It is not only the throughput of the systems which is effective on large amounts of data, but Hadoop is also build on the concept of low latency processing. It should be possible to serve information quickly to the user.

The Hadoop ecosystem also focuses a lot on failure handling, such that even in the case of machine breakdowns the frameworks should be able to gracefully handle it and be able to replay, reroute, or abandon the process, and the developers are able to specify which approach should be taken as to how to restore the data of the crashed machine.

For a lot of developers it has also been a deciding factor that the Apache foundation requires its projects to be open-source, which encourages developers to help find bugs, and ensures that the system does not require any proprietary frameworks, operating systems, or hardware.

To conclude on this it becomes obvious that if a system is going to scale, it is a good idea to use the Hadoop ecosystem since other systems might not be able to handle the same amounts of data, but if the system is of limited scale, the overhead of using the Hadoop ecosystem is quite high.

2 Question 2

2.1 A)

> *"Consider the data set from project 3. How much of the work you did in project 2 to clean data could be reused to clean the data set from project 3? Explain your answer."*

From a code perspective it would be difficult to reuse the source code of Project 2 to clean the data from Project 3, mostly because we used the serialization framework Avro, which then requires the data to be in a certain format, and outputs data in a specific format to project 2.

We used streaming to clean the data in Project 2 so in some sense it would be possible to reuse the streaming part, since by lazily streaming the data, it is possible to handle the 80GB of data in Project 3. Then by rewriting the logic of what to remove, label or ignore, the program would eventually complete.

Referring back to question 1C it should be noted that as soon as we clean data, we can no longer (unless the cleaning only tags, or ignores) assume that the derived data is the same as the primary data. Because of this an approach to backing up the original data or otherwise it should be made very clear that whatever analysis is made on the data will always be done from derived data.

We choose to not remove or delete any data in the cleaning process but simply ignoring it and allowing the batch computations to decide whether or not to use data. This was done to make it possible for future batch views to use the outliers and missing values for other computations.

For example if the operation system was unknown we did not use that WiFi client in the view, but it might be interesting in the future to see how many WiFi clients had unknown OS types. Having information about the what data is not there can be interesting in itself and therefore we did not remove it from the master dataset.

2.2 B)

> *"Describe a cleaning process for the data set in project 3. Describe the design of a system that implements this cleaning process."*

A cleaning process over this data could include checking for valid values, such as speed values that are negative. Then checking whether or not each entry falls into one of the two well defined categories, Vehicle-type or Person-type. Another step in the cleaning process would be to check for missing information or information which should not exist for an entity.

One of specific parts of cleaning the data of Project 3 is the fact that sometimes, when cars have been staying still for too long they are randomly teleported to other parts of the map. If some analysis would be done on location, some cleaning process needs to handle this. This could be done by checking the delta of x and y coordinates compared to the last position of the car. The new value could in case of a teleportation detection, be labelled such that the batch view could handle the entry correctly.

To create such a cleaning process, I would create a Map-Reduce program such that it can handle the large amounts of data. Then I would create a mapper for each of the different procedures, and checks, which each output to the next mapper in a pipeline fashion. By doing this it is possible for map reduce achieve higher concurrency. A reducer could then be placed at the end, aggregating all the results into two lists of entities, one for vehicles and one for people. At the end the results could be stored on HDFS, ready for batch or streaming analysis.

One could also implement this process as a Hive job, which would have the obvious advantage that Hive itself would split the process into multiple stages of mappers and reducers automatically. Though one disadvantage of this approach is that the data would first have to be imported into Hive tables and then the result would have to be put into another table, or the original data removed.

3 Question 3

3.1 A)

> *"Assume that the data from project 3 is not a massive data set, but a data stream. Every time step, a large collection of vehicles and persons is generated (based on the attributes contained in the ¡vehicle¿ and ¡person¿ elements of the XML file given in project 3). How would you proceed to characterize such a data stream?"*

To characterize a stream of data I would look at at least three factors; structure, mean throughput and peek throughput.

In project 3 the data of the stream contains structured data since it is in XML format. The structure is as follows:

```
<Timestep>
    <vehicle, id, x, y, angle, type, speed, pos, lane, slope/>
</Timestep>
```

or

```
<Timestep>
    <person, id, x, y, angle, speed, pos, edge/>
</Timestep>
```

depending on which type the input has. In general when working with XML which is semi-structured, the structure of the stream is easily obtained since the structure of the data pr definition is part of the data itself. In some cases the structure of the XML is even specified in an explicit XML schema. Since JSON also is a semi-structured format it would also be possible to define the structure from the data, but JSON does not have an XML schema counterpart. Had the data been unstructured it would have been even more difficult, since one should try to create and fit a schema at the same time.

The mean throughput can be described as the average of the number of discrete elements over a timeunit. Often when working with infinite streams we need to specify and limit that time to base this over. This could be calculated either with batch jobs, but it could also be calculated using windowed stream analysis. By examining a window in the stream one could approximate the average amount of entities pr timesteps. This approximation of the overall mean throughput would probably also be more describing than an overall mean, since streams are infinite and ever changing. The overall mean would not be representative for the stream currently, and it would therefore be a bad foundation to base decisions on.

The third characterization could be the peek amount of elements in the stream, which could be approximated by comparing the current mean throughput with the last mean throughput. By using the mean instead of hard values, the computation is less sensitive to very short peeks, which might or might not be desired. This value could have some kind of fallout value (for example a day or a week), such that it continues to be representative and a former peek does not continue to overshadow the current reality. The peek value can be used to help the developers decide on whether to scale the system, or rent extra computation in small periods of time, as it is for example seen possible with AWS[4] or Azure[5].

3.2 B)

> *"Describe a meaningful view based on the data set from the Project 2 data set. How do you obtain that view? Describe the problems you faced obtaining such views in project 2 and how you fixed them."*

I have chosen to showcase the second view from our Project 2 as I find that the most interesting. The view was described as follows:

> *"How can Wi-Fi data be used for tracking an individual at ITU?"*

The hypothesis is that information about what access points a client has been connected to makes it possible to track a single person at ITU. Since each Wi-Fi client has an unique ID in the data set, tracking that ID around the ITU through various access points in certain rooms, one could connect this information to teaching activities. One could essentially build a schedule corresponding to a person, the holder of the unique ID, and by cross referencing the public course base, identify any student or teacher.

It is necessary to assume that every person is connected to Wi-Fi whenever they are at ITU and even more important that they are connected to the access points in the rooms that they have courses in.

To obtain the data we created a map-reduce program. The main part of the analysis is done in a mapper. The map method can be seen in figure 2. The mapper joins the reading with its WiFi Client if it can, then it filters, the reading based on whether it measures the Access Point. Then for each reading, it joins it with the reading with its Access Point and outputs the WifiClients id as the key and the location and time of that reading as the value. The mapper filters on null values for Access Points, and Locations.

[4]https://aws.amazon.com/

[5]https://azure.microsoft.com/en-us/?b=16.48

```
public void map(AvroKey<Readings> key, NullWritable value, Context context) throws
     IOException, InterruptedException {
  Readings readings = key.datum();
  // Where
  if(wifiMap.containsKey(readings.getUUID().toString()))
  {
        // join
    WifiClient wifiClient = wifiMap.get(readings.getUUID().toString());
    // where
    if(wifiClient.getTypeOfMeasure().equals(WifiClientMeasure.AccessPoint))
    {
          // select
      for(Reading reading : readings.getReadings())
      {
            // join
        AccessPoint ap = apMap.get(reading.getValue());
        // where
        if(ap != null && ap.getLocationId() != null && locationMap.containsKey(ap.
            getLocationId()))
        {
          Date date = new Date(reading.getTimeStamp());
          Location location = locationMap.get(ap.getLocationId());

          context.write(new Text(readings.getUUID().toString()), new Text(location
              .getRoom() + "-" + dateFormat.format(date)));
        }
      }
    }
  }
}
```

Figure 2: The map method for batch view 2 in project 2.

It becomes obvious that having this as one mapper does use the map-reduce framework to its full potential. It would have been a better idea to split this into multiple mappers, for example in the situation where the mapper iterates over the readings, it would have been smarter to use a mapper for that job. This could greatly increase the parallelism and therefore the scalability of the batch job.

The reducer simply aggregates the rooms, times together to a list over each specific WiFi client. A snippet of the output can be seen in figure 3

```
...
fd958189-5ad3-5586-a7ad-d3fe4e6f4695
 4A32-2016-10-12:11, AUD44A60-2016-10-24:08, AUD44A60-2016-10-24:09,
 AUD32-3A56-2016-10-04:09, AUD32-3A56-2016-10-04:08, 5A60-2016-10-12:07,
 4A58-2016-10-24:08, AUD44A60-2016-10-10:09, AUD32-3A56-2016-10-04:10,
 3A12-2016-10-06:11, 3A12-2016-10-06:12, 5A07-2016-10-12:11,
 AUD32-3A56-2016-10-13:09, 5A05-2016-10-31:11, 5A07-2016-10-12:10,
 3A52-2016-10-25:11, 3A52-2016-10-25:10, AUD44A60-2016-10-24:10,
 4A58-2016-10-24:09, AUD44A60-2016-10-31:10, 4A16-2016-10-24:14,
 5A07-2016-10-05:08, 4A16-2016-10-24:11, 4A16-2016-10-24:13,
 4A16-2016-10-24:12, 5A07-2016-10-12:08, 5A07-2016-10-12:07,
 AUD32-3A56-2016-10-25:10, AUD44A60-2016-10-10:10, 4A58-2016-10-24:10,
 5A07-2016-10-12:09, 4A16-2016-10-10:12, 4A16-2016-10-10:11,
 4A16-2016-10-10:14, 4A16-2016-10-10:13, 4A22-2016-10-31:13,
 4A05-2016-10-12:11, AUD32-3A56-2016-10-06:09, AUD32-3A56-2016-10-13:10,
 5A07-2016-10-05:09, AUD32-3A56-2016-10-25:09,
...
```

Figure 3: Resulting data

Which can be represented a bit better visually in a schema as seen in figure 4.

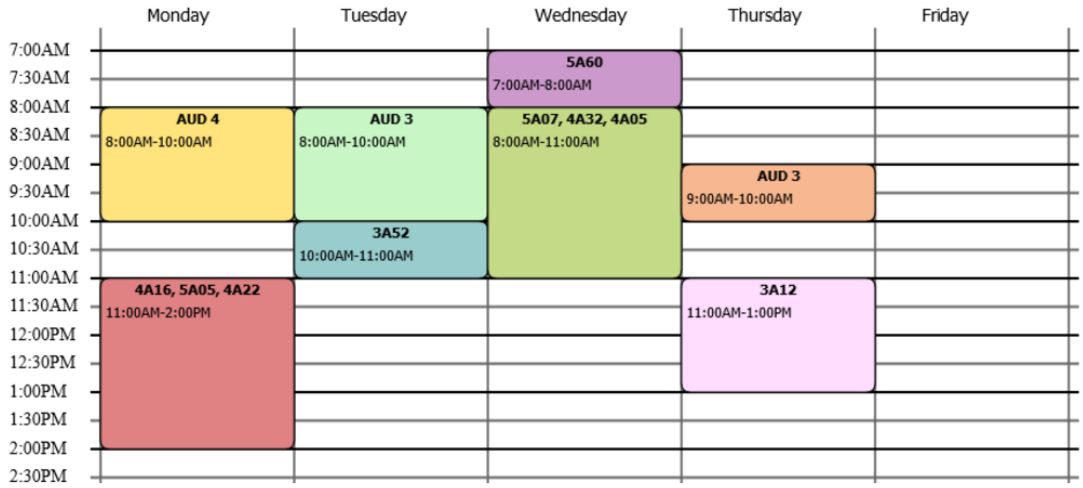


Figure 4: Resulting Schema

With these results, the schema of a Wifi Client can be correlated to the schema of a student at ITU by matching it with course information and TimeEdit. If the Wifi Client has been to

the rooms that match a student's schema then we might be able to specify which person, or at least which programme, a specific WiFi Client matches. For our result snippet shown above, the most plausible result is that the WiFi Client corresponds to a 1st year GBI student, since their schemas(see figure 5) overlap nicely.

Figure 5: Timeedit Schema for a 1st year GBI student

One of the most difficult parts of creating this batch view was handling the difference between WiFi Clients and Access Points, and how readings should be mapped to these. The design we chose was to use Map-reduce over the readings, and make the mapper read in all the meta data in the beginning of the job. Then when mapping the readings the meta data is looked up in two local list, one for WiFi clients and one for Access Points. We decided on this approach since the meta data file was of limited size and did not grow as fast as the readings data, of which there came one new file each day. But this approach is not the most effective and if we had for example used Hive, the process of combining meta data to readings would have happened in a parallized fashion.

We also spend some time figuring out how to use Avro serialization and how to integrate Avro with Map-Reduce. Avro uses a JSON like schema to define the binary format, the data is in. Making lists of lists was especially difficult which was the case for the readings, and furthermore we needed to define two Avro schemas for the different types of data, before and after it was cleaned and transformed into the different types.

A Appendix: Project 2

Project 2: Group 4 + 40

Master set data - Technical

A. How do you store this master data set? Explain your answer.

In order to store the master data set, it was necessary to examine and define the data. We categorized data from the metadata file as either Wi-Fi clients or access points and locations, and data in the time series files, is specific readings of either of those types of measurements. There exist multiple other types of metadata but we have chosen to ignore either by scope or because the data is incomplete - more on that in the next section. We use the serialization framework Apache Avro[1] in order to define schemas and (de)serialize these objects. We defined an Avro-schema both for the provided data files and for data model, such that cleaning could be done efficiently and dynamically.

Parsing the data is conducted using the Google GSON[2] library with the defined Avro schemas and serializing the parsed, cleaned and transformed data into files of serialized Avro objects. By creating these schemes it makes it easier to perform analysis as it is possible to have access to related data easily.

We wanted to store data in Hive[3] tables using Kite SDK[4] on top of Avro, but because of difficulties with other parts of the exercise we chose to postpone this implementation, and eventually, we did not reach a point with time enough left to implement this. Some of our choices with regards to splits of metadata still looks like it is meant to be placed in tables though.

The resulting metadata contains roughly the same information as in the provided data files. We did parse the path-property into multiple properties though, and remove some of the data that are unused for our batch views.

Furthermore the schema contains a definition of the readings that contains the UUID and a collection of Reading-objects that each contain a timestamp and a value, rather than being a list of numbers with length 2.

The schema of the master data set has been drawn in Figure 1 below. Note that AccessPointMeasure and WiFiClientMeasure are enumerations where each value is in the comma separated list.

[1] http://avro.apache.org/
[2] https://github.com/google/gson
[3] https://hive.apache.org/
[4] http://kitesdk.org/

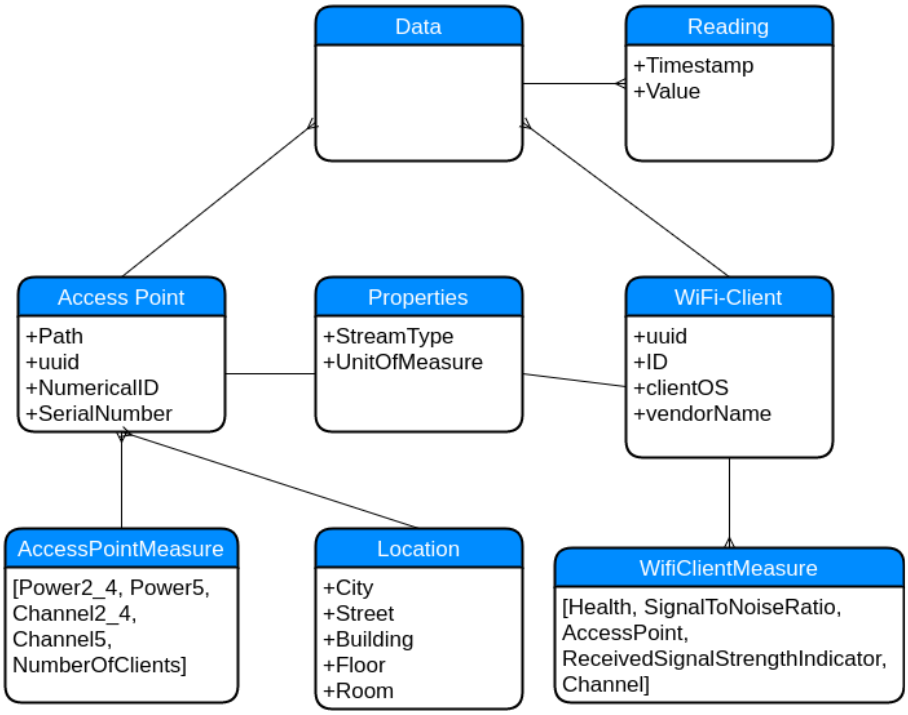


Figure 1 - The master dataset model

Finally Avro supports serializing to binary files, so the data is stored in 4 different binary files, with each type of data. It would have been useful to have a partitioning framework to be able to partition the data for each day, but this was not done in time for the project handin.

B. What is the sampling interval of the data? Are there missing data in the dataset?

The sampling interval of access point information is approximately one sample every four minutes. This calculation is only made on a subset of data, by applying the following formula:

$$sampling\ interval = \frac{time\ of\ latest\ reading - time\ of\ earliest\ reading}{\frac{number\ of\ readings\ for\ access\ points}{number\ of\ access\ points}}$$

As mentioned in the first section we ignore certain types of data. In the data files information concerning student course base, calendar bookings, thermostat readings, is available. In the case of the thermostat readings, the data is incomplete, since it is only present for one room. In the imported data, we also have missing information, for example is the information about the vendor name and the OS name for Wi-Fi clients often unknown. Furthermore, we have chosen not to include the student course base and calendar bookings, not because data is missing but because we see it as out of scope for the analysis we want to perform.

2

C. **Define a procedure to clean the data set and handle the missing data. Give arguments for your approach.**

There are multiple approaches on how to handle missing data. We have chosen to both remove, by not including data, and ignore certain types of missing data. When we read the raw data and transform it to our master data format, we do not include any entities not related to Wi-Fi clients or access points. This is done for both the metadata file and the readings files. This effectively removes information about the student course base, thermostat readings and calendar bookings.

For the other type of missing information, we simply ignore it when performing the analysis. By keeping data concerning these entities but ignoring them it is possible to perform analysis on the rest of the entity even if this specific information is missing. This approach has the obvious downside by requiring more work and thought to complete analysis.

D. **Generate a clean data set.**

Do you use Hadoop for this batch process? Explain your answer.

We run the cleaning of data as a simple java program, which does not use any Hadoop[5] features. As of now running the program, and moving the data is a manual task but it would be possible to change the program to use Hadoop and MapReduce, which would make it possible to automate the process at some point. This is not done, due to time pressure and failed tries to do so.

That said, at the moment our solution is not prepared for performant data ingestions and data partitioning. In the future we are planning to use Kite together with Apache Crunch[6] to import, clean, partition and ingest data into our master data set. This solution would utilize all the features offered by Hadoop.

How many instances of missing data did you find?

We do not have a specific number of how much data was missing but as mentioned before, thermostat readings were only available for one room, therefore implying that data was missing from all the other rooms at ITU. Furthermore, there is a lot of missing data regarding vendor and OS name used by Wi-Fi clients.

Personal data - Critical

A. **What are these data about? What is known/not known about Wi-Fi use in this dataset? What is made obvious/visible? What is overlooked?**

The data are about Wi-Fi use at the IT University of Copenhagen (ITU). The nature of the data is continuous and not real-time. It comes in batches every day making us able to say it

[5] http://hadoop.apache.org/
[6] https://crunch.apache.org/

is about the Wi-Fi use over the course of 7 days, i.e. the daily Wi-Fi use at the university. We employ, the term ‘use’ is in a broad sense in this report so far. From a technical viewpoint, the data tells us that activity is occurring at the hardware level, a kind of Wi-Fi infrastructure at the university (based on metadata on the Wi-Fi Clients as identified in the technical investigation). Devices are connecting to the Wi-Fi access points throughout the building, but we do not know for what purpose they are doing so. We can for instance pinpoint at the room level where activity on the access points is occurring and many other characteristics of this so-called use. The context, however, is unknown--or made invisible. The visible is that these data can say something about the behaviour of the people at ITU as a collective. What is more hidden is the fact that with the right analyses, we can begin to track and identify individuals.

B. What kinds of stories can these data tell about people at the ITU (what can these data reveal about individuals if anything)?

Lack of context brings us to the question of the stories the data can tell us as well as what privacy concerns may be tied into these. Stories have main characters, or protagonists, whose actions and behavior tie seamlessly into the progression of the story. From data about Wi-Fi use at ITU we are arguably not able to know much about the individuals, whereby the stories we can tell become quite flat. We also cannot know what the Wi-Fi is enabling the users to do, that is, what they are using it for, for example the websites they are browsing or the files they are down or uploading. The ‘who’ remains at the aggregate level, and the ‘what’ entirely unknown. So asking what stories we can tell is relevant, but the answer would most likely be: not very compelling ones. This can can be argued if you just take the data at face value. However, if we apply a critical view, we may be able to tell the compelling stories: stories that highlight how this data can be used for activities that puts assumptions about privacy, identifiability and ethics at stake. We will do so later in this project paper.

C. What can these data reveal about all occupants at ITU in general? Can you say anything about things other than the devices connecting to Wi-Fi access points and the locations of these access points in the building?

The data can reveal frequency of use. We may see spikes in activity on the access points and Wi-Fi Clients during weekdays where the ITU would be host to more people than in the weekends. The activity may also be spread out throughout the building to account for the heterogenous sites of activity (students as well as faculty). On weekdays more rooms will be in use and thus more access points will be connected to from various devices. On afternoons in the weekdays, the usage may be isolated to a few rooms and locations throughout the building with students undertaking study activities such as group work in single rooms. Outliers in activity can occur from the ITU hosting conferences and other such events, and in the same vein, Friday nights (until 2 AM) may also see a spike in the area around Scroll Bar. The argument can be made, that this information puts privacy at stake to some extent, as the concept of privacy exists and is negotiated in many different ways in theory. A definition of privacy can be the individual’s right “to control, edit, manage, and delete information about them[selves] and decide when, how, and to what extent information is communicated to others” (Westin 1967). As such, privacy functions as a timeout from the

practices taking place around an individual at ITU, which this individual feels is compromising this right. To better illustrate the potential of this dataset as it is subjected to more and various analyses and tracking, we will define three views or scenarios in which tracking can have implications of different sorts.

Batch layer - Technical/Critical

A. **Define three views that can be used to get insights about this data set**.

View 1: How can Wi-Fi data be used for purposes of evaluation of teaching staff?

With this view we are implicitly concerned with how use of the Wi-Fi at the room level can say anything about the turnout of students for a particular teaching activity. Our hypothesis is that the number of registered participants should match the number of Wi-Fi clients connected to the access points in the room of the lecture. If the access point in the room has a number of Wi-Fi clients connected which does not match the expected number, this might indicate that students do not show up and in turn indicate if course is popular or not. More specifically, connecting to the access point could indicate how many users connected their laptops, presumably for note-taking and participation of other activities in the course. Tracking the data over time, could be a way to see if fewer students are attending throughout a semester.

Attendance may in some evaluations be seen as indicators of the success of a course. This assumes that a successful course draws a lot of students to the lectures consistently, and therefore this indicator ultimately reflects on the quality of the teaching staff affiliated with the course.

There are a lot of assumptions which must be made to conclude this. First of all we assume that a Wi-Fi client is one attendance. Furthermore we assume that everyone attending is using the Wi-Fi, which might not be the case for all courses. It also assumes that every room the course has is used only by students of that course. Even though some of these assumptions seems like a stretch one could take some of these problems into account and use it in collaboration with the qualitative analysis of the course evaluation.

View 2: How can Wi-Fi data be used for tracking an individual at ITU?

We have a hypothesis that information about what access points a client has been connected to makes it possible to track a single person at ITU. Since each Wi-Fi client has an unique ID in the data set, tracking that ID around the ITU through various access points in certain rooms, one could connect this information to teaching activities. One could essentially build a schedule corresponding to a person, the holder of the unique ID, and by cross referencing the public course base, identify any student or teacher.

It is necessary to assume that every person is connected to Wi-Fi whenever they are at ITU and even more important that they are connected to the access points in the rooms that they have courses in.

Interestingly it should be noted that if a person stalked another person around ITU, writing down where and when a person was there with his/her device, it would be possible to identify the ID of that person with the current data alone. As soon as a person has identified the ID to a person, it is possible to track that person for the rest of the lifetime of that device. The fact that this is possible is not visible directly by looking at the data, but it raises interesting ethical questions about the data. One could imagine ITU tracking a specific student to make sure that they participate in the courses they are registered in.

View 3: How can Wi-Fi data be leveraged as a commodity by ITU to sell to third parties with commercial interests?

The data can reveal details about the devices connected to the access points. In a situation where ITU would wish to monetize the data in question, it could be sold to third parties with commercial interests so that these may target advertising towards people using particular brands of software. Through analysis where the device is correlated with the unique Wi-Fi Client ID and the courses the holder of this ID attends, a commercial third party may be able to pinpoint and target the creative Design and Communication student that uses Apple's Macbook Pro for completing design projects. If this individual has strong brand loyalty, advertisers can target ads for other Apple products towards them, but the advertiser could also choose to base their marketing on other factors of the segmentation of software users of a given course (by referencing the course base). The critical question one should raise in this regards is "If I am exposed to this content, what content will/am I then then not exposed to?" (Turow 2011). The mediation of communication we as individuals are exposed to, in the example ads, determines how our perception of the world is shaped. However the notion of the *world being shaped* is critical since it encourage certain behaviours, leaving the freedom of will to being no more than an illusion, played out by internalized practice by marketers.

If the hypothesis is true, then the buyers of the information would be the ones who had to make assumptions. Of course there would still be assumptions about Wi-Fi clients being people, and people taking specific courses and so on, to be able to sell the data. The point here being: The moment the data shift hands, it also shifts domains, thus the domain one works in also determines what questions one should and can ask. In this instance, the data goes from an ITU domain to a marketing/advertising domain,however the assumptions and thereby also the disciplines that are build up around each domain will be different from each other even though the data remains the same.

B. Implement the corresponding batch processes that take the clean data as input.

Implementing the analysis is done using MapReduce. In each example the readings are taken as keys. For each view a mapper is created which filters and transforms the readings while also adding relevant information from the corresponding meta data files. The reducer is then often used to aggregate or sum up the information. The batch view is simple text files, but it would be possible to serve this in another way in an extension of the project. The result of the batch views is still huge amount of data, but it is transformed in such a way that a query could easily retrieve interesting information about specific entities. For example the tracking view shows where every Wi-Fi client has been to at every point in time, and

therefore a query on a specific Wi-Fi client will be necessary to track a single person. As of now running the batch views is a manual process but it would be easy to automate to run once a day for example.

Three examples of the outputs of the batch views can be seen below:

Course popularity: ***number of "people" in room Aud3 at 8, 9 AM on a tuesday***

```
...
AUD32-3A56: 2016-10-25_08 74
AUD32-3A56: 2016-10-25_09 86
...
```

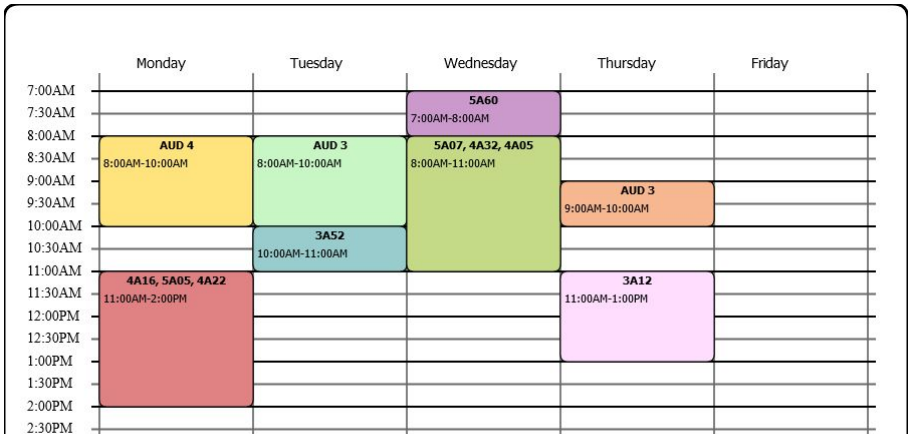
This result implies, that the course, Big Data Management (Technical), have 74-86 people showing up to the course the 25th of October '16. However, the course base states that only 65 people are taking this course[7]. This could be explained by people having multiple devices, or people sitting outside the auditorium being connected to the same access point. Therefore as expected, our assumptions does not hold in all cases.

Tracking info: ***snippet of tracking info of a person who's been around room Aud3 at 8, 9 AM on a tuesday.***

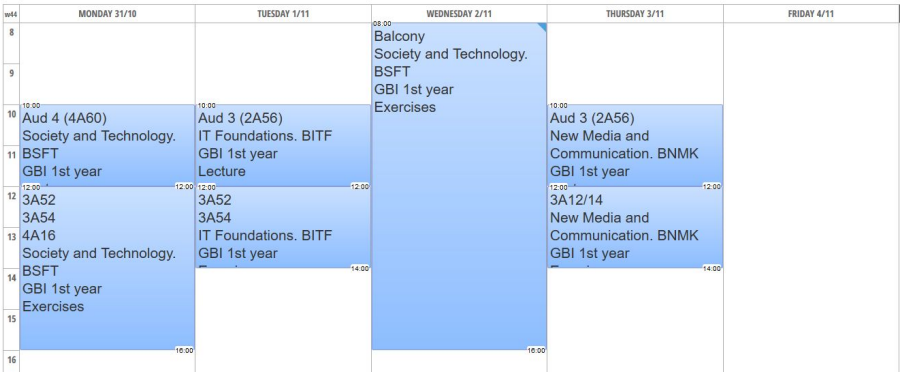
```
...
fd958189-5ad3-5586-a7ad-d3fe4e6f4695  4A32-2016-10-12:11,
AUD44A60-2016-10-24:08,              AUD44A60-2016-10-24:09,
AUD32-3A56-2016-10-04:09,          AUD32-3A56-2016-10-04:08,
5A60-2016-10-12:07,                      4A58-2016-10-24:08,
AUD44A60-2016-10-10:09,            AUD32-3A56-2016-10-04:10,
3A12-2016-10-06:11,  3A12-2016-10-06:12,  5A07-2016-10-12:11,
AUD32-3A56-2016-10-13:09,                5A05-2016-10-31:11,
5A07-2016-10-12:10,  3A52-2016-10-25:11,  3A52-2016-10-25:10,
AUD44A60-2016-10-24:10,                  4A58-2016-10-24:09,
AUD44A60-2016-10-31:10,                  4A16-2016-10-24:14,
5A07-2016-10-05:08,  4A16-2016-10-24:11,  4A16-2016-10-24:13,
4A16-2016-10-24:12,  5A07-2016-10-12:08,  5A07-2016-10-12:07,
AUD32-3A56-2016-10-25:10,            AUD44A60-2016-10-10:10,
4A58-2016-10-24:10,  5A07-2016-10-12:09,  4A16-2016-10-10:12,
4A16-2016-10-10:11,  4A16-2016-10-10:14,  4A16-2016-10-10:13,
4A22-2016-10-31:13,                      4A05-2016-10-12:11,
AUD32-3A56-2016-10-06:09,          AUD32-3A56-2016-10-13:10,
5A07-2016-10-05:09, AUD32-3A56-2016-10-25:09,
...
```

[7] https://mit.itu.dk/ucs/cb/course.sml?course_id=1835814&mode=search&semester_id=1820419

With these results, we can put together a schema for the person and match it with course information and TimeEdit, to find the rooms the person have been around. The most plausible result is, that it is a 1st year GBI student, since the schemas match.



Schema composed by Wi-Fi readings



1st year GBI schema from TimeEdit [8]

[8] http://bit.ly/2ebuIGt

8

Vendor info: ***room Aud3 at 8, 9 AM on a tuesday***

```
...
AUD32-3A56: 2016-10-25_08-Android 9
AUD32-3A56: 2016-10-25_08-Apple iOS 2
AUD32-3A56: 2016-10-25_08-Mac OS X 3
AUD32-3A56: 2016-10-25_08-Windows 7/Vista 2
AUD32-3A56: 2016-10-25_08-Windows 8 3
AUD32-3A56: 2016-10-25_08-unknown 55
AUD32-3A56: 2016-10-25_09-Android 7
AUD32-3A56: 2016-10-25_09-Mac OS X 6
AUD32-3A56: 2016-10-25_09-Windows 7/Vista 3
AUD32-3A56: 2016-10-25_09-Windows 8 2
AUD32-3A56: 2016-10-25_09-unknown 68
...
```

This batch view is actually very much alike the course popularity, and only required small changes in the code. Adding the ClientOS to the output key of the mapper was the only thing needed. This showcases the flexibility of the MapReduce and the batch setup. Unfortunately it is visible that a large amount of data is not clean since the OS is unknown. In this batch view the unclean data is not ignored but shown, since Windows 10 for example obviously should be present but is not.

C. Do you use Hadoop to answer Q3B? Explain your answer.

The implementation uses Hadoop since it is implemented with MapReduce. This is done to make it possible to scale the results. Even though the amount of data right now is not that big, a lot of data gets added every day and at some point it will be difficult to handle with regular relational databases. Therefore, using Hadoop will ensure that the system can stay in place and if needed, it is possible to extend the system horizontally. Furthermore, Hadoop and MapReduce, makes it easier to extend the solution with different frameworks should the scope of the project extend. In the implemented batch views, it was possible for us to do all the work in one mapper and one reducer. Hence, we did not need more features, like pipelining.

Log - Technical/Critical

List the problems/challenges you faced during this project and explain how you tackled them.

Technical:

- We had high expectations at the start. We had planned to try to use Kite to create Avro schemas for us and then use kite to create a Hive dataset and import the data into Hive, and then either query directly on Hive or use Crunch[9] on top to query. But as looked more into the frameworks and tried Kite, we saw that it was harder to do than expected.
- The Kite tutorials seemed to be easy at first, but the further in we got, the more complex it became. The tutorials either seemed too easy and broad or too specific and complex to be used in our case.
- Errors and problems regarding Kite CLI in Hadoop with Hive were too specific to be googled, making it harder to progress.
- We had several problems regarding Kite CLI
    - It took a long time to run the program, because the tutorial didn't explicitly tell how with Hadoop. Thereafter, it also took a lot of time to figure out the right commands.
    - We had problems with Hadoop, which seemed like we were missing some dependencies either in the Kite .jar-file or in Hadoop. E.g. ClassDefNotFound, ClassNotFound
    - Furthermore, the newest Kite .jar-file from their own site doesn't seem to work. So we ended up downloading an older version from another site.
        - The older version could not format or make schemas with json-files (only csv), but unfortunately the data was in json.
        - The older version could only be used to complete the tutorial
    - Hive wasn't installed, when we tried to make the dataset in Hive with Kite. Instead we stored it in HDFS; but that way it couldn't be queried.
- The Hadoop installation seems to be missing classpaths and dependencies, and so in MapReduce we had to explicitly include every dependency in the compiled jar of the MapReduce job. The weird part is that our Parser job uses the same libraries, where this is not an issue. The difference between the jobs is that the MapReduce job runs on the cluster, where the Parser job runs on the Hadoop client machine that we have access to.
- The project description and assignment is so vague, that it makes it hard to decide which tools to use and why
    - We spent a lot of time on deciding what to do, and in the end we couldn't get Kite, Hive etc. to work within the time available, so we opted for raw MapReduce instead.
- It was hard to integrate the technologies / frameworks together
- When we got the project it was hard to know, where to start.

[9] https://crunch.apache.org/

- One of the difficulties was to communicate the data model with critical students, because they are lacking technical knowledge within data modeling and manipulation. Hence, they were not able to generate questions for which our big data infrastructure needs to provide answers. In order to make this gap smaller we needed to create an appendix explaining the data model and the provided data we are interpreting.
- We haven't learned anything about cleaning yet, and so it was hard to know how to.
- Hard to do all parts of the project together, which means that our knowledge is scattered.
    - Some did MapReducing
    - Some looked more into frameworks
    - Some did a little bit of both
- Initially a lot of time was spent understanding what the data was about and how different data was related.

Critical:

- First C&T Group Work: Prepared data was delayed
    - The plan for Project 2 was to have the Technical (T) team prepare the dataset prior to our first meeting. Due to miscommunication at the course administration level, this did not happen. But it gave us a chance to gain valuable insights into aspects of the data preparation process. The first meeting on Project 2 gave us a chance to look over the shoulders of T while they gained an overview of the data's properties as well as took initial steps to model the data on a whiteboard for both T and C to see. All the while they were answering any questions we had about data preparation in general and the concrete data we were given to work with. It was also an occasion where we could directly question any choices to exclude or include data that T would be taking--and be given, what we must and do assume, valid reasoning for either or.

- Views and close collaboration (C&T) is more productive for C
    - It is difficult to do the work on this project separate from T. At the very least, it seems we are more productive when the work is done together. Though we are C, we enjoy staying close to the data so we avoid being too abstract and trying to answer the project questions too broadly (as done in the earlier project), forgetting the scope we are given. Defining scenarios and views helps, though. And it reflects the approach we are taking to the data: we are looking at the data and seeing what we can do with it, as opposed to stating a question and finding the data that can answer this question.

- A quite creepy discovery
    - The iterative process of working with the data meant that interesting discoveries could be made throughout the project. From the data, T managed to compile a schedule for an individual whose data was present in the dataset. This brings about aspects of ethics to the fore, which are arguably

just as interesting to dive into as the view we chose to focus on in the 5th part (evaluation of teachers and their courses), but time constraints make us unable to treat them with our critical expertise. Nevertheless, the discovery makes us feel that we might be dealing with very creepy data that certainly warrants some critical thoughts.

Ethics/consent - Technical/critical

A. If you were charged with a problem of coming up with ways to make these data useful to ITU in new ways what might be some options for doing so? Select and describe one potential way you might implement a way to use these data - what kind of system would this be?

The batch views defined above all provides some interesting ways for ITU to gain information. Especially the course popularity view could be used to support the qualitative analysis currently done through the course evaluation. We consider this a plausible extension of the current system at ITU, more so than the two other views. The view should of course try to mitigate the insecure assumptions mentioned in the section, by doing extra analysis, by for example making sure that they only count unique Wi-Fi clients, with specified operating systems and so on. Mitigating the assumptions in this way is taking a critical approach in order to avoid drawing conclusions based solely on correlations, because "[…] Big Data correlations suggest causations where there might be none. We become more vulnerable to having to believe what we see without knowing the underlying whys." (Zwitter, 3 2014).

The system should be automated to take batches of data each day from the IT-department's wifi data. These data should then automatically be cleaned and stored with the corresponding batch view. Storage of the batch view could be done in a relational database, which could be queried from mit.itu.dk or a similar intranet. This way both students, teachers and administration could check the course popularity based on both the course evaluation and the wifi use.

B. What would you implement as your consent procedure? Do you even need consent here?

It is interesting to notice that at this point it is not transparent that the collection of data is happening and is available to ITU staff members, and furthermore that it is possible to use the data for analysis on this topic. By actually performing the analysis, it is clarified that privacy in some sense is violated, since the users of the Wi-Fi does not know that the data is used for analysis. Therefore implementing a consent procedure if ITU chose to perform any of these batch views, would be a good idea. Though it is difficult to see why ITU would track the individual student, it is easy to imagine why ITU would try to track the course popularity, which would still have some individual effect.

Consent when you set up Wi-Fi on your device. The data is largely anonymized. And that fact is what is visible. But we have now seen that actually a lot is possible if you implement analysis.

C. What are some of the ethical issues you would need to think about? (Critical students, think about the "unraveling effect" - week 7 lecture).

Since tracking the popularity of a course is indirectly tracking an individual's performance, raises ethical questions. The derived information can have personal implications if the teacher is fired from doing so.

There are aspects of the unraveling effect (Peppet 2011) that become crucial in the ethical concerns of using the Wi-Fi data for evaluating teaching staff. We may imagine that decision-makers at ITU are able to argue for using the data as an indicator of popular courses that by their popularity reflect on the capabilities of teachers. In a show of concern for ethics regarding data use, these same decision-makers could ask teachers to give consent to having the data be used. Decision-makers would contend that the data will be used to improve evaluation, leaving teachers to take a stand to this claim when they consider whether to give consent to the collection and use or not. Teachers that give consent will be in line with the interests of the administration, whereas teachers who do not appear to stand in opposition and may see consequences because of it. In this scenario such consequences could ultimately be the loss of the job. Not because they are poorly evaluated, but because they do not consent to the practices that decision-makers incentivize and claim are a part of the evaluation process itself.

Appendix

Literature

1. Peppet, S. R. (2011). Unraveling privacy: The personal prospectus and the threat of a full-disclosure future. Nw. UL Rev., 105, 1153.
2. Turow (2011) The Daily You. Yale University Press - Introduction & Ch 1
3. Zwitter, A. (2014). Big Data ethics. Big Data & Society, 1(2).

B Appendix: Project 3

Project 3: Group 4 + 40

Question 1 (C): Defining your goals

We have chosen to work with scenario 1.

A city may be thought of in namely two ways. In one way it is a place. It is given a name and it describes a what and where in representations of many other whats and wheres, for example on a map. But we may also think of a city in terms of a space that engenders relationships between objects and humans. Some of these relationships come to be managed within the context of city planning and traffic management. A city has infrastructure and facilities that support the lives of residents in various ways. As such, a scenario that the city planners and traffic managers have to deal with could be improving mobility infrastructure for these residents with attention to the forms of transport that exist, which in the case of this paper is pedestrians and vehicles. These are the means through which residents travel from A to B for whatever purpose. Mobility tracking and transport visibility is in itself certain view of a city and a certain understanding of the relationship between its objects.

There are, however, limitations to the movement that pedestrians and vehicles can make within the city. For vehicles, movement has to take place via designated roads. Pedestrians may travel more freely and deviate from the roads. In this way there are things we know about the mobility of residents in our city space because we have designed the space physically. But there is arguably value in knowing how the space is being used because this can help to improve the infrastructure for residents (or other stakeholders) if the insights make their way to those in power to make decision that money should be spent towards this goal of improvement.

An important step towards mobility improvement, then, is tracking mobility and making the movement of our relevant actors, visible, so we can make sense of it as a reality that exists in our city. What we really want is to see movement, even just partially. A way to do this could be having cameras scattered across the city. Another way, which we are able to imagine in our city, is to have our residents and their vehicles equipped with data-producing technology that can be retrieved by the city government for use.

In our case tracking is enabled by the fact that a fair percentage of vehicles have networking capabilities. They are able to produce location data in the form of longitude and latitude matched to a vehicle. These will be visible in the data. Vehicles without such capabilities will not be visible, which poses a problem of representation for the residents driving these vehicles, but also for the accuracy of whatever further knowledge is made based on the data. The behavior of these un-networked vehicles has an effect on the city mobility in the physical environment, but as the behavior cannot be seen in the data, we also cannot know how this behavior impacts the behavior that IS seen. This would have to be mitigated somehow.

Pedestrians are also producing data and are visible in the dataset in question. The same point can be made about pedestrians as above: for whatever reasons, some pedestrians may not be equipped with the technology to produce and send data to the city government. Only the data-producing and those contributing to the view of data are represented. In the case that and they are contributing data it is detrimental to their representation as citizens, and for those other pedestrians the accuracy of improvements for those who ARE represented.These are the human actors represented in the data. The other aspects of infrastructure visible is location in the form of longitude and latitude. We cannot know where every resident is at any given time. The data we have is timestep data that records the location and other data points of a certain pedestrian of vehicle at a point in time. Using the data effectively means we have to look at throughout recorded or captured time or in real time.

Areas where data can be used for improving person mobility and transport management are for example congested areas. Roads that are bottleneck at certain times of day for a period of time. We may also adopt an environmentally friendly strategy and aim to improve how the city is planned to have green areas further away from congested areas so that pollution does not take away from the positive environmental effects of parks and green spaces. A city could look into having areas be car-free at certain times, maybe even permanently. Signs could be implemented to provide better updated information on congested roads.

We want to identify patterns of stopping of vehicles. This could indicate that they break a lot and that the road is congested. This can will lead to higher fuel consumption and time consumption for the people driving. We can propose improvements so the city can become more green and more efficient.

Question 2 (C/T): Batch layer

1. **Car density**

How can we use the data to gain insight to the density of cars, and why does such insight matter? It can matter if you infer that a high density of cars is cause for elevated pollution levels in areas affected by this. Knowing where car density is high can allow urban planners to take this factor into account when implementing green spaces and outdoor recreation facilities. Two approaches present themselves when you have the insight: you can either have more green spaces like parks in areas with high car density to counter the emissions' effect on air quality and aural environment in those same areas. The parks and recreational areas would then essentially act as safe havens for residents of noisy and polluted neighborhoods. The other way to take action upon the insight would be to use car density measurements for mapping areas where green spaces and recreational areas should not be implemented, and instead moving clusters of such facilities to areas further from the noise and pollution, so the benefits of green and recreational spaces are not hamstrung. In this view we would have to take the lanes as best indicator of geographical space, which is arguably a poor indicator of such. Since lanes is a part

of an edge, which then may only be a stretch of an entire road, based on our analysis of that value.

Data: lane, AVG(lat, lng), COUNT(cars), grouped by lanes, ordered by count desc
Extract: Top 1000
After: Put top 1000 in map (tableau) and look if there's any recreational areas around

2. **Person density**

Car free days and car free areas are popular concepts in city planning. In order to utilize the city space we want to implement car free areas in Copenhagen. In the process of choosing a car free area, we want to create a view that enables us to identify areas with high density of pedestrians. This could also be combined with the density of cars, so the disturbance of car traffic is minimised.

Data: edge, average (lat, lng), count of persons, grouped by edges, ordered by count desc
Extract: Top 1000
After: Put top 1000 in map (tableau)

3. **Count number of times a car stops and starts on specific edges**

We want to optimise the flow of traffic through the city. This is desirable both because of increased efficiency for the commuting people and good for the environment, since the cars are spending less time on the road queuing. In order to establish where in the city there are bottlenecks in the traffic flow, we are establishing a view where we can calculate the starts and stops on a given road. Our hypothesis here is that fewer stops and starts equals a smoother flow of the traffic.

So what is a possible way to gauge the number of stops for several vehicles?

With a real time overview of the traffic flow, we could be able to send direction information to the smart cars in the grid and guiding them to take the optimal route through the city. With machine learning the algorithm can be optimised with historical data of driving patterns in order to optimise the route for the specific car and the specific driver.

Data: Count the number the cars stop and start again on lanes, ordered by average speed and number of cars
Extract: Top 1000

Question 3 (T): Master data set

The data is made available in XML.

A. Describe the pros and cons of two different systems to store and manipulate this data (to answer this question you will need to make assumptions about the kind of processing required that are consistent with your answers for Question 2)

1. Solution with plain mapreduce

Pros	Cons
● Very low-level in the sense that we have absolute control of the mapping and reducing logic and direct access to the HDFS. ● Automatically allow parallelization. ● Does not have to load in all data for processing, if the master data set is partitioned.	● Very low-level, because we have to take care of details that could be hidden from the use of frameworks (for instance setting up the files that are used in the job). ● Specifics of the storage of data has to be taken care of directly. ● If multiple MapReduce iterations must be used, the output from one reducer, must be written to HDFS, before it can be used from the next mapper.

2. Solution with Hive

Pros	Cons
● Tez provides pipelining ● Easy batch processing - Batch views can be written as HiveQL queries, which looks like SQL. ● Benefits from MapReduce ● Easy abstraction of data as structured data (high level abstraction)	● harder to integrate frameworks with each other ● harder to inject data, because there's more restrictions defined by the frameworks ● hard to understand the underlying processes, since its high level

B. Ingest the data into the system of your choice in a way that supports the definition of views defined in Question 2.

We decided on implementing the solution with Hive. To approach the difficult problem of ingesting 80GB of XML data, two things must be taken into account. First of all 80GB of data cannot be put into memory and probably neither into swap space, therefore we need to make sure that when we ingest the data, it is done in a manner which splits the input into chunks and handles them individually. This leads us to the second problem. The XML format is a tree format structure, and it is difficult to split into parts which can be handled by different processes, therefore a flatter format will do better. Because of this we decided to convert the XML data file to CSV which is a flat file format. Then we used Hive SerDe to import the CSV file into a table format in Hive. By using Hive we are able to express the batch view via an SQL interface, but still using map-reduce underneath such that we harness the power of that framework. We used the following command to create the tables in Hive and import the data:

```
CREATE TABLE IF NOT EXISTS sumo_data (timestep_time DOUBLE,
vehicle_id INT, vehicle_x DOUBLE, vehicle_y DOUBLE, vehicle_z
DOUBLE, vehicle_angle DOUBLE, vehicle_type VARCHAR(25),
vehicle_speed DOUBLE, vehicle_pos DOUBLE, vehicle_lane VARCHAR(25),
vehicle_slope DOUBLE, person_id INT, person_x DOUBLE, person_y
DOUBLE, person_z DOUBLE, person_angle DOUBLE, person_speed DOUBLE,
person_pos DOUBLE, person_edge VARCHAR(25), person_slope DOUBLE)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH
SERDEPROPERTIES ( "separatorChar" = "\;", "quoteChar" = "'",
"escapeChar" = "\\" ) STORED AS TEXTFILE;
```

Creating the table with SerDe import options

```
LOAD DATA LOCAL INPATH 'project3/FCDOutput.csv' INTO TABLE
sumo_data;
```

Loading data into the created table

This command turned out to import all columns as strings when using SerDe to import the CSV, although we did create a typed table beforehand. We could probably achieve a noticeable speedup by making the column types correct, and splitting the input into two tables, one for persons and one for vehicles, or by creating indexes on the most used columns.

It should be noted that we were not successful in ingesting the data in such a way that view 3 is easily computable (though it is possible), but as we will see in the next section an approximation of it is still possible.

C. Implement the derivation processes that produce the views defined in Question 2

For computing the batch views we used Hive SQL scripts, which are stated below.

View 1: **Vehicle density**

```
SELECT COUNT(DISTINCT vehicle_id) AS vehicle_count,
       AVG(CAST(vehicle_x AS DOUBLE)),
       AVG(CAST(vehicle_y AS DOUBLE)),
       vehicle_lane
FROM   sumo_data
WHERE  vehicle_id IS NOT NULL
GROUP BY vehicle_lane;
```

This query calculates the vehicle counts on different lanes (which can be mapped to actual roads if wanted) over the entire data set (2 hours). Furthermore we include an average of the latitude and longitude (vehicle_x and vehicle_y) of the cars on that lane. This information is only included to have a rough idea of where the lane is located, without merging the data with map information. The reason for the cast to double is because the data is stored as strings as written in the previous section. A snippet of the results can be seen below.

vehicle_count	vehicle_x	vehicle_y	vehicle_lane
648	12.48229964	55.72693068	122492607#3_0
557	12.49144177	55.72215528	119507350_0
555	12.54158527	55.66268498	27409296#1_0
531	12.5412276	55.66199099	26482048#0_1
528	12.54150479	55.66246743	:10437896_1_0
523	12.51613423	55.6636757	25912891#2_0

This table for example shows us that lane *10437896_1_0* has *528* distinct cars in total over the two hours. We can also see that the lane with the most traffic is the first with about 20% more cars than the other lanes. It is clear that given this view a number of interesting queries can be made.

View 2: Person density

```
SELECT COUNT(DISTINCT person_id AS person_count,
       AVG(CAST(person_x AS DOUBLE)),
       AVG(CAST(person_y AS DOUBLE)),
       person_edge
FROM   sumo_data
WHERE  person_id IS NOT NULL
GROUP BY person_lane;
```

This query is almost identical to the one above. It just calculates number of people on edges, rather than vehicles on lanes.

person_count	person_x	person_y	person_edge
3208	12.58723354	55.67475359	-1694109
2970	12.58780263	55.67437389	-322164666
2903	12.58865936	55.67388881	-1694110#1
2876	12.57654324	55.67056482	225964063#0
2863	12.51608017	55.66418092	115678974#0
2861	12.51618019	55.66432078	-78412354#2

The table shows us that on edge *-1694109* there have been a total of *3208* people, and that edge is the edge with the most people in the table above. Just like the table above it is clear that given this view a number of interesting queries can be made.

View 3: Count number of times a car stops and starts on specific edges

```
SELECT AVG(vehicle_speed) AS avg_speed,
       COUNT(DISTINCT vehicle_id) AS vehicle_count,
       AVG(CAST(vehicle_x AS DOUBLE)),
       AVG(CAST(vehicle_y AS DOUBLE)),
       vehicle_lane
FROM   sumo_data
WHERE  vehicle_id IS NOT NULL
GROUP BY vehicle_lane
ORDER BY vehicle_count DESC, avg_speed ASC;
```

This query calculates the average vehicle speed per lane. As the other queries it includes an estimate of the center of the lane. This view should have counted the number of times a car

stop on a given lane, but this is hard to express in HiveQL if we do not want to include the same car multiple times, if it is stopped for more than one second. This table might not be interesting in itself, but if we match this information with data about speed limits for the given lane (or road), then we can see roads where the majority of cars is driving at a much slower speed than the limit, which might indicate a problem on the road.

avg_speed	vehicle_count	vehicle_x	vehicle_y	vehicle_lane
1.735935597	648	12.48229964	55.72693068	122492607#3_0
0.7336198794	557	12.49144177	55.72215528	119507350_0
2.339899297	555	12.54158527	55.66268498	27409296#1_0
13.08320411	531	12.5412276	55.66199099	26482048#0_1
7.8700208	528	12.54150479	55.66246743	:10437896_1_0
1.25324843	523	12.51613423	55.6636757	25912891#2_0

The table above shows that the average speed of vehicle lane 119507350_0 is 0.734, which is by far the lowest value of the lanes shown above. This might indicate that the vehicles are stopping often, for example due to traffic or intersections.

Interestingly enough, the originally planned batch view would have been more easily expressed in pure Map-Reduce with java since it requires some fine manipulation to find consecutive speeds of zero for the same cars and mapping them to one entity.

Question 4 (C/T): Data processing

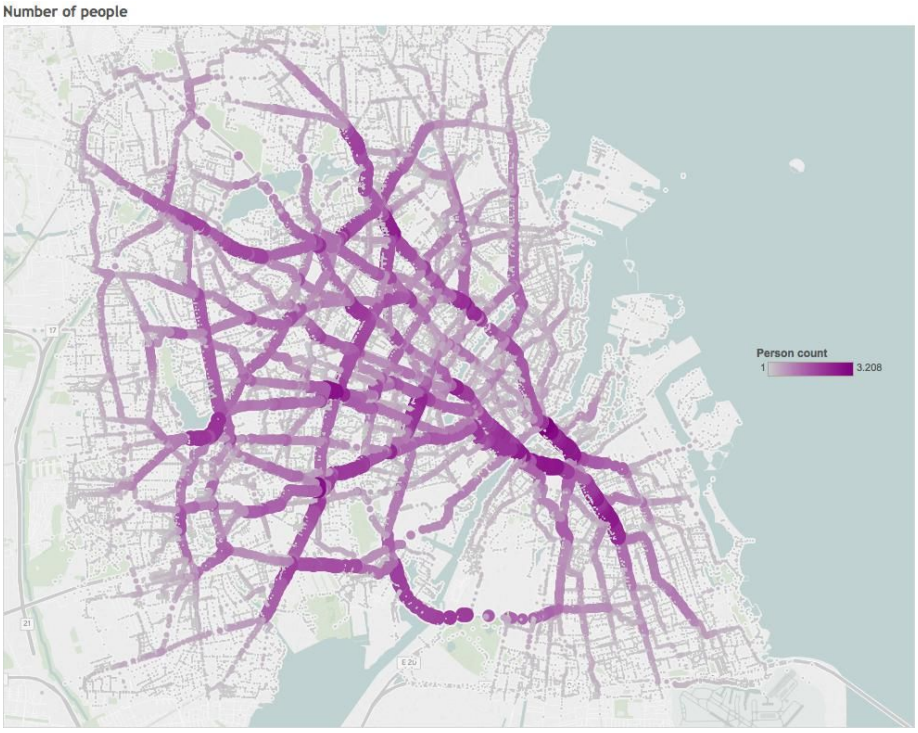
Car density

With the number of cars on each lane, we can visualize where in the city the car density is highest and lowest. The visualization of number of vehicles below is made without a background map, but you are still able to see the outline of the city roads. As expected the main gateways to the city carry more cars, but you can also use the map to investigate which of the smaller roads have unexpected high usage.

Number of vehicles

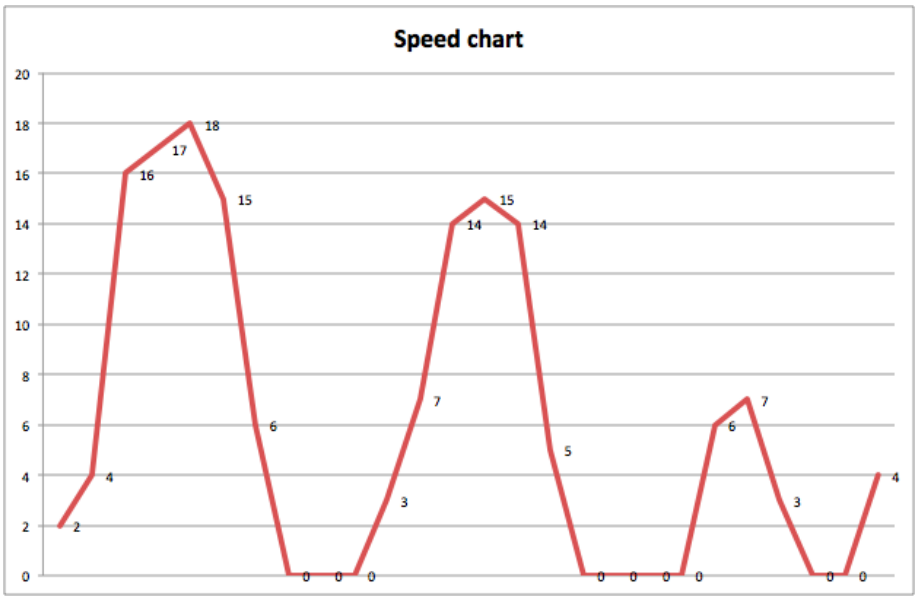
Person density

Combined with the previous vehicle count map, we can use the person count map to see where we could potentially implement car free zones. If we identify an area with many cars, the implications of making this area car free will be too immense and disruptive for the traffic flow. On the other hand, if there are no people in the area, not many would be reaping the benefits of the car free zone, which makes the intervention redundant.



Count number of times a car stops and starts on specific edges

In order to count the number of stops and starts, we need to establish a way to define a stop. On the 'Speed chart' figure we have exemplified data from a vehicle. On the y-axis we have speed and then we have time on the x-axis. When we see the data like this, we can see that there are three events where the vehicle is stopping, but there are nine events of the speed being zero. We can therefore not just count the number of zeroes, but we have to add the condition that if the previous speed were zero, then the event is not a stop, and if the previous event is not zero, then the event will be defined as a stop.



In our final iteration of the data processing we did not manage to produce the stop/start count. We did manage to extract the average speed on the top 100 lanes in the dataset. This gives us an idea of how we can use the data. In the average speed visualization, we can see how the speed distribution is on a given road. If this view could be converted to a real time visualization, we would be able to analyse events that reduces speed.

Average speed

Question 5 (C/T): Log

Critical:

- 08-11-16: We have chosen a scenario (1), and can start defining mobility tracking etc. and the focus of our committee in developing this plan (i.e. if we want to be greener, smarter, more efficient). **But to choose points of action** we need to look at the data which is still being downloaded.
- 08-11-16: How to go from source data to **data we can use in visualization programs** (we discuss Tableau). This is for the purpose of storytelling in the pitch/proposal (Question 6).
- 08-11-16: We start on C questions while T discuss **how to store and handle the data**.
- 17-11-16: Defining views within the “green vision”--maybe expanding to a smart city discourse. Batch views that are the same for T, we see as different stories still.
- 17-11-16: Discussing with T what is possible in terms of batch views correlated with what might also be interesting, given that we have to frame, storytell and pitch. Given the attributes in the data, how can we combine them, and potentially outside sources and data, to do something interesting.
- 17-11-16: T are experiencing a bottleneck with some of the tools and platforms they have to use, which means we also stall at some points, or may come to. But we have defined our views into the data and we, C, will now work on articulating them further.
- 17-11-16: There will be some sort of “data handover” on Thursday next week, so we can start to clean data for visualization purposes.
- 22-11-16: We work separate from T today, because we have managed to arrange other times for meeting this week. They work on the batch views, while we work on framing the views. We agree to be “less critical” in rhetoric when storytelling and proposing/pitching, but make sure to note the critical reflections for the designated paragraph. We hope to have data available for visualization later this week.
- 29-11-16: “Getting data” for visualization can be something entirely different than “getting it” and using it for analysis in other ways (as we have done in previous projects). Data for visualization in a program like Tableau means a lot of cleaning data, is our assumption. This creates a kind of missing link in the workflow, which is then not exclusive to either T or C.

Technical:

- The entire cluster had a lot of down time (including the host of the virtual machines).
- The different frameworks on the cluster, that we were using were down on the server.
- After importing the csv to Hive with SerDe, we realised that SerDe overwrites every data type to String.
- We first tried to compute the XML to CSV in memory, because of wrong instructions on how the xml2csv-tool worked, but obviously it didn't work.
- Then we tried using the CSV Philippe had made, but we were missing some columns for our views.
- We had to adapt the XML schema to get the columns we needed. For instance, in the schema-file that was referenced from the data-xml-file, there was no representation of persons at all.
- When preparing for this course another time, please be sure to have space enough for all groups. We know that some of it is for illustrative purposes, but it does not make sense to block out groups from doing projects, because other groups are filling up the entire drives.
- It is quite problematic that we only have a single entry point to the HortonWorks cluster. If one group are running a job on the server (not the cluster), then it is impractical for other groups to do anything on that server.
- It is practically impossible to do any work on the cluster during the exercise lessons, so instead we have opted for working in the weekends sometimes, or use our own laptops for computation, which was not the intent for this course. It is also hard to coordinate, because we have to work, when no one else is using the cluster, and it is hard to know when that's the case before trying.
- We tried to import the CSV to Hive in one of the exercise lessons, but we had to wait to import it until the following Sunday, because the cluster and/or Hadoop, YARN, Tez, HDFS, Hive or similar was down untill then.

Question 6 (C/T): Selling it

The visualization above makes city mobility visible in one way. It focuses on a crucial part of how our citizens move through our city, with the use of cars. Representing the city in this way gives us the ability to make sense of and intervene in traffic management.

Cars making multiple stops on the roads is bad for both the flow of vehicular traffic and air quality in our city. In that way it is bad for all residents of our city. Wanting to improve mobility with a green view in mind needs special attention focused on minimizing the stopping. We need to stop stopping and start driving. The cars that move through our city are increasingly equipped with networking capabilities, which provides a good foundation of existing infrastructure that just needs to be integrated with the correct data practices and analyses.

From data to algorithms
To give you an idea of how data harnessed from pervasive data-producing technologies can be used towards the goal of stop stopping let us turn our attention towards the simulation dataset. Scaling such a dataset provides the means for analysis that can determine the amount of stops cars have on certain roads. But what is the use of such a metric? We argue that knowing where stopping is most prevalent should be integrated in traffic optimization algorithms. We would use these algorithms to direct the movement of vehicles throughout the city, recommending alternate routes of making it from A to B. This would happen in real-time given we have the means of implementing a speed layer into our existing data capturing infrastructure.

From algorithms to drivers
The envisioned scenario when the algorithms and the infrastructure are in place is that we have a automated system that simultaneously analyse traffic flow while directing vehicles by the optimal route through the city. The system will be able to learn from previous data in order to optimize its recommendations. This combination of Big Data and machine learning is bringing our city infrastructure into the future of transportation.

Benefits for pedestrians
We have not forgotten about citizens who choose to walk our streets by foot. There are certainly more aspects to managing city space than the flow of vehicular traffic. Our plans to optimize the flow of cars on the roads has positive effects for foot pedestrians as well, because minimizing stopping is at the same time a way to minimize emissions so that air quality becomes better. We are not only concerned with bettering the roads we travel, but also the air we breathe.

Some challenges
Some of the additions to the current big data processing infrastructure are arguably very costly. The best use of data for traffic flow management in the form that we propose requires a speed layer that gives access to data real-time. There is no doubt that we need to have a system of data governance in place. Ethical concerns includes not getting consent to capture data for learning about the amounts of, but also for the aspect of the system that aims to communicate with drivers of cars. If we want to push information about alternate routes to drivers. In setting up practice for getting consent, we can anticipate an unravelling effect which needs to be addressed. We need support in the form of policies that continue to incentivize buying cars that have the necessary technological capabilities to provide us data as well as receive our notifications of redirection. We do need to be aware of how changed mobility can affect other stakeholders than just drivers. Certain areas, such as residential ones, may be planned and commodified in ways that rely on the roads being less quiet, polluted and trafficked. This is an environmental impact that warrants concern.

If we are able to critically reflect on the above we have no doubt that the system can be designed to best alleviate the concerns while having a positive impact on the flow of traffic in our city.

Critical reflection

Our proposal is arguably very focused on increasing mobility of traffic for vehicles. Although we argue for some benefits for pedestrians, we can question whether there is too heavy a focus on benefits for those driving cars. There is also the concern that we use this data to design a system with which we aim to make better use of less trafficked roads to relieve the most trafficked ones. But in doing so we should take into account that the areas surrounding the former will be impacted by the new flow of traffic. Some citizens may have taken up residency in areas with less trafficked roads precisely because they are quieter and cleaner. The environmental impact, although briefly addressed in the above, is no small matter. It suggests that traffic management should happen closely with urban planning.

In our proposal we argued for pushing transportation policy towards increasing the number of cars that can take part in the improved mobility infrastructure. There might be social drawbacks to this, because it requires that citizens are able to spend the money needed to either equip their cars with the necessary technology or buy cars that are already equipped. Regarding consent, we draw attention to this in our proposal. But the fact remains, that there is a real chance of marginalizing those who do not give consent to have their data collected, stored and used. It takes the form of an unraveling effect (Peppet 2011) that makes those that agree to give consent viewed as in line with the interests of policy-makers, while does who do not wish to participate in the data handover find themselves underrepresented.

Designing and implementing a system where an algorithm is tasked with directing the flows of a city means we are introducing a new knowledge logic (Gillespie 2014) into how mobility is practiced within the city. This logic relies on classifying certain instances of data on traffic and making decisions upon those classifications means that other data and instances of traffic are omitted (Bowker 2005). This is effectively a way of simplifying a very complex reality of mobility in our city.

In terms of how we reach out to potential investors and adopters of our proposed system, we use the aid of visually representing data. *“Visualization can allow humans to interface with and make sense of a large amount of ever-changing big data”* (Kim et al 2013, p. 5.) We wanted the audience or intended readers of our proposal to think of mobility in a way that blends well with how mobility data could be visualised, by the use of maps.

References

Bowker, G. Introduction - from Memory Practices in the Sciences (2005) MIT Press

Kim, J., Lund, A., & Dombrowski, C. (2013). Telling the story in big data. interactions, 20(3), 48-51.

Gillespie, T. (2014). The Relevance of Algorithms. in Gillespie, T., Boczkowski, P. and Foot, K. (Eds) Media technologies: Essays on communication, materiality, and society, Cambridge, MA: MIT Press

Peppet, S. R. (2011). Unraveling privacy: The personal prospectus and the threat of a full-disclosure future. Nw. UL Rev., 105, 1153.

References

[1] High-throughput, low-latency, and exactly-once stream processing with apache flink. http://data-artisans.com/high-throughput-low-latency-and-exactly-once-stream-processing-with-apache-flink/.

[2] Official website for apache flink. https://flink.apache.org/.

[3] Real-time stream processing: The next step for apache flink. https://www.confluent.io/blog/real-time-stream-processing-the-next-step-for-apache-flink/.