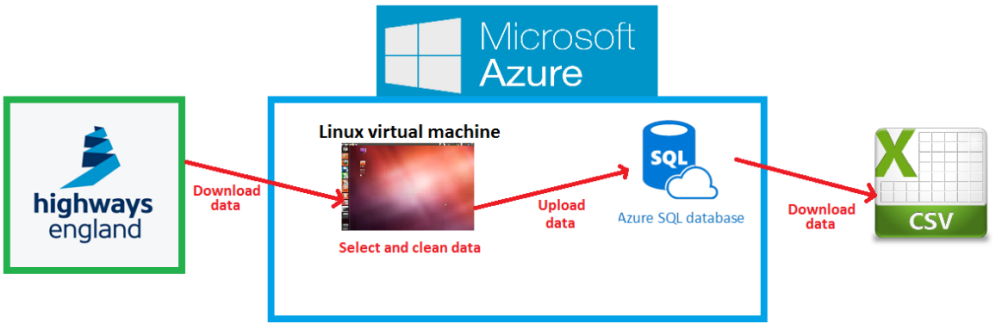
Summary and step by step documentation of traffic data acquisition

This is a summary and a step by step documentation of how the Thales RSG created the SQL database in the cloud containing the Highways England data of M6 and M11 from March to May 2016, and how the CSV files with the traffic data were created from the SQL database.

Summary

In Microsoft Azure a Linux virtual machine and a SQL database were created. The Linux virtual machine ran Python code to download all traffic data of March to May 2016 from the Highways England website called Traffic England. Then, the data from the M6 and M11 motorways was selected, and the rest discarded. After this, the data was cleaned and uploaded to the SQL database. Finally, the tables made were stored in CSV files. The next picture summarizes this process in a visual way.



Step by step documentation

The following are the technical steps done to accomplish all the tasks described in the summary. All the files referenced here are in the directories where this file is located.

A. Downloading the NTIS model and the Highways England traffic data for one day
   - Access to the traffic data is provided by Highways England via Traffic England website. We logged in to the website using this link: https://trafficengland.info/app/login and a username and password given by Thales UK.
   - There is a section called “DATD” (Daily Aggregated Traffic Data), in which information about the daily traffic data in all UK for the past three months can be found. Initially, we just downloaded the data for the most recent day available (March 29th, 2016). The data for each day consumes a little more than 2 GB.
   - There is other section called “NTIS Model”, where the current and past versions of the NTIS models can be downloaded. The NTIS model shows the information of all the nodes, links, sites and measurements of the UK highways network. We downloaded the most recent version (March 29th, 2016).

B. Setting up the cloud

- A virtual machine running Linux and an empty SQL database were created in Microsoft Azure (https://azure.microsoft.com/en-gb/), which is a platform of integrated cloud services.

C. Creating SQL empty tables in the cloud
   - The DBeaver software (a universal SQL client) was installed in a laptop to enable a direct connection to the cloud SQL database.
   - Using this software, a SQL Server script called "database.sql" was written to create the main tables of the SQL database. The data downloaded for March 29th, 2016 was used to understand the structure of data. The following is the summary of what this script does.
     - The first four empty tables created in the code store the information of the NTIS model. The description of each table is:
       - Nodes table: made to contain all nodes in the NTIS model, with their id, latitude and longitude.
       - Links table: made to contain all links in the NTIS model, with their id, type, direction (e.g. northbound or southbound), location and the two nodes to which each link is connected.
       - Sites table: made to contain all sites (places where measurements take place) in the NTIS model. Each site has id, site reference, link id (i.e. link where the site is located), location and number of measurements.
       - Measurements table: made to contain information of the types of measurements being done by each site.
     - The rest of the file has the code for creating empty motorway tables in the SQL database that will contain the measurement information for the motorways of interest, which are M6 and M11 (M25 is included in the code because in the beginning we considered it). Each motorway has four associated tables:
       - Midas table: made to contain the actual measurements (mainly flow and speed) done in each site every minute for the selected dates.
       - PTD table: made to contain data of traffic flow, traffic concentration, traffic speed, and traffic headway.
       - Travel time table: made to contain travel time, free flow time and profile time for each link for every minute of the selected dates.
       - Events table: made to contain information about events in every link of the NTIS model (e.g. maintenance, weather or accident events)

D. Filling SQL empty tables with Highways England data
   - At this point, we had two types of empty tables to fill: NTIS model and motorways tables. Several Python 2 scripts that run in the Linux virtual machine of Microsoft Azure cloud were used to fill the tables. Note: the virtual machine was used because running these scripts in a laptop is very demanding.
   - Filling NTIS model tables

- The file called “db_interface.py” connects to the database, and then calls different “insert” functions to insert the nodes, links, sites and measurements in the respective SQL tables. These functions are defined in the file “db_module.py” (located in same directory).
- Each “insert” function defined in “db_module.py” makes two things:
    - 1. Use a “get” function to extract the NTIS model information from the NTIS model files downloaded from Highways England: “NTISModel-MeasurementSites-2016-03-29-v4.1.xml” and “NTISModel-PredefinedLocations-2016-03-29-v4.1”. The “get” functions are defined respectively in the scripts: “nodes.py”, “links.py”, “sites.py” and “measurements.py” (located in same directory).
    - 2. Use the “pymssql” library to connect to the cloud and write the data obtained by the “get” functions into the SQL tables.
- Note: the “tags.py” (located in same directory) contains auxiliary functions used by the other files.

- Filling motorway tables
    - The filling of the motorway tables was managed by “scrape.py” and other scripts imported by it which are in the same directory. “scrape.py” does the following:
        - Creates a connection to SQL database in the cloud and downloads the information of the NTIS model.
        - Then it makes a loop through all the dates (March to May), and for each, connects to Traffic England server to download, unpack and read the info for the given date. Then for each date the script extracts the relevant info and stores it in the SQL database tables in the cloud.
        - The following is a description of the files imported by “scrape.py” (located in same directory).
            - “get_file.py”: definition of a function for downloading the traffic data from Traffic England for a given date.
            - “events.py”: definition of a function to insert events columns in the motorway tables.
            - “get_links.py”: functions for each motorway are defined, used to obtain all the links belonging to the motorway using a latitude/longitude window.
            - “get_sites.py”: functions for each motorway are defined, used to obtain all the sites belonging to a motorway (+++ the other function).
            - “days.py”: functions are defined to be able to insert the date in the motorways tables.
            - “directory.py”: functions for extracting and erasing zip files are defined.
            - “ptd.py”: a function for inserting the data to the PTD table is defined. The PTD table contains data of traffic flow, traffic concentration, traffic speed, and traffic headway.

- "travel_time.py": a function for inserting the travel time in a motorway table is defined.
- "midas.py": a function for inserting the midas measurements in a motorway table is defined.
- "helpers.py": auxiliary functions are defined.

E. Creating final motorway tables made specially made to convert to CSV files
   - We created SQL tables specially to download to CSV files. The CSV files with the data are the ones used by the R code that runs the algorithms. The file called "QUERY_DATA_v2.py" does this job. The file must run twice: for M6 and M11.The following describes the code in the file:
     - In the first 15 lines a "pymssql" library is imported, the connection to the SQL database is made, and the motorway and dates are selected.
     - In section "#Creation of m#_data (only once)" a query for creating a table for the selected motorway is defined. The colums of the table for each motorway are: link_id, m_date, absolute_time, travel_time, free_flow, profile_time, traffic_concentration, traffic_speed, traffic_flow, traffic_headway, congestion_event, poor_event, other_event, day_week. Note that the columns agree to the ones in the CSV file.
     - In section "#Insertion of links and dates" a query for filling the colums: link_id, m_date and absolute_time is defined.
     - In section "#Insertion of data" a query for filling the columns: travel_time, free_flow and profile_time, traffic_concentration, traffic_speed, traffic_flow and traffic_headway is defined.
     - In sections "#Creation of m#_events_temp (temporal)" and "#Creation of m#_data_events (temporal)…" queries for doing temporal tables are defined. The tables are meant to be used by the queries in the next section to fill the events columns.
     - In section "#Updating data with events information" a query for filling the columns: congestion_event, poor_event and other_event is defined.
     - In section "#Drop temporal tables" the temporal tables created are removed.
     - In section "Execution" the queries defined are executed.

F. Creating queries to download CSV files
   - The file called "query.py" has a template function called "execute_to_csv" for making queries to the SQL database. By editing this file, two files with queries for downloading the data into CSV files were created: "QUERY_PRINT_DATA_v2.py" and "QUERY_PRINT_LINKS.py".
   - "QUERY_PRINT_DATA_v2.py" initially defines a function called "execute_to_csv" made to write the data from the SQL table of a motorway to the CSV file of the motorway. The columns to fill in the CSV file are: link_id, m_date, absolute_time, travel_time, free_flow, profile_time, traffic_concentration, traffic_speed, traffic_flow, traffic_headway, congestion_event, poor_event and other_event. Then a main function is defined to select the motorway and write the data in the CSV file. The two files created are "m6_data.csv" and "m11_data.csv".

- In a similar way than “QUERY_PRINT_DATA_v2.py” (since a template was used), the file “QUERY_PRINT_LINKS.py” is made to download all the links of a motorway into a CSV file. The columns in the CSV file are: link_id, link_type, link_length, link_direction, link_location, from_node, to_node, from_lat, from_lon, to_lat, to_lon.