CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT

BIG DATA PROCESSING

ASSIGNMENT 2: SPARK STREAMING & SPARK STRUCTURED STREAMING

BACKGROUND.

The bus 208 is a common way of doing the commute between CIT and City Centre.
In particular it is the transport system used by myself last year and perhaps used now by most of you as students of this module. If this is the case, as you have probably experienced by yourself the bus is a bit unreliable in terms of its timetable schedule.

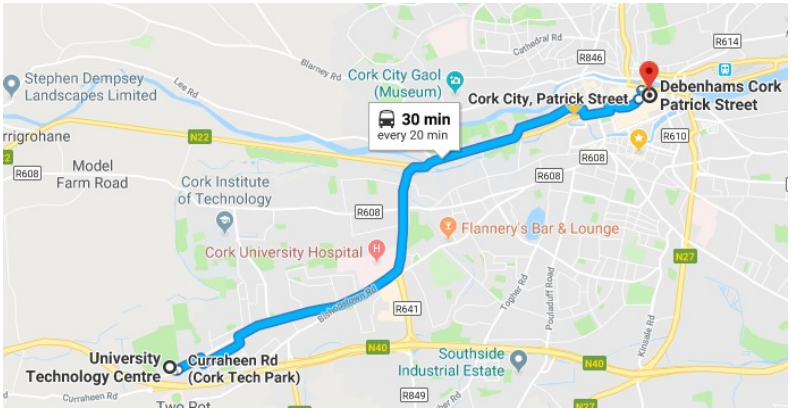
Bus Éireann

While historic information on the location of the buses of the line is not available, real-time information can be obtained through the LED displays at bus stops and through the Transport for Ireland Phone App: https://www.transportforireland.ie/available-apps/

MY_DATASET_ FOLDER

My former colleague Michael O'Keefe collected data from mid June 2016 to late September 2017 by creating a service quering the API every 3 minutes from 6:30am to midnight and gathering all entries of a day into a file.

The original dataset has been cleaned and enriched for the assignment:
- The files for the period 01/09/2016 – 31/08/2017 have been selected.



- The entries for the following 6 stops have been selected:
  - Patrick Street (Debenhams) – Curraheen – 240491.
  - – Lotabeg – 240171.

- UCC – Western Gateway Building – Curraheen – 240561.
  – Lotabeg – 240101.
- CIT – Technology Park – Curraheen – 240001.
  – Lotabeg – 241111.

- Only GPS-based entries have been selected.
- Only entries for the following time periods have been selected:
  - Morning: 7am – 10am.
  - Evening: 4pm – 7pm.
- The timetable scheduled for the buses has been extracted from:
http://www.buseireann.ie/inner.php?id=406&form-view-timetables-from=&form-view-timetables-to=&form-view-timetables-route=208&form-view-timetables-submit=1
Note: Thanks to Diarmuid for letting me know this info was available.

All in all, the entries have the following schema:
num;name;direction;day_of_week;date;query_time;scheduled_time;expected_arrival_time

- Finally the dataset has been re-formatted to simulate its arrival over time (streaming).
  12 files have been created for the complete dataset:
  - file_00_(07_00_16_00).csv --> All year entries for the intervals [7am, 7.15am) and [4pm, 4.15pm)
  - file_01_(07_15_16_15).csv --> All year entries for the intervals [7.15am, 7.30am) and [4.15pm, 4.30pm)
    ...
  - file_11_(09_45_18_45).csv --> All year entries for the intervals [9.45am, 10.00am) and [6.45pm, 7.00pm)

I can pass you both the original dataset and the script "A02_dataset_cleaning.py" for turning it into the provided dataset if you want to take a look at it again very geeky implementation :)

The folder **my_dataset_single_file** (just contained the file_00_(07_00_16_00).csv) is provided for Part1 and Part3.
The folder **my_dataset_complete** (containing the 12 files) is provided for Part2 and Part4.

MY_MONITORING FOLDER

This folder is automatically generated by the programs, to simulate the streaming arrival of the files of MY_DATASET.

MY_CHECKPOINT FOLDER

This folder is automatically generated by the programs to provide fault recovery capabilities during the computations.

MY_RESULT FOLDER

While Part1, Part2 and Part3 are expected to print the result by console, the Structured Library has a bug in Databricks (not allowing to do so). Thus, Part4 is expected to print the results to files in FileStore.
For this latter the script "structured_streaming_post_process_my_result.py" is provided (another very geeky implementation re-formatting the solution provided by the Structured Streaming library into human legible format- :)

MARK BREAKDOWN.

The assignment is worth 50 marks. It consists on 4 parts with 5 exercises each: 20 exercises. All exercises are worth the same: 2.5 marks per exercise.

The submission of each exercise is optional (it is perfectly fine to not submit some exercises). Marks will be given to each submitted exercise in the following basis:

A. Exercise is correct and the code is efficient: 100% of marks.
B. Exercise is correct but the code has some efficiencies: 75% of marks.
C. Exercise is not correct due to a single error in the code: 50% of marks.
D. Exercise is not correct due to multiple errors in the code: 25% of marks.
E. Exercise has not been attempted or has been barely attempted: 0% of marks.

SUBMISSION DETAILS.

Submission deadline: Sunday 22th of December, 11:59pm.

Please submit to Canvas (folder 5. Submission Links) the following files:

- Part 1 -> A02_Part1_RDD.py
- Part 2 -> A02_Part2_Streaming_RDD.py
- Part 3 -> A02_Part3_DF.py
- Part 4 -> A02_Part4_Streaming_DF.py

Assignment Demo.

A demo for the assignment will take place from Week 15 onwards.
I will organise a brief individual interviews with each student to run the code and discuss about it (student is expected to explain the approach followed when tackling each exercise and explain the code being submitted).

The demo is mandatory for the assignment to be evaluated.

ASSIGNMENT 2– PART 1 (SPARK CORE) (Week 10)

Use only RDD-based operations.

Use the dataset folder my_dataset_single_file to work just with the file_00.

Exercise 1:

- Number of measurements (lines).

Exercise 2:

Let's focus just on the measurements of the station 240101 (UCC WGB – Lotabeg).

- Amount of days in the calendar year (01/09/2016 - 31/08/2017) for which data is collected.

Exercise 3:

Let's focus just on the measurements of the station 240561 (UCC WGB – Curraheen).

- Amount of measurements where *scheduled_time* <= *expected_arrival_time*.
- Amount of measurements where *scheduled_time* > *expected_arrival_time*.

Exercise 4:

Let's focus just on the measurements of the station 241111 (CIT Tech. Park – Lotabeg).

- Get a list with the scheduled_time values found for each day of the week.

Exercise 5:

Let's focus just on the measurements of the station 240491 (Patrick Street – Curraheen).

- Consider only the months of Semester 1 (i.e., months 09, 10 and 11) and aggregate the measurements by month and day of the week to compute the average waiting time (i.e., expected_arrival_time - query_time). Sort the entries by decreasing average waiting time.

ASSIGNMENT 2– PART 2 (SPARK STREAMING) (Week 11)

Use only RDD-based operations.

Use the dataset folder my_dataset_complete to simulate the streaming arrival of the 12 files and their processing in separate batches.

Exercise 1:

- Number of measurements (lines).

Exercise 2:

Let's focus just on the measurements of the station 240101 (UCC WGB – Lotabeg).

- Amount of days in the calendar year (01/09/2016 - 31/08/2017) for which data is collected.

Exercise 3:

Let's focus just on the measurements of the station 240561 (UCC WGB – Curraheen).

- Amount of measurements where *scheduled_time* <= *expected_arrival_time*.

- Amount of measurements where *scheduled_time* > *expected_arrival_time*.

Exercise 4:

Let's focus just on the measurements of the station 241111 (CIT Tech. Park – Lotabeg).

- Get a list with the scheduled_time values found for each day of the week.

Exercise 5:

Let's focus just on the measurements of the station 240491 (Patrick Street – Curraheen).

- Consider only the months of Semester 1 (i.e., months 09, 10 and 11) and aggregate the measurements by month and day of the week to compute the average waiting time (i.e., expected_arrival_time - query_time). Sort the entries by decreasing average waiting time.

ASSIGNMENT 2 – PART 3 (SPARK SQL) (Week 12)

Use only DataFrame-based operations.

Use the dataset folder my_dataset_single_file to work just in the file_00.

Exercise 1:

- Number of measurements (lines).

Exercise 2:

Let's focus just on the measurements of the station 240101 (UCC WGB – Lotabeg).

- Amount of days in the calendar year (01/09/2016 - 31/08/2017) for which data is collected.

Exercise 3:

Let's focus just on the measurements of the station 240561 (UCC WGB – Curraheen).

- Amount of measurements where *scheduled_time* <= *expected_arrival_time*.
- Amount of measurements where *scheduled_time* > *expected_arrival_time*.

Exercise 4:

Let's focus just on the measurements of the station 241111 (CIT Tech. Park – Lotabeg).

- Get a list with the scheduled_time values found for each day of the week.

Exercise 5:

Let's focus just on the measurements of the station 240491 (Patrick Street – Curraheen).

- Consider only the months of Semester 1 (i.e., months 09, 10 and 11) and aggregate the measurements by month and day of the week to compute the average waiting time (i.e., expected_arrival_time - query_time). Sort the entries by decreasing average waiting time.

ASSIGNMENT 2 – PART 4 (SPARK SQL STRUCTURED STREAMING) (Week 13)

Use only DataFrame-based operations.

Use the dataset folder my_dataset_complete to simulate the streaming arrival of the 12 files and their processing in separate batches.

Exercise 1:

- Number of measurements (lines).

Exercise 2:

Let's focus just on the measurements of the station 240101 (UCC WGB – Lotabeg).

- Amount of days in the calendar year (01/09/2016 - 31/08/2017) for which data is collected.

Exercise 3:

Let's focus just on the measurements of the station 240561 (UCC WGB – Curraheen).

- Amount of measurements where *scheduled_time* <= *expected_arrival_time*.

- Amount of measurements where *scheduled_time* > *expected_arrival_time*.

Exercise 4:

Let's focus just on the measurements of the station 241111 (CIT Tech. Park – Lotabeg).

- Get a list with the scheduled_time values found for each day of the week.

Exercise 5:

Let's focus just on the measurements of the station 240491 (Patrick Street – Curraheen).

- Consider only the months of Semester 1 (i.e., months 09, 10 and 11) and aggregate the measurements by month and day of the week to compute the average waiting time (i.e., expected_arrival_time - query_time). ~~Sort the entries by decreasing average waiting time.~~

Note: Cannot sort as sorting is only supported after an aggregation and Complete mode (not append mode).