Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel

Exercise Sheet 12
Big Data Analytics
WiSe 2016/2017
Due Date: 2017-02-03, 300 P

Please document your code, without sufficent documentation you won't recive any points.

1 Identifying Movie Quotes for the Wikipedia (Python/Spark) (180 P)

Goal of this task is to create suitable movie quotes for a set of Wikipedia articles. Therefore, we use the IMDb movie quotes as input and identify – according to a distance metrics – the five closest quotes for an article. The creation of an appropriate model for fitting quotes is the difficult part of this task. Consider the available information from the quotes and the Wikipedia articles. You may start with the distance metrics you have developed previously and tweak them towards this goal.
Create a Python script for Spark. Run your script on the Wikipedia articles given in `/users/bigdata/wikipedia-text-tiny-clean500` and the quotes `/users/bigdata/imdb-quotes`. Document a few created quotes with their article.

Submission:

`1-imdb-quotes.py`	Your Python program for Spark.
`1-imdb-quotes-output.txt`	The example articles and created quotes.

2 Streaming Data Model for Crime Data (Theory) (120 P)

In this task, we build a data model for streaming crime data. Consider we are managing the police forces and delegate our limited (resources) to crime places according to the severity of the crime, the location of our forces and the crime location, the number of required officers and their equipment. We receive all this information in real-time from various streaming sources. Officers in the management should be provided with the refined information to delegate efficiently. In this task, have to treat the input to have low quality, e.g., a crime reported by several people is more likely to be real and, thus, important than one only reported by a single person. Albeit the concepts and model you develop are applicable for other streaming engines, we focus on Storm in this task.
We assume to have at least the following input sources available (you may extend the list):

- Officer GPS, this source reports: time, officer ID, location (GPS coordinates)
- Officer supply, reports: time, officer ID, location, equipment (list)
- Social media source, we automatically try to filter relevant messages that are indicators for a potential crime and tread it like a reported crime. It reports: time, text, reporter, location
- Reported crime, this is a single unconfirmed crime reported by phone, it reports: time, location, type, text

Create a topology for this task and document the transformations on the individual bolts. Describe a simple heuristics to identify relevant crimes and how the topology could suggest an operator useful candidates to assign police force.
Output of the topology could be a tuple of (crime, location, severity, list of officers to send, time until destination). Note that officers could always move from a less important crime to a more severe crime.
You may want to use Bolts that access/manipulate persistent information (such as the HBase*Bolts). How could the different nodes be parallelized? What kind of grouping would be necessary?
Could we use DRPCs in this scenario for something useful (for what?)?

Submission:

`2-streaming-model.pdf`	The data model with description and answers to the questions.