sentiance
From mobile data to smart life

Sentiance NV
Korte Lozanastraat 20-26 | 2018 Antwerp | Belgium
www.sentiance.com | +32 3 369 96 96 | @sentiance

Data Engineer Recruitment Exercise

Software Development

Overview

The programs below can be written in either Java or Python. If Java is chosen, make use of the Streams API where applicable.

1.1 – Data generation

Write a program that generates a random Master Dataset as a set of files on disk.

The program should accept three arguments:

1. The first argument represents the location where the Master Dataset should be written.
2. The second argument represents a file size in megabyte.
3. The third argument has the following format *<name1>,<size1>,<name2>,<size2>* where *<name>* represents a folder name and *<size>* represent a size in megabytes (for example: locations,64,sensors,138,devices,24).

The program should create a Master Dataset based on the input parameters in the following structure:

```
<input-folder>/<name1>/file1
                      /file2
                      /file3
                      /…
               /<name2>/file1
                      /file2
                      /file3
                      /file4
                      /file5
                      /…
               /<name3>/file1
                      /file2
                      /…
```

The size of an individual file should be close the file size given in the second argument, the last file is allowed to have a significantly smaller size. The total size of a subfolder should be close to the size in the third argument.

In the example, the size of the locations folder should be close to 64 MB.

The content of an individual file should be randomly generated strings of alphanumeric characters of varying sizes, one string per line.

1.2 – Data update

Write a program that updates a given Master Dataset.

The program should accept two arguments:

1. The first argument represents the location of an existing Master Dataset.

Sentiance NV
Korte Lozanastraat 20-26 | 2018 Antwerp | Belgium
www.sentiance.com | +32 3 369 96 96 | @sentiance

2. The second argument has the following format *<name1>,<size1>,<name2>,<size2>* where *<name>* represents a folder name and *<size>* represent a size in megabytes (for example: locations,12,sensors,23,devices,10).

The program should enlarge the existing Master Dataset where each subfolder mentioned in the second argument should be enlarged by the amount after it.
In the example, if the current size of the locations-folder is 64 MB, then the program should enlarge it to 76 MB.

The constraints of the first program should remain valid:

- All files roughly the same size, except for the last one that can be smaller.
- Newly added content should be randomly generated strings of varying size, one per line.

1.3 – Data backup

Write a program that can backup a Master Dataset.

The program should accept two arguments:

1. The first argument represents the location of an existing Master Dataset.
2. The second argument represents the location of a backup folder.

This program should put a backup of the Master Dataset in the backup folder. A previous backup might still be present in the backup folder.

Software/Systems Design

2.1 – Design a data refresh system

Have a look at the Foursquare API, more specifically, the Venues search route at https://developer.foursquare.com/docs/venues/search

Design a system that can receive locations (as a [latitude, longitude, accuracy] mobile GPS fix) that represent a place that a user has visited and outputs information (that comes from the Foursquare API-call above) about that place. Include a way to cache Foursquare results. Also include a way to refresh Foursquare results as data retrieved from Foursquare can only be kept for a maximum of 30 days.

Keep in mind the definition of accuracy for a mobile GPS fix. Accuracy is represented in meter and means that there is a 68% probability that a user is within a radius of that amount of meter around the provided [latitude, longitude] coordinates. See Android docs:
https://developer.android.com/reference/android/location/Location.html#getAccuracy()

2.2 – Large scale data handling

Think back on the implementations you made in the first section.

- Would these change once the data size gets larger? If so, how?
- How would you handle a data size that becomes too large to hold on a single machine on local disks?
- Would there be libraries or tools that you could use to help out? If so, which ones and how could they help?