Short Programming Project Description

Alexandra Ciobica S3257061
Andrei Isac S3257053

1. Project description

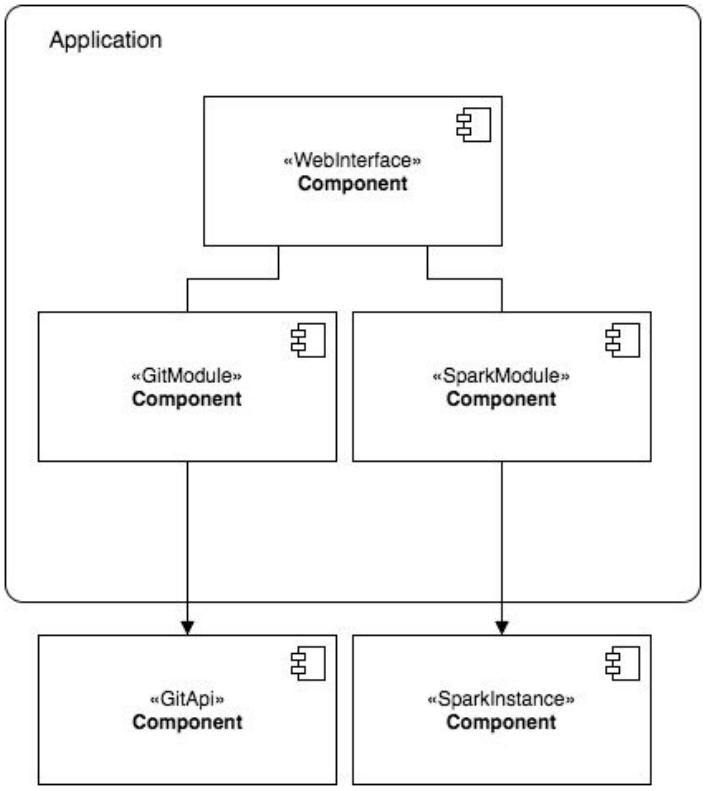
Project name: **GitHub Data Analytics**

The goal of this project is to develop an application which analyzes data from projects and attempts to find a correlation between the quality of the project and the input data such that it can be modelled. Input data is provided by a large number of GitHub repositories. Potentially interesting data includes the number of commits, lines of codes, pull request count and others. The programming language that should be used for this project is Scala. The data analytics should be performed using the Spark cluster computing framework.

Supervisors: Brian Setz, Alexander Lazovik

2. Software architecture

- Logic view



- Component description

2.1. GitHubApi

The version available now and the one we are using is GitHub API v3. All API access is over HTTPS, and accessed from the *https://api.github.com.* The data that is sent and received through the API is in the form of JSON. In order to use the API you need authentication that can be done through OAuth2 Token.
When you get a list of repositories from an organization, you get the summary representation of each repository. The summary can contain information about the `"owner"` -- like`"avatar_url"` or `"repos_url"`-- ,or `"contributors_url"`,`"comments_url"`,`"commits_url"` etc.

2.2. GitModule

For the GitModule we use json4s and restClient from the rugds framework that is explained in the next section. We communicate with the GitHub API and we receive JSON objects for our calls. We parse the JSON objects and get only the information we need. The access to the API data is done using a token that is saved in a config file and is sent in each request.

The function in this module are:

- `getOrganisationRepos(token: String, org: String)` through which we get the list of the repositories available for the organization and a big amount of information for each repo.
- `getNoLanguages(token: String, org: String, repo: String)` through which we get the programming language used in each repo. This is an important feature as the final grade depends on whether Scala is used or not in the project.
- `getContributorsStats(token: String, org: String, repo: String)` we get how many contributors the repository has and how many commits each contributor has. The number of contributors can be 2 or 3. This is important as the number of commits should be bigger for the projects with three contributors than for the ones with two.

The information that is retrieved through the API is stored in a file that is accessed by the SparkModule.

2.3. SparkInstance

Apache Spark is a cluster-computing system for data processing. Apache Spark has a scalable machine learning library named MLlib that contains common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction etc.

Our project uses native Spark cluster (standalone), meaning we run a single Spark instance, locally on our computer. We also use linear regression from MLlib for training the data set.

2.4. SparkModule

SparkModule is the one responsible for :

- taking the data from the file resulted from GitModule,
- processing the data so that it can be inputted in the linear regression algorithm,
- taking the grades of the projects in the test set from another file,
- preparing the data for the train by creating a LabeledPoint with grades that are the output and the repositories information that is the input,
- training the data,
- print the result in a file.

2.5. WebInterface

For now, the WebInterface module only reads the file. In the future, we want to make an actual web interface where you can add your token and your organisation name and it returns the list with you repositories graded.

3. Technologies used

3.1. Rugds framework

The framework suggested by our coordinator is rugds. The framework is a modular one thus a reusable one that saves the developer the time of setting the environment.

3.2. Scala

Scala is a programming language that combines object-oriented programming with functional programming. We chose scala as the rugds framework is scala-based.

4. Results

We have managed to make a linear regression for only two projects, our project and the main WaCC project, “course-2016”. The use of this repository is not relevant for the results, but because we have not yet received access to all the repositories it was used for test purposes.

These results only prove the concept and show that the MLlib Linear Regression training can lead to close results (same order of magnitude). For now the final results are stored in a file as follows:

“[{"course-2016":9.145140823278584},{"2016-Group-03-Flavius-Andrei-Isac_Elena-Alexandra-Ciobca":4.812436449541319}]”

The initial grading for the two courses was 7.0, respectively 9.0.

5. Difficulties

- Slf4j-log4j and log4j-over-slf4j libraries don’t work together, they are both included in the project and have been hard to diagnose the problem, we still don’t know why the error occurs sometimes.

- Incompatible Jackson version for the com.fasterxml.jackson.databind.JsonMappingException, this occurs when trying to run the GitAnalysisMain, so the main project app. We did not yet solve this problem, but we will move on from writing jsons in files to database use in the next iteration.
- restClient.get[String](...).onComplete cannot return anything , this forced us to write in file for every request from the API, therefore we want to switch to a database.
- For some reason, our Scala app does not write to file until the process is finished, leading to not being able to run all the required steps in one run. This is the main reason why we want to integrate a database.

6. Evolution
    a. User interface
    b. Improve modules
    c. Add database or other solution to solve writing text files problem
    d. Spark Streaming