[CS209A-22Fall] Final Project Report

11912021 Li Jiajun 12011526 Sun Xun

December 20, 2022

Abstract

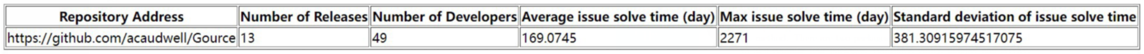
Github is a website for developers to store and manage their code. Developers could also use GitHub to track the releases, versions, issues, commits (code changes) and discussions of their projects.We take the user name as ”acaudwell” and the repositories as ”Gouce” as our research objects. For a given GitHub repository, we will crawl and analyze the data.You can click here to find more details.

1 Basic Requirements

1.1 Developers

1.1.1 How many developers have committed to this repo?

We can directly know how many developers there are in the repositories through visual icons.In the table, we can fine how many users by ”Number of developers”.



Repository Address	Number of Releases	Number of Developers	Average issue solve time (day)	Max issue solve time (day)	Standard deviation of issue solve time
https://github.com/acaudwell/Gource	13	49	169.0745	2271	381.30915974517075

Figure 1: Table

1.1.2 Which developers are the most active(who committed the most)?

Through the histogram ”quantitative relationship between development and commit”, we can see which development is the most active or contributes the most.

1.2 Issues

1.2.1 How many issues are open and how many are close?

From the number of ”switch issues” in the pie chart, we can know how many issues are open and how many are closed.Black indicates the number of closed issues and red indicates the number of open

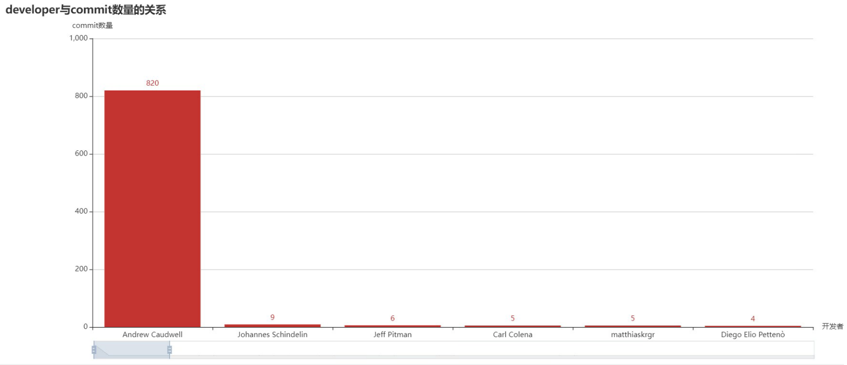
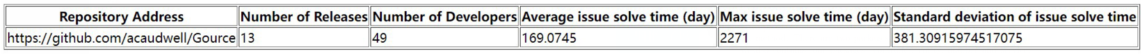


Figure 2: Develop-commit

issues.

1.2.2 What is the typical issue resolution time(the duration between issue open time and issue close time) for this repo?

The table shows typical processing of issue resolution time, including average value, maximum value and standard deviation.



Repository Address	Number of Releases	Number of Developers	Average issue solve time (day)	Max issue solve time (day)	Standard deviation of issue solve time
https://github.com/acaudwell/Gource	13	49	169.0745	2271	381.30915974517075

Figure 3: Table

1.3 Releases and Commits

1.3.1 How many releases are there in this repo?

The number of releases in this repository can be known through the table.

1.3.2 How many caommits are made between each release?

Through the histogram, we can know the number of new commits before the release. Through simple calculation, we can know how many commits each release has.

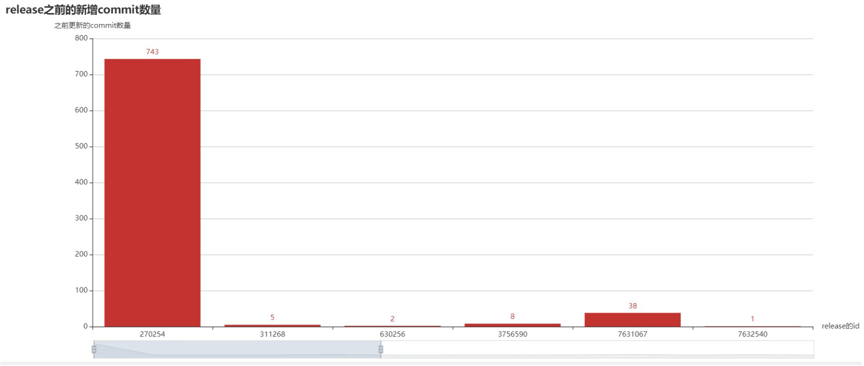


Figure 4: Caption

1.3.3 At which time(weekday,weekend,morning,etc) do developers made commits?

Through the histogram, we can see the quantitative relationship between the date and the commitment, and we can know the date on which developers committed.

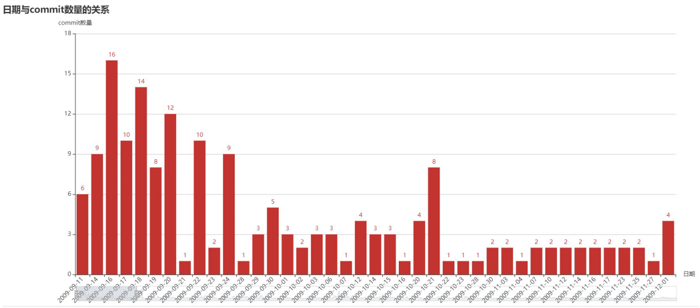


Figure 5: Caption

2 Requirements for Implementation

2.1 Data Collection and Storage

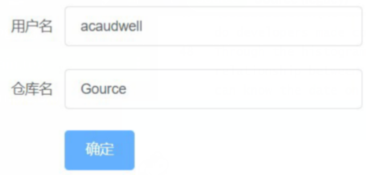
We use Java to collect data and use MySQL to store the data.In this way, data can be stored and calculated conveniently.

2.2 Web Framework

The web framework we used is Spring Boot.

2.3 Fronted

The data in the front end is visualized for user interaction. Users only need to enter Github's user name and a repositories to query the information of various data that meet the requirements. []



用户名 acaudwell

仓库名 Gource

Figure 6: User

3 Advanced Requirements

3.1 Multiple repositories

Our web application can handle multiple repositories querying GitHub You only need to enter different user names and repositories in the same web interface to query the corresponding results.

3.2 REST services

The Web server can provide at least three different RESTful API endpoint.

3.3 Issues topics

4 Documentstion

Architecture design:There are two packages in the root directory: main and test. The main methods are in main, and test is the test method. There are two packages under the main package: java and resource. The package of Java also contains four packages: controller, entity, mapper and service. Each package contains corresponding implementation methods.

To put it simply, our process can be divided into three steps. The first step is to read data from github, the second step is to import the data into the database, and the third step is to analyze and visualize the data in the database.

- Important classes:
  - ReleaseController:This. class file describes the crawling process of some data. Read web page information through Java language and store it in JSON file in specified format, then filter and read from JSON file as required.
  - Commit:The entity class of Commit is created to facilitate importing into the database.
- Important fields:
  - url

  - token

- Important methods:

  - double getSolveTimeAVG():Get the average time to solve the problem.
  - void insertReleases(List¡Release¿ releases):Import the obtained release data into the database.

Insights: When we want to compare the specified user names and repositories, or if we want to know the specific information of our own or others' repositories, we can read it through this page, which is simple and intuitive.

Visualization: we use tables, histograms and pie charts to express the relationship between data, which is more concise and intuitive. At the same time, it has high flexibility. If you need to know other information, you can adjust slightly to achieve the goal.