

Инструмент для поиска плагиата в исходном коде

Куратор	Дмитрий Иванов, 6304
Лидер	Корытов Павел, 6304
Разработчики	Артём Бутко, 8304 Дмитрий Перелыгин, 8303 Александр Алтухов, 8304 Александр Рыжиков, 8304

Постановка задачи

✓	Подключение выгрузки репозитория к REST API
✓	Завершение работы над фронтэндом
✓	Загрузка нескольких файлов на проверку
✓	Оптимизация алгоритма
✓✗	Модульное тестирование
✓	Развертывание в Docker

Методы решения

- Язык программирования - Python
 - BeautifulSoup4 для парсинга датасета StackOverflow
 - Flask - веб-сервер
- PostgreSQL + fuzzystmatch для поиска похожего кода в БД
 - Конструктор запросов к SQL - PyPika
- Vue.js + Bootstrap - фронтэнд
- Предобработка языков программирования
 - jsbeautifier
 - yapf
 - astyle

Предобработка



```
def _set_absolute_path(self, attr_path, obj):
    if len(attr_path) > 1: return self._set_absolute_path(attr_path[1:], obj[attr_path[0]])
    else: obj[attr_path[0]] = os.path.normpath(os.path.join(get_project_root(), obj[attr_path[0]]))

def __str__( self ) :
    return str(self.configs)
```



```
def _set_absolute_path(self, attr_path, obj):
    if len(attr_path) > 1:
        return self._set_absolute_path(attr_path[1:], obj[attr_path[0]])
    else:
        obj[attr_path[0]] = os.path.normpath(
            os.path.join(get_project_root(), obj[attr_path[0]]))

def __str__(self):
    return str(self.configs)
```

Демонстрация алгоритма

Для примера были взяты лабораторные по web-программированию предыдущих лет. С помощью данного инструмента была найдена работа, не отличающиеся уникальностью.

Оригинал работы был подвергнут небольшому рефакторингу, однако это никак не повлияло на результат работы.

```
5  const ctx = nextEl.getContext('2d');  
6  ctx.scale(20, 20);  
7  
8  
9  const pieces = 'TJLOSZI';  
10 var rotate = new Audio('rotate.mp3');  
11 var line = new Audio('line.mp3');  
12 var move = new Audio('move.mp3');  
13 var drop = new Audio('drop.mp3');  
14 var theme = new Audio('sound.wav');  
15  
16
```

Results from database

Similar to string:

var lineSound = new Audio('line.mp3');

Source of match:

*/Users/artembutko/Developer/GitHub/
mse_plagiarism_search/backend/uploads/
Petrov_lab1.js*

Lines successively:

83

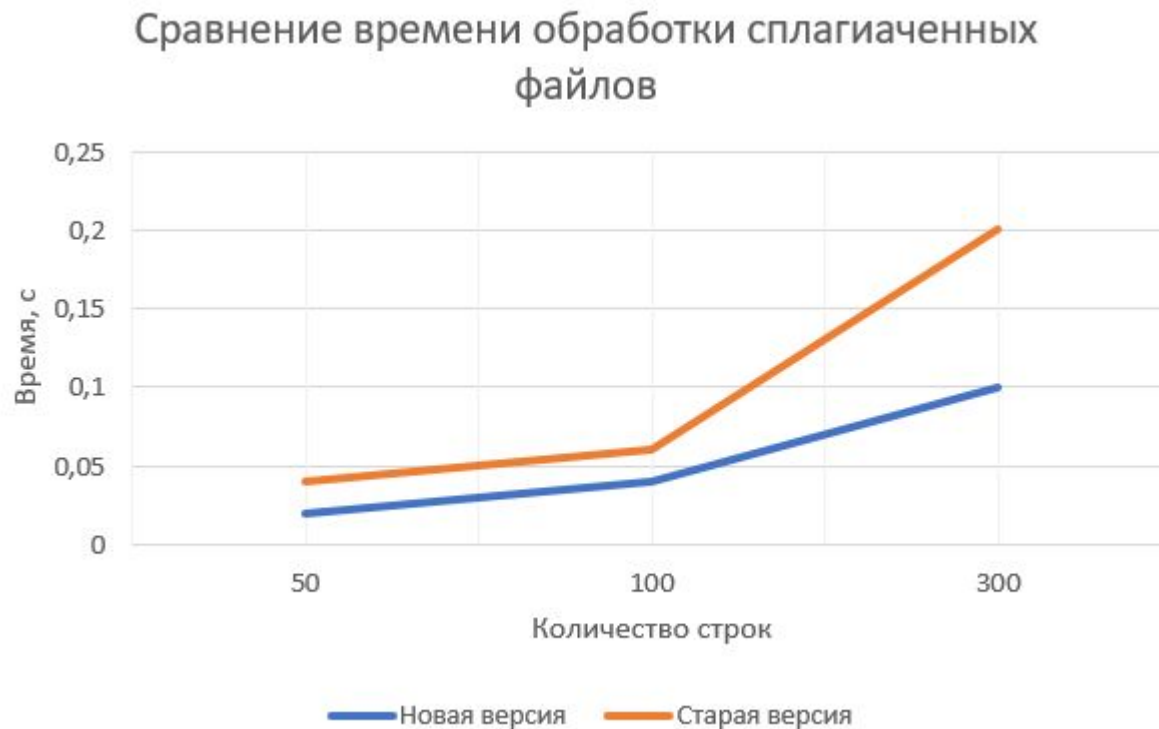
Оптимизация алгоритма

Была проведена работа по оптимизации времени работы алгоритма. Порядок анализа файлов подвергся переработке, стал формироваться динамически: если в каком-либо файле была найдена похожая на искомую строка, то при обработке следующей строки поиск будет произведен в первую очередь в этом файле.

Также при поиске плагиата будут анализироваться файлы, имеющие то же расширение, что у целевого файла.

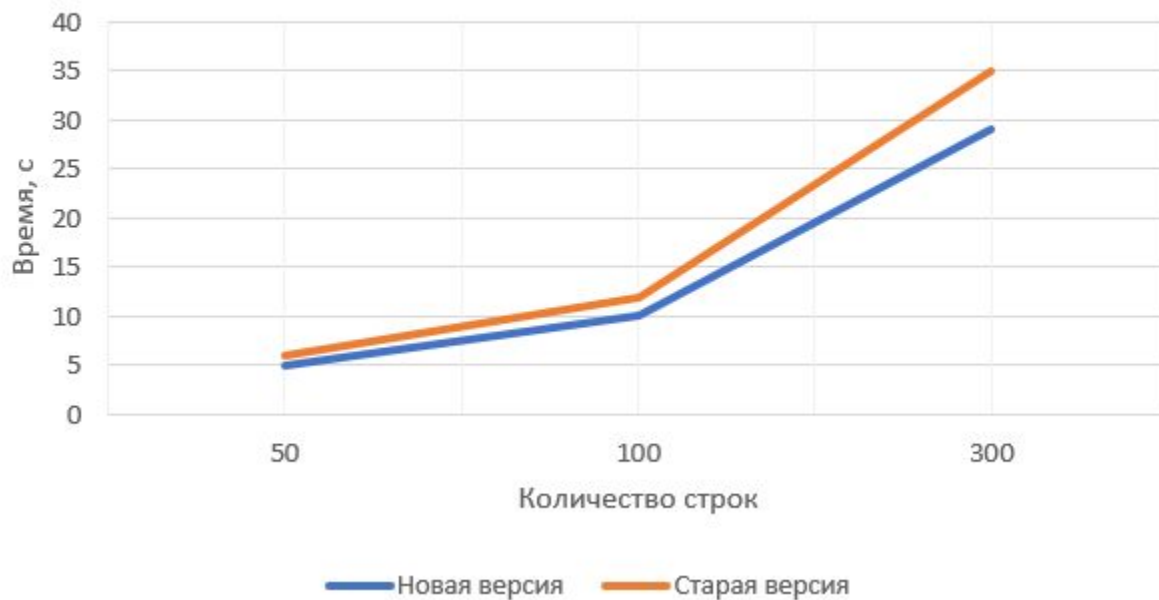
Для наглядного представления результата оптимизации сравниваются результаты работы программы с предыдущей итерации и доработанной версией на файлах, различных по уникальности и длине. В базе данных при этом находится две тысячи файлов объемом в шестьсот тысяч строк.

Файлы с минимальной уникальностью



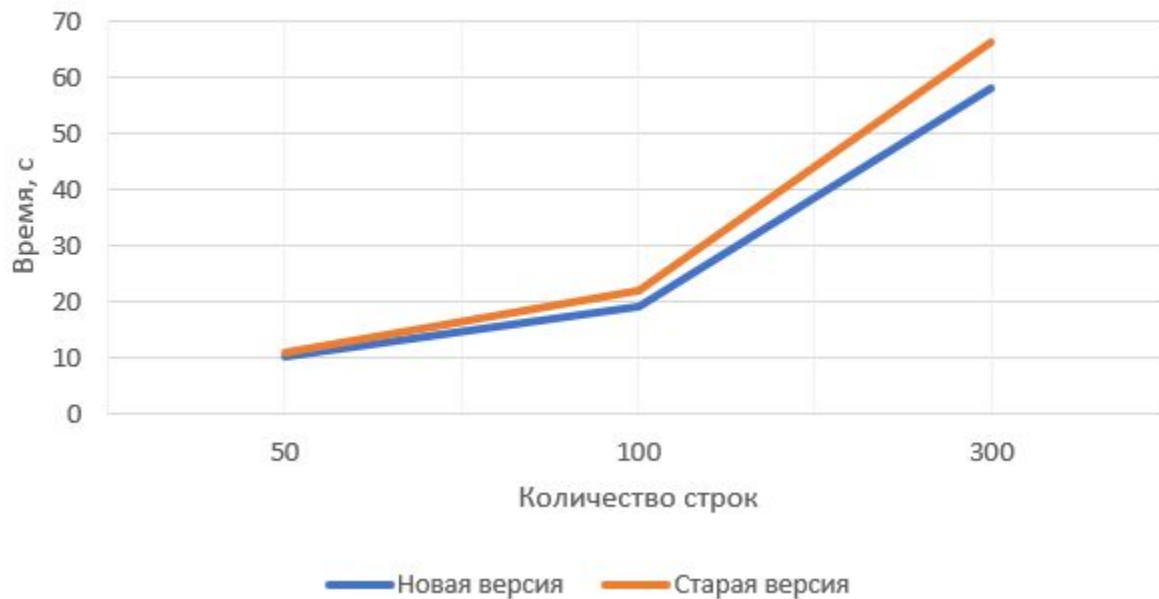
Файлы со средней уникальностью

Сравнение времени обработки файлов средней уникальности



Уникальные файлы

Сравнение времени обработки уникальных
файлов



Вывод

Ощутимый прирост производительности коснулся, как и ожидалось, неуникальных файлов, так как копирование производится в основном из небольшого числа источников. Но небольшое улучшение присутствует и в ситуации с фактическим отсутствием плагиата, так как в любом коде есть похожие синтаксические конструкции.

Модульные тесты

С использованием программы Postman было проведено тестирование. Результаты представлены на скриншоте.

The screenshot displays the Postman interface showing test results for a collection named 'anti plagiarism'. The top bar indicates 10 tests passed and 0 failed. The test suite is titled 'Iteration 1' and contains 10 individual test cases. Each test case is represented by a row with a status icon (green for pass, red for fail), a method, URL, and various metrics like status code, response time, and body size. The tests are as follows:

Method	URL	Status	Response Time	Body Size
POST	http://localhost:5000/upload	200 OK	3740 ms	157 B
Status code is 200				
POST	http://localhost:5000/loadAndCheckFile	200 OK	9047 ms	90.673 KB
Status code is 200				
Files exists				
GET	http://localhost:5000/getAllFiles	200 OK	831 ms	12.266 KB
Status code is 200				
Database is not empty				
GET	http://localhost:5000/checkFile/1	200 OK	992 ms	3.569 KB
Status code is 200				
GET	http://localhost:5000/getExtensions	200 OK	507 ms	180 B
Status code is 200				
Extensions found				
DELETE	http://localhost:5000/deleteResult/3	200 OK	515 ms	157 B
Status code is 200				
DELETE	http://localhost:5000/deleteEntry/5	200 OK	516 ms	157 B
Status code is 200				

Инструкция по запуску

1. Склонировать репозиторий

```
git clone git@github.com:moevm/mse_plagiarism_search.git  
cd mse_plagiarism_search
```

2. Запустить docker

- `docker-compose up`

3. Будут запущены:

- Фронтэнд на порту 8080
- pgAdmin 4 на порту 81

Демонстрация работы программы

<http://sqrtminusone.ddns.net:8080/>

