Московский авиационный институт
(национальный исследовательский университет)

Выпускная квалификационная работа на тему:

«Поиск алгоритмов»

Студент: Шичко Алексей Игоревич
Группа: М8о-407б-18
Научный руководитель: Морозов Александр Юрьевич

Москва, 2022

Введение.

Алгоритмы в наше время решают судьбу многих компаний. 1 миллисекунда улучшения отклика сервиса может значительно повысить вовлеченность пользователей или клиентов. От вовлеченности в свою очередь, зависит прибыль и отношение самих пользователей.

Представив сервис, в котором программисты не знают и не умеют пользоваться алгоритмами, мы фактически представим себе большие накладные расходы на амортизацию вычислительных мощностей (т.к. их потребуется гораздо больше, чем на самом деле надо).

Проблема

- Требуется возможность поиска по типу алгоритма;
- Необходима возможность сохранять результаты выдачи сервиса;
- Пользователь должен иметь возможность выбрать короткий ответ из списка предложенных.

Актуальность проблемы

- Количество вещей, которое должен запоминать средний программист, с каждым днем растет;
- Программист не всегда знает алгоритм конкретно по имени, но знает что алгоритм должен делать (работать со строками, работать с числами)
- Существующие системы сложны в использовании и не предоставляют необходимого функционала.

Цель

Разработать приложение для поиска алгоритмов по типу алгоритма (например: на графах, на строках) или по имени алгоритма. Должна иметься возможность отслеживать состояние системы и каждой компоненты.

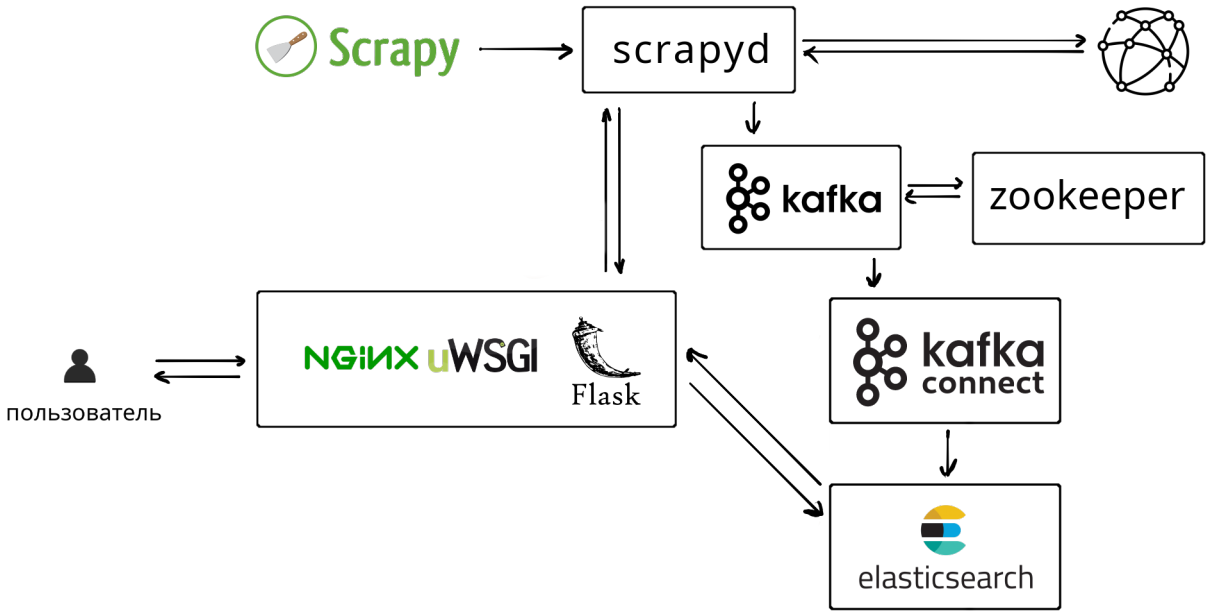
Задачи

- Проанализировать предметную область.
- Проанализиировать актуальность поставленной задачи.
- Проанализировать существующие на рынке решения.
- Проанализировать существующие инструменты для написания решения.
- Составить макет решения.
- Протестировать его на подвыборке данных.
- Разработать итоговое решение.

Постановка задачи

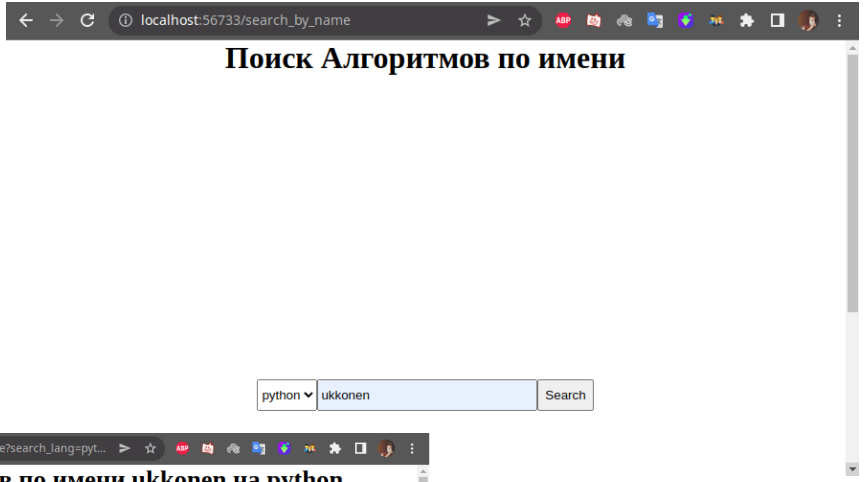
Найти оптимальные с точки зрения масштабирования и эксплуатации компоненты системы. Настроить их коммуникацию с дальнейшей виртуализацией. На основе построенных сервисов написать серви для поиски алгоритмов на интернет ресурсах без использования API последних.

Результаты: архитектура проекта



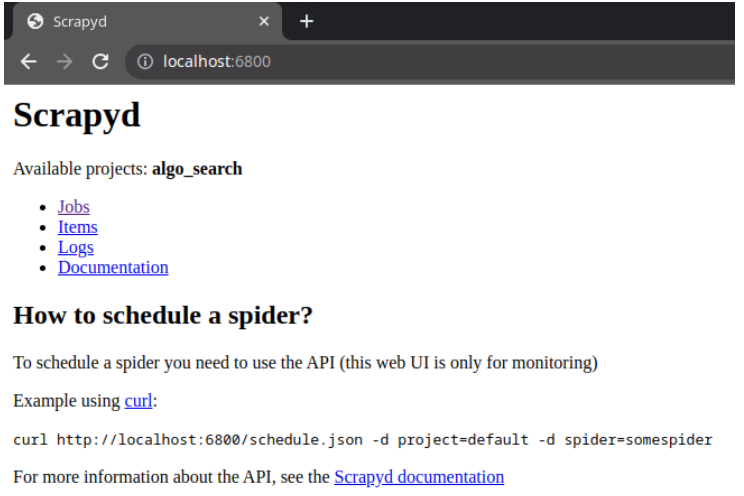
Результаты: компоненты системы: Flask

- Flask: позволяет писать сервера на python, использует jinja как систему шаблонов html. В общей структуре выполняет роль интерфейса пользователя к системе.



Результаты: компоненты системы: Scrapy + Scrapyd

- Scrapy: позволяет в формате “веб пауков” описать что нужно достать из страницы и куда положить. Остальное ложится на плечи самой библиотеки;
- Scrapyd: сервис для разветки “пауков”.



Результаты: компоненты системы: Kafka + Zookeeper

- Kafka: распределенный менеджер сообщений
- Zookeeper: центр синхронизации распределенной системы, необходим для kafka.

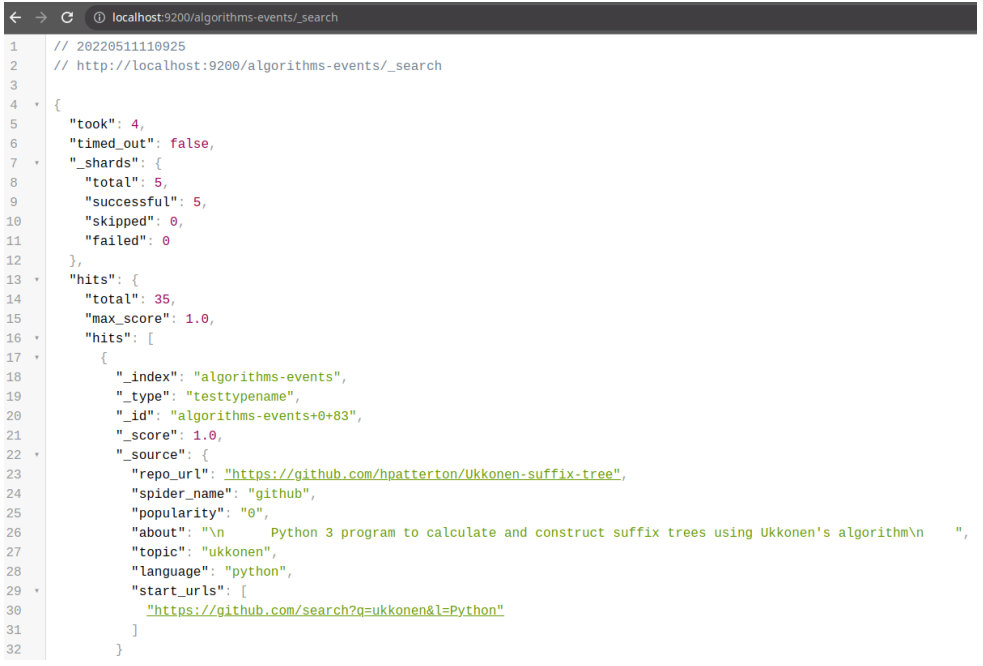
```
~/files/vus/diplom/solution(master*) » docker exec -it kafka kafka-topics --describe --topic algorithms-events --bootstrap-server localhost:9092
Topic: algorithms-events        TopicId: tuM0Ai20R_y2jmSuxHTS0Q PartitionCount: 1       ReplicationFactor: 1    Configs:
        Topic: algorithms-events        Partition: 0    Leader: 1       Replicas: 1     Isr: 1
```

Результаты: компоненты системы: Kafka Connect

- Сервис для интеграций с базами данных. Есть возможность написать свою интеграцию.

Результаты: компоненты системы: Elastic Search

- Распределенная документоориентированная СУБД.



Заключение

- Инфраструктура написанна полностью в микросервисной архитектуре. Работу каждой отдельной компоненты можно проследить войдя в необходимый сервис либо с помощью команды верхнего уровня системы виртуализации. Результаты инфраструктуры легко достижимы либо из командной строки, либо путем перехода в браузере.
- С помощью написанных микросервисов программисты смогут облегчить себе жизнь и ускорят цикл написания ПО.