

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА И АНАЛИТИКИ ДЛЯ
ЧАТ-БОТА**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Машкиной Дианы Александровны

Научный руководитель
старший преподаватель

М. И. Сафрончик

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

Саратов 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Решения для управления процессами, мониторинга и аналитики данных	6
1.1 Вопросы обработки данных в современном мире	6
1.2 Apache Airflow	6
1.2.1 Общие сведения	6
1.2.2 Архитектура платформы	6
1.2.3 Основные понятия	6
1.2.4 Планировщик задач	7
1.2.5 Исполнитель задач	7
1.2.6 Экземпляры задач	7
1.3 Альтернативные платформы для управления процессами	8
1.3.1 Luigi	8
1.3.2 Google Cloud Platform	8
1.4 Визуализация в системах аналитики и мониторинга	8
2 Реализация системы мониторинга и аналитики нераспознанных со- общений для бота социальной сети ВКонтакте «Скриба»	9
2.1 Установка и настройка Apache Airflow	9
2.2 Настройка группы ВКонтакте для администрирования бота	9
2.3 Разработка сервиса мониторинга нераспознанных сообщений	9
2.3.1 Разработка оператора для отправки сообщений ВКонтакте	9
2.3.2 Разработка DAG для мониторинга нераспознанных со- общений	9
2.3.3 Использование сервиса мониторинга нераспознанных со- общений	10
2.4 Разработка сервиса для рассылки статистики соотношения нерас- познанных сообщений относительно месяцев	10
2.4.1 Разработка DAG для рассылки статистики соотношения нераспознанных сообщений относительно месяцев	10
2.4.2 Использование сервиса рассылки статистики соотноше- ния нераспознанных сообщений относительно месяцев	10
2.5 Разработка сервиса для мониторинга статуса SSL-сертификата	10
2.5.1 Алгоритм работы сервиса	10
2.5.2 Разработка оператора для работы с протоколом SSH	11

2.5.3	Разработка DAG для мониторинга статуса SSL-сертификата	11
2.5.4	Использование сервиса мониторинга статуса SSL-сертификата	11
2.6	Разработка сервиса для мониторинга состояния сервера.....	11
2.6.1	Разработка DAG для мониторинга состояния сервера.....	11
2.6.2	Использование сервиса мониторинга состояния сервера ...	11
3	Разработка системы визуализации статистики нераспознанных сообщений	12
3.1	Разработка сайта с визуализацией статистики нераспознанных сообщений	12
3.2	Использование системы визуализации статистики нераспознанных сообщений	12
ЗАКЛЮЧЕНИЕ		13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ		14

ВВЕДЕНИЕ

Своевременная помощь пользователям приложения позволяет сформировать позитивный пользовательский опыт и улучшить репутацию сервиса. Поэтому кроме добавления новых функций важно разработать системы мониторинга и аналитики, позволяющие эффективно реагировать на запросы пользователей, а также выявлять потенциал для улучшения сервиса.

Поэтому многие крупные компании не только внедряют готовые решения аналитики, мониторинга и автоматизации процессов в свои проекты, но и разрабатывают собственные сервисы. При большом объёме данных целесообразно использование облачных решений, однако для лучшего контроля среды и разработки может быть принято решение построения систем с нуля.

Новизна исследования состоит в том, что многие процессы могут быть автоматизированы, ускорены и отлажены с помощью систем управления процессами, систем аналитики и систем мониторинга, что может положительно сказаться на сервисе, предоставляемом компаниями и отдельными лицами. Для успешного внедрения и благоразумного использования систем важно всесторонне изучить их и с теоретической, и с практической точки зрения.

В практической части исследования проводится разработка системы мониторинга и аналитики для бота социальной сети ВКонтакте «Скриба», который позволяет узнавать расписание занятий студентов и преподавателей СГУ. Взаимодействие с сервисом осуществляется с помощью текстовых сообщений. Приложение сопоставляет команды с заданными шаблонами и подбирает нужный ответ. При наборе текста на клавиатуре возможны опечатки, и бот не распознает сообщение с ошибкой. Также пользователь может решить пообщаться с ботом в свободной форме или задать вопрос напрямую администратору.

Было принято решение разработать систему автоматического оповещения администраторов о нераспознанных входящих сообщениях, а также сервис для визуализации статистики.

Актуальность внедрения обусловлена возрастанием количества пользователей сервиса и сообщений, при котором ручной мониторинг становится затруднительным.

Периодическая автоматическая доставка сообщений позволит администраторам гораздо быстрее реагировать на сообщения пользователей и при корректной работе гарантирует, что администратор увидит сообщение и при-

мет решение о реагировании на ситуацию.

Кроме непосредственно отслеживания нераспознанных сообщений будет администраторам будет удобно получать оповещения о соотношении количества нераспознанных сообщений в текущем месяце и в предыдущем.

В работе сервера возможны сбои, поэтому важно отслеживать состояние инфраструктуры, где развернут проект. Было принято решение разработать системы мониторинга состояния некоторых элементов сервера.

Платформа сообществ ВКонтакте предоставляет некоторую статистику использования сообщества, однако не может отражать специфические для пользователей конкретного бота характеристики, например, факультет и номер группы. Поэтому сторонний разработанный сервис визуализации статистики позволит получить дополнительную актуальную информацию о пользователях.

Соответственно, целью бакалаврской работы является разработка системы аналитики и мониторинга для бота ВКонтакте «Скриба».

Поставлены следующие задачи:

- исследовать систему управления процессами Airflow и её аналоги;
- подготовить к использованию систему управления процессами Airflow;
- разработать программу рассылки нераспознанных сообщений бота администраторам;
- разработать программу рассылки статистики соотношения нераспознанных сообщений относительно месяцев на базе Airflow;
- разработать программу мониторинга состояния SSL-сертификата сервера, где расположен бот;
- разработать программу мониторинга статуса сервера, где расположен бот;
- разработать сервис визуализации статистики бота;
- оценить влияние разработанных средств на опыт использования бота.

В данной работе для реализации практической части используются следующие программные средства:

- язык программирования Python 3;
- платформа управления процессами Apache Airflow;
- библиотека VK API для языка Python 3;
- фреймворк для создания визуализаций Dash.

1 Решения для управления процессами, мониторинга и аналитики данных

1.1 Вопросы обработки данных в современном мире

В современном мире приложения обрабатывают всё больше данных, что приводит к усложнению программ и повышению требований к ним. Выделяются три основных свойства приложений, интенсивно оперирующих данными [1]: надёжность, масштабируемость, удобство.

Существует три подхода к построению систем работы с данными, основанные на регулярности поступления и обработки данных [1]: онлайн-сервисы, автономные системы пакетной обработки и системы поточной обработки.

1.2 Apache Airflow

1.2.1 Общие сведения

Apache Airflow — платформа для создания, планирования и мониторинга рабочих процессов. Максим Бушечин начал разрабатывать её в октябре 2014 года, а в марте 2016 года проект присоединился к программе-инкубатору от Apache Software Foundation [2]. Apache Airflow является приложением с открытым исходным кодом, используется лицензия Apache-2.0 License. [3].

В качестве языка программирования используется язык Python. Airflow предоставляет большое количество готовых операторов, которые интегрированы с разными сервисами [4].

1.2.2 Архитектура платформы

Если рассматривать архитектуру платформы Airflow на глобальном уровне, можно выделить следующие компоненты: планировщик задач, исполнитель задач, веб-сервер, директория с файлами DAG и база данных.

1.2.3 Основные понятия

Одним из центральных понятий платформы Apache Airflow является «DAG», что расшифровывается как «directed acyclic graph» и переводится как «ориентированный ациклический граф». Он описывает набор задач процесса, их организацию и зависимости.

Иногда может понадобиться передавать данные между задачами. В Airflow для этих целей существует два механизма. Во-первых, можно загружать файлы, особенно большого размера, на диск своего сервера или в облачное храни-

лище. Во-вторых, данные небольшого размера удобно передавать с помощью системы XCom.

1.2.4 Планировщик задач

Планировщик задач Airflow следит за всеми задачами и графами процессов, запускает новые шаги, если предыдущие завершились. Планировщик запускает побочный процесс, который производит синхронизацию всех графов в указанной в настройках директории для хранения DAG. [5].

1.2.5 Исполнитель задач

Исполнители задач — это механизмы, которые отвечают непосредственно за выполнение задач [6].

Локальными исполнителями задач являются следующие исполнители: `DebugExecutor` используется в качестве средства отладки [7], `LocalExecutor` обрабатывает задачи в соответствии с выбранной стратегией параллелизма [8], `SequentialExecutor` обрабатывает задачи последовательно [9].

Удалёнными исполнителями задач являются следующие исполнители: `CeleryExecutor` на основе Celery [10], `KubernetesExecutor` на основе API Kubernetes [11], `CeleryKubernetesExecutor` совмещает в себе `CeleryExecutor` и `KubernetesExecutor` [12], `DaskExecutor` позволяет выполнять задачи на кластере Dask [13].

1.2.6 Экземпляры задач

Задачи, входящие в процесс, также имеют экземпляры собственных запусков, как и DAG. Экземпляр задачи — это конкретный запуск задачи для конкретного DAG за конкретную дату запуска. Одной из характеристик экземпляров задач является состояние, которое отражает определённый этап жизненного цикла задач Airflow [14].

Airflow предоставляет функционал для соблюдения соглашения о максимально допустимом времени, которое может быть отведено для выполнения шага. Также Airflow отслеживает и по возможности удаляет зависшие задачи.

1.3 Альтернативные платформы для управления процессами

1.3.1 Luigi

Одной из альтернатив Apache Airflow является Luigi — платформа с открытым исходным кодом, которая управляет рабочими процессами с помощью визуализации через DAG-конвейеры. По статистике веб-сервиса для хостинга IT-проектов GitHub количество звёзд (отметок заинтересованности пользователей сервиса в репозитории) Luigi среди платформ управления процессами уступает только Airflow [15, 16].

Luigi изначально предназначался для управления сложными процессами с использованием Hadoop [17]. Одним из основных отличий Luigi от Airflow можно назвать отсутствие планировщика задач как отдельного компонента, что ухудшает масштабируемость.

1.3.2 Google Cloud Platform

Одной из самых популярных облачных платформ является Google Cloud Platform, которая содержит обширный набор сервисов.

Google Cloud Composer предоставляется Google в качестве управляемого сервиса для оркестрации процессов. Основан на платформе Apache Airflow [18]. Размещение в облаке позволяет сконцентрироваться на разработке, внедрении и отслеживании процессов, сокращая трудовые затраты на управление ресурсами.

Однако построение процессов может потребоваться людям, которые не имеют достаточных навыков программирования. В качестве альтернативы сервису Composer может рассматриваться Data Fusion, разработка процессов в котором ведётся с помощью визуального пользовательского интерфейса [19].

1.4 Визуализация в системах аналитики и мониторинга

Визуализация в системах аналитики и мониторинга позволяет представлять результат более наглядным способом и помогает обнаружить закономерности. Одна из классификаций [20] делит данные на следующие типы:

- информация о категории, метки без числовых значений;
- упорядоченная информация, схожая с информацией о категории, однако должна воспроизводиться в установленном порядке;
- количественная информация, которая представляет собой исчисляемые данные.

2 Реализация системы мониторинга и аналитики нераспознанных сообщений для бота социальной сети ВКонтакте «Скриба»

2.1 Установка и настройка Apache Airflow

Устанавливается Python версии 3.6, менеджер пакетов для него [21], компоненты Airflow и сопутствующие библиотеки. Для хранения метаданных Airflow была выбрана СУБД PostgreSQL. Так как в репозитории CentOS 7 самой новой версией является 9.2, а минимальная совместимая версия — 9.6 [22], устанавливается СУБД из стороннего репозитория. Осуществляется конфигурация платформы, после чего она запускается.

2.2 Настройка группы ВКонтакте для администрирования бота

Для отправки оповещений администраторам проекта была создана частная группа ВКонтакте «Scriba Admin», для которой была настроена функция отправлять сообщения с помощью бота.

2.3 Разработка сервиса мониторинга нераспознанных сообщений

2.3.1 Разработка оператора для отправки сообщений ВКонтакте

У Airflow отсутствует встроенный оператор отправки сообщений в ВКонтакте, поэтому был разработан оператор, который позволяет отправлять сообщения в ВКонтакте.

К методу инициализации оператора применяется декоратор получения параметров по умолчанию [23]. Airflow использует Jinja в качестве средства для работы с шаблонами [24], и оператору подключается возможность обрабатывать поле с содержимым сообщений. Оператор может принимать на вход как одно сообщение, так и несколько. Внедрение данного оператора позволило ускорить разработку.

2.3.2 Разработка DAG для мониторинга нераспознанных сообщений

Нераспознанные сообщения хранятся в базе данных бота. Согласно классификации видов данных [25] эти данные являются одновременно структурированными и данными на естественном языке. Для запроса данных из базы данных создаётся новое подключение в модуле Airflow «Connections» [26].

Может быть два сценария: список сообщений пуст, и тогда ничего не происходит, или в списке присутствуют сообщения, которые необходимо обработать. Информация о нераспознанном сообщении и пользователе-отправителе

получается из SQL-запроса к базе данных бота. Если список нераспознанных сообщений получился не пустым, то он публикуется через XCom.

На шаге форматирования информации формируются оповещения. Некоторые данные получают не из базы бота, а из переменных Airflow [27]. Для отправки сообщений ВКонтакте используется `VkSenderOperator`.

2.3.3 Использование сервиса мониторинга нераспознанных сообщений

Введение в эксплуатацию новой системы мониторинга позволило администратору своевременно получать оповещения о новых нераспознанных сообщениях в удобном для анализа виде.

2.4 Разработка сервиса для рассылки статистики соотношения нераспознанных сообщений относительно месяцев

2.4.1 Разработка DAG для рассылки статистики соотношения нераспознанных сообщений относительно месяцев

С внедрением системы отслеживания нераспознанных сообщений бота появляется потребность оценить прогресс относительно количества нераспознанных сообщений. Для этого была разработана программа, отслеживающая соотношение нераспознанных сообщений текущего и предыдущего месяца.

Выполняются SQL-запросы для получения количества нераспознанных сообщений за текущий и предыдущий месяц, вычисляется разница между этими результатами. Сообщение формируется и отправляется администраторам.

2.4.2 Использование сервиса рассылки статистики соотношения нераспознанных сообщений относительно месяцев

Внедрение этого сервиса позволит администраторам отслеживать изменение количества нераспознанных сообщений с течением времени.

2.5 Разработка сервиса для мониторинга статуса SSL-сертификата

2.5.1 Алгоритм работы сервиса

Для корректного взаимодействия с API ВКонтакте необходимо, чтобы сервер имел SSL-сертификат. Протокол SSL является популярным протоколом для шифрования и дешифрования данных для протоколов прикладного уровня стека TCP/IP [28].

Так как человек может забыть вовремя обновить сертификат, было принято решение разработать программу для мониторинга статуса SSL-сертификата и его обновления. Был определён алгоритм действий.

2.5.2 Разработка оператора для работы с протоколом SSH

Для отправки запросов на сервер бота используется подключение по SSH. В Airflow присутствует функционал для работы с SSH, однако было принято решение разработать более легковесный оператор. Он подключается к удалённому серверу, выполняет требуемый скрипт и публикует результат его выполнения через XCom.

2.5.3 Разработка DAG для мониторинга статуса SSL-сертификата

Создаётся DAG для мониторинга статуса SSL-сертификата. Проверяется, не истёк ли срок действия сертификата, при необходимости предпринимается попытка обновления. Проверяется, было ли обновление корректным, отправляется сообщение о результатах проверки.

2.5.4 Использование сервиса мониторинга статуса SSL-сертификата

Введение в эксплуатацию новой системы мониторинга статуса SSL-сертификата уменьшает вероятность внезапной потери доступа к функционалу ВКонтакте для бота, связанной с устаревшим сертификатом.

2.6 Разработка сервиса для мониторинга состояния сервера

2.6.1 Разработка DAG для мониторинга состояния сервера

Сбои в работе оборудования могут привести к остановке того или иного процесса инфраструктуры. Было принято решение разработать программу для мониторинга состояния сервера. Был создан DAG, проверяющий запущенные процессы и объём доступной памяти на сервере.

2.6.2 Использование сервиса мониторинга состояния сервера

Автоматизированная проверка состояния сервера позволяет администраторам быстрее реагировать на сбои в работе инфраструктуры бота.

3 Разработка системы визуализации статистики нераспознанных сообщений

3.1 Разработка сайта с визуализацией статистики нераспознанных сообщений

Данные, представленные в виде графиков, обычно являются более наглядным источником информации, чем описанные в виде таблиц. В связи с этим было принято решение разработать сервис для визуализации статистики по нераспознанным сообщениям бота на основе фреймворка Dash. Он позволяет создавать веб-приложения для визуализации с использованием только языка Python [29].

Необходимые данные получаются из базы данных бота и преобразуются с помощью библиотеки Pandas. На их основе был сформирован ряд графиков: график количества нераспознанных сообщений в день, график количества нераспознанных сообщений по подразделениям университета, график количества нераспознанных сообщений по месяцам и другие. Полный список графиков описан в бакалаврской работе.

3.2 Использование системы визуализации статистики нераспознанных сообщений

После запуска приложения можно посетить адрес, указанный в конфигурационном файле, и увидеть сайт с графиками.

Было проведено исследование большого количества нераспознанных сообщений за определённую дату.

В статистике распределения нераспознанных сообщений по подразделениям лидируют факультет компьютерных наук и информационных технологий и колледж радиоэлектроники, потому что студенты этих подразделений помогали тестировать новый функционал. Однако стоит обратить внимание и на количество пользователей, которые зарегистрировались в боте в каждом из подразделений. Бот имеет много пользователей, обучающихся в этих подразделениях, значит, отношение количества сообщений к пользователям будет далеко не самым большим.

Графики статистики нераспознанных сообщений относительно часов и времени суток показывают степень активности пользователей в течение дня. Логично отметить, что наименьшая активность наблюдается ночью.

ЗАКЛЮЧЕНИЕ

Целью бакалаврской работы являлась разработка системы мониторинга и аналитики для бота ВКонтакте «Скриба». Создание и внедрение данной системы позволило многократно ускорить работу с нераспознанными сообщениями пользователей бота, а также найти места для улучшения интерфейса.

Были рассмотрены теоретические аспекты системы управления процессами Apache Airflow и её аналогов.

Были выполнены следующие задачи:

- исследована система управления процессами Airflow и её аналоги;
- установлена система оркестрации Airflow на VPS-сервере под управлением CentOS 7;
- разработана программа рассылки нераспознанных сообщений бота администраторам на базе Airflow;
- разработана программа рассылки статистики соотношения нераспознанных сообщений относительно месяцев на базе Airflow;
- разработана программа мониторинга состояния SSL-сертификата сервера, где расположен бот, на базе Airflow;
- разработана программа мониторинга статуса сервера, где расположен бот, на базе Airflow;
- разработан сервис визуализации статистики бота с использованием фреймворка Dash;
- оценено влияние разработанных средств на опыт использования бота.

В дальнейшем на основе набора данных о нераспознанных сообщениях можно создать классификацию нераспознанных сообщений с помощью средств машинного обучения и внедрить её в систему рассылки сообщений администраторам. На данном этапе модель будет работать слишком нестабильно, потому что набор данных, на которых она может обучиться, слишком мал.

Сервис для визуализации статистики бота можно расширять с помощью новых графиков, а также с помощью добавления фильтрации данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. — СПб.: Питер, 2018. — С. 640.
- 2 Project [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/project> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 3 apache/airflow: Apache Airflow [Электронный ресурс]. — URL: <https://github.com/apache/airflow/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 4 Apache Airflow [Электронный ресурс]. — URL: <https://airflow.apache.org/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 5 Scheduler [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/concepts/scheduler.html> (Дата обращения 06.05.2021). Загл. с экр. Яз. англ.
- 6 Executor [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/index.html> (Дата обращения 06.05.2021). Загл. с экр. Яз. англ.
- 7 Debug Executor [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/debug.html> (Дата обращения 06.05.2021). Загл. с экр. Яз. англ.
- 8 Local Executor [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/local.html> (Дата обращения 06.05.2021). Загл. с экр. Яз. англ.
- 9 Sequential Executor [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/sequential.html> (Дата обращения 06.05.2021). Загл. с экр. Яз. англ.
- 10 Celery Executor [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/celery.html> (Дата обращения 07.05.2021). Загл. с экр. Яз. англ.

- 11 Kubernetes Executor [Электронный ресурс].— URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/kubernetes.html> (Дата обращения 07.05.2021). Загл. с экр. Яз. англ.
- 12 CeleryKubernetes Executor [Электронный ресурс].— URL: https://airflow.apache.org/docs/apache-airflow/stable/executor/celery_kubernetes.html (Дата обращения 07.05.2021). Загл. с экр. Яз. англ.
- 13 Dask Executor [Электронный ресурс].— URL: <https://airflow.apache.org/docs/apache-airflow/stable/executor/dask.html> (Дата обращения 07.05.2021). Загл. с экр. Яз. англ.
- 14 Tasks [Электронный ресурс].— URL: <https://airflow.apache.org/docs/apache-airflow/stable/concepts/tasks.html> (Дата обращения 08.05.2021). Загл. с экр. Яз. англ.
- 15 Star history [Электронный ресурс].— URL: <https://star-history.t9t.io/#spotify/luigi&argoproj/argo&kubeflow/kubeflow&mlflow/mlflow> (Дата обращения 10.05.2021). Загл. с экр. Яз. англ.
- 16 Airflow vs. Luigi vs. Argo vs. MLFlow vs. KubeFlow [Электронный ресурс].— URL: <https://www.datarevenue.com/en-blog/airflow-vs-luigi-vs-argo-vs-mlflow-vs-kubeflow> (Дата обращения 10.05.2021). Загл. с экр. Яз. англ.
- 17 Airflow vs. Luigi [Электронный ресурс].— URL: <https://www.astronomer.io/guides/airflow-vs-luigi> (Дата обращения 10.05.2021). Загл. с экр. Яз. англ.
- 18 Cloud Composer [Электронный ресурс].— URL: <https://cloud.google.com/composer> (Дата обращения 11.05.2021). Загл. с экр. Яз. англ.
- 19 Cloud Data Fusion [Электронный ресурс].— URL: <https://cloud.google.com/data-fusion?hl=ru> (Дата обращения 11.05.2021). Загл. с экр. Яз. англ.
- 20 *Wexler, S. The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios / S. Wexler, J. Shaffer. — USA: John Wiley & Sons, Inc., 2017. — С. 451.*

- 21 Installation [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/installation.html> (Дата обращения 12.05.2021). Загл. с экр. Яз. англ.
- 22 Set up a Database Backend [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/howto/set-up-database.html> (Дата обращения 12.05.2021). Загл. с экр. Яз. англ.
- 23 Creating a custom Operator [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/howto/custom-operator.html> (Дата обращения 13.05.2021). Загл. с экр. Яз. англ.
- 24 DAG Writing Best Practices in Apache Airflow [Электронный ресурс]. — URL: <https://www.astronomer.io/guides/templating> (Дата обращения 14.05.2021). Загл. с экр. Яз. англ.
- 25 Дэви, С. Основы Data Science и Big Data. Python и наука о данных / С. Дэви, М. Арно, А. Мохамед. — СПб.: Питер, 2017. — С. 336.
- 26 Managing Connections [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/howto/connection.html> (Дата обращения 13.05.2021). Загл. с экр. Яз. англ.
- 27 Managing Variables [Электронный ресурс]. — URL: <https://airflow.apache.org/docs/apache-airflow/stable/howto/variable.html> (Дата обращения 14.05.2021). Загл. с экр. Яз. англ.
- 28 Олифер, В. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. / В. Олифер, Н. Олифер. — СПб.: Питер, 2016. — С. 992.
- 29 Introduction to Dash [Электронный ресурс]. — URL: <https://dash.plotly.com/introduction> (Дата обращения 17.05.2021). Загл. с экр. Яз. англ.