|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по рубежному контролю №** | 1 |

**Название:** Разработка современных витрин данных

**Дисциплина:** Искусство аналитической работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | Д.В. Березкин |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2024

**Содержание**

[Введение 3](#_Toc3)

[1. Проблематика построения и понятие витрин данных 4](#_Toc4)

[2. Версионность витрин данных 7](#_Toc5)

[3. Apache Airflow 12](#_Toc6)

[Заключение 17](#_Toc7)

[Список использованных источников 18](#_Toc8)

# Введение

В настоящее время в контексте словосочетания “большие данные” наиболее популярны темы, связанные с машинным обучением, нейронными сетями и искуственным интеллектом.

Действительно, искать скрытые закономерности, визуализировать заранее собранные датасеты, применять к ним методы машинного обучения и математической статистики может быть довольно занимательно (и, конечно, же полезно для бизнеса).

Однако, зачастую, даже люди, профессионально занимающиеся информационными технологиями, не осознают всей технической сложности, скрывающейся за «расчетом» обычных витрин данных.

В данном реферате освещается основная проблематика рассчета витрин данных в современных реалиях, а также использование ПО Apache Airflow для оркестрации расчета витрин данных и ETL-процессов.

# 1. Apache airflow

В предыдущем реферате мы уже рассмотрели базовые функции и задачи Apache Airflow. Очевидно, основная идея Airflow довольно проста, однако реализация данного ПО имеет крайне высокое качество и множество небольших полезных функций, которые позволили ему обрести мировую популярность. Давайте рассмотрим детали работы и функции Apache Airflow подробнее [книга по Airflow].

***Планирование и выполнение конвейеров.*** После того как вы определили структуру вашего конвейера (конвейеров) в виде DAG, Airflow позволяет вам определить параметр schedule\_interval для каждого графа, который точно решает, когда ваш конвейер будет запущен Airflow. Таким образом, вы можете дать указание Airflow выполнять ваш граф каждый час, ежедневно, каждую неделю и т. д. Или даже использовать более сложные интервалы, основанные на выражениях, подобных Cron.

Чтобы увидеть, как Airflow выполняет графы, кратко рассмотрим весь процесс, связанный с разработкой и запуском DAG. На высоком уровне Airflow состоит из трех основных компонентов (рисунок 1):

* планировщик Airflow – анализирует DAG, проверяет параметр schedule\_interval и (если все в порядке) начинает планировать задачи DAG для выполнения, передавая их воркерам Airflow;
* воркеры1 (workers) Airflow – выбирают задачи, которые запланированы для выполнения, и выполняют их. Таким образом, они несут ответственность за фактическое «выполнение работы»;
* веб-сервер Airflow – визуализирует DAG, анализируемые планировщиком, и предоставляет пользователям основной интерфейс для отслеживания выполнения графов и их результатов.

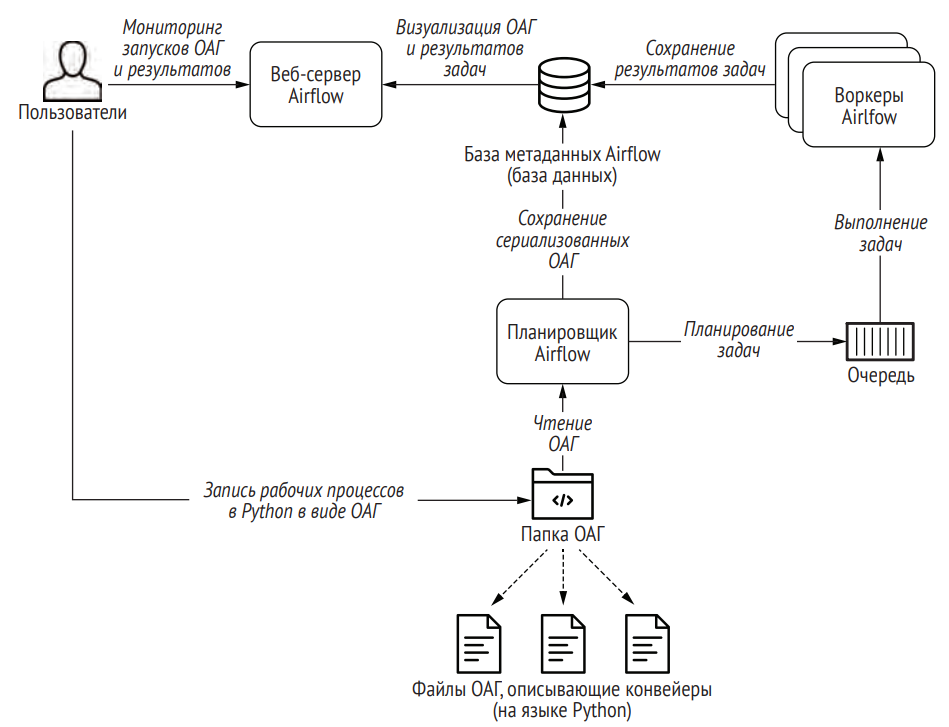


Рисунок 1 — обзор основных компонентов Airflow

Сердцем Airflow, вероятно, является планировщик, поскольку именно здесь происходит большая часть магии, определяющей, когда и как будут выполняться ваши конвейеры. На высоком уровне планировщик выполняет следующие шаги (рисунок 2):

После того как пользователи написали свои рабочие процессы в виде DAG, файлы, содержащие эти графы, считываются планировщиком для извлечения соответствующих задач, зависимостей и интервалов каждого DAG.

После этого для каждого графа планировщик проверяет, все ли в порядке с интервалом с момента последнего чтения. Если да, то задачи в графе планируются к выполнению.

Для каждой задачи, запускаемой по расписанию, планировщик затем проверяет, были ли выполнены зависимости (= вышестоящие задачи) задачи. Если да, то задача добавляется в очередь выполнения.

Планировщик ждет несколько секунд, прежде чем начать новый цикл, перескакивая обратно к шагу 1.

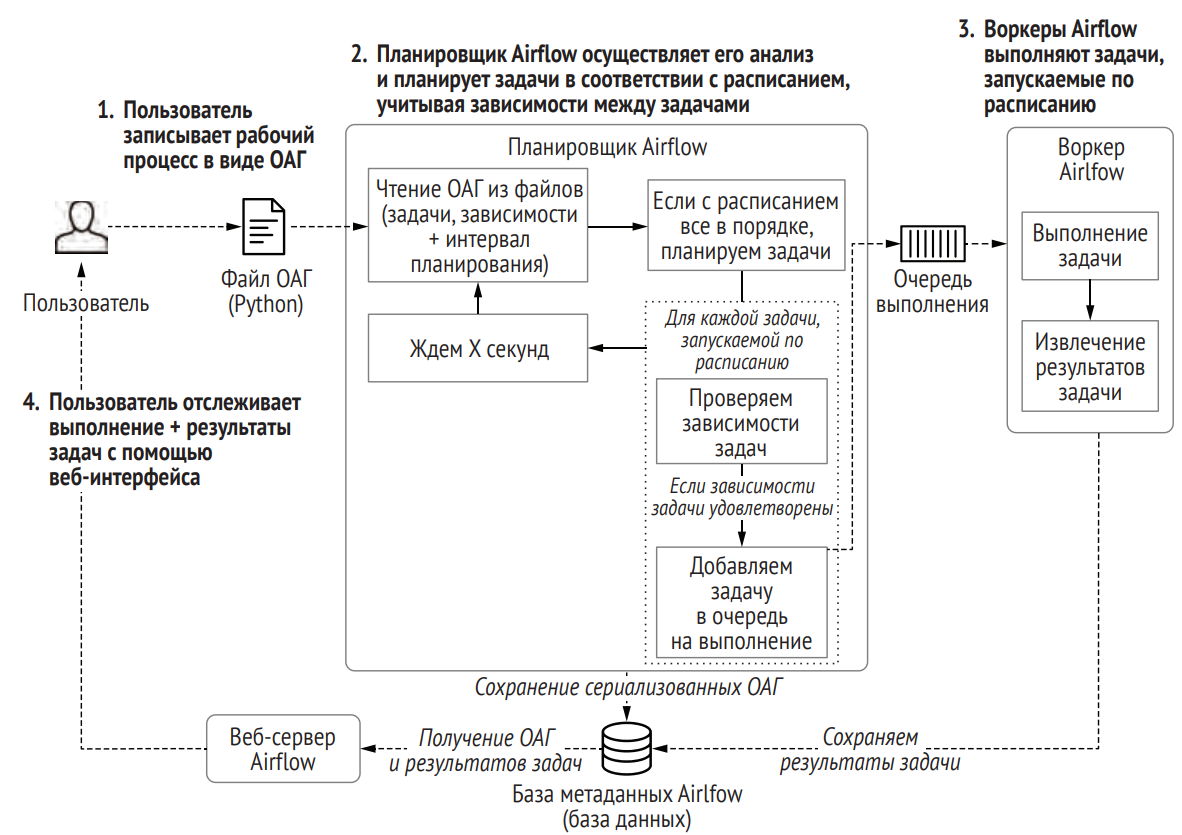


Рисунок 2 — Обзор основных компонентов Airflow (в динамике)

После того как задачи поставлены в очередь на выполнение, с ними уже работает пул воркеров Airflow, которые выполняют задачи параллельно и отслеживают их результаты. Эти результаты передаются в  базу метаданных Airflow, чтобы пользователи могли отслеживать ход выполнения задач и  просматривать журналы с  помощью веб-интерфейса Airflow (интерфейс, предоставляемый веб-сервером Airflow).

***Инкрементальная загрузка и обратное заполнение (backfilling).*** Одно из мощных функций семантики планирования Airflow состоит в  том, что вышеуказанные интервалы не только запускают DAG в определенные моменты времени (аналогично, например, Cron), но также предоставляют подробную информацию о них и (ожидаемых) следующих интервалах.

По сути, это позволяет разделить время на дискретные интервалы (например, каждый день, неделю и т. д.) и запускать DAG с учетом каждого из этих интервалов. Такое свойство интервалов Airflow неоценимо для реализации эффективных конвейеров обработки данных, поскольку позволяет создавать дополнительные конвейеры.

В  этих инкрементных конвейерах каждый запуск DAG обрабатывает только данные для соответствующего интервала времени (дельта данных), вместо того чтобы каждый раз повторно обрабатывать весь набор данных. Это может обеспечить значительную экономию времени и  средств, особенно в случае с большими наборами данных, за счет предотвращения дорогостоящего пересчета существующих результатов.

Эти интервалы становятся еще более мощными в сочетании с концепцией обратного заполнения (рисунки 3-4), позволяющей выполнять новый DAG для интервалов, которые имели место в прошлом[https://bigdataschool.ru/blog/dag-backfilling-in-airflow.html]. Эта функция позволяет легко создавать новые наборы архивных данных, просто запуская DAG с учетом этих интервалов. Более того, очистив результаты прошлых запусков, вы также можете использовать эту функцию Airflow, чтобы повторно запускать любые архивные задачи, если вы вносите изменения в код задачи, что при необходимости позволяет повторно обрабатывать весь набор данных.

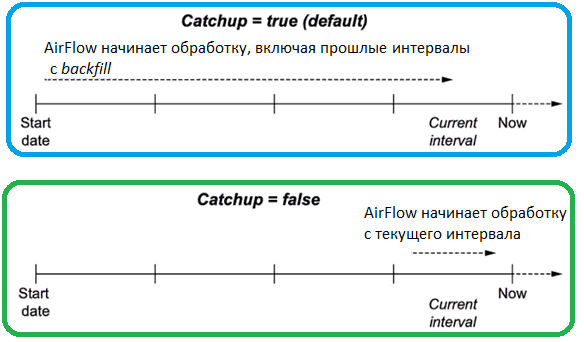


Рисунок 3 — обратное заполнение в Airflow

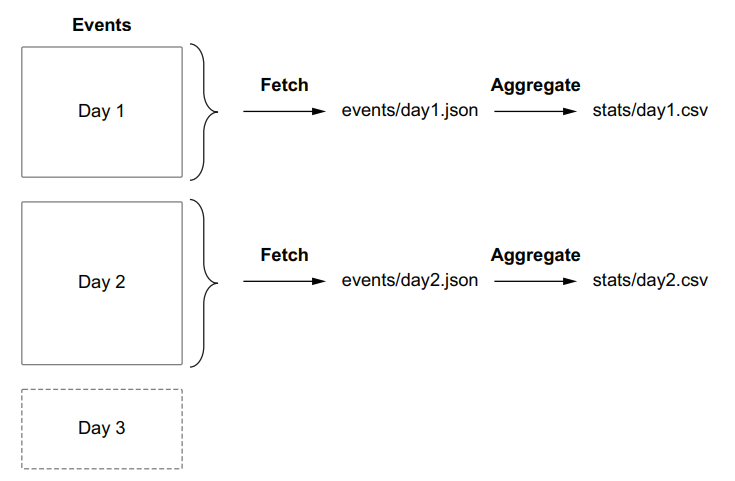


Рисунок 4 — инкрементальная обработка данных

***Рекомендации по разработке задач — атомарность.*** Хотя Airflow выполняет большую часть работы по повторному запуску задач, мы должны убедиться, что наши задачи соответствуют определенным ключевым свойствам для получения надлежащих результатов. Два ключевых свойства: атомарность и идемпотентность [https://bigdataschool.ru/blog/atomicity-and-idempotency-in-airflow.html].

Термин атомарность часто используется в СУБД, где атомарная транзакция рассматривается как неделимая последовательность операций с базой данных, при которой либо выполняются все операции, либо не выполняется ничего.

Аналогично, в Airflow задачи должны быть определены таким образом, чтобы выполняться либо успешно и приводить к какому-либо надлежащему результату, либо выдавать сбой, не влияющий на состояние системы.

В качестве примера, допустим, что мы решили добавить к примеру из рисунка 4 нотификацию пользователей по email. Мы можем записать csv файл...

# Заключение

Несмотря на растущую популярность таких направлений обработки больших данных, как машинное обучение и нейронные сети, в настоящее время остается актуальной проблема проектирования высоконагруженных хранилищ данных, ETL-процессов и витрин данных.

В данном реферате были рассмотрены:

* определение витрины данных;
* актуальные технические проблемы построения высоконагруженных витрин данных;
* организация версионного хранения данных;
* использование Apache Airflow в построении ETL-процессов.

# **Список использованных источников**

1. Простор для данных // habr : сайт. – URL: https://habr.com/ru/articles/650237/ (дата обращения: 15.03.2024)
2. Drupal : сайт. – URL: https://www.drupal.org/node/2360815 (дата обращения: 18.03.2024)
3. Версионность и история данных // habr : сайт. – URL: https://habr.com/ru/articles/101544/ (дата обращения: 21.03.2024)
4. Harenslak, B. Data Pipelines with Apache Airflow / B. Harenslak, J. de Ruiter. – Shelter Island : Manning Publications Co., 2021. – 482 с. – ISBN 9781617296901.
5. Как Apache Airflow помогает дирижировать данными компаний // Яндекс.Практикум : сайт. – URL: https://practicum.yandex.ru/blog/apache-airflow/ (дата обращения: 26.03.2024)