МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

ЗАПУСК ЗАДАЧ ETL ПРОЦЕССА С ПОМОЩЬЮ ФРЕЙМВОРКА APACHE AIRFLOW В КЛАСТЕРЕ KUBERNETES

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Вязкова Андрея Андреевича

Научный руководитель
к. т. н.,доцент _____ В. М. Соловьев

Заведующий кафедрой
к. ф.-м. н.,доцент _____ А. С. Иванов

Саратов 2020

ВВЕДЕНИЕ

Большие данные являются популярным трендом в современном IT. С помощью них можно прогнозировать события, угадывать поведение пользователя, обнаруживать мошенническую активность. Большие данные используются в таких отраслях как: финансовые услуги, здравоохранение, физика.

Однако не всегда данные могут быть однородными и готовыми для дальнейшего анализа. Чтобы они были готовы их необходимо предварительно обработать. Подготовка данных для дальнейшего анализа называется ETL процессом от английского Extract — извлечение, Transform — трансформация, Load — загрузка. В данной работе показан пример построения архитектуры этого процесса с помощью современных инструментов в облаке Amazon Web Services.

1 Краткое содержание работы

Первый раздел «Большие данные» посвящен основной теории, связанной с Большими данными.

Большие данные — обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами.

Термин «Большие Данные» [1] вызывает множество споров, многие полагают, что он означает лишь объем накопленной информации, но не стоит забывать и о технической стороне. К данной сфере именно относится обработка и хранение большого объема информации, для которого традиционные способы являются неэффективными, а также сервисные услуги.

Большие данные характеризуются следующими признаками:

— **Volume** — объем, накопленная база данных представляет собой большой объем информации, который трудно хранить и обрабатывать традиционными способами;

— **Velocity** — скорость, данный признак указывает как на увеличивающуюся скорость накопления данных, так и на скорость обработки данных, в последнее время стали более востребованы технологии обработки данных в реальном времени.

— **Variety** — многообразие, возможность одновременной обработки структурированной и неструктурированной разноформатной информации. Главное отличие структурированной информации – это то, что она может быть классифицирована. Неструктурированная информация включает в себя видео, аудио файлы, свободный текст, информацию, поступающую из социальных сетей. Данная информация нуждается в комплексной обработке для дальнейшего анализа.

— **Veracity** — достоверность, насколько точны полученные данные? [2]

Большие Данные получили широкое распространение во многих отраслях. Их используют в здравоохранении, телекоммуникациях, торговле, логистике, в финансовых компаниях, а также в государственном управлении.

В подразделе «Технологии Больших данных» описаны основные технологии, которые используются в больших данных, такие как: SQL, MapReduce, Apache Spark. Также было написано о сервисах для больших данных, предо-

ставляемых компаниями Amazon, Google и Microsoft.

Во втором разделе «ETL процесс. Его задачи и технологии» было дано определение ETL процесса, а также описаны задачи, которые он решает.

ETL (Extract, Transform, Load) — одна из главных процедур копирования данных из одного или нескольких источников в конечную систему. Данные в конечной системе имеют общий структурированный вид. Термин набрал популярность в 1970-х годах и преимущественно используется при построении хранилищ данных. [3].

ETL процесс состоит из трех фаз. Ниже представлено описание каждой из них [4]:

— Extract — извлечение данных из различных источников. Источниками могут выступать: результаты работы программ, логи этих программ, копии таблиц базы данных, любой внешний набор данных;
— Transform — выполнение преобразований над данными, их фильтрация, группировка и агрегация. На этом этапе сырые данные превращаются в готовый для анализа датасет;
— Load — загрузка обработанных данных в место конечного использования, например в хранилище данных. Эти данные могут быть использованы конечными пользователями или их можно подать на вход другому ETL процессу.

ETL придает данным значительную ценность. Это не просто копирование данных из исходного источника в хранилище данных. ETL решает такие проблемы как:

— Удаляет ошибки и исправляет недостающие данные;
— Настраивает данные из нескольких источников для совместного использования:
— Структурирует данные для использования конечными инструментами:

В подразделе «Технологии для решения задач ETL» были описаны технологии для реализации ETL процесса. Данные технологии были разделены на несколько классов.

— Хранилища данных;
— Планировщики ETL процессов;
— Хранилища общего назначения;
— Среды обработки данных;

В каждом из классов были описаны инструменты и технологи как отдельные части системы, так и сервисы предоставляемые облачными провайдерами.

Третий раздел «Подробное описание архитектуры» посвящен подробному описанию архитектуры, созданного ETL процесса и описанию технологий, которые не были описаны в предыдущем разделе, однако использованы при построении архитектуры.

В подразделе «Описание архитектуры в общем виде» описана архитектура построенного ETL процесса в общем виде. Откуда поступают данные, где хранятся сырые данные, как они обрабатываются и куда записывается результат. Схема архитектуры представлена на рисунке 1

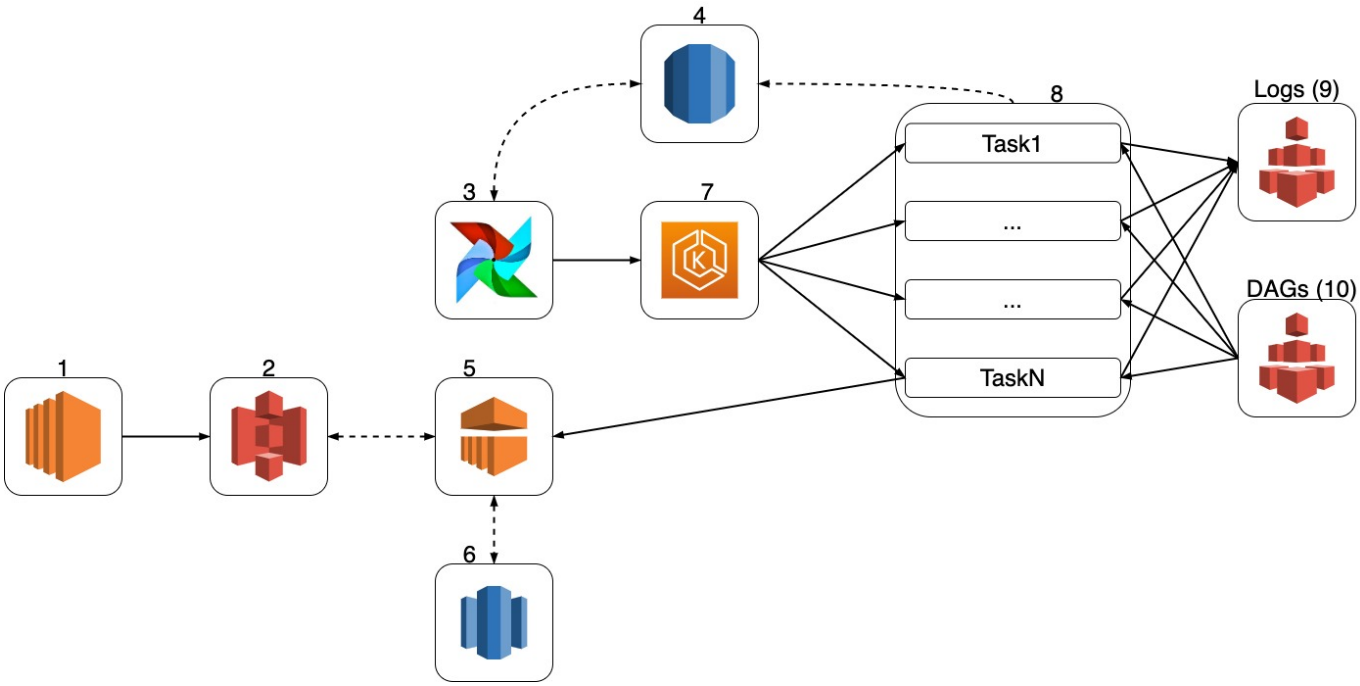

Рисунок 1 – Архитектура ETL процесса

Ключевым элементом архитектуры является Apache Airflow, который запускает задачи обработки данных. Вся архитектура расположена в виртуальной частной сети, предоставленной AWS. Необработанные данные хранятся на отдельном инстансе (1), который моделирует источник данных. При запуске графа на Airflow инстансе (3) происходит проверка наличия необработанных данных на инстансе. Если данные присутствуют, начинается загрузка в S3 корзину (2) для их дальнейшей обработки. Затем запускается задача трансформации данных. Она выполняется в кластере EMR (5). Код программы для обработки написан с помощью фреймворка Spark на языке

Python. Данная программа производит расчет метрик и конвертацию исходных файлов в формат Apache Parquet [5]. Метрики сохраняются в хранилище данных Amazon Redshift (6), а файлы Parquet сохраняются в отдельную папку в S3 корзине (2). Задачи Airflow запускаются в Kubernetes кластере, представленном в виде сервиса Amazon EKS (7). В качестве хранилища метаданных для Airflow используется инстанс базы данных, представленный сервисом Amazon RDS (4). Файл с описанием задач хранится в сетевой файловой системе Amazon EFS (10), а логи выполнения записываются в другую файловую систему того же типа (9).

В подразделе «Серверы с необработанными данными» описана подробная конфигурация сервера на котором хранятся необработанные данные.

Серверы с необработанными данными являются EC2 инстансами типа t2.micro с одним виртуальным ядром и 1Гб оперативной памяти. Инстансы используют дистрибутив Amazon Linux 2, основанный на дистрибутиве Red Hat Enterprise Linux компании Red Hat. Кроме стандартного пакета программ, на инстансы установлен Python интерпретатор версии 3.6, а также интерфейс командной строки AWS CLI, позволяющий управлять сервисами AWS прямо из командной строки. Это необходимо для копирования данных с сервера в S3 корзину. Необработанные данные хранятся в папке `data/csv` в домашней директории стандартного пользователя на каждом сервере соответственно.

В подразделе «Среда запуска Spark приложений» описана среда, где будут запускаться приложения для обработки данных.

Кластер для выполнения трансформации данных работает в сервисе AWS EMR. На кластере установлен фреймворк Apache Spark версии 2.4.4, а также в конфигурации добавлена поддержка Python третьей версии. Кластер состоит из двух узлов: одного главного узла и одного основного. Узлы являются инстансами типа m4.large с 4 виртуальными ядрами и 8Гб оперативной памяти, в качестве файловой системы кластера используется S3 корзина. Также при старте кластера на каждый узел будет установлен Python библиотека boto3 с помощью, которой можно осуществить доступ к другим сервисам AWS из кода.

В качестве файловой системы кластера и объектного хранилища для необработанный выступает S3 корзина. Все объекты в ней имеют класс хра-

нения Standard. В корзине расположены следующие папки:

— `application` — содержит код Spark приложений;
— `data` — содержит необработанные данные;
— `dependencies` — папка с зависимостями для Spark приложения. В ней расположены `.jar` файлы необходимых библиотек:
— `emr-logs` — папка для логов. Все логи, производимые кластером записываются в эту папку;
— `parquet` — папка, в которую сохраняются конвертированные файлы в формате Parquet;
— `tmp`— временная папка для кэширования результатов записи в хранилище данных;

В подразделе «Хранилище данных» описана конфигурация хранилища данных для результатов обработки.

В качестве хранилища данных выступает кластер AWS Redshift. Кластер состоит из одного узла типа dc2.large с двумя виртуальными ядрами и 15Гб оперативной памяти, размер дискового пространства в кластере — 160Гб. В кластере была создана база данных `metrics`, в которую записываются результаты работы Spark приложения.

В подразделе «Apache Airflow» описана конфигурация мастер сервера, хранилища метаданных и кластера для запуска и выполнения задач.

В качестве главного сервера на котором работает Airflow, выступает инстанс типа t3.medium с двумя виртуальными ядрами и 4Гб оперативной памяти. На сервер с помощью Ansible устанавливаются интерпретатор Python версии 3.6, интерфейс коммандной строки AWS CLI. Затем производится установка Airflow и создание рабочей директории для него, туда копируется файл конфигурации. Также для возможности запуска задач в Kubernetes кластере устанавливается утилиты `aws-iam-authenticator` и `kubectl` и производится их настройка. С их помощью и при правильной конфигурации Airflow сможет запускать задачи в Kubernetes.

В качестве хранилища метаданных для Airflow использует инстанс базы данных в сервисе AWS RDS типа db.t2.micro с 1 виртуальным ядром, 1Гб оперативной памяти и 10Гб дискового пространства.

Kubernetes кластер кластер, в котором будут запускаться задачи создан с помощью сервиса Amazon Elastic Kubernetes Serivce (EKS). EKS предо-

ставляет полностью управляемый кластер последней версии с возможностью создания групп узлов любого типа. В качестве узла был использован один инстанс типа t3.medium.

В четвертом разделе «Программная реализация» представлено подробное описание программного кода ETL системы. Раздел состоит из следующих подразделов:

— Скрипты создания инфраструктуры;
— Пример Spark приложения;
— Скрипт загрузки данных;
— Airflow граф;
— Конфигурации;

ЗАКЛЮЧЕНИЕ

В ходе написания данной работы были достигнуты следующие цели:

— Ознакомление с таким понятием как ETL процесс;
— Построение его архитектуры;
— Изучение фреймворка Apache Spark;
— Знакомство с платформой облачных вычислений Amazon Web Services;

Также для создания ETL системы было проведено ознаколение с платформой для разработки, планирования и мониторинга рабочих процессов Apache Airflow. В результате ознакомления был написан простой граф, реализующий ETL процесс. А также была выполнена конфигурация Airflow, которая позволяет запускать задачи в кластере Kubernetes, что позволяет более граммотно утилизировать неиспользуемые ресурсы и оптимизировать расходы.

Для создания инфраструктуры в облаке, был изучен инструмент HashiCorp Terraform. С его помощью были созданы все необходимые элементы инфраструктуры, а именно:

— Серверы хранения необработанных данных;
— Airflow сервер;
— Хранилище метаданных для Airflow сервера;
— Кластер для обработки данных;
— Хранилище данных;
— Хранилище общего назначения;
— Кластер Kubernetes;
— Сетевые файловые системы;

ETL система является важным компонентом при построении хранилищ данных. Она позволяет обработать данные из различных источников и привести их к общему виду.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Аналитический обзор рынка Big Data. [Электронный ресурс]. — URL: https://habr.com/ru/company/moex/blog/256747/ (Дата обращения 21.05.2020). Загл. с экр. Яз. рус.

2 *Cilen, D.* Introducing Data Science. Big Data, Machine Learning and more, using Python tools / D. Cilen. — Manning, 2016.

3 ETL — Wikipedia. [Электронный ресурс]. — URL: https://en.wikipedia.org/wiki/Extract,_transform,_load (Дата обращения 21.05.2020). Загл. с экр. Яз. англ.

4 Введение в Data Engineering. ETL, схема «звезды» и Airflow. [Электронный ресурс]. — URL: https://habr.com/ru/company/newprolab/blog/358530/ (Дата обращения 21.05.2020). Загл. с экр. Яз. рус.

5 Apache Parquet. Национальная библиотека им. Н. Э. Баумана. [Электронный ресурс]. — URL: https://ru.bmstu.wiki/Apache_Parquet (Дата обращения 24.05.2020). Загл. с экр. Яз. рус.