DATA ENGINEERING

Архитектура и
актуальные инструменты BigData

АРТЁМ БАДАНОВ

Data Engineer, 1T, AmberData

О СПИКЕРЕ

- Владею следующими языками программирования: Scala, SQL, Bash, Java
- Закончил МГТУ «СТАНКИН», кафедра: «Прикладная информатика»
- Финалист хакатона 2019 - UrbanTechChallenge

ВВЕДЕНИЕ В BIGDATA

Big Data — это раздел прикладной информатики, которая отвечает за грамотный подход к данным, их обработку и трансформацию и обеспечение дальнейшего доступа к данным.

- Актуальность темы вызвана увеличением данных с 2000-х годов, в связи с появлением множества источников данных
- Big — значит много. Несколько тысяч гигабайтов, терабайтов, петабайтов

ОТКУДА БЕРУТСЯ ДАННЫЕ?

В настоящее время бесперебойными источниками данных являются сайты, мобильные операторы, данные компаний, медицинские учреждения

Неструктурированные данные, такие как музыка, тексты, картинки, попадают в озеро данных (Data Lake).

Data Lake — это сырая информация, обработав которую, бизнес сможет получить выгоду.

Данные — это ответственность.

Любое распространение необезличенных данных — это уголовная ответственность.

ГДЕ ХРАНИТЬ ДАННЫЕ?

После получения данных в Data Lake, информацию необходимо перенести на хранение в базы и хранилища.

В качестве классических хранилищ часто выступают **Greenplum и Terradata**.

Но, на практике чаще всего можно увидеть массив серверов с операционной системой Linux и возможностью горизонтальной масштабируемости. Эту связку массивов будем называть **кластером.**

GREENPLUM DATABASE

teradata.

Файловая система на кластере — HDFS (Hadoop Distributed File System)

HDFS — ФАЙЛОВАЯ СИСТЕМА

Файловая система, которая позволяет хранить данные на разных узлах компьютеров поблочно.

Стандарт блока — 64 Мб.

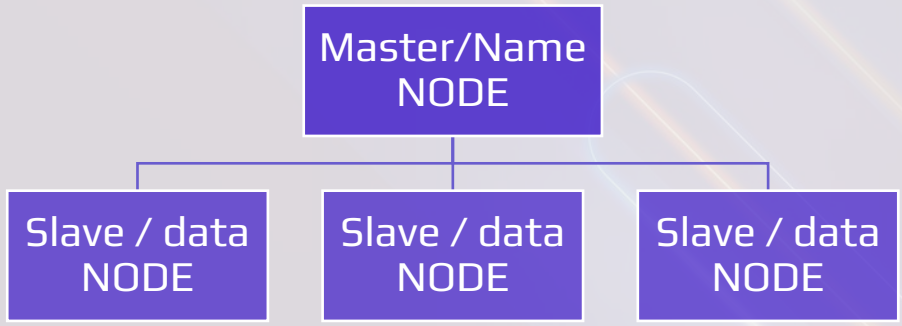
Админами этой файловой системы выступают data node и master node.

Master / Name

Родитель всех data node. В нем содержится информация о работоспособности каждого узла, количестве блоков в каждой data node, размер памяти каждой data node.

HDFS

Это отказоустойчивая система, поскольку каждый блок содержит дубликат на другой data node. В случае если data node или весь сервер выйдет из строя — данные не потеряются



Master = name NODE
Slave = data NODE

HDFS — ФАЙЛОВАЯ СИСТЕМА

При правильной архитектуре кластера скорость взятия данных очень быстрая.

Это обусловлено тем, что данные никуда не перемещаются, а рассчитываются на этом же компьютере.

Результат, полученный на множестве узлов кластера объединяется.

Такая технология работы с данными называется MapReduce.

Map

Это шаг, который позволяет брать данные с разных узлов локально и параллельно.

Reduce

Это шаг, который объединяет полученные результаты и хранит их на одном узле.

SPARK, КАК ИНСТРУМЕНТ РАЗРАБОТКИ

Spark — это движок, который позволяет обращаться к данным и выполнять их трансформацию на кластере.

Доступные языки программирования — scala, java, python. Имеется возможность внедрить скрипты Bash и SQL.

Так как технология MapReduce собирает данные на одном узле и хранит ее в памяти, перенести эти данные для обработки далее будет затруднительно и небезопасно.

Появляется выход — написать программу и внедрить ее в кластер.

Данные останутся там, где и лежали.

HIVE И HDFS

Hive — это оболочка над HDFS, которая позволяет работать с данными в виде таблиц. Hive поможет разбить таблицы на бакеты, партиции для более быстрого доступа к ним.

Но как файловая система поймет высокоуровневый программный код на Scala?

Ведь HDFS — это файловая система, которая понимает только поток MapReduce задач. Именно для перевода программного кода и оптимизации запросов внедряется Hive.

Hive помогает кластеру понять наш код, путем перевода SQL запросов на цепочку MapReduce задач.

OOZIE ИЛИ AIRFLOW?

Oozie и Airflow — оркестратор задач. Через них можно строить логику запуска приложения, настраивать начало старта приложения на кластере, выставлять частоту запуска приложения.

Код написан, базы данных готовы, остается ждать кейса от заказчика.

И вот, прилетает долгожданный кейс от заказчика. Необходимо найти клиентов из таблицы, которые хотят рефинансирования ипотеки. Частота запуска программы — каждый день.
Это связано с тем, что таблицы обновляются, а код программы остается таким же.

YARN И SPARK HISTORY?

Заказчик получил свои данные и за последнюю неделю приложение работало корректно. Но, вдруг, приложение остановило свою работу. Что делать?

Для таких случаев были созданы вспомогательные приложения Spark History и Yarn.

Если посмотреть логи нашего приложения в **Yarn'e,** можно увидеть точную ошибку, если ошибка связана с нехваткой памяти.

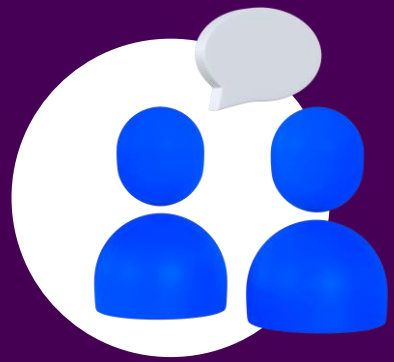
Если посмотреть логи приложения в **Spark History,** можно увидеть, как выполнялась задача и на каком моменте она упала. Возможно, был изменен код.

SKILLS

Для работы дата инженером необходимо обладать некоторым набором навыков, которые позволят работать приятнее и удобнее

Hard Skills

Это набор скиллов, который показывает профессиональный уровень как дата инженера.
Это может быть знание языков программирования, алгоритмов, структур данных, знание баз данных и многое другое.

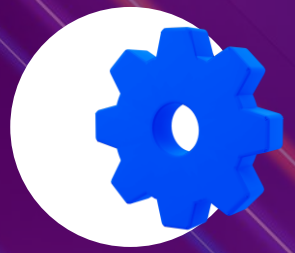


Soft Skills

Эти скиллы нужны для того, чтобы уметь правильно работать в команде, поддерживать здоровые рабочие отношения.
Это базовые умения и качества, такие как ответственность, честность, открытость.

ГРЕЙДЫ

Дата инженер, как и другие профессии, связанные с написанием кода и логики приложения, имеют уровни своего развития. Существует три основных грейда программиста, ниже приведены знания того, что необходимо знать.



Junior

Необходимо понимать предметную область. Знание Scala, Python, Java на уровне ООП. Строить и понимать алгоритмы. Базовое владение SQL. Умение работать с Git, Linux.

Middle

Знание Scala, Python или Java на продвинутом уровне. Знание SQL с использованием оконных функций, подзапросов, транзакций. Понимать как все работает «под капотом».

Senior

Контроль качества написания кода других разработчиков. Знание всех основных конструкций ЯП.

СПАСИБО ЗА ВНИМАНИЕ

Баданов Артем
Data Engineer

Telegram : @artem5240
+7 (977) 699-82-41