Apache Spark. Введение

Введение в технологию, без которой немыслим современный мир данных. Почему Apache Spark является важным инструментом для работы с данными?

Большие данные — это не только «большие»

Часто думают, что Big Data — это просто много данных. **На деле всё гораздо сложнее:** данные льются с бешеной скоростью, приходят в самых разных форматах — от структурированных таблиц до неструктурированных текстов и изображений. Классическая база данных или простой скрипт на Python здесь просто пасуют. Нужен принципиально другой подход — распределённая обработка данных.

Volume (Объем)

Терабайты и петабайты данных, с которыми нужно работать ежедневно

Velocity (Скорость)

Непрерывные потоки из социальных сетей, логов приложений и микросервисов, бизнес-события

Variety (Разнообразие)

Структурные таблицы, неструктурные тексты и картинки, полуструктурные JSON и логи

Именно здесь на сцену выходит герой нашего курса, **Apache Spark** — инструмент, который стал стандартом для компаний по всему миру.

Что такое Apache Spark? Простыми словами

Apache Spark — это открытый распределённый вычислительный «движок»

Он берёт вашу задачу (например, «посчитать статистику по 100 млн пользователей»), разбивает данные на куски, распараллеливает вычисления на кластере компьютеров и возвращает результат.

Самое важное преимущество — он держит промежуточные результаты в оперативной памяти, а не на медленном диске, что даёт просто взрывную производительность по сравнению с традиционными подходами.

Распределённый

Работает на кластере из множества компьютеров, объединяя их вычислительную мощь

In-Memory

Данные обрабатываются в оперативной памяти, что обеспечивает невероятную скорость

Универсальный

Одна платформа для SQL, стриминга, машинного обучения и графовых вычислений

Когда использовать Spark? Сравнение с другими инструментами

Spark — не серебряная пуля для всех задач. Давайте разберём реальные сценарии использования.

Ваша задача	Когда выбрать Spark?	Пояснение	Компонент Spark
Пакетная обработка 1 ТБ логов (ETL-пайплайн)	Всегда	Это 80% использования Spark в продакшене! Инженеры данных ежедневно используют Spark для: • Очистки и трансформации сырых данных • Агрегации данных из разных источников • Подготовки витрин для аналитиков	**Spark Core, Spark SQL**
Частые запросы к одним и тем же большим данным	Когда нужна скорость повторных запросов	Кеширование (**Caching**) — ключевая фича **Spark**. Загрузили 1 ТБ данных в память кластера один раз, а потом 100 аналитиков могут делать по ним быстрые SQL-запросы. Это основа для систем типа Lakehouse (**Delta Lake**).	**Spark SQL**
Обработка кликов в реальном времени	Для стриминговой аналитики	**Structured Streaming** позволяет обрабатывать бесконечные потоки данных (события с сайта, телеметрия, транзакции) с той же логикой, что и пакетную обработку. Используется для: • Мониторинга мошенничества • Персонализированных рекомендаций «здесь и сейчас» • Аналитики в реальном времени	**Structured Streaming**
Подготовка данных для модели либо базовый ML	Да, здесь тоже	**MLlib** — это действительно мощно, но важно понимать: Spark не учит модели лучше, чем специализированные фреймворки (**PyTorch**, **TensorFlow**). Его сила — в подготовке данных для обучения. Подготовка данных для ML-моделей — вот где Spark незаменим. Само обучение часто выносят на GPU.	MLlib
Аналитика на 1 ГБ CSV	Почти никогда	**Spark** — это распределённая система, у неё есть «накладные расходы» на запуск задач, обмен данными между узлами. Для 1 ГБ **Pandas** в 10-100 раз быстрее и проще. Порог входа для **Spark** — обычно от 10-100 ГБ.	—

Как видите, Spark — это в первую очередь инструмент для обработки и трансформации больших данных. Универсальность Spark в том, что он покрывает весь жизненный цикл работы с данными: от сырых логов до готовых моделей.

Ключевые суперсилы Apache Spark

Именно этот уникальный набор возможностей сделал Spark безусловным королём экосистемы Big Data. Вы получаете скорость, как у компилируемых языков, гибкость Python, мощь SQL и надёжность промышленного решения — всё в одном флаконе. Давайте разберём, что делает эту технологию настолько востребованной.

Скорость

Универсальность — единый стек

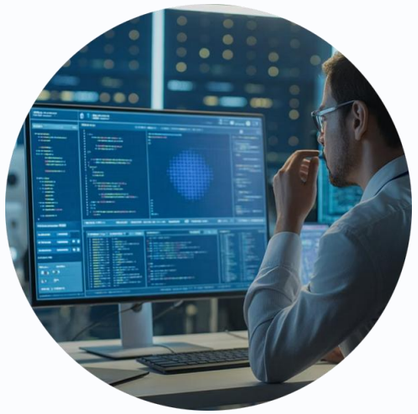
Встроенная отказоустойчивость

Мультиязычность и интеграция

Кому нужен Apache Spark?

Data Engineer

«Мне нужно строить надёжные и быстрые пайплайны обработки данных, которые будут работать годами без сбоев»



Data Scientist

«Мне нужно готовить фичи и обучать модели машинного обучения на огромных выборках в сотни гигабайт»

Data Analyst

«Мне нужно выполнять сложные SQL-запросы и аналитику по терабайтам исторических данных»

DevOps / Software Engineer

«Мне нужно внедрить систему обработки событий в реальном времени в нашем продукте и обеспечить её масштабируемость»

Заключение и следующие шаги

Остались вопросы?

Пишите в комментариях — отвечу каждому! Ваша обратная связь помогает делать курс лучше

Смотрите весь курс

После выхода 8 выпусков объединю в одно видео как гайд

Код на GitHub

Все примеры кода, конфигурации и датасеты доступны в репозитории

Следи за обновлениями

В следующем видео мы уже засучим рукава и запустим Spark у вас на компьютере — будет очень практично!

Подписывайся и оставайся на связи

- Telegram-канал: t.me/marat_notes
- Репозитарий: https://github.com/MaratNotes/marat_notes
- Обучающие видео:
  https://vkvideo.ru/@club231048746
  https://www.youtube.com/@marat_notes