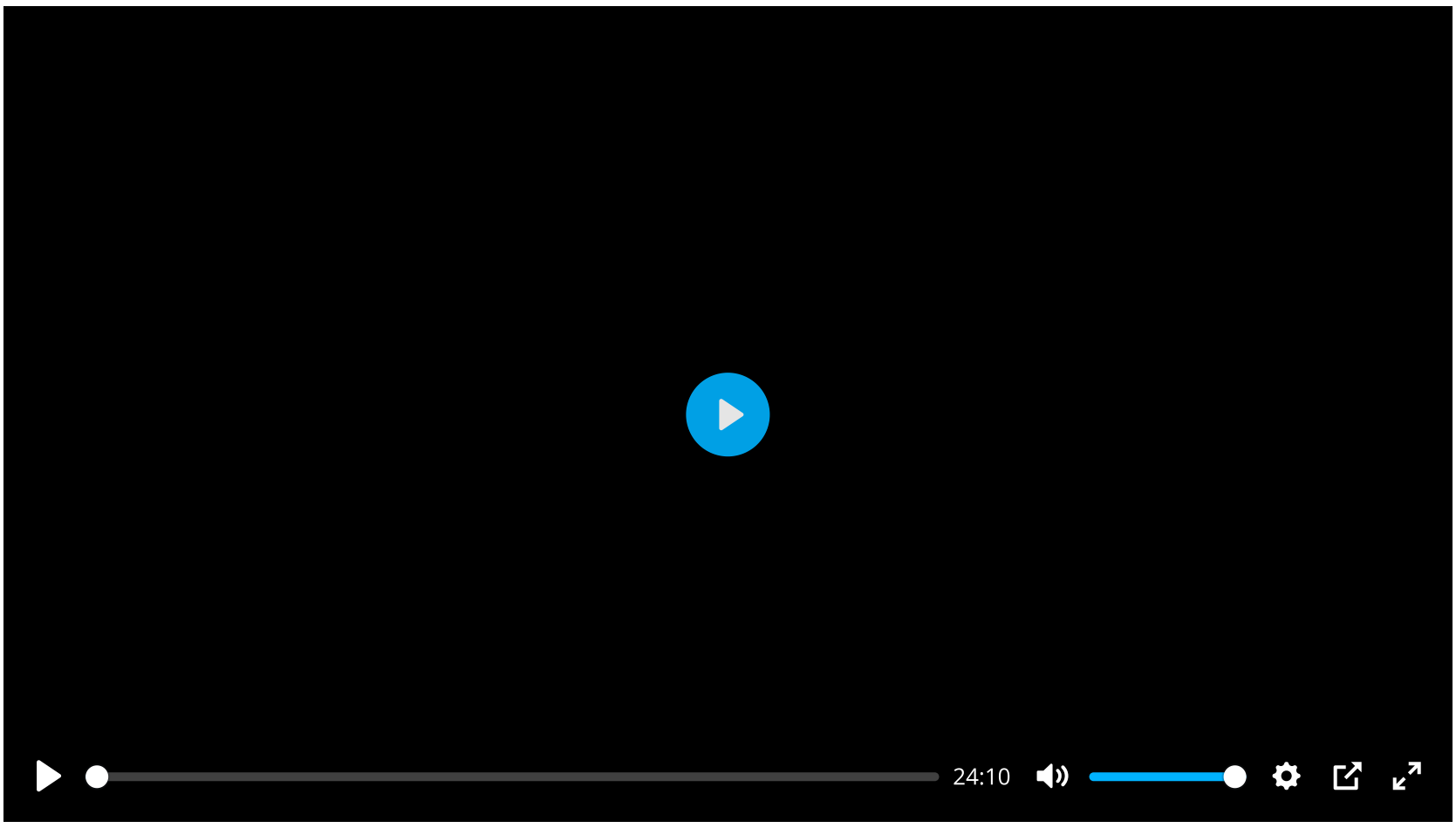
3.2 Экосистема Hadoop – краткий обзор и возможности



Текстовая расшифровка видео:

АНАЛИЗ БОЛЬШИХ ДАННЫХ

План:

- YARN;
- Выбираем данные – Hive, Impala;
- Виртуализация данных – Trino;
- Распределенные вычисления – Spark, Tez;
- Смотрим глазами – Hue, Zeppelin;
- Охраняем данные – Keycloak, Knox, Ranger;
- Что осталось за бортом;
- Как поживает Hadoop в 2023 году?

YARN

Когда мы считаем что-либо распределено, нам хочется отслеживать прогресс, иметь возможность справляться с проблемами (например, падение машины), отслеживать загруженность определенных нодов. В первой версии Hadoop менеджмент ресурсов осуществлялся самим фреймворком **MapReduse**, однако фреймворк был неудобен в ряде случаев, поэтому во второй версии **Hadoop** придумали **YARN (Yet Another Resource Negotiator)**, сделав его отдельным продуктом, отвечающим за запуск распределенных задач.

Большинство современных фреймворков (например, Spark) позволяют использовать внутри себя YARN.

В Spark и других фреймворках есть поддержка аналогий YARN. Одним из основных конкурентов YARN был Mesos, который прекратил существование из-за популярности YARN.

Плюсы YARN:

- Хорошо масштабируется (можно масштабировать кластер на 10 000+ машин);
- Совместим с другими фреймворками;
- Поддерживает динамическое выделение ресурсов.

Выбираем данные – Hive, Impala

Идея проектов Hive и Impala: они могут взять SQL и запустить его поверх встроенного движка.

- **Hive** менялся и поддерживал разные движки запросов, начиная с MapReduce и заканчивая Spark.
- **Impala** была разработана в Cloudera, как «улучшенный Hive» на C++ с поддержкой HiveQL, YARN.
- **Hive** имеет ниже время старта и latency, но выше throughput и отказоустойчивость.
- **Hive Metastore** – полезный компонент, который может использовать сторонние фреймворки.

Узнать больше о сравнении вы можете, перейдя по ссылке: https://www.bigdataschool.ru/blog/hive-vs-impala-sql-on-hadoop.html

Виртуализация данных – Trino

Поверх Hive был создан проект «Presto». Сейчас развивается его более современная версия – Trino.

Изначально смысл Hive заключался в возможности делать запросы по данным в Hadoop. Однако данные могут находиться в совершенно разных источниках, соответственно, с каждым из источников необходимо взаимодействовать отдельно. Из этого родилась **концепция виртуализации данных**.

Суть концепции: у нас есть единый интерфейс, который позволяет писать SQL-запросы, доставая данные из разных мест самостоятельно. Так, мы можем джойнить файлы, лежащие, например, в HDFS с файлами PostgreSQL.

В Hadoop существуют еще два продукта, связанные с SQL и запросами – Apache Phoenix и Apache Drill.

Распределенные вычисления – Spark, Tez

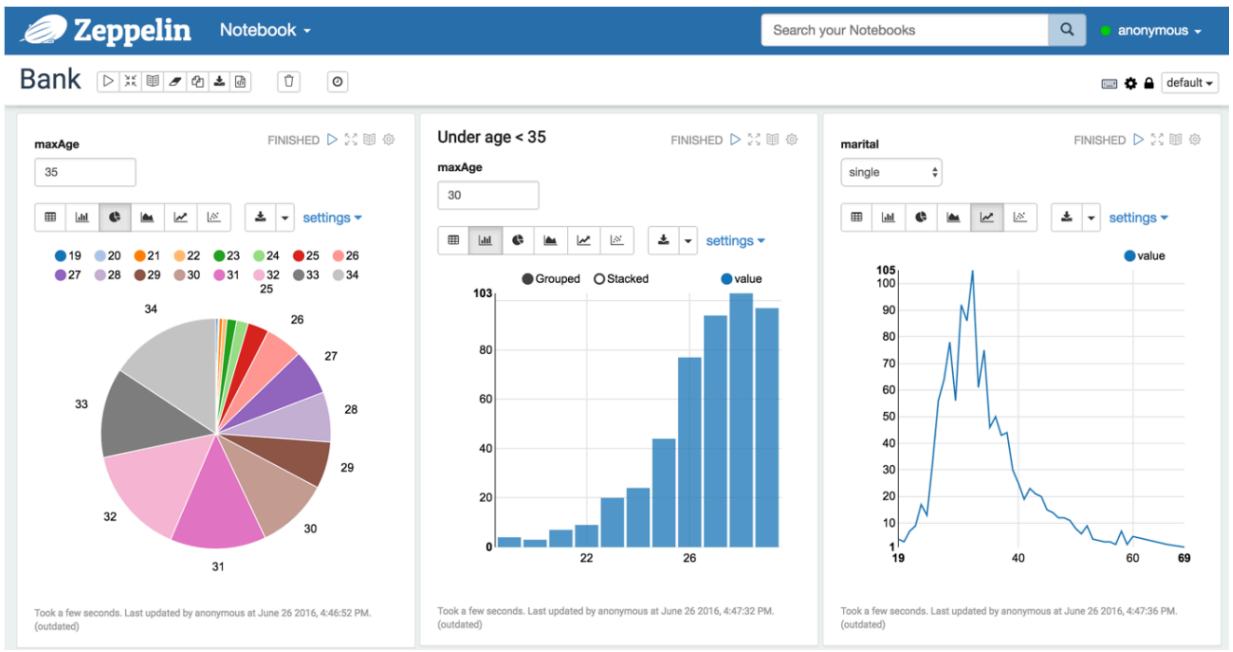
В свое время Tez на равных сосуществовал со Spark.

- Tez разрабатывался раньше, он теснее интегрирован в Hadoop-инфраструктуру, например, в тот же YARN.
- Оба (Tez и Spark) умеют держать данные в оперативке.
- Tez не требует держать контейнеры запущенными и занимать ресурсы.
- Spark более популярен и имеет расширения в виде GraphX и MLLib.

Узнать больше о сравнении вы можете, перейдя по ссылке: https://www.integrate.io/blog/apache-spark-vs-tez-comparison/

Смотрим глазами – Hue, Zeppelin

Результаты запросов можно найти в опенсорсных программах – **Hue, Zeppelin**.



Zeppelin практически «калька» с Jupiter. Zeppelin более заточен под Hadoop.

Охраняем данные – Keycloak, Knox, Ranger

Keycloak чаще всего используется в кластере для авторизации и аутентификации.

Не все сервисы Hadoop поддерживают эти механизмы самостоятельно, часто ставят Knox – дополнительный API Gateway для слоя защиты.

Apache Ranger дает интерфейс для управления доступами и PBAC, в том числе с поддержкой **LDAP**.

Что осталось за бортом

Предлагаем ознакомиться с такими платформами, как:

- Oozie;
- Apache Pig;
- Mahout;
- Druid;
- Apache Camel;
- Apache Storm.

Как поживает Hadoop в 2023 году?

HDFS используется в подавляющем большинстве случаев, как основа для Data Lake.

Вендорные пакеты нужны все реже – почти любой компонент можно заменить.

Большие MPP-базы и движки работают быстрее запросов поверх HDFS, но они сложнее в поддержке.

Вопрос: есть ли смысл в 2023 году брать Hadoop для построения современного решения?

Ответ: скорее, да. Это хорошая основа для Data Lake, большинство фреймворков его поддерживают.