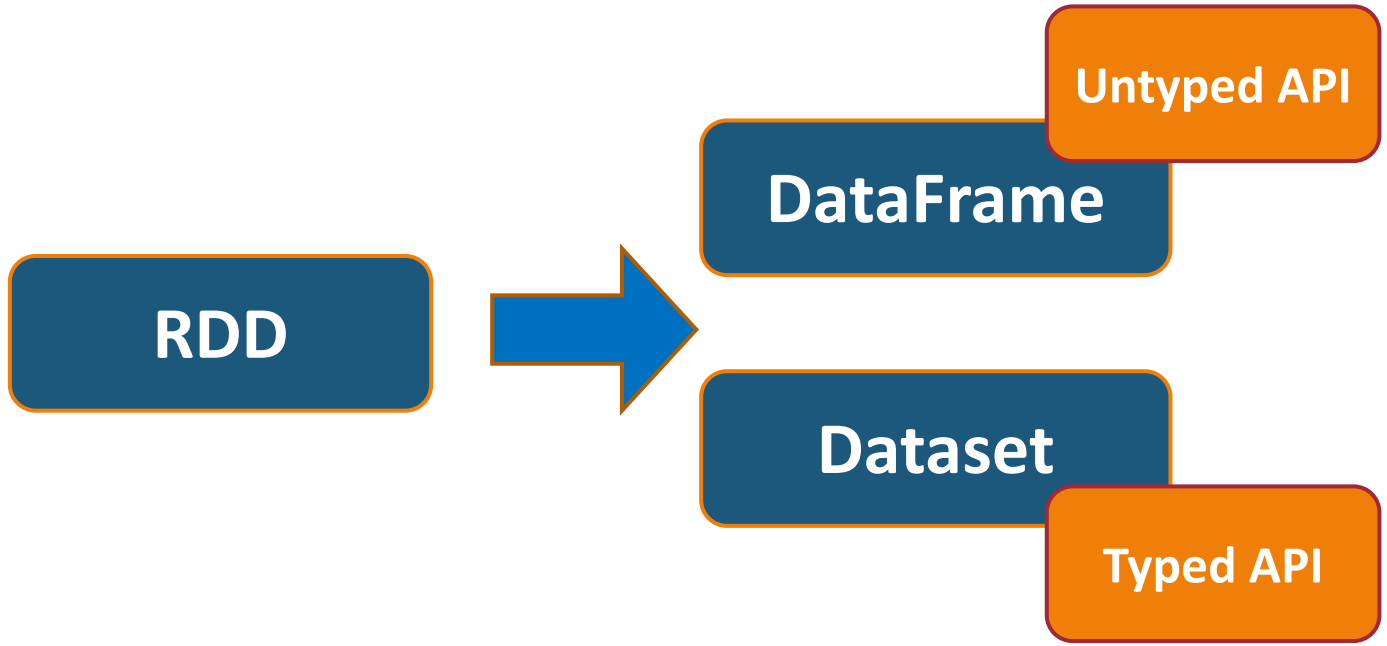
Основные абстракции Spark

МОДУЛЬ 2

ШКОЛА БОЛЬШИХ ДАННЫХ

Основные абстракции



BIG DATA
ШКОЛА БОЛЬШИХ ДАННЫХ

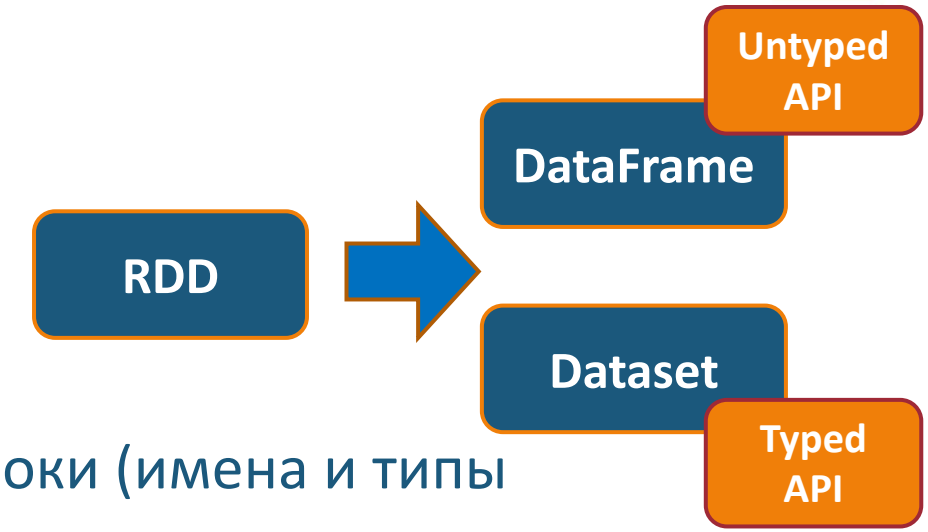
Основные абстракции

- **RDD (Resilient Distributed Dataset)**: набор объектов, разбитых на разделы (“**partitions**”)
- **Dataset**: набор типизированных записей, разбитых на разделы
- **Dataframe**: набор записей типа “**Row**”, разбитых на разделы
- **Distributed shared variable**: переменные двух типов (“**broadcast**” и “**accumulator**”)

API:

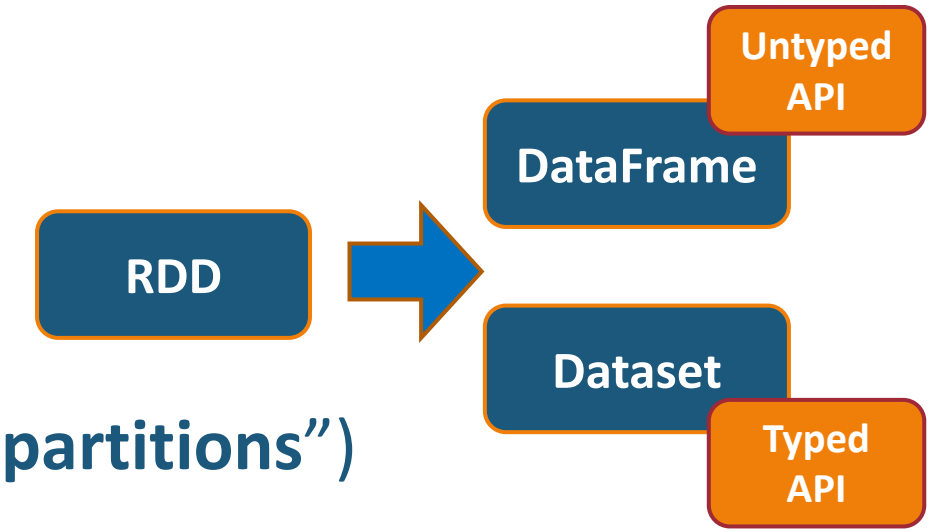
- **Low level API** (**RDD**)
- **Structured API** (**Dataset**, **Dataframe**)
  - ✓ **Dataset**: только в Scala и Java API

Dataframe



- **таблица**, состоящая из строк и столбцов
  - ✓ похожа на датафреймы в pandas и R
- “**schema**”: список, определяющий структуру строки (имена и типы колонок)
- “**partition**”: набор строк датафрейма, расположенных на одном узле кластера
  - ✓ определяют возможность параллелизма (executors + partitions)
- “**Row**”: строка датафрейма (объект)
- “**Column**”: элемент строки (объект)
- spark работает со своими типами данных (**ByteType, IntegerType, StringType**, ...)

RDD



- набор **объектов**, разбитых на разделы (“**partitions**”)
- часть “**Low Level API**”
- “**dataframe**” "компилируются" в “**RDD**”
- **менее** функционально полная абстракция “**spark**”

Трансформации и действия

- **методы** spark делятся на трансформации (“**transformation**”) и действия (“**action**”)
- **все объекты** (Dataframe, RDD, ...) неизменяемы (“**immutable**”)
- “**transformation**”: алгоритм преобразования объекта
  - ✓ чтобы изменить объект нужно задать инструкцию его изменения
  - ✓ результат трансформации - объект (например, Dataframe)
- “**action**”: вычисление, возвращающее результат
  - ✓ подразумевает перемещение данных между экзекьюторами и драйвером
  - ✓ Примеры (посмотреть данные на консоли, собрать данные на драйвере, сохранить данные в файл
- “**lazy evaluation**”: трансформации выполняются только во время действия (“**action**”)
  - ✓ используется оптимизатор (“**Catalyst**”)
  - ✓ план исполнения - алгоритм построения объекта (**dataframe**)

Lazy Evaluation

Lazy evaluation (or call-by-name) is an evaluation strategy which delays the evaluation of an expression until its value is needed

Основные действия

Основными действиями (**“action”**) являются:

- ✓ **collect()** - возвращает dataframe/rdd на драйвер
- ✓ **take()** - аналогично collect(), только первые N элементов
- ✓ **first()** - возвращает первый элемент
- ✓ **show()** - "показывает" первые несколько строк dataframe
- ✓ **count()** - возвращает количество элементов в dataframe/rdd
- ✓ **saveAs...()** - сохраняет dataframe/rdd в файл (в HDFS)
- ✓ **saveAsTable()** - сохраняет dataframe в таблицу (Hive)

Вопросы?

ШКОЛА БОЛЬШИХ ДАННЫХ