Tutorial 03
Spark for batch and streaming processing

FREDERICK AYALA-GÓMEZ

PHD STUDENT IN COMPUTER SCIENCE, ELTE UNIVERSITY
VISITING RESEARCHER, AALTO UNIVERSITY

Agenda

Spark at a Glance
- Summary from lecture 09
- Spark Applications
- Spark's APIs: Low-level API, high-level structured API

Getting Started
- The SparkSession
- Spark UI

Spark's low-level API
- RDD
  - Transformations (Lazy)
  - Actions (Eager)
  - Reduction Operations
  - Pair RDDs
  - Join
- Shuffling and Partitioning
- Streaming Context and DStream

Spark's high-level structured API
- SQL and DataFrame

In the previous lecture:

Spark is a distributed big data processing framework.

- Distribution brings new concerns: Node failure and latency

Uses Resilient Distributed Datasets (RDD) to distribute and parallelize the data.

- RDDs are *lazily-created* and *ephemeral*
- Caching and persistence is used to preserve a RDD in memory, disk, or both

RDDs are fault tolerant

- Able to recover the state of an RDD using **coarse-grained transformations** and **lineage.**

Transformations are **lazy** (e.g., map, filter, groupBy, sortBy, reduceByKey)

Actions are **eager** (e.g., take, collect, reduce, first, foreach)

The topology of the cluster matters

Working with RDDs implies **shuffling** and **partitioning**

- Impact on performance due to latency

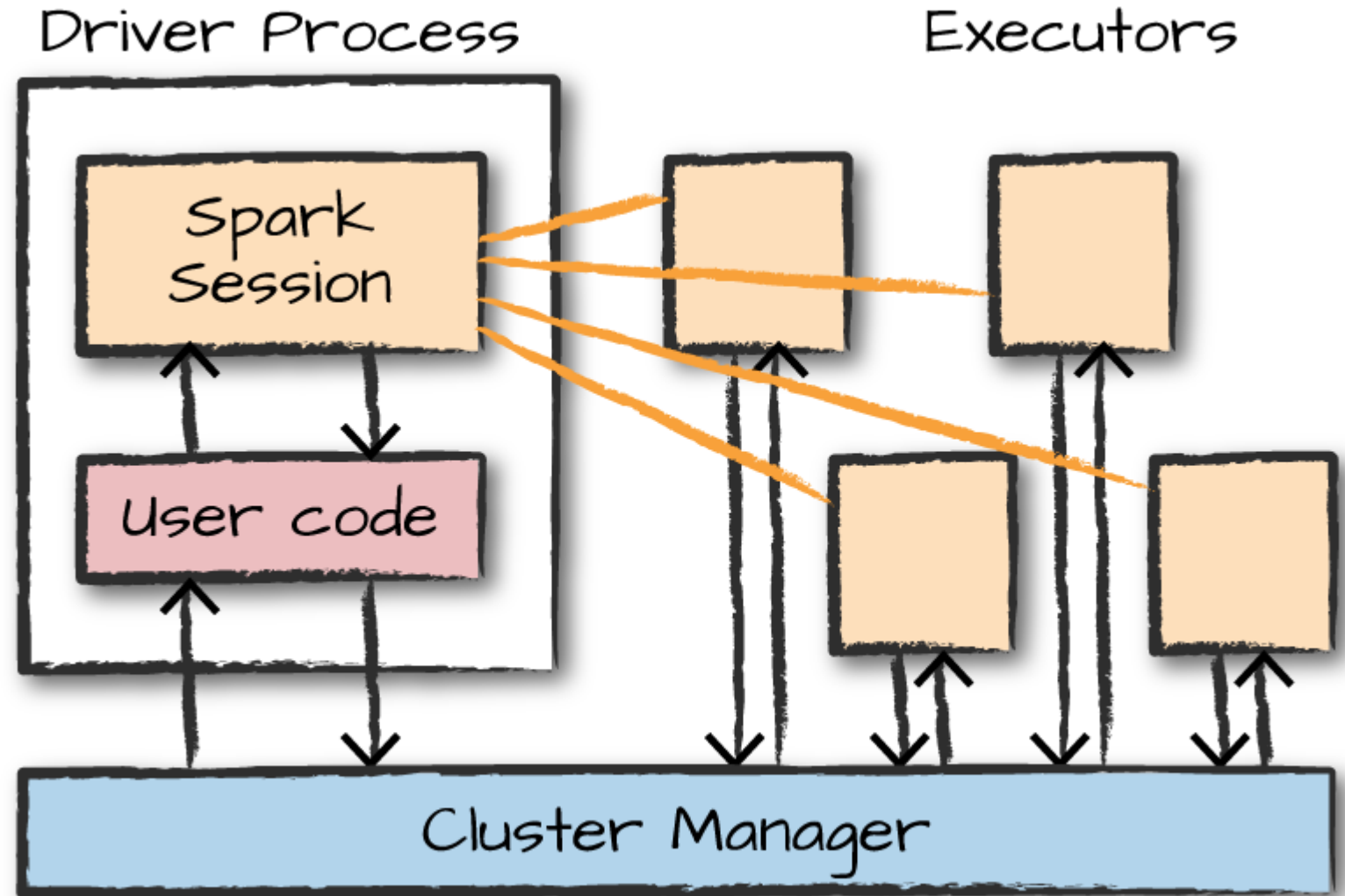Spark provides Big Data Streaming processing via **DStreams**

Spark Architecture

Driver:

- Maintains information about the Spark Application.

- Responding to a user's program or input.

- Analyzing, distributing, and scheduling work across the executors.

Executors:

- Executing code.
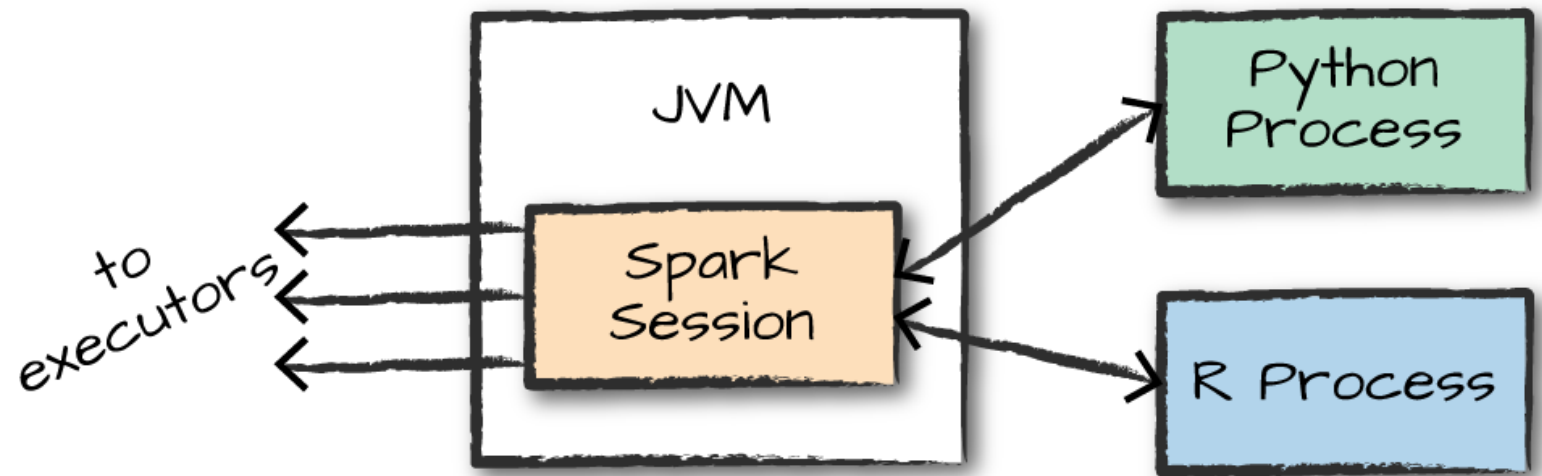
- Reporting the state of the computation.

Spark APIs

Available in many languages:
Scala, Java, Python, R.

These APIs offer a **SparkSession** for running Spark code:

- This is the driver

- Executes user-defined manipulations across the cluster.

- One-to-one correspondence between a SparkSession and a Spark Application.

Spark's toolset

LECTURE 09 SPARK FOR BATCH AND STREAMING PROCESSING
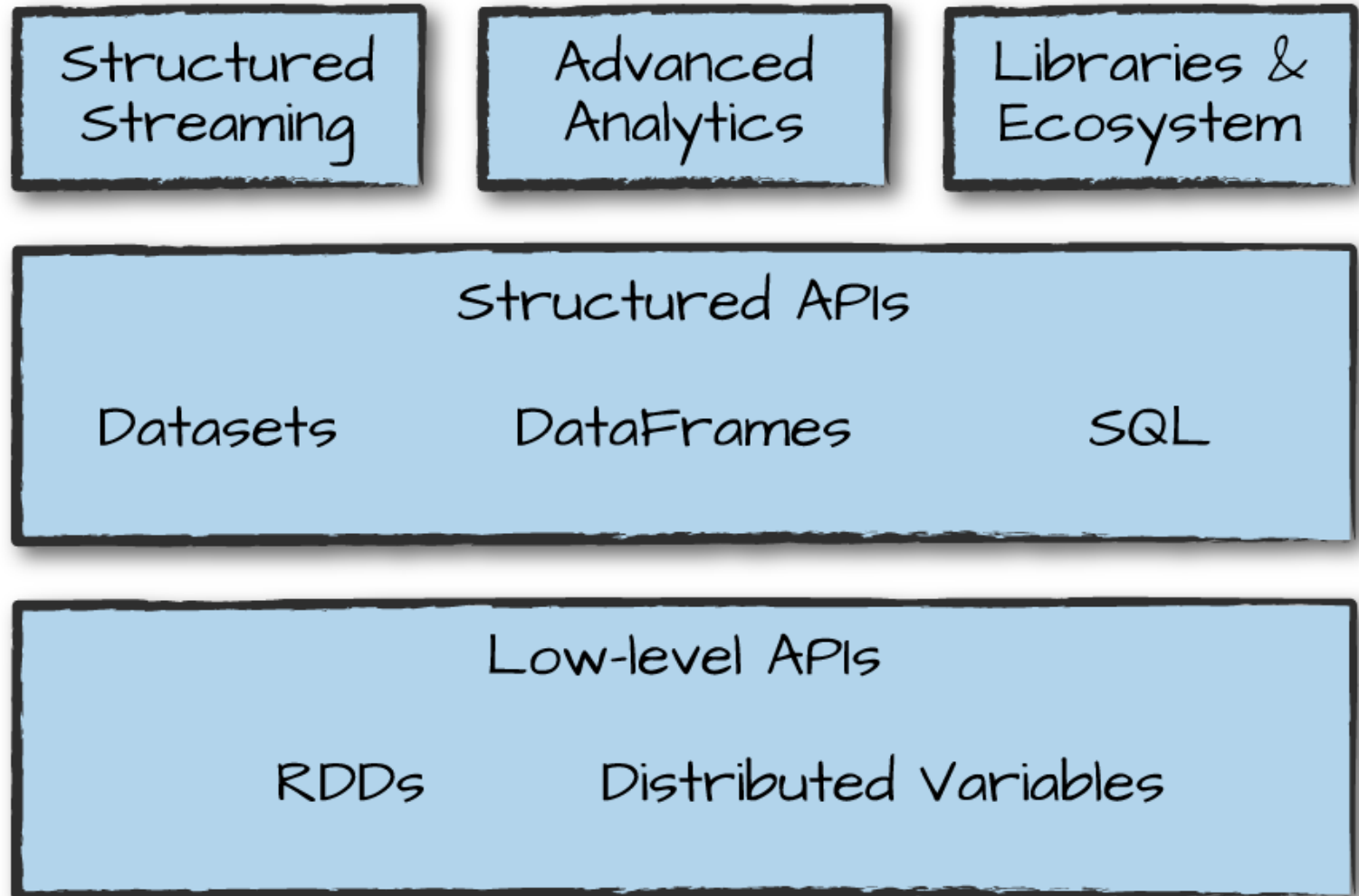
Hands-on: Getting started, RDDs, and DStreams

Issues with RDDs

As developers, we use RDDs to tell Spark **how** to do transformations and actions. Spark will do as we instruct.

Spark cannot optimize our transformations because does not know **what** we are doing.

What is in an RDD?

> Dependencies

> Partitions (with optional locality info)

> Compute function: **Partition => Iterator[T]**

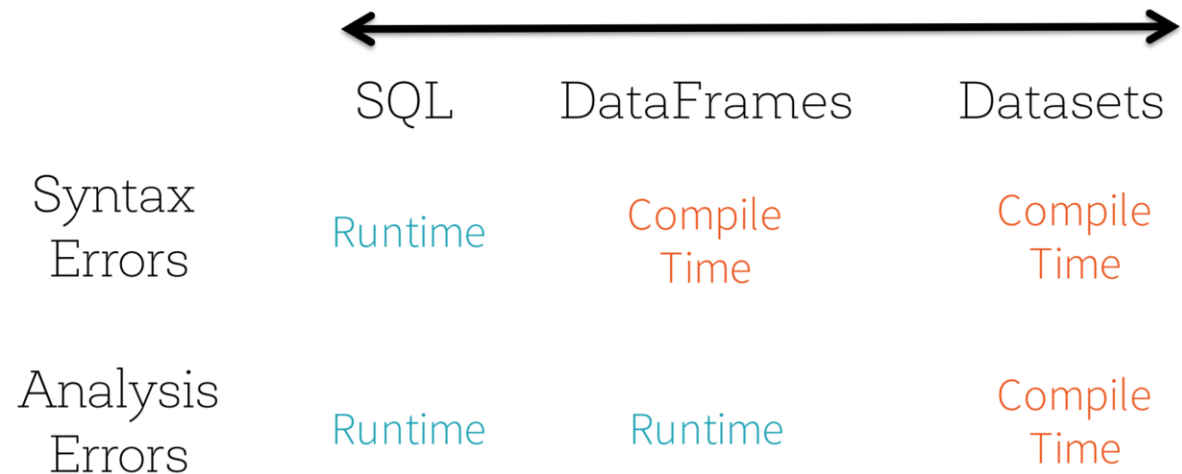- **Partition => Iterator[T]:** Opaque computation
- **[T]:** Opaque data

Structured APIs

Help us to express what we want to achieve.

By doing this, Spark understands the goal and is able to optimize the execution plan.

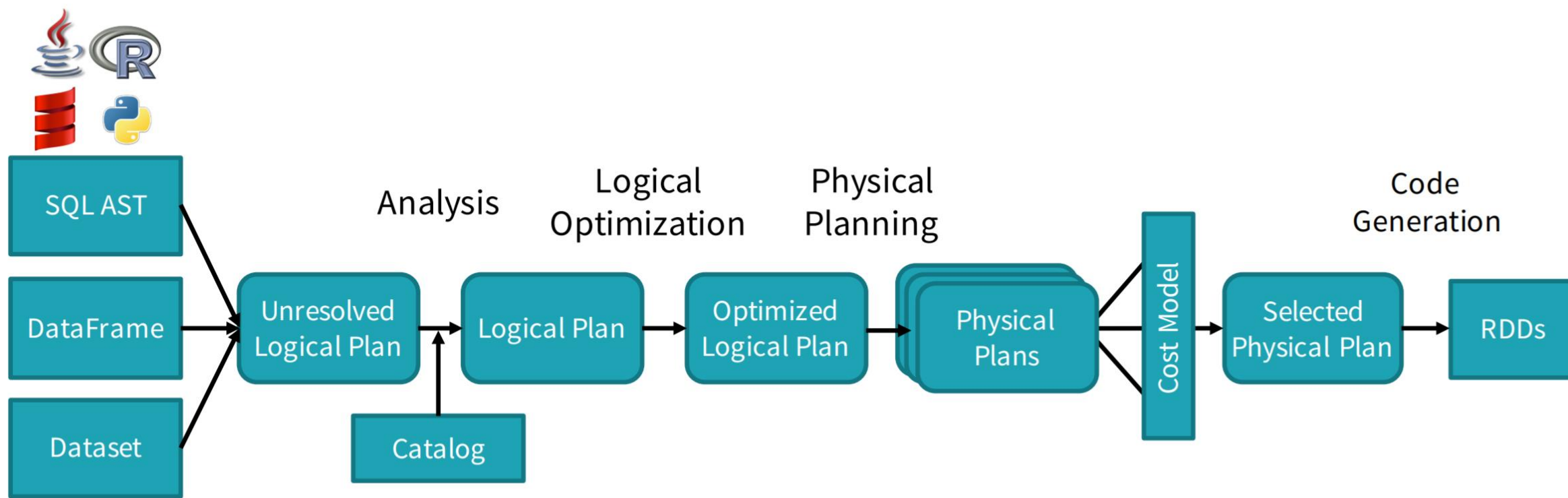Moreover, it can help us to find errors before execution!

Structured APIs In Spark

	SQL	DataFrames	Datasets
Syntax Errors	Runtime	Compile Time	Compile Time
Analysis Errors	Runtime	Runtime	Compile Time

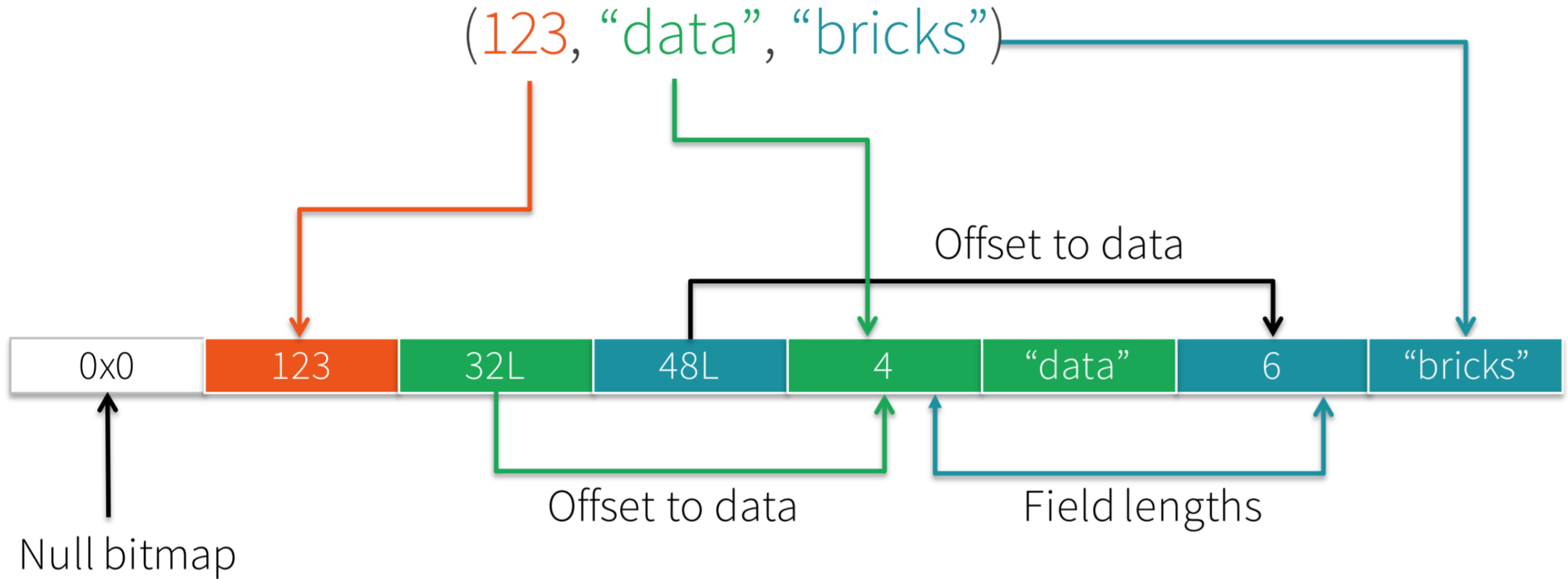**databricks** **Analysis errors reported before a distributed job starts** 7

DataFrame = Dataset[Row]

Shared Optimization & Execution

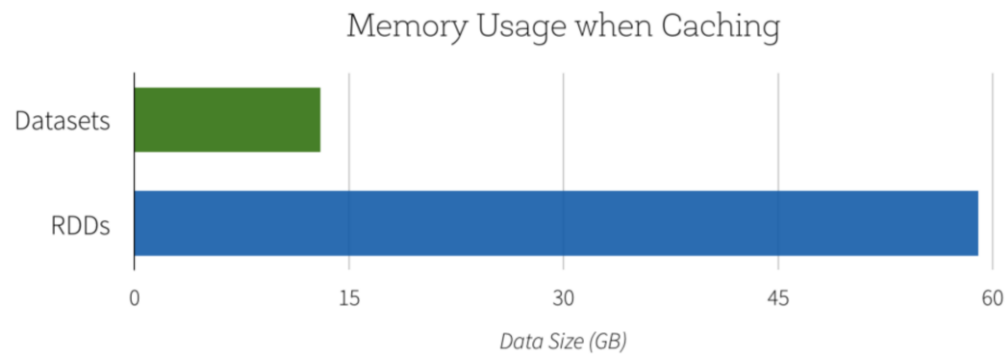DataFrames, Datasets and SQL
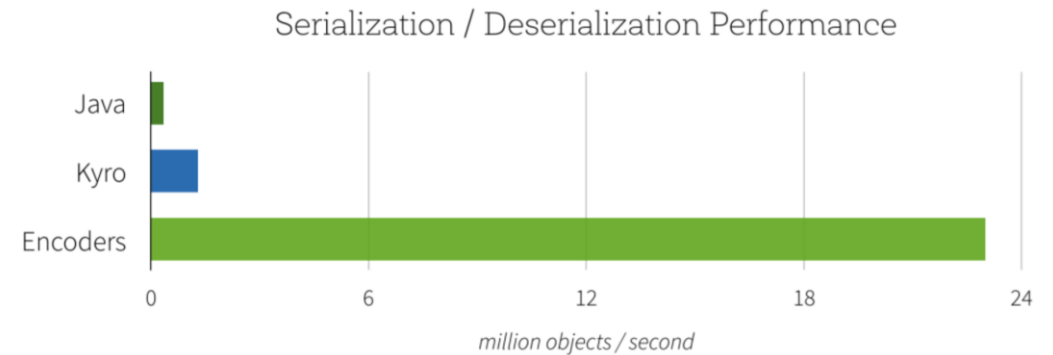share the same optimization/execution pipeline

Tungsten's Compact Encoding

(123, "data", "bricks")

| 0x0 | 123 | 32L | 48L | 4 | "data" | 6 | "bricks" |

Offset to data

Offset to data

Field lengths

Null bitmap

Tungsten's compact encoding improves efficiency

Space Efficiency

Memory Usage when Caching

Datasets	(green bar ~13 GB)
RDDs	(blue bar ~58 GB)

Data Size (GB): 0, 15, 30, 45, 60

Serialization performance

Serialization / Deserialization Performance

Java	(green bar ~0.5)
Kyro	(blue bar ~1)
Encoders	(green bar ~22)

0, 6, 12, 18, 24

million objects / second

DataFrames are faster

Because of the optimization, they tend to outperform RDDs.

Chart from Databricks
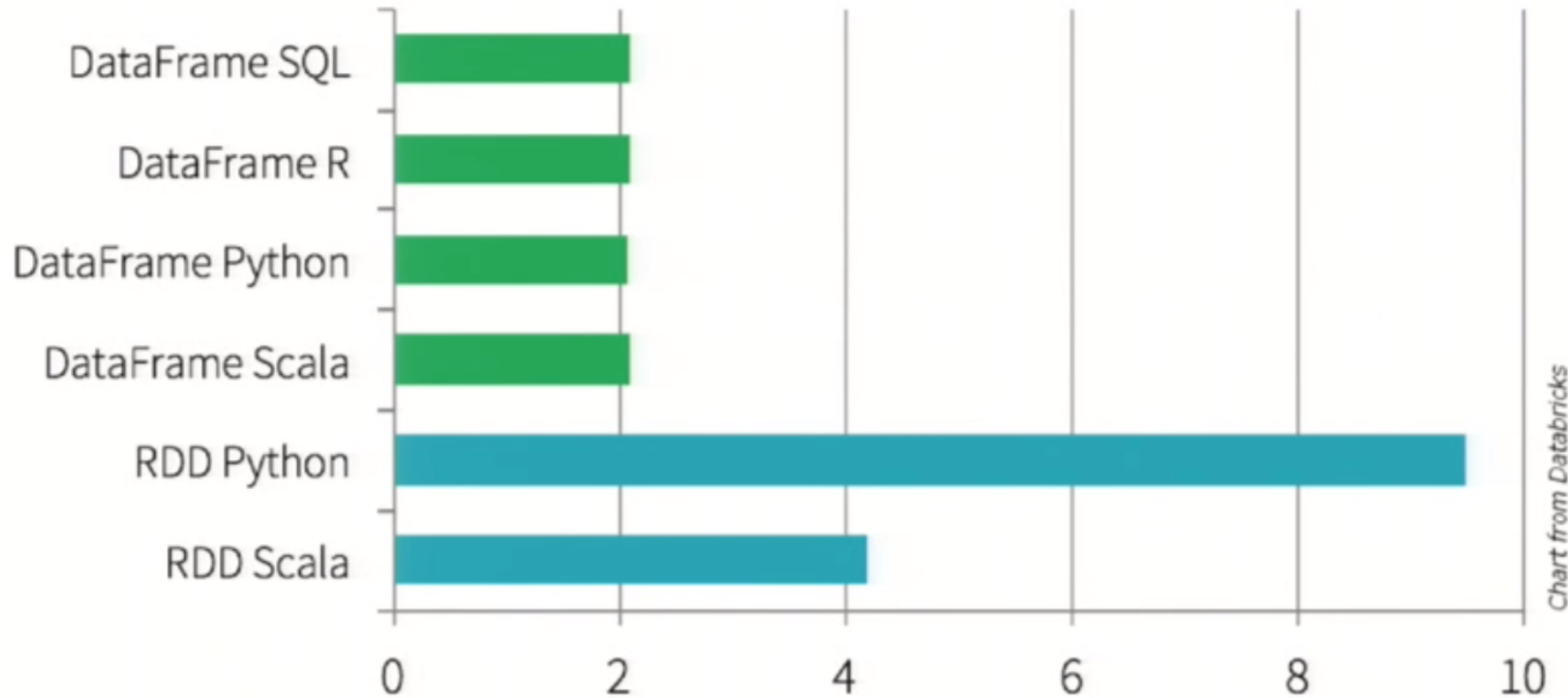
Time to aggregate 10 million integer pairs (in seconds)

Hands-on: Getting started, RDDs, and DStreams

That's all!

Thanks!

Questions?

Frederick Ayala-Gómez
frederick.ayala@aalto.fi

Credits and References

- **Learning Spark** by Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia (O'Reilly). Copyright 2015 Databricks, 978-1-449-35862-4.
- **Spark: The Definitive Guide** by Bill Chambers and Matei Zaharia (O'Reilly). Copyright 2018 Databricks, Inc., 978-1-491-91221-8."
- **Structuring Apache Spark SQL, DataFrames, Datasets, and Streaming**, Michael Armbrust. Spark Summit 2016.