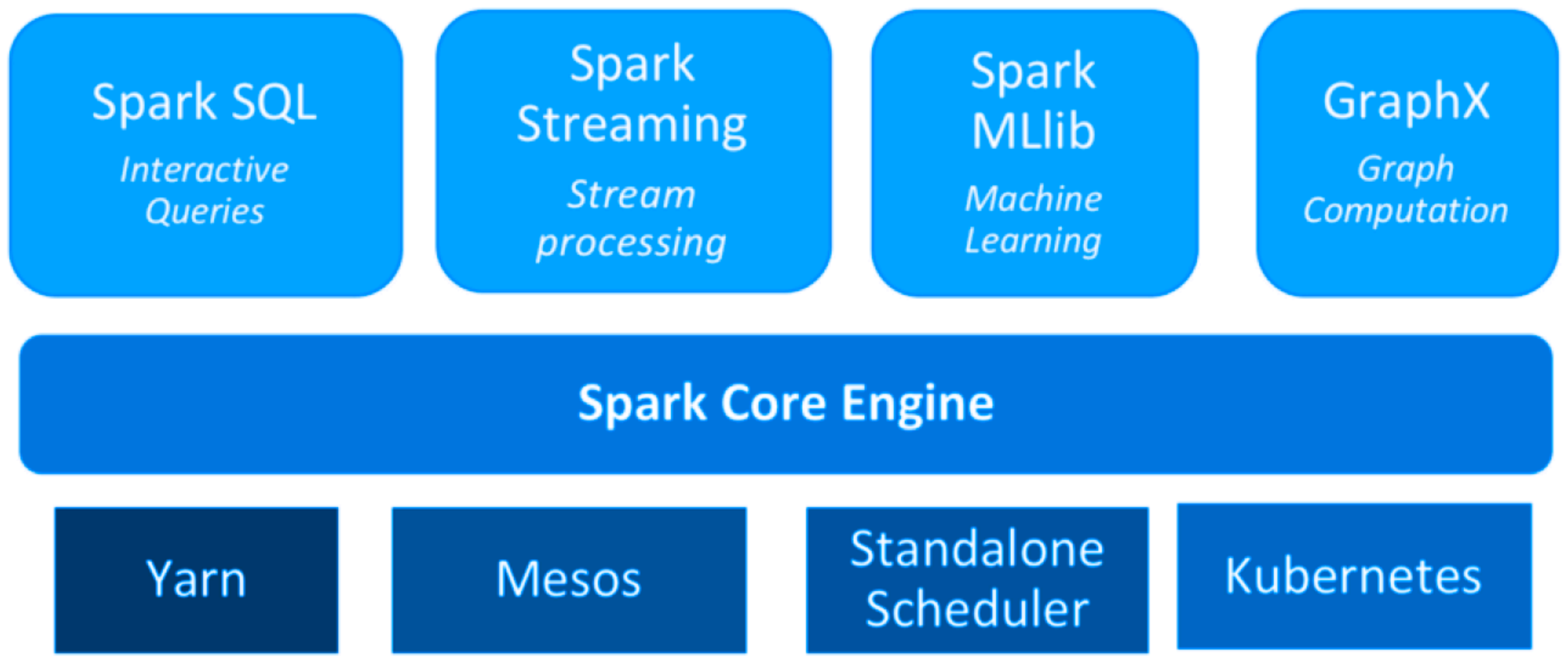
Spark & Scala

Developing highly distributed applications

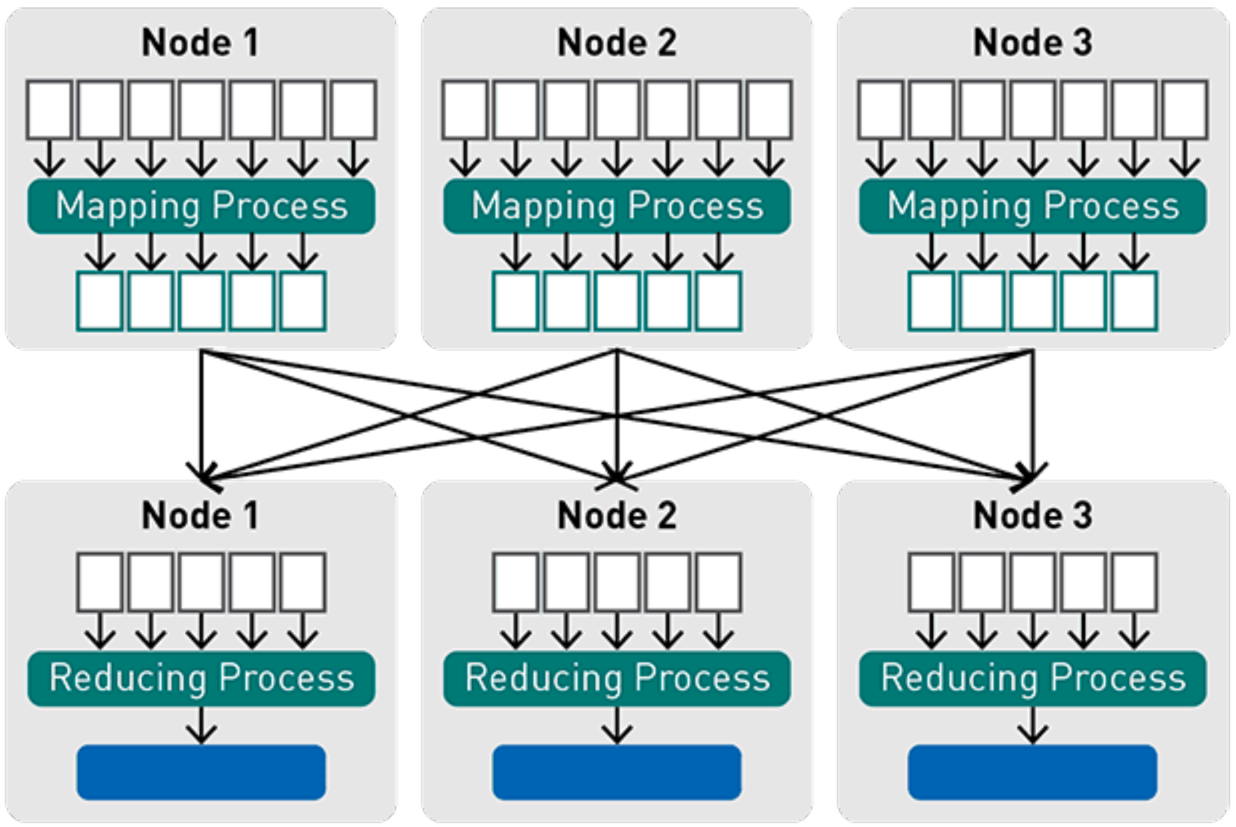
What is Spark?

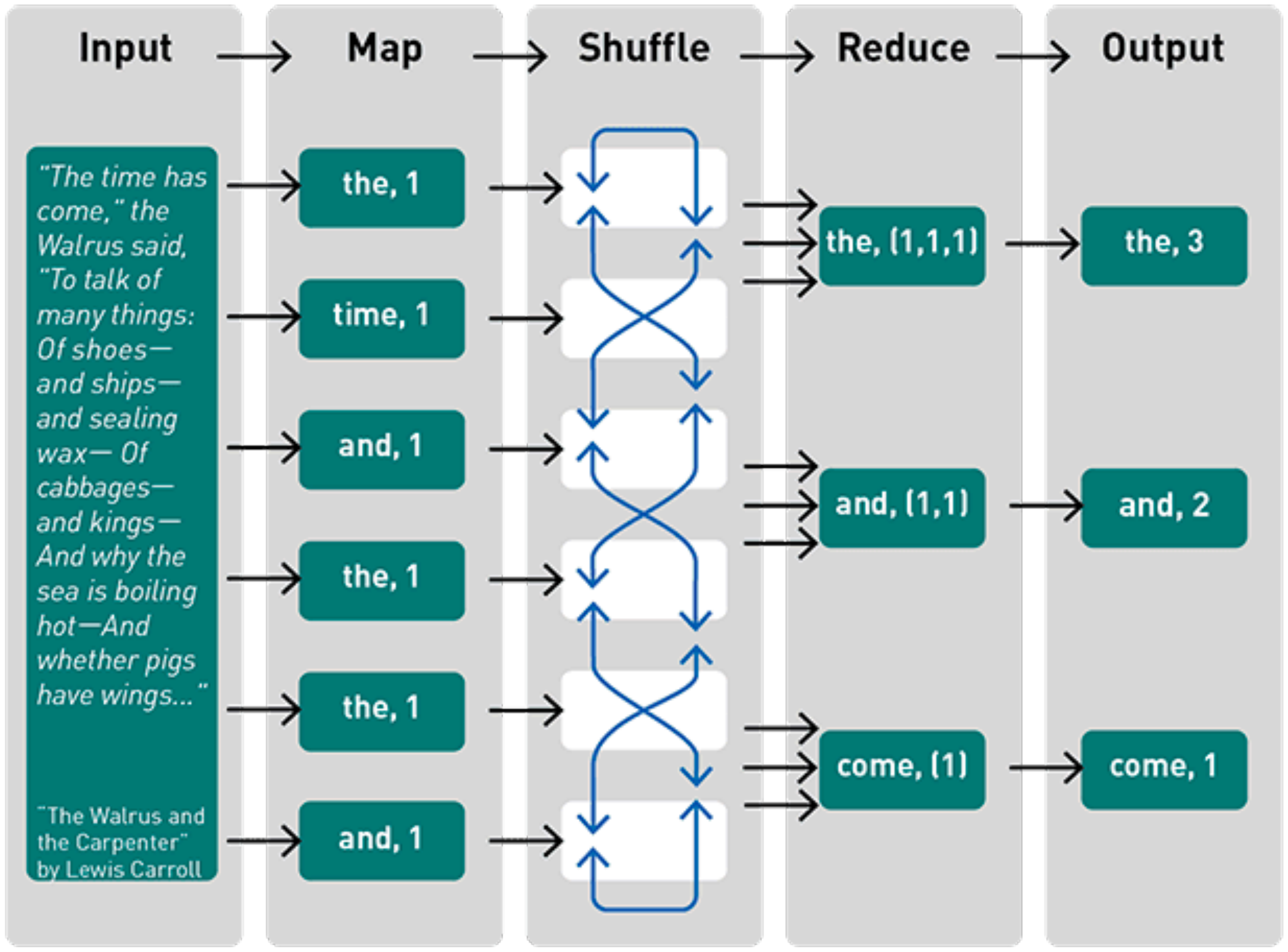
- unified analytics engine
- large-scale data processing
- fault tolerant
- highly distributed
- 'easy to use'
- runs 'everywhere'



MapReduce

- Chunk data in store
- Distribute compute
  - Map nodes to data chuncks
  - Reduce the results
- Fault tolerant




Input
Map
Shuffle
Reduce
Output
"The time has come," the Walrus said, "To talk of many things: Of shoes— and ships— and sealing wax— Of cabbages— and kings— And why the sea is boiling hot—And whether pigs have wings..."
"The Walrus and the Carpenter" by Lewis Carroll
the, 1
time, 1
and, 1
the, 1
the, 1
and, 1
the, (1,1,1)
and, (1,1)
come, (1)
the, 3
and, 2
come, 1

SparkContext

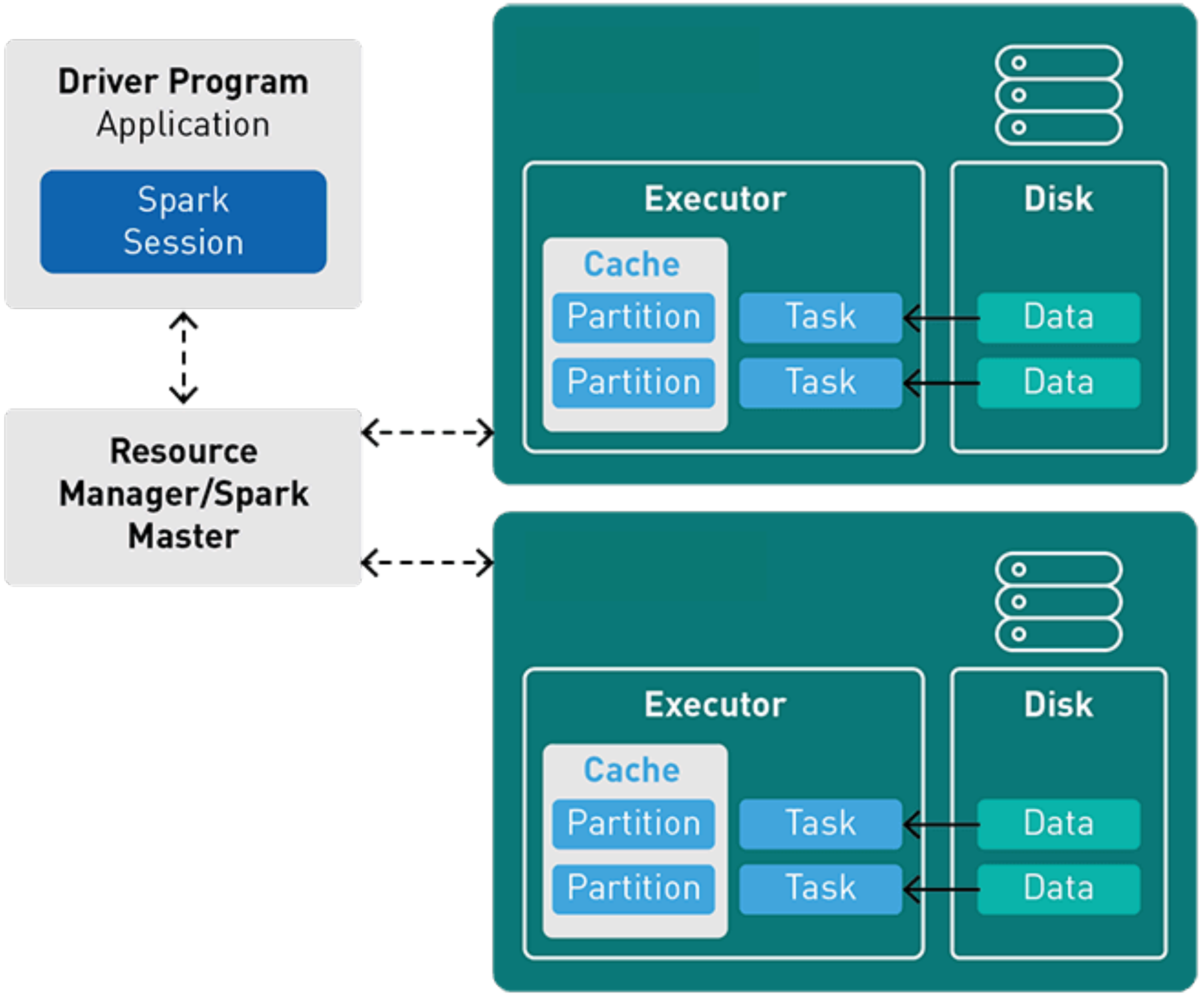
- Independent process coordinated by **SparkSession** in **Driver**

- **Resource Manager** assigns **Tasks** to **Executors** on **Worker** nodes

- Each **Task** is assigned to a **Partition** of data

- A **Task** applies a unit of work to the **Partition** and outputs a new **Partition**

- Results are sent back to the **Driver**

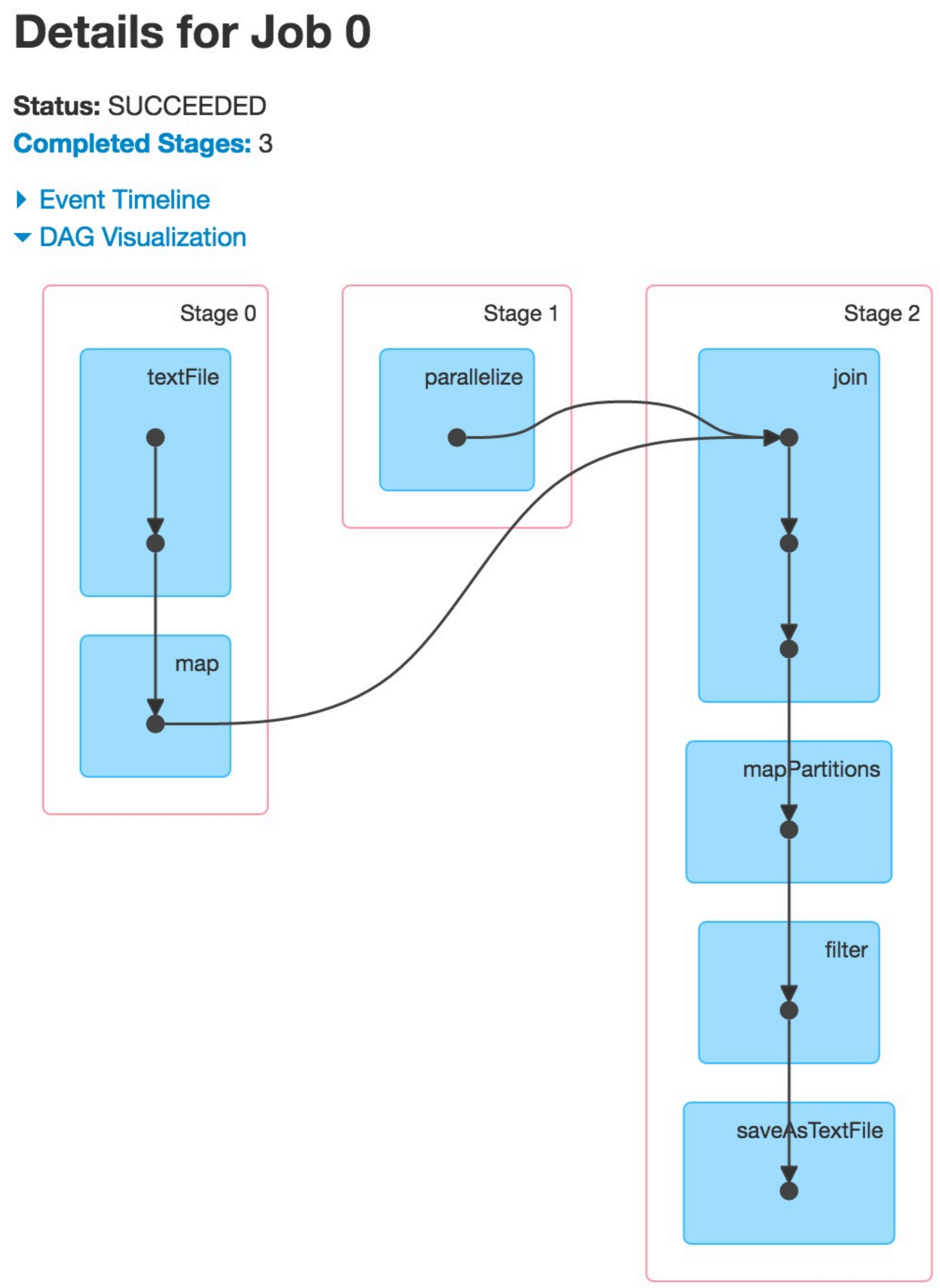


Spark Jobs

- **Job** is an individual action

- **Job** is split in **Stages**, based on reading, caching and joining steps or shuffling events

- **Tasks** are the minimum execution unit



Details for Job 0

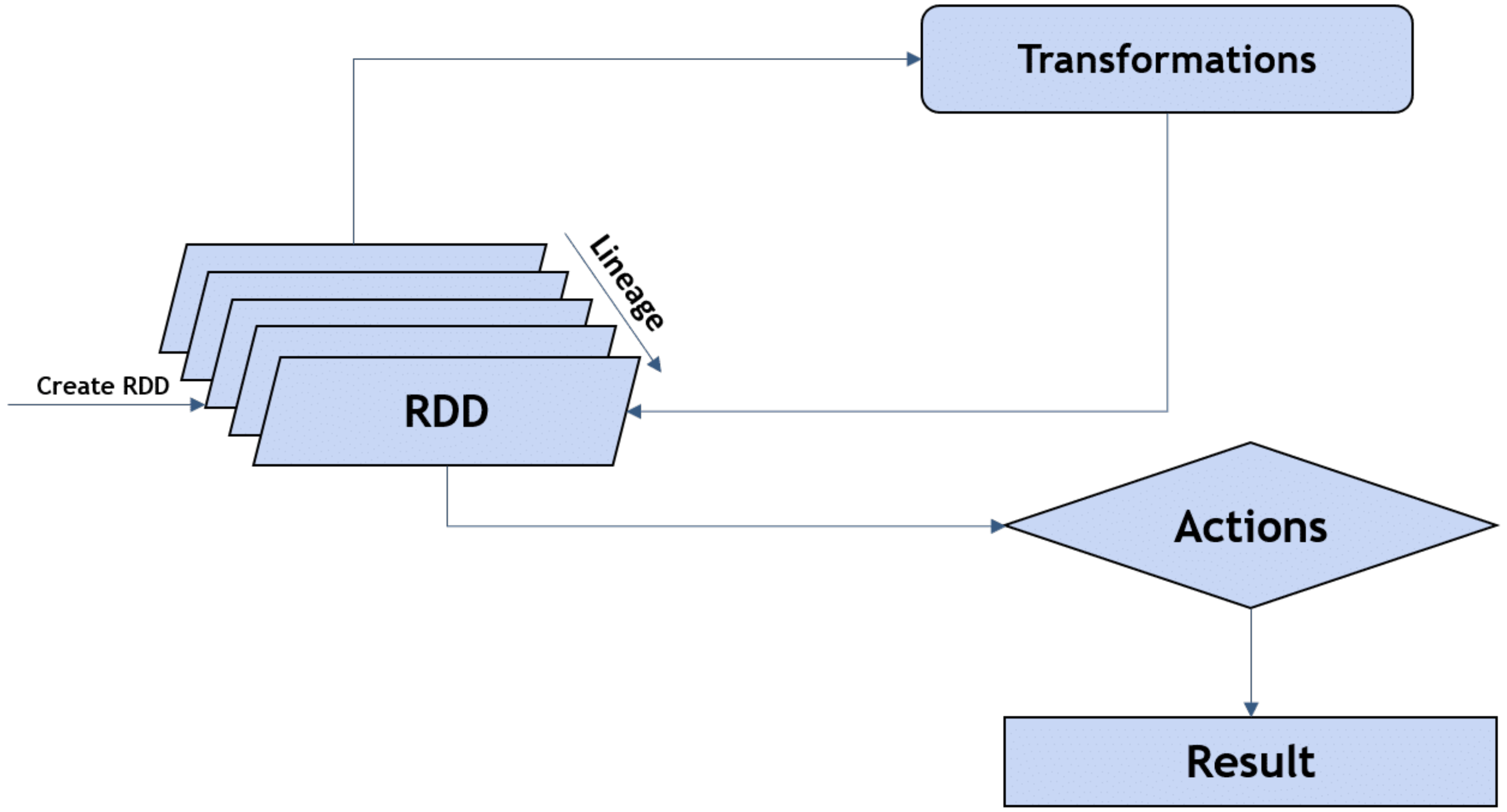
Status: SUCCEEDED
Completed Stages: 3

▸ Event Timeline
▾ DAG Visualization

demo JobsContext & Spark UI

RDD

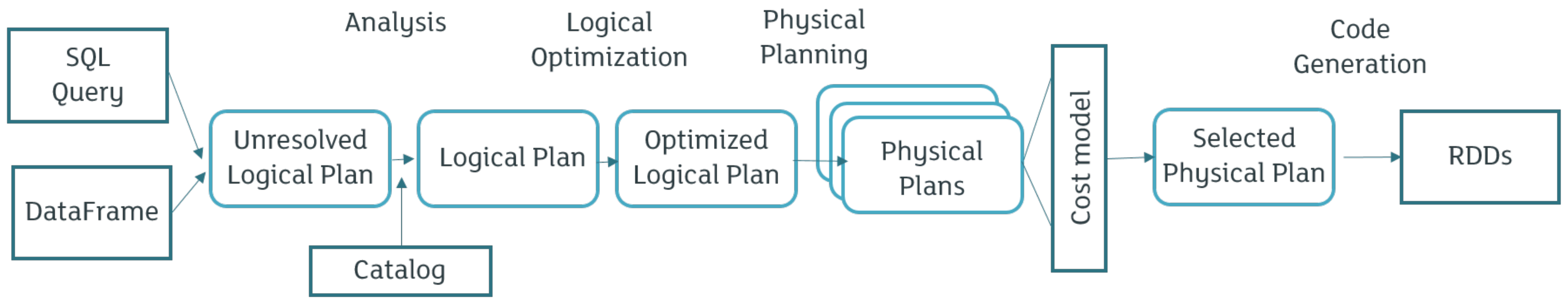
- **R**esilient **D**istributed **D**ataset
- Spark Core
- Huge List split in chunks across machines
- Lazy evaluated
- Composed of
  - Partitions
  - Functions
  - Dependencies other RDD

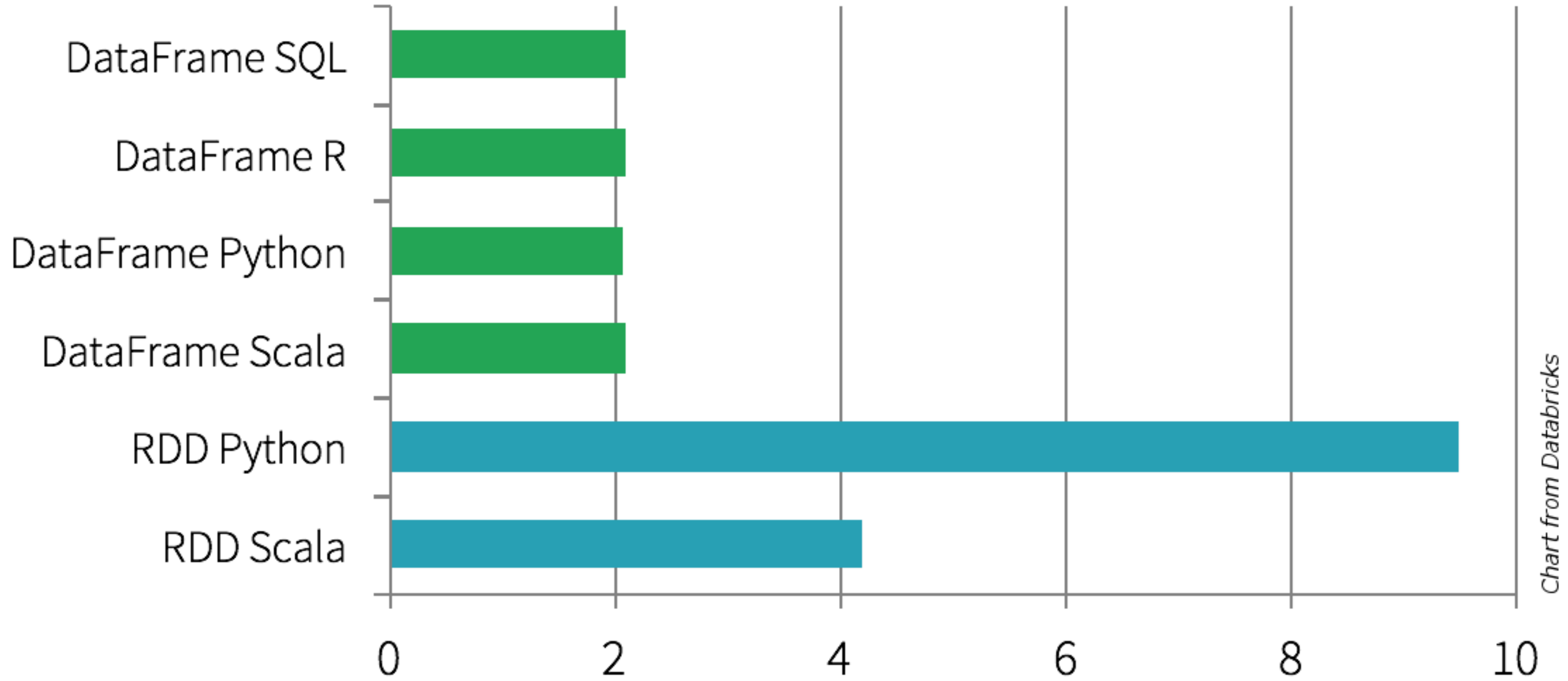


demo RDD & PairRDD

SparkSQL

- Data sources unified
  - Json / Parquet / Avro / CSV / Text / ...
  - Postgres / MongoDB / Kafka / ElasticSearch / ...
- **DataFrame**
  - DataFrame = RDD[Row]
  - Row holds column & types
  - Good for analysis of messy data
- **Tungsten** Custom Memory Management (off heap binary)
- **Catalyst Engine** (Optimised Plans)

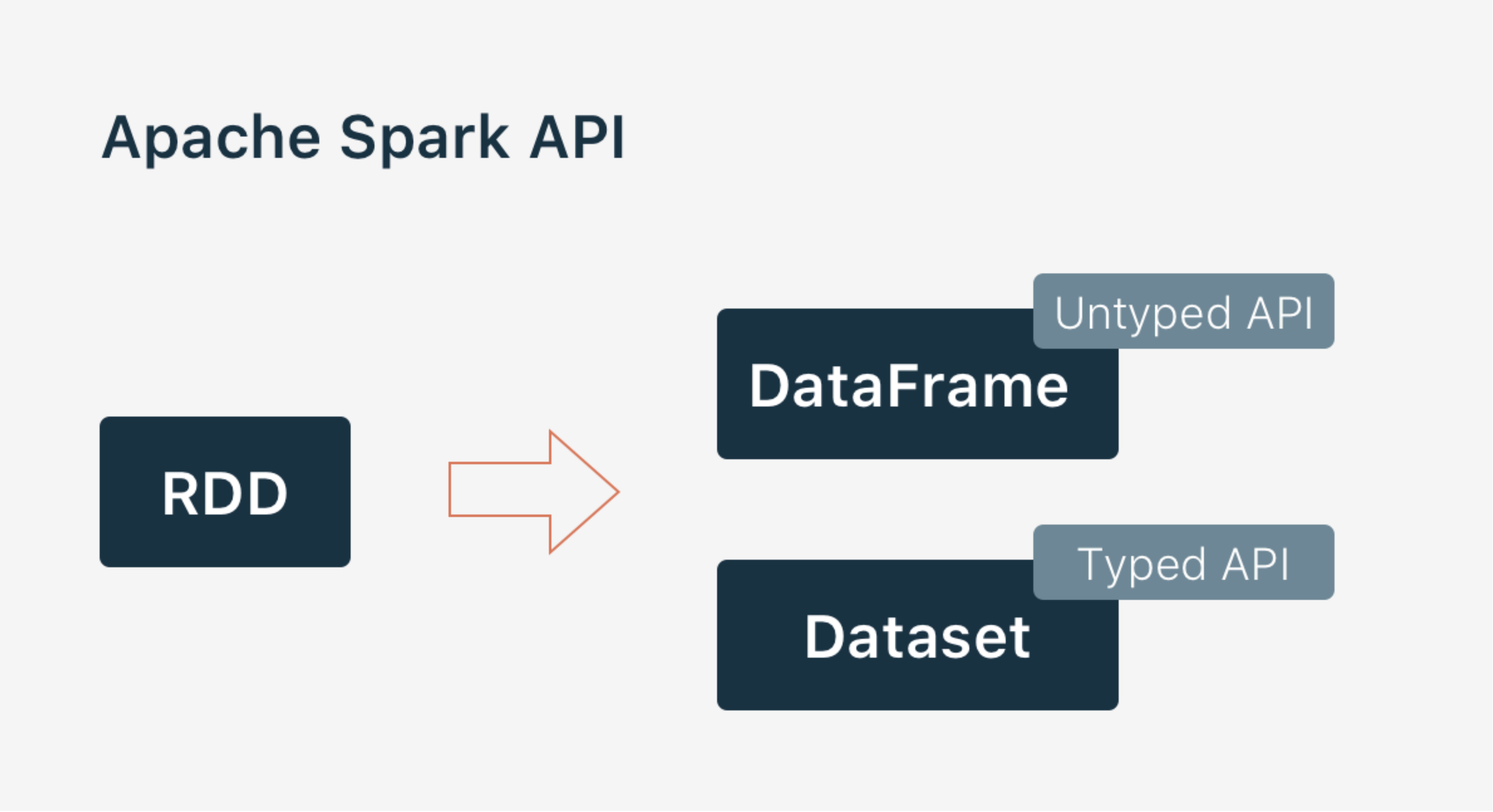




Time to aggregate 10 million integer pairs (in seconds)

SparkSQL: Dataset

- Dataset[T]
  - All benefits of DataFrame
  - DataFrame = Dataset[Row]
  - Typed

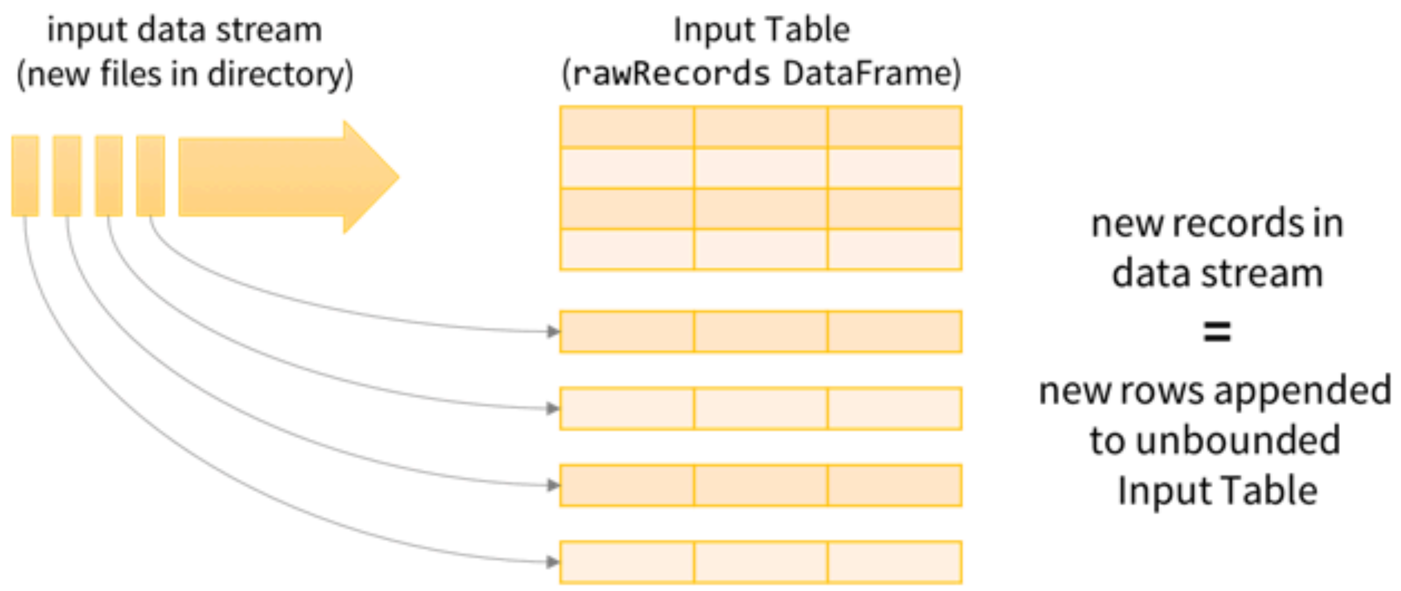


	SQL	DataFrames	Datasets
Syntax Errors	Runtime	Compile Time	Compile Time
Analysis Errors	Runtime	Runtime	Compile Time

demo DataFrame & Dataset

Structured Streaming

- Similar to Dataset / DataFrame processing
- end-to-end exactly once
  - checkpointing
  - write ahead logs
- < 100ms latency
  - or <1 ms in continuous mode

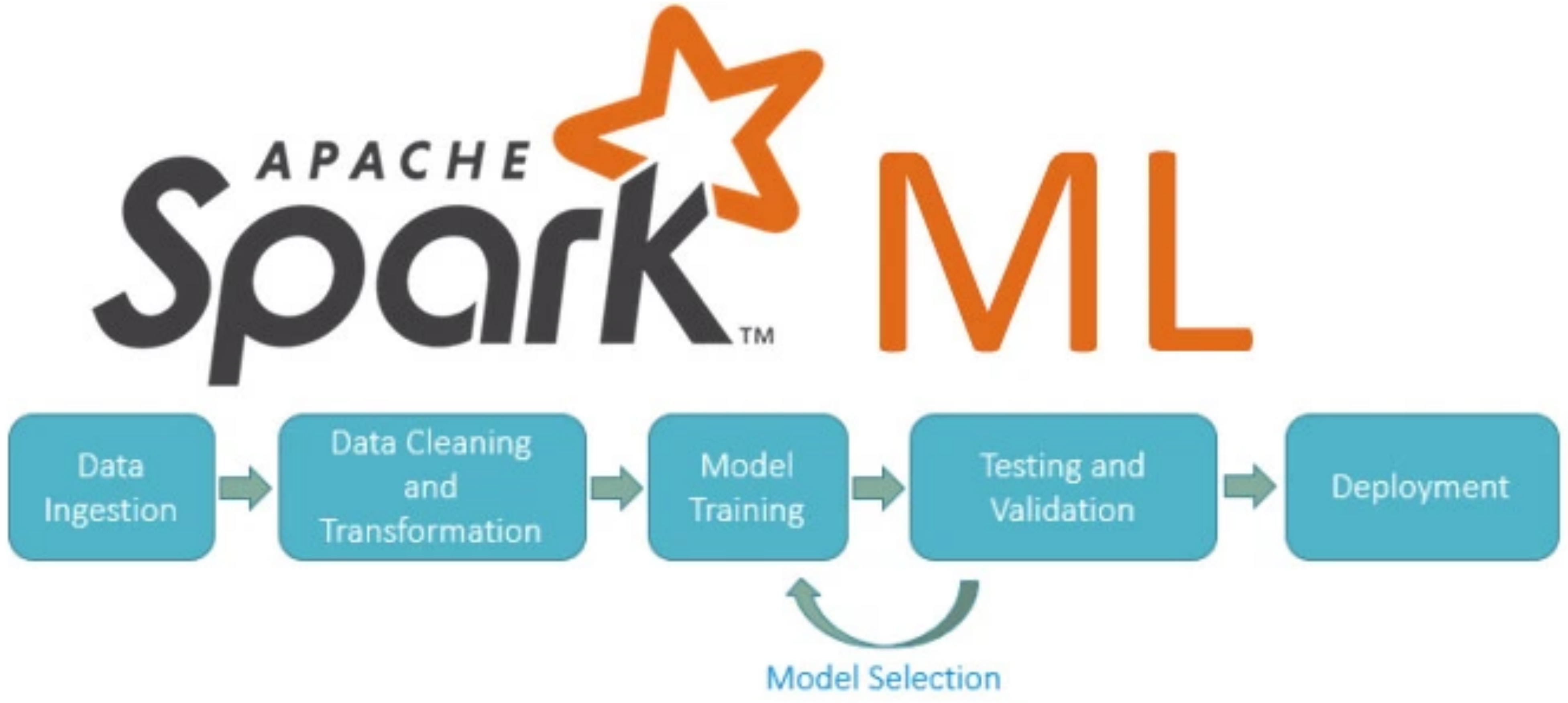


Structured Streaming Model
treat data streams as unbounded tables

demo Structured Streaming

MLlib

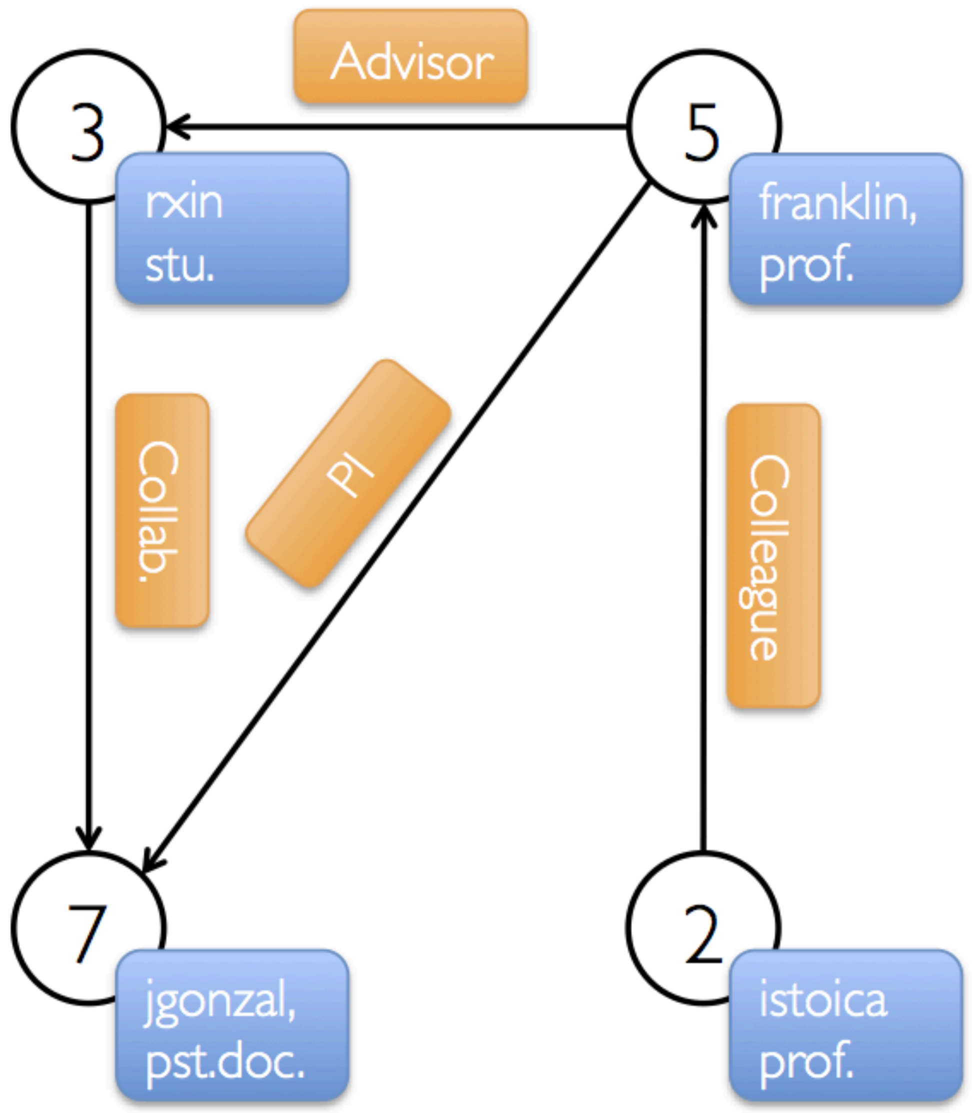
- ML Algorithms
  - classification, regression, clustering, and collaborative filtering
- Featurization
  - feature extraction, transformation, dimensionality reduction, and selection
- Pipelines
- Persistence
- Utilities
  - linear algebra, statistics, data handling



GraphX

- Abstracts over RDD
- Directed multigraph (Vertex / Edge)
- Powerful distributed functions
  - Connected Components
  - Pagerank
  - Triangle Counting
  - Pregel API

Property Graph



Vertex Table

Id	Property (V)
3	(rxin, student)
7	(jgonzal, postdoc)
5	(franklin, professor)
2	(istoica, professor)

Edge Table

SrcId	DstId	Property (E)
3	7	Collaborator
5	3	Advisor
2	5	Colleague
5	7	PI

Deployment

Local

- master=local[n]
- Spark driver is launched in same process
- Workers are launched in same process

Standalone

- master=spark://host
- Spark driver is launched on a cluster machine
- Workers are deployed as separate machines
- submit jar to nodes

demo MLLib & deployment

Done

You can now build your own streaming predictor of stocks :-)