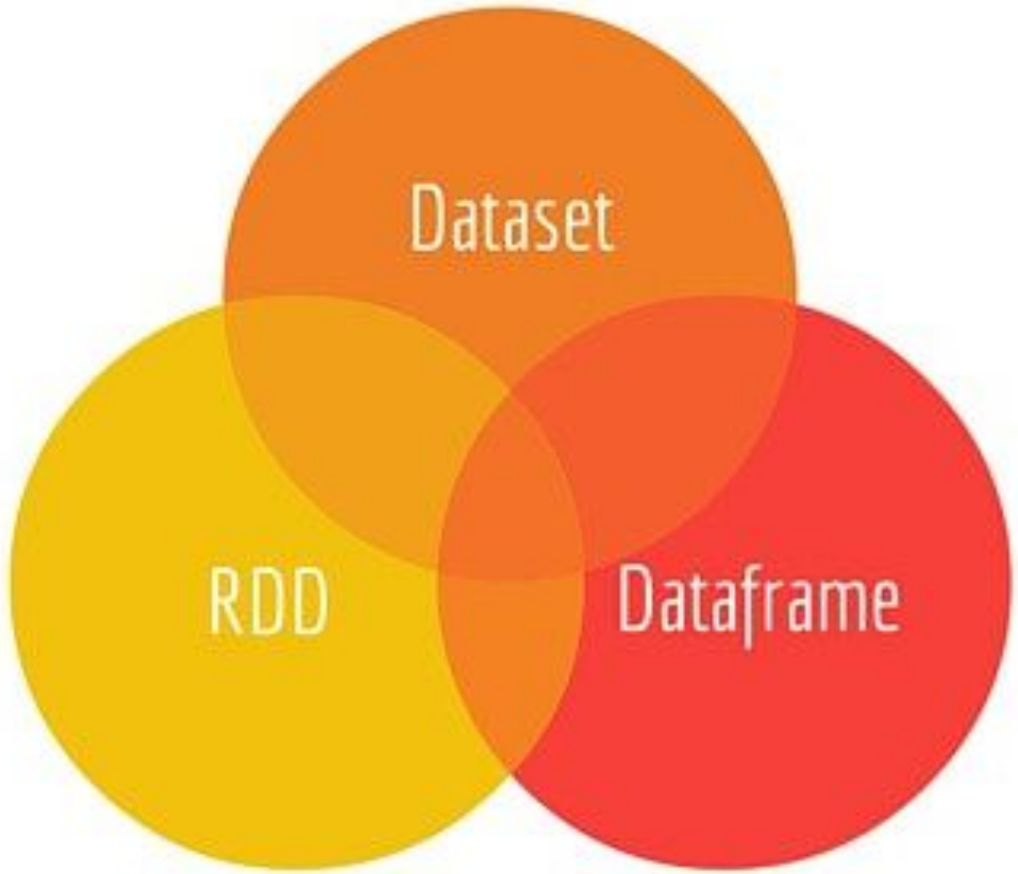

APACHE
Spark™


Dataset
RDD
Dataframe

Distribute collection of JVM objects

Functional Operators (map, filter, etc.)

Distribute collection of Row objects

Expression-based operations and UDFs

Logical plans and optimizer

Fast/efficient internal representations

Internally rows, externally JVM objects

Almost the "Best of both worlds": **type safe + fast**

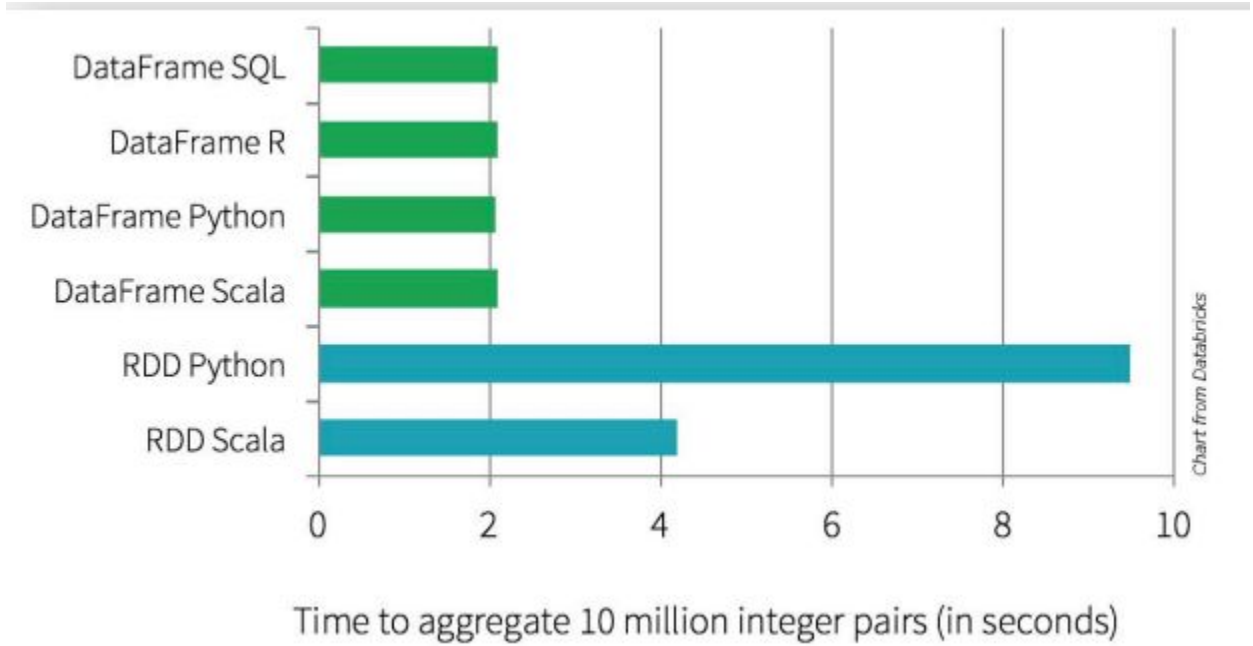
But slower than DF
Not as good for interactive analysis, especially Python

SPARK Interface API

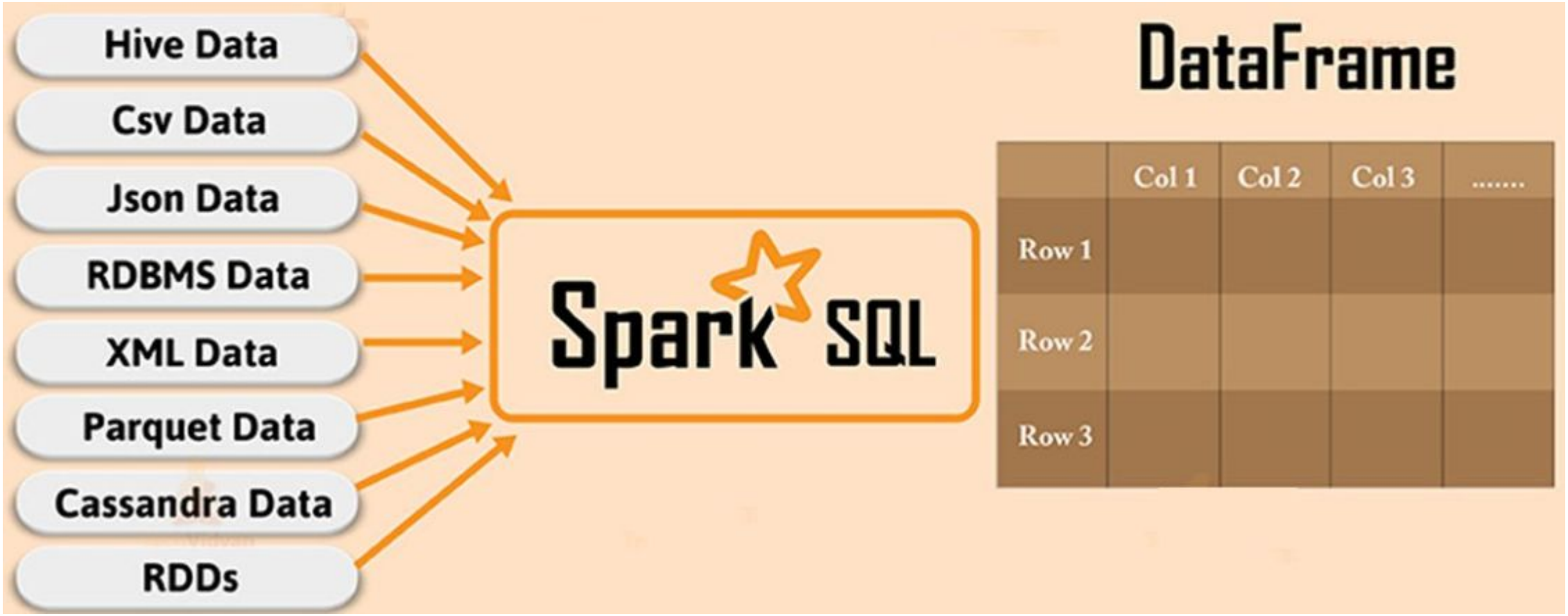
RDD	DataFrame	Dataset
Fault Tolerant	Fault Tolerant	Fault Tolerant
Distributed	Distributed	Distributed
Immutabilty	Immutabilty	Immutabilty
No schema	Schema	Schema
Slow on Non-JVM languages	Faster	Faster
No Execution optimization	optimization Catalyst optimizer	optimization
Low Level	High Level	High Level
No SQL Support	SQL Support	SQL Support
Type Safe	No type Safe	Type Safe
Syntax Error detected at Compile Time	Syntax Error detected at Compile Time	Syntax Error detected at Compile Time
Analysis Error Detected at Compile time	Analysis Error Detected at Run time	Analysis Error Detected at Compile time
JAVA,SCALA, Python,R	JAVA,SCALA, Python,R	JAVA, SCALA
Higher memory is used	Higher memory is used	Low memory is used. Tungsten encoders provide great benefits
We can seamlessly move between DataFrame or Dataset and RDDs by using simple syntax		

```
Schema = student (
            name :String,
            age :Int,
            department : String,
            status : String,
            remark : String
             )
```

```
Data =
student(A,17,DS,Pass,Awesome),
student(B,19,DE,Pass,Excellent),
student(C,17,AE,Pass,Excellent),
student(D,18,BD,Pass,Awesome)
```

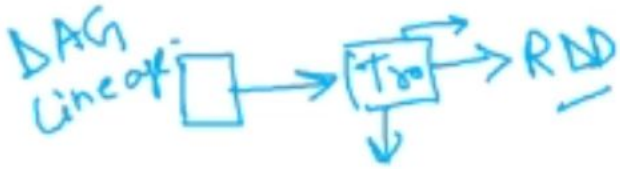

DataFrame SQL
DataFrame R
DataFrame Python
DataFrame Scala
RDD Python
RDD Scala
0
2
4
6
8
10
Chart from Databricks
Time to aggregate 10 million integer pairs (in seconds)

Ways to Create A DATAFRAME

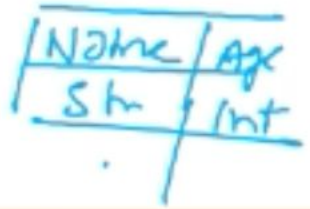


Type Safe

Data Set = DataFrame + RDD



SPARK Interface API



RDD	DataFrame	Dataset
Fault Tolerant	Fault Tolerant	Fault Tolerant
Distributed	Distributed	Distributed
Immutabilty	Immutabilty	Immutabilty
No schema	Schema	Schema
Slow on Non-JVM languages	Faster	Faster
No Execution optimization	optimization Catalyst optimizer	optimization
Low Level	High Level	High Level
No SQL Support	SQL Support	SQL Support
Type Safe	No type Safe	Type Safe
Syntax Error detected at Compile Time	Syntax Error detected at Compile Time	Syntax Error detected at Compile Time
Analysis Error Detected at Compile time	Analysis Error Detected at Run time	Analysis Error Detected at Compile time
JAVA,SCALA, Python,R	JAVA,SCALA, Python,R	JAVA, SCALA
Higher memory is used	Higher memory is used	Low memory is used. Tungsten encoders provide great benefits
We can seamlessly move between DataFrame or Dataset and RDDs by using simple syntax		