Spark UI, Optimizer & Internal concepts (PhysicalPlan, Jobs, Stages, Tasks)

arnaud.nauwynck@gmail.com

This document:
https://github.com/Arnaud-Nauwynck/presentations/
/pres-bigdata/12-sparkui-internals-plan-job-stage-task

Outline

What’s visible from Spark-ui

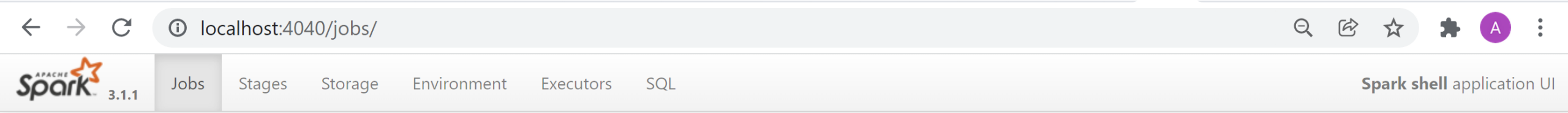
The concepts behinds Dataset/RDD execution : Job, Stage, Task

The architecture of deployment : Client, Driver, Executor

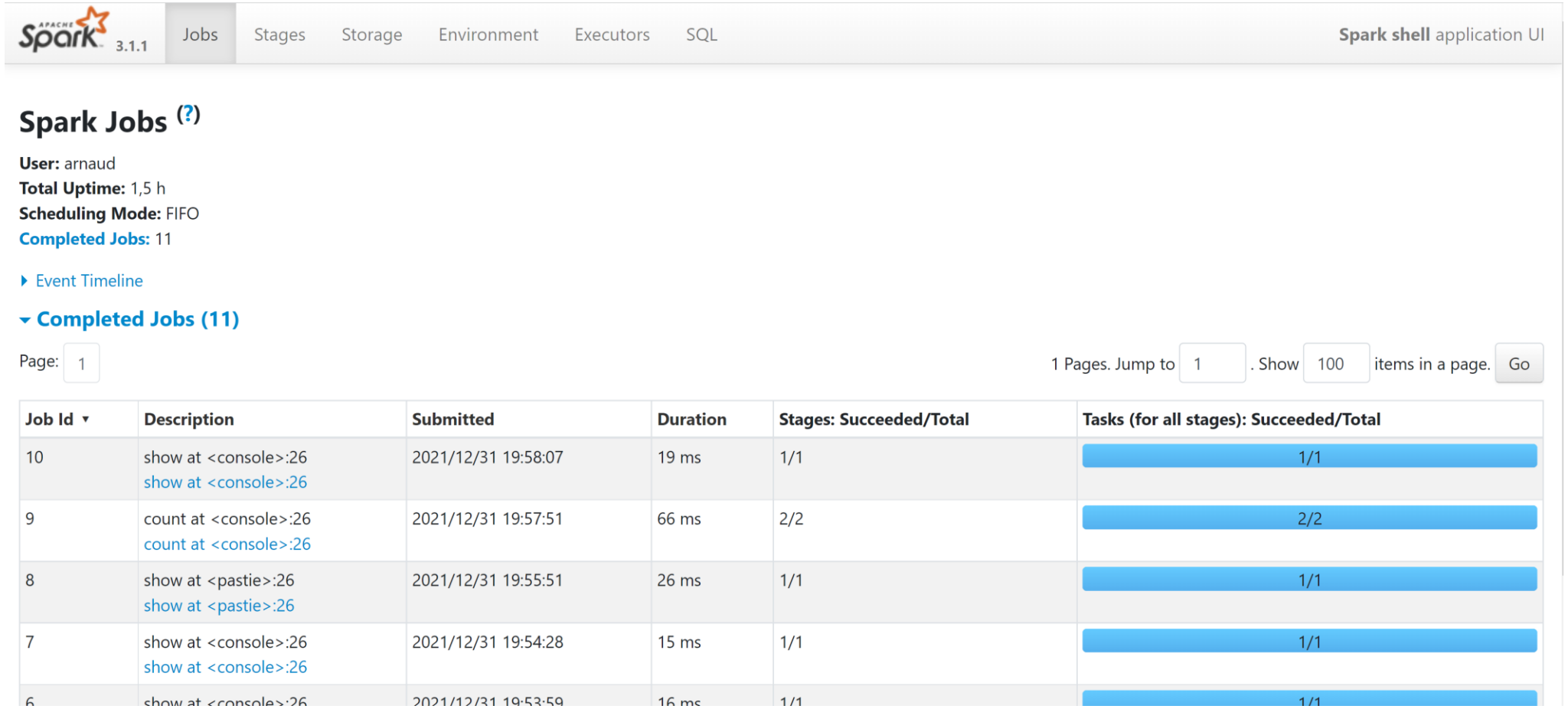
Spark-ui

```
Spark context Web UI available at http://DESKTOP-2EGCC8R:4040
Spark context available as 'sc' (master = local[*], app id = l
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/
```

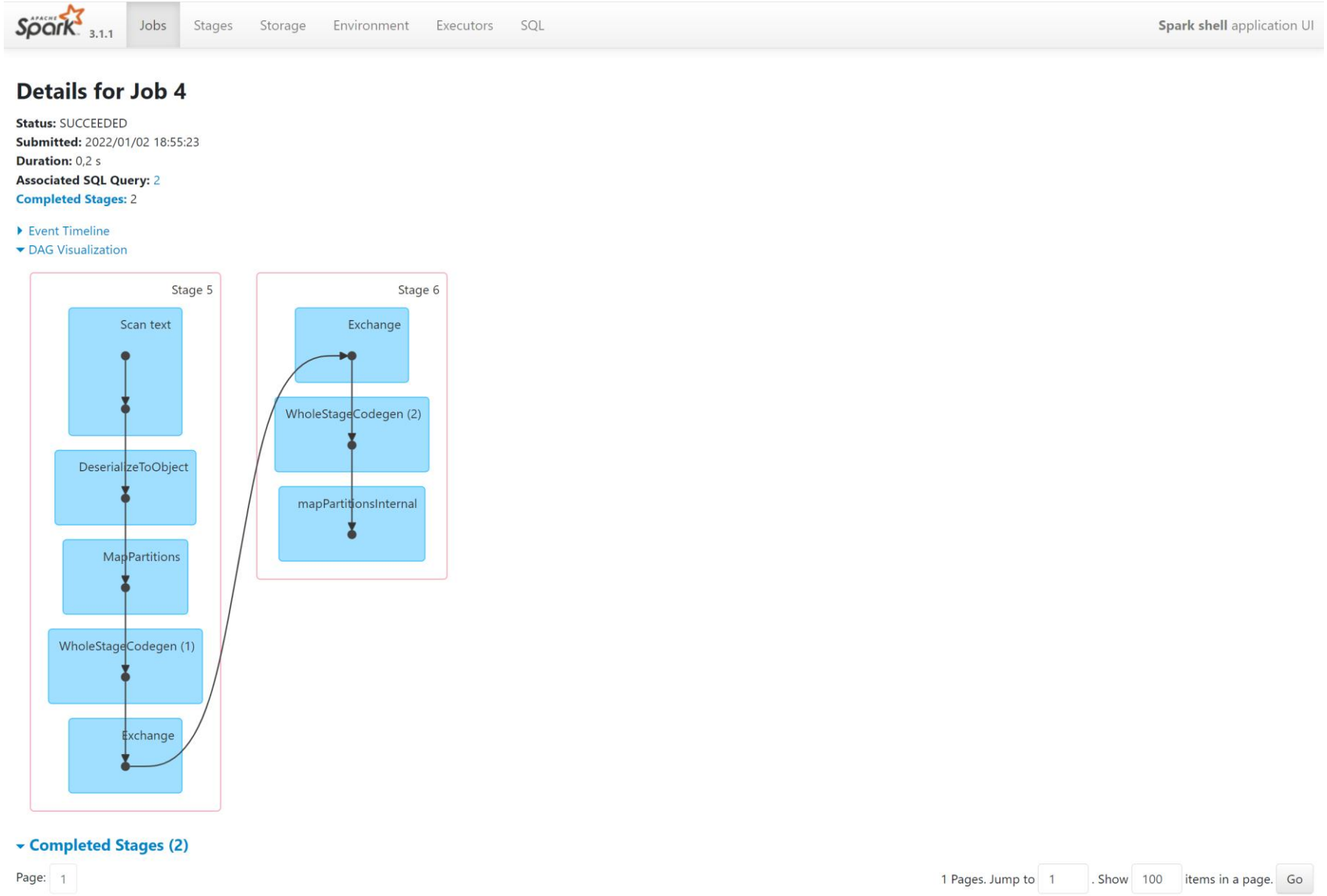
http://localhost:4040



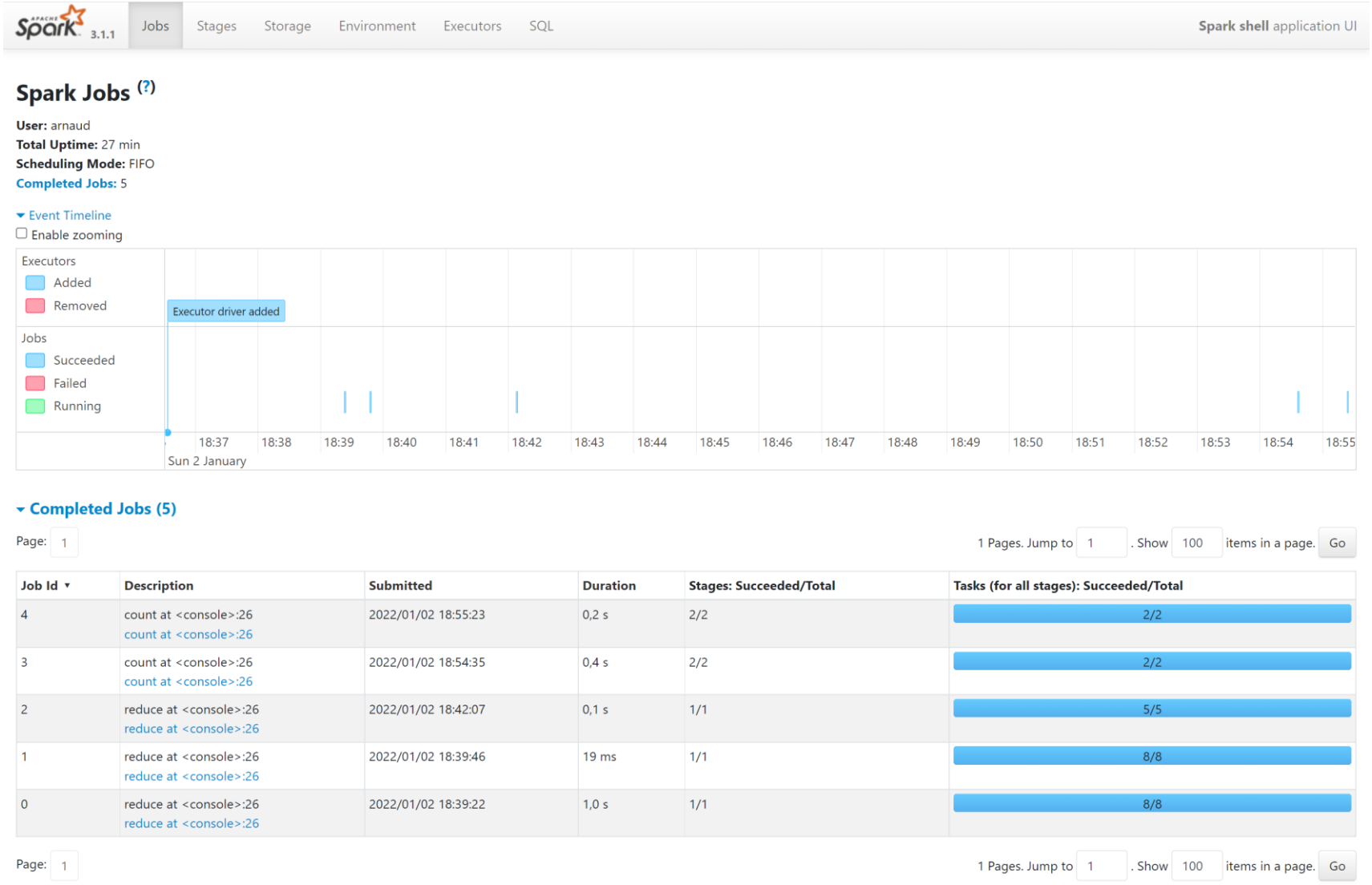
spark-ui [1/6]: Jobs



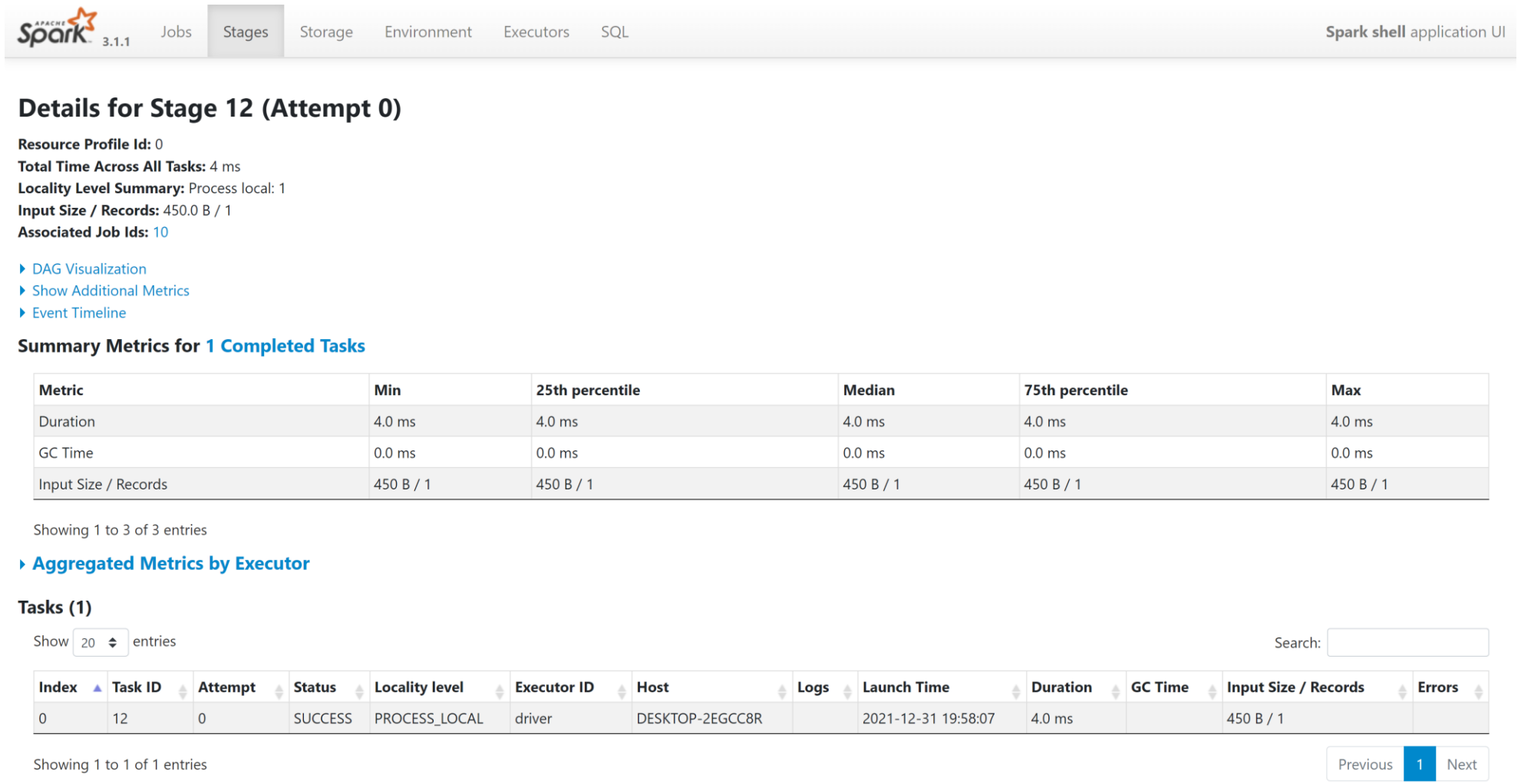
Spark-ui [1/6]: Job details DAG



Spark-ui [1/6] detail Job timeline + statistics



Spark-ui [2/6]: detail Job > Stage > Task



Spark-ui [3/6] Storage (cache, persist)

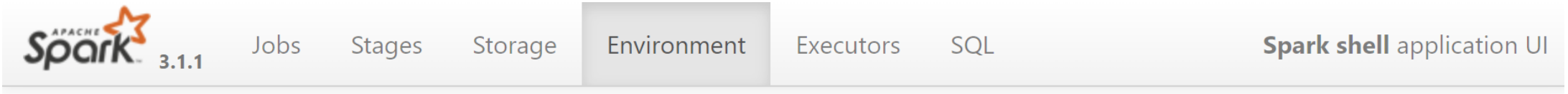
Spark 3.1.1 Jobs Stages Storage Environment Executors SQL **Spark shell** application UI

Storage

RDDs

ID	RDD Name	Storage Level	Cached Partitions	Fraction Cached	Size in Memory	Size on Disk
28	*(1) SerializeFromObject [staticinvoke(class org.apache.spark.unsafe.types.UTF8String, StringType, fromString, input[0, java.lang.String, true], true, false) AS value#30] +- MapPartitions org.apache.spark.sql.Dataset$$Lambda$4450/17435982@68a3d4, obj#29: java.lang.String +- DeserializeToObject value#21.toString, obj#28: java.lang.String +- FileScan text [value#21] Batched: false, DataFilters: [], Format: Text, Location: InMemoryFileIndex[file:/c:/data/loremIpsum.txt], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<value:string>	Disk Memory Deserialized 1x Replicated	1	100%	952.0 B	0.0 B
40	FileScan text [value#21] Batched: false, DataFilters: [], Format: Text, Location: InMemoryFileIndex[file:/c:/data/loremIpsum.txt], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<value:string>	Disk Memory Deserialized 1x Replicated	1	100%	960.0 B	0.0 B

Spark-ui [4/6] Environment: jvm+spark+hadoop+..



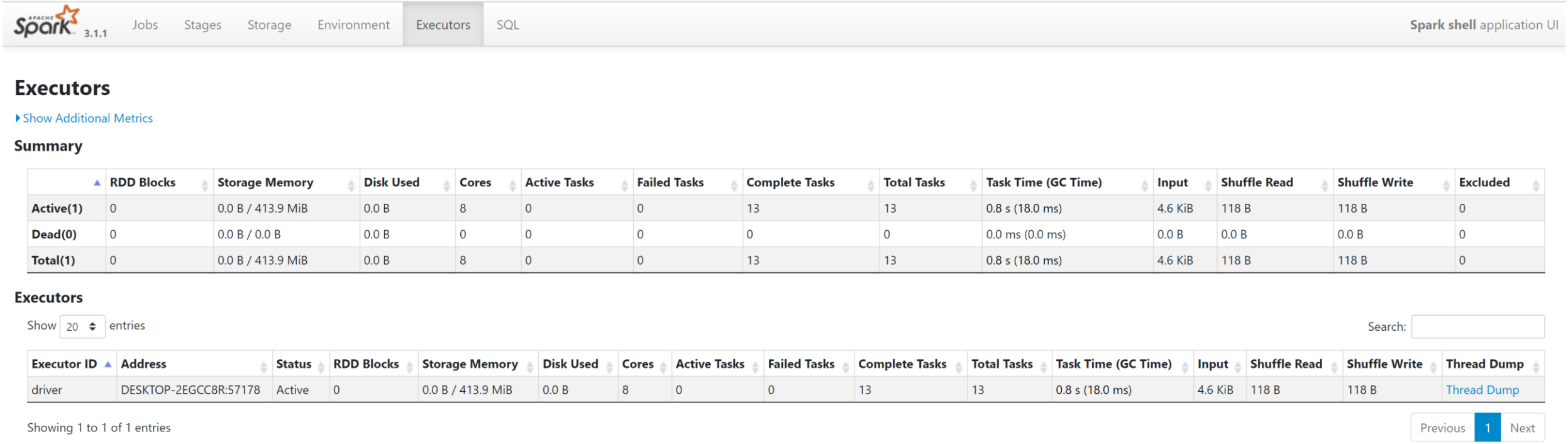
Environment

- Runtime Information
- Spark Properties
- Resource Profiles
- Hadoop Properties
- System Properties
- Classpath Entries

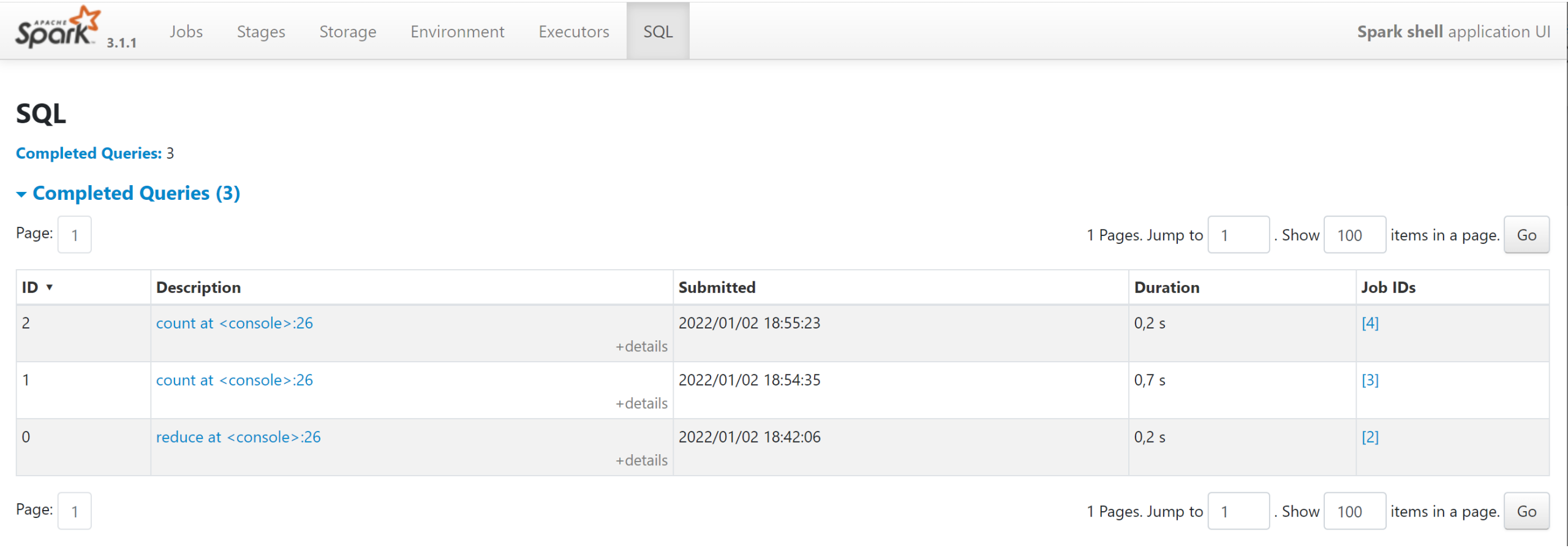
▾ Spark Properties

Name	Value
spark.app.id	local-1640975191762
spark.app.name	Spark shell
spark.app.startTime	1640975189843
spark.driver.host	DESKTOP-2EGCC8R
spark.driver.port	57122
spark.executor.id	driver
spark.home	C:\apps\hadoop\spark-3.1.1
spark.jars	
spark.master	local[*]
spark.repl.class.outputDir	C:\Users\arnaud\AppData\Local\Temp\spark-169a28ae-4479-4585-ab47-1e354d2d1347\repl-ff2cda14-38ec-4b1e-9c63-c4f02e84bf14
spark.repl.class.uri	spark://DESKTOP-2EGCC8R:57122/classes
spark.scheduler.mode	FIFO
spark.sql.catalogImplementation	hive
spark.submit.deployMode	client
spark.submit.pyFiles	
spark.ui.showConsoleProgress	true

Spark-ui [5/6]: Executors / Driver



Spark-ui [6/6] SQL



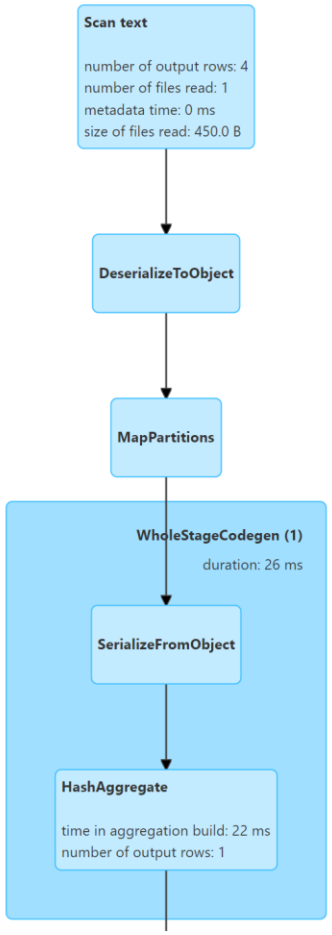
Spark-ui [6/6] Sql detail for query

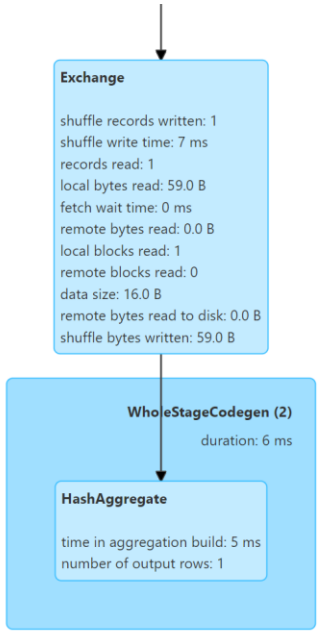
Spark 3.1.1 | Jobs | Stages | Storage | Environment | Executors | SQL

Details for Query 2

Submitted Time: 2022/01/02 18:55:23
Duration: 0,2 s
Succeeded Jobs: 4

☐ Show the Stage ID and Task ID that corresponds to the max metric





▾ Details

```
== Physical Plan ==
* HashAggregate (7)
+- Exchange (6)
   +- * HashAggregate (5)
      +- * SerializeFromObject (4)
         +- MapPartitions (3)
            +- DeserializeToObject (2)
               +- Scan text  (1)


(1) Scan text
Output [1]: [value#21]
Batched: false
Location: InMemoryFileIndex [file:/c:/data/loremIpsum.txt]
ReadSchema: struct<value:string>

(2) DeserializeToObject
Input [1]: [value#21]
Arguments: value#21.toString, obj#28: java.lang.String

(3) MapPartitions
Input [1]: [obj#28]
Arguments: org.apache.spark.sql.Dataset$$Lambda$4450/17435982@68a3d4, obj#29: java.lang.String
```

```
(4) SerializeFromObject [codegen id : 1]
Input [1]: [obj#29]

(5) HashAggregate [codegen id : 1]
Input: []
Keys: []
Functions [1]: [partial_count(1)]
Aggregate Attributes [1]: [count#35L]
Results [1]: [count#36L]

(6) Exchange
Input [1]: [count#36L]
Arguments: SinglePartition, ENSURE_REQUIREMENTS, [id=#55]

(7) HashAggregate [codegen id : 2]
Input [1]: [count#36L]
Keys: []
Functions [1]: [count(1)]
Aggregate Attributes [1]: [count(1)#32L]
Results [1]: [count(1)#32L AS count#33L]
```

Spark Internals

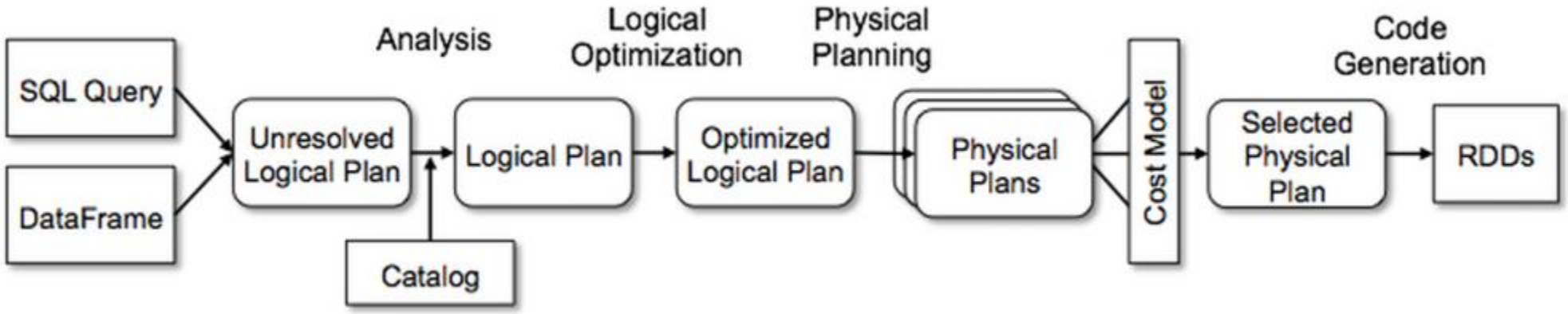
Logical/PhysicalPlan

Jobs - Stage - Tasks

Sql -> parsed AST -> Dataset + expression API

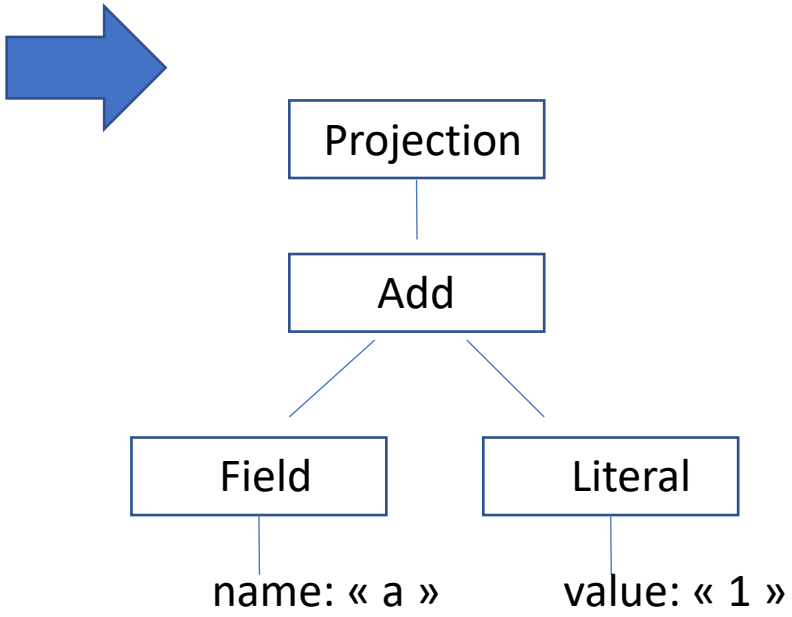
Optimisation: re-write
from LogicalPlan to PhysicalPlan

SQL / DataSet -> AST -> Catalyst -> Code + RDD



Compiler chain: Parser -> AST -> ..

Sql> select **(a+1)** from …




```scala
case class Add(left: Expression, right: Expression) extends BinaryArithmetic {
  override def symbol: String = "+"

  lazy val numeric = dataType match {
    case n: NumericType => n.numeric.asInstanceOf[Numeric[Any]]
    case other => sys.error(s"Type $other does not support numeric operations")
  }

  override def eval(input: Row): Any = {
    val evalE1 = left.eval(input)
    if(evalE1 == null) {
      null
    } else {
      val evalE2 = right.eval(input)
      if (evalE2 == null) {
        null
      } else {
        numeric.plus(evalE1, evalE2)
      }
    }
  }
}
```

Catalyst AST -> CodeGenerator

apache / spark Public

Watch 2.1k | Fork 25k | Star 31.7k

Code | Pull requests 241 | Actions | Projects | Security | Insights

fedbfc7074 spark / sql / catalyst / src / main / scala / org / apache / spark / sql / catalyst / expressions / codegen / CodeGenerator.scala

Go to file ...

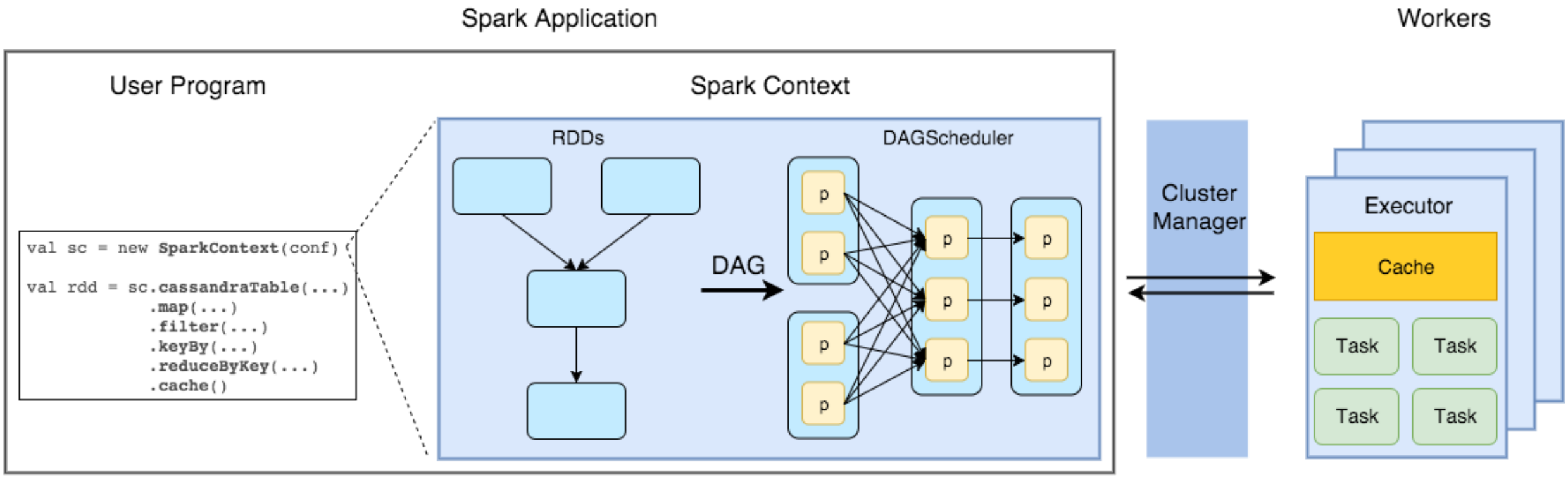
```
33  /**
34   * A base class for generators of byte code to perform expression evaluation.  Includes a set of
35   * helpers for referring to Catalyst types and building trees that perform evaluation of individual
36   * expressions.
37   */
38  abstract class CodeGenerator[InType <: AnyRef, OutType <: AnyRef] extends Logging {
```

```
363         case Add(e1, e2) =>      (e1, e2) evaluate { case (eval1, eval2) => q"$eval1 + $eval2" }
364         case Subtract(e1, e2) => (e1, e2) evaluate { case (eval1, eval2) => q"$eval1 - $eval2" }
365         case Multiply(e1, e2) => (e1, e2) evaluate { case (eval1, eval2) => q"$eval1 * $eval2" }
```

```
640     }
641   }
```

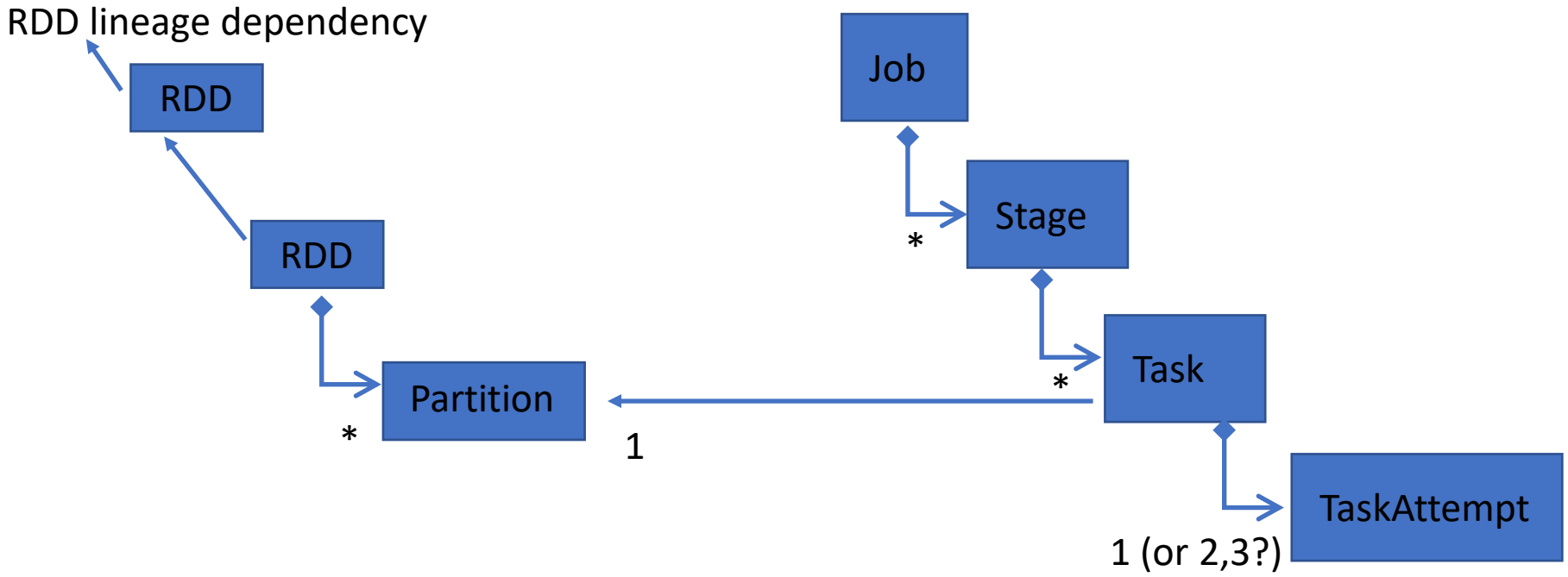
Runtime Execution:

Dataset -> RDD -> Dag Scheduler
-> Jobs,Stages,Tasks

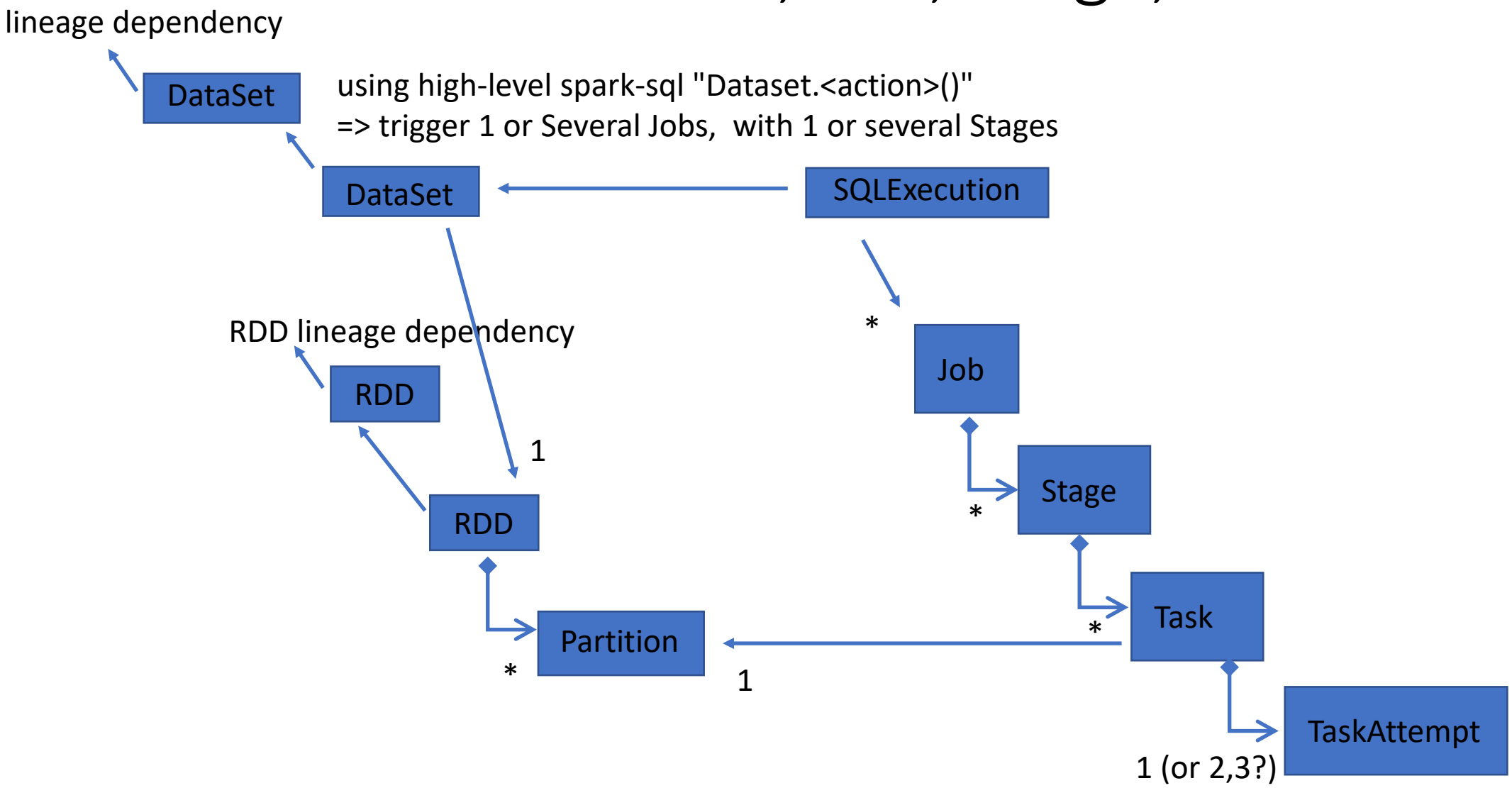

Spark Application
Workers
User Program
Spark Context
val sc = new SparkContext(conf)
val rdd = sc.cassandraTable(...)
.map(...)
.filter(...)
.keyBy(...)
.reduceByKey(...)
.cache()
RDDs
DAGScheduler
DAG
p
Cluster Manager
Executor
Cache
Task
Task
Task
Task

Spark-core Internal Concepts…
RDD, Job, Stage, Task

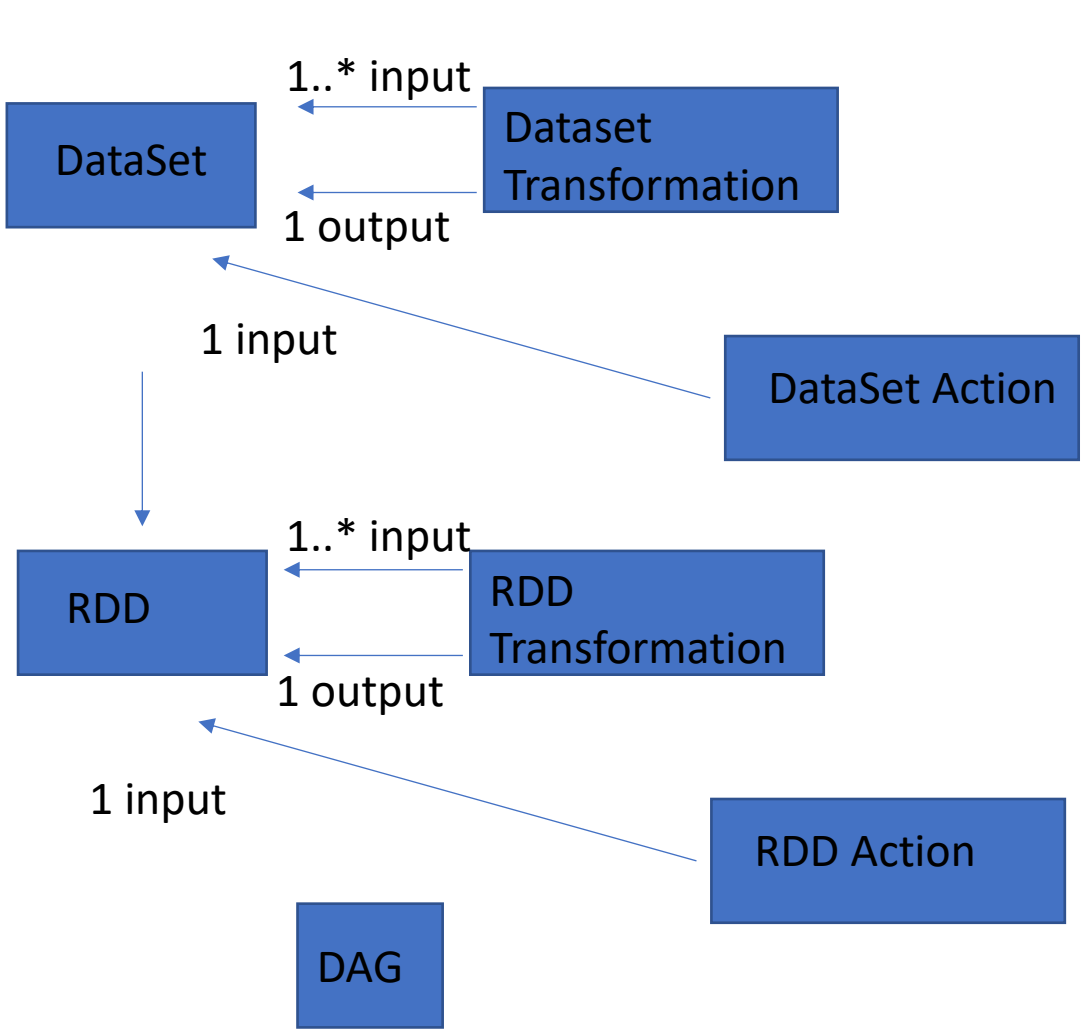
using only low-level spark-core "RDD.eval()"
=> trigger 1 Job, with 1 or several Stages

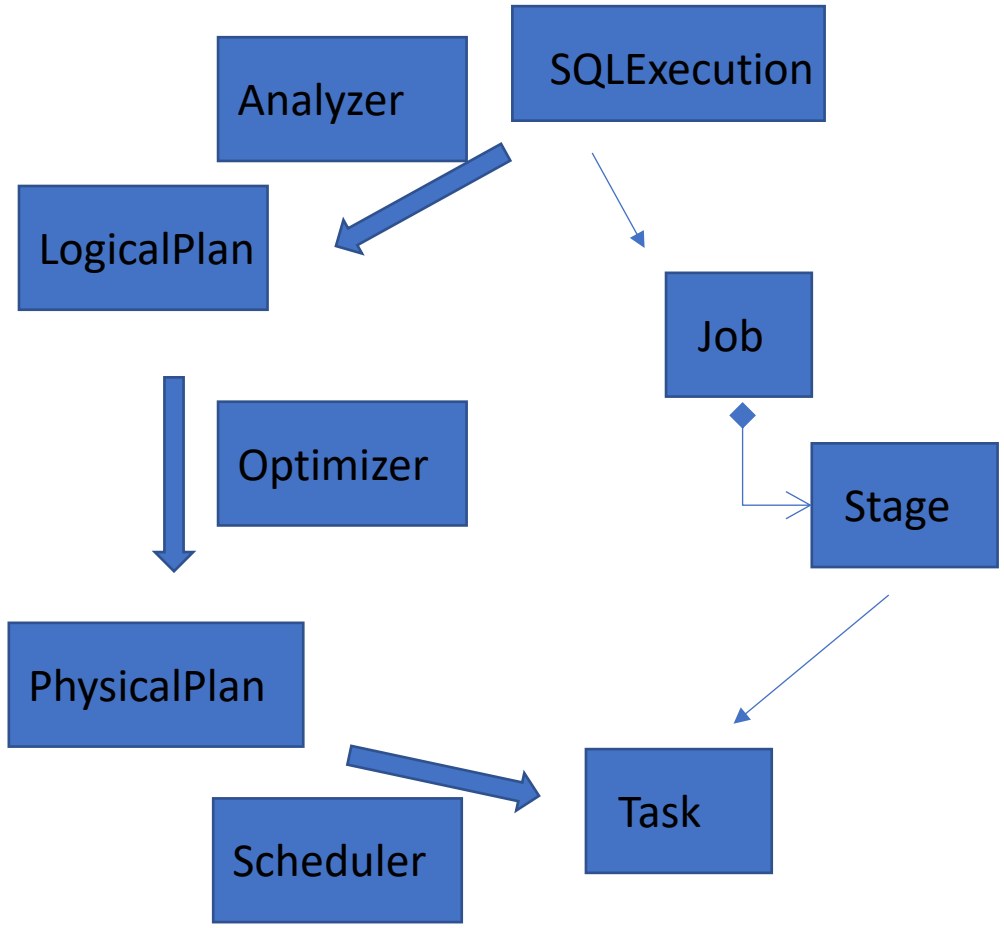


Spark-sql Internal Concepts… SQLExecution, Job, Stage, Task



Spark Internal Concepts…
SQL, Job, Stage, Task

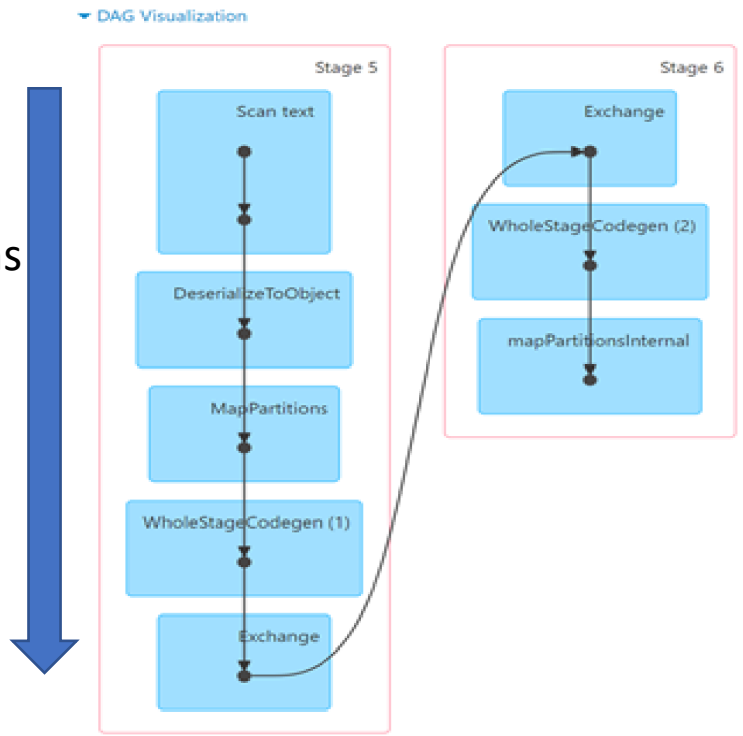




DAG Job = list of Stages
each Stage = N "narrow" transformations
between Stages = Exchange Shuffle ("Wide" transfo.)

1 stage =
Successive **Narrow** transformations

Bytecode generated
… executed within same
Thread/Process



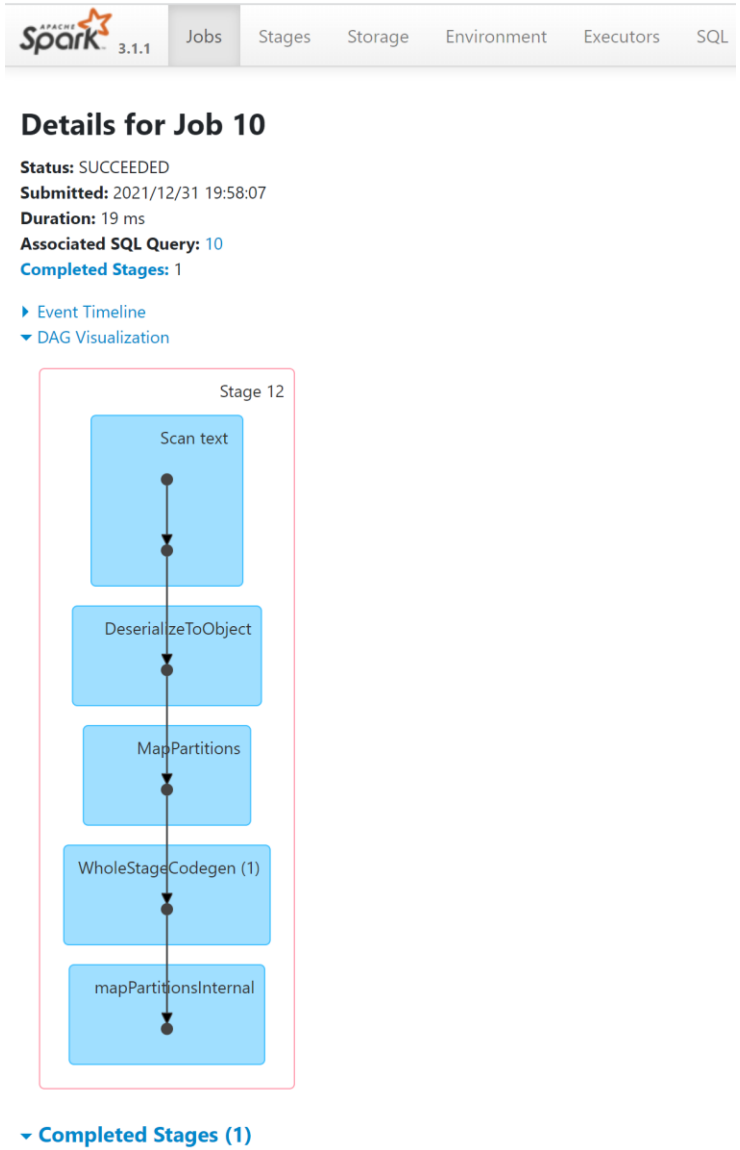
Job =
Full DAG of Stages

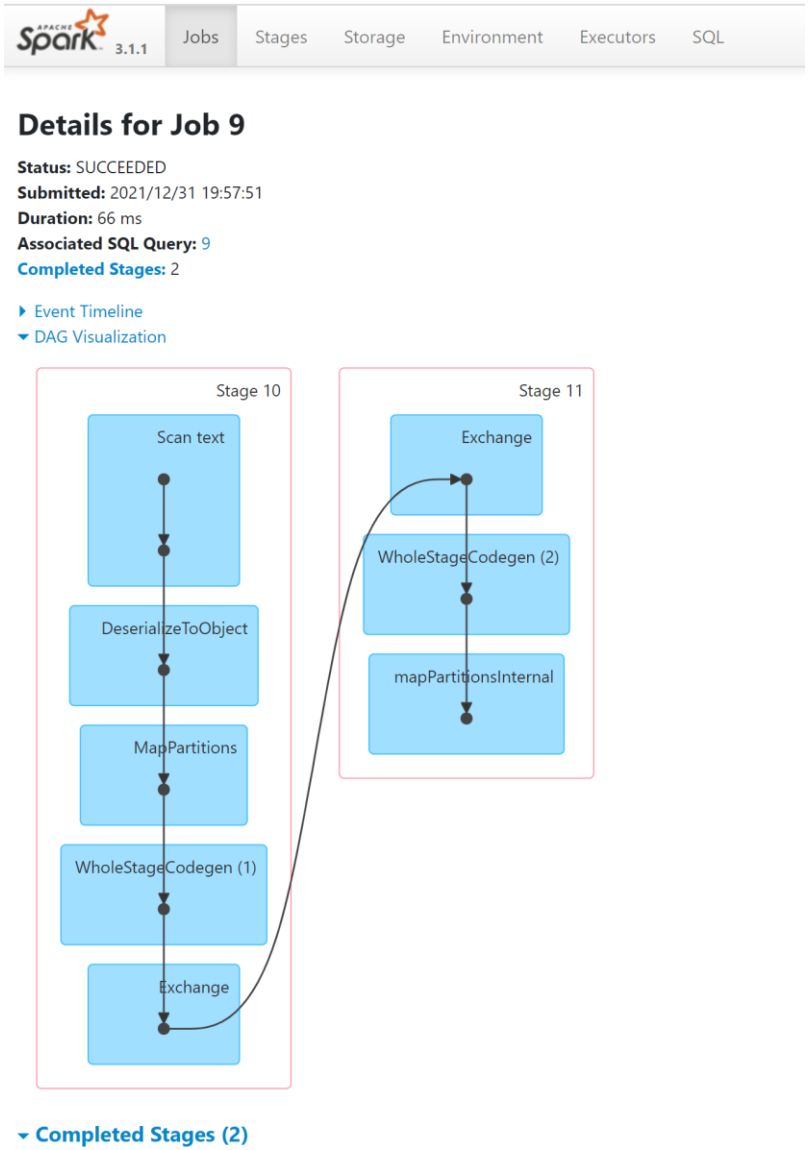
Exchange
Output

Exchange
Input

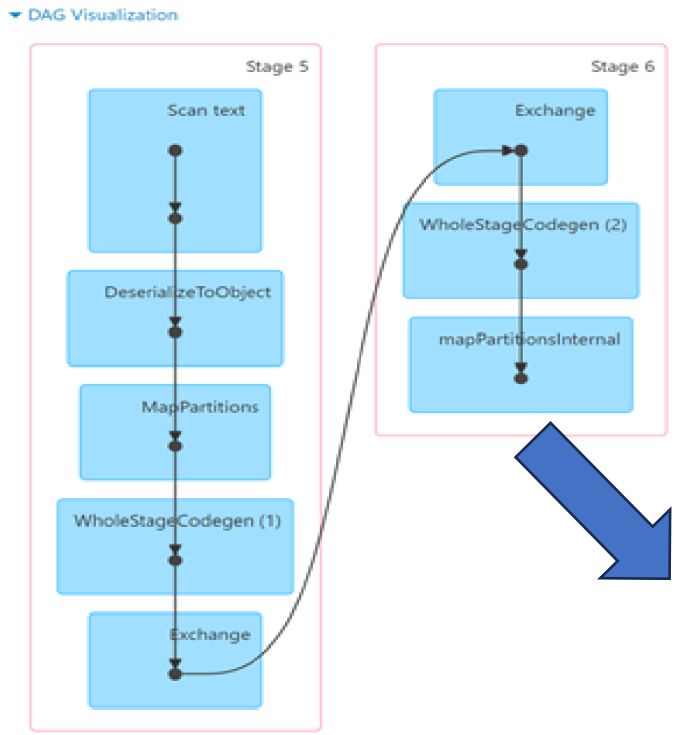
Wide transformation … separation between successive **Stages**

example Word Count DAGs
simple -vs- repartition()

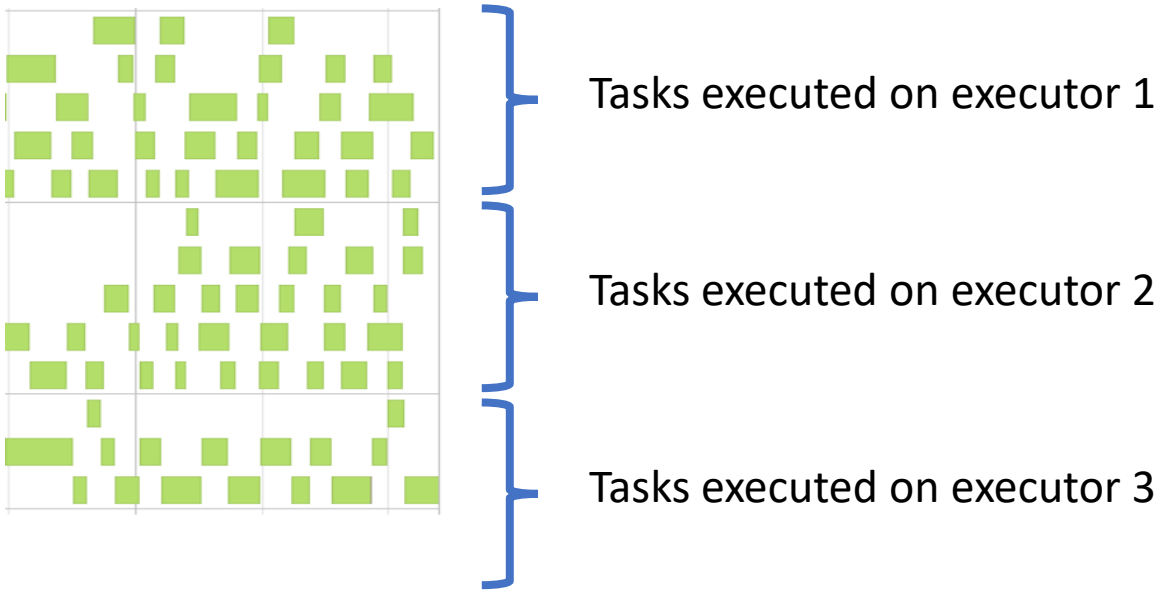




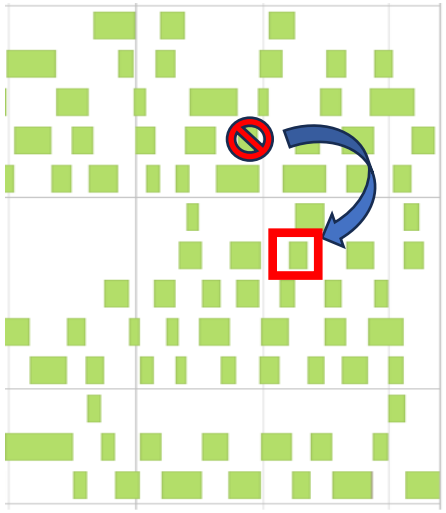
Stage = list of Tasks : 1 per Partition
… each scheduled on an executor



Tasks "**eventTimeline**" for stage



Tasks may fail, Retry individually

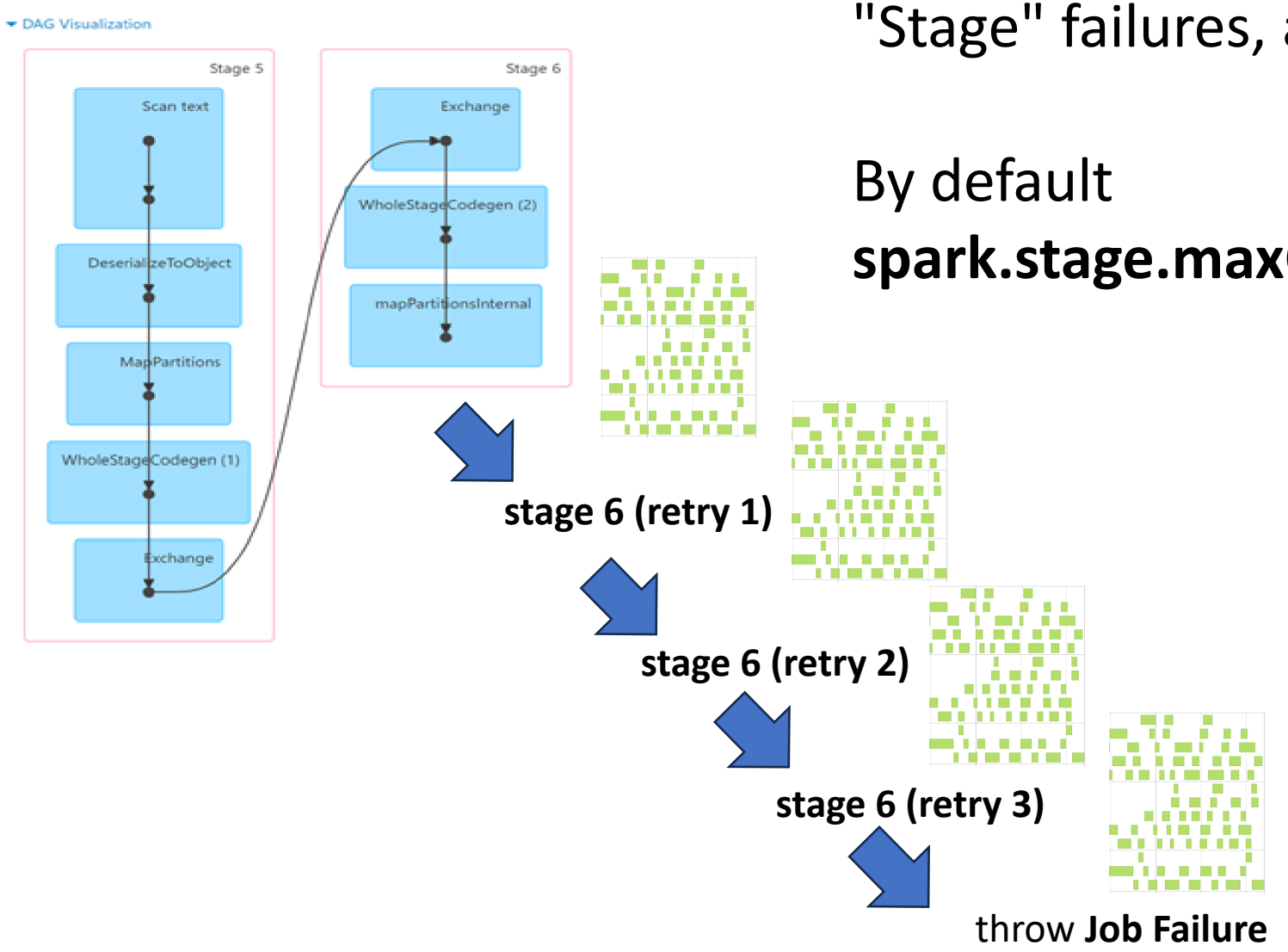


Tasks errors are automatically retryed (rescheduled)

By default,

spark.task.maxFailures = 4

Stage Retry



"Stage" failures, are also retryed

By default

spark.stage.maxConsecutiveAttempts = 4

Architecture Components

Client-Cluster … Driver & Executors

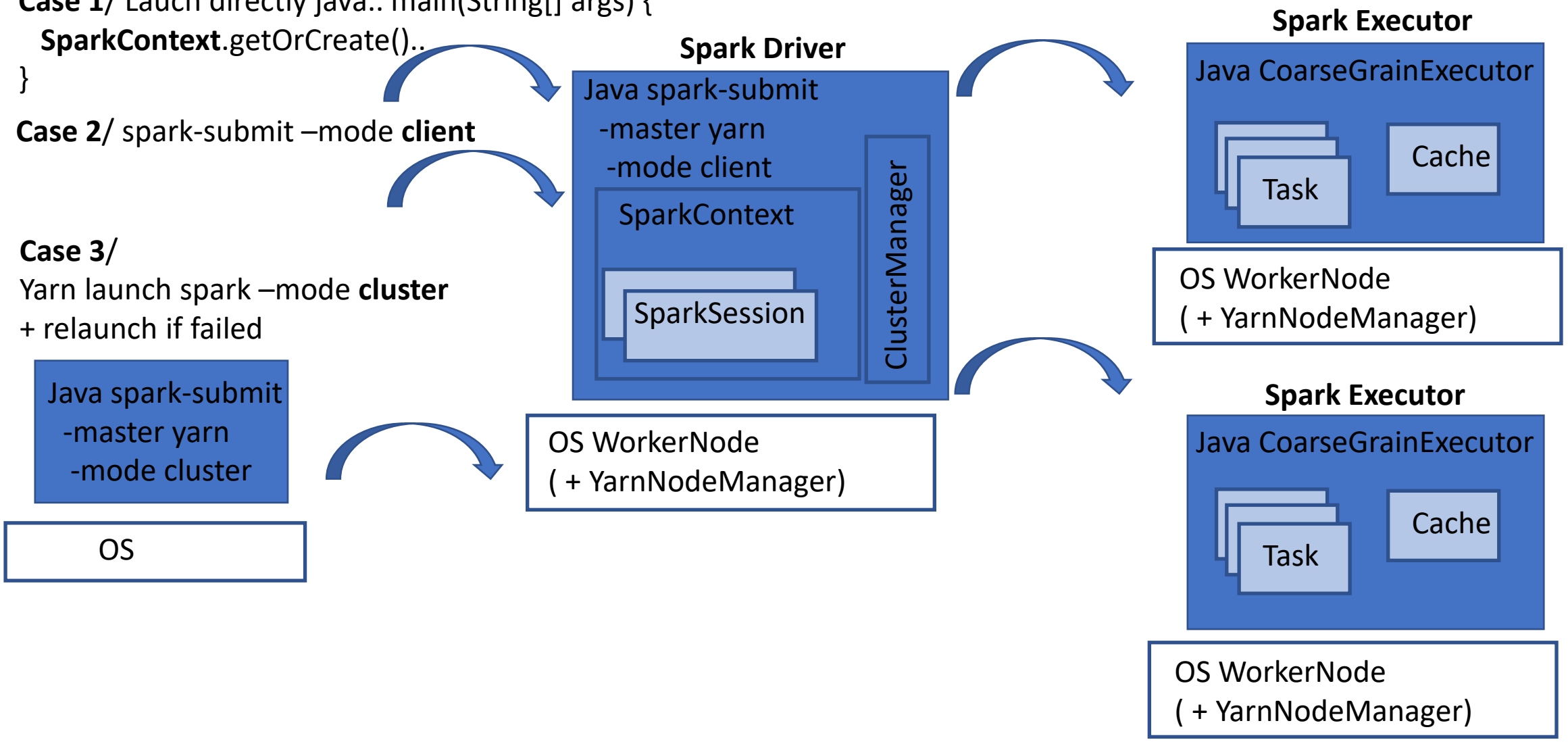
program/library/servers

Cluster Client -> Driver (-> SparkContext) -> Executor

Case 1/ Lauch directly java.. main(String[] args) {
SparkContext.getOrCreate()..
}

Case 2/ spark-submit –mode **client**

Case 3/
Yarn launch spark –mode **cluster**
+ relaunch if failed



Questions ?

… Next Steps ?