Системы анализа больших данных
(САБД)

What is Apache Spark™?

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Ли Ицзя

2023.12

CONTENT

History of Apache Spark

2009

Started as a project at UC Berkley AMPLab

2010

Open sourced under a BSD license

2013

Spark became an Apache top level project

2014

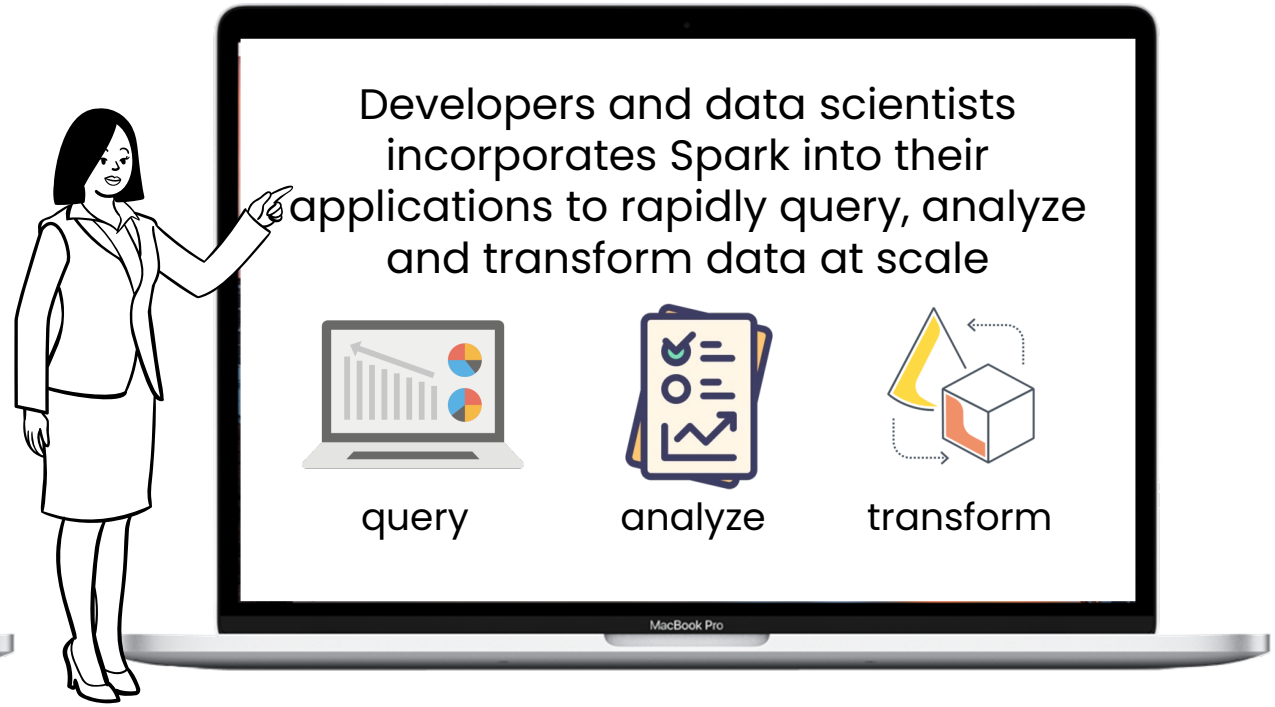
Used by Databricks to sort large-scale datasets and set a new world record

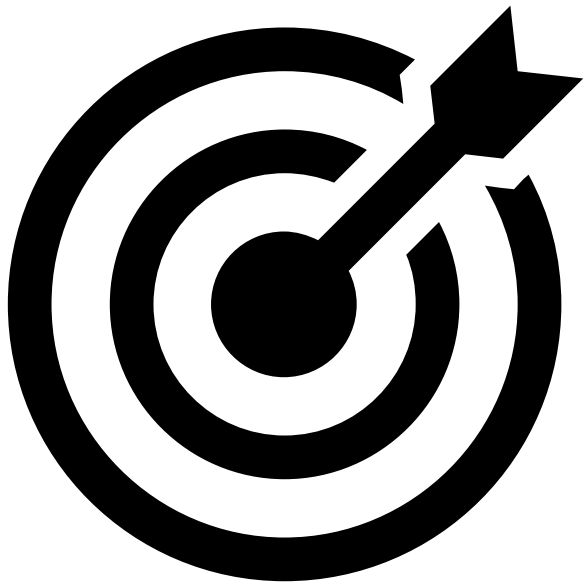
Now

Exists as a nect generation real-time and batch processing frameword

What is Apache Spark?

Apache Spark™ is an open-source data processing engine across clusters of computers using simple programming constructs.





Spark Features

Spark Features

Fast processing

Spark contains Resilient Distributed Dataset (RDD) which saves them time taken in reading, and writing operations and hence, it runs almost ten to hundred times faster than Hadoop

In-memory computing

In Spark, data is stored in the RAM, so it can access the data quickly and accelerate the speed of analytics

Flexible

Spark supports multiple languages and allow the developers to write applications in Java, Scala, R or Python

Fault tolerance

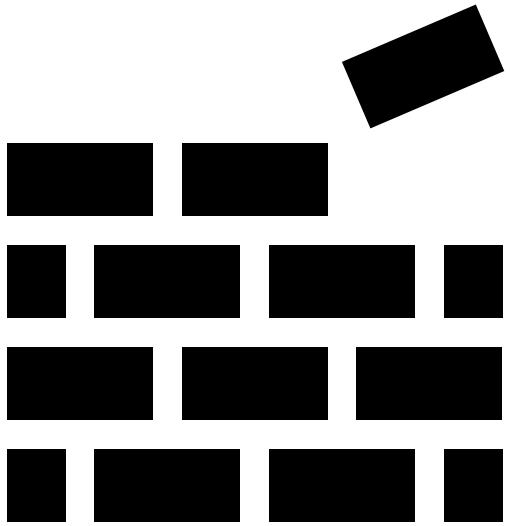
Spark contains Resilient Distributed Dataset (RDD) that are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero

Better analytics

Spark has a rich set of SQL, queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed better

Hadoop vs Spark

Processing data using MapReduce in Hadoop is slow	Spark processes data 100 times faster than MapReduce as it is done in-memory
Hadoop has more lines of code. Since it is written in Java, it takes more time to execute	Spark has fewer lines of code as it is implemented in Scala
Performs batch processing of data	Performs both batch processing and real-time processing of data
Hadoop relies on data replication and checkpointing to ensure fault tolerance	Spark provides fault-tolerance through RDD, which duplicates nothing



Components of Spark

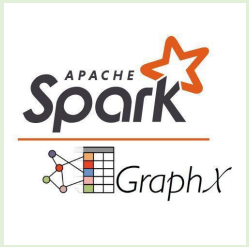
Components of Apache Spark

Spark Core

Spark SQL

Spark Streaming

MLlib



GraphX

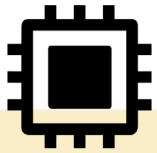
Components of Apache Spark – Spark Core

Spark Core

Spark core is the base engine for large-scale parallel and distributed data processing

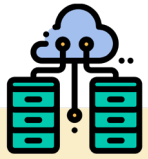
Spark Core

It is responsible for:



Memory management

Fault recovery



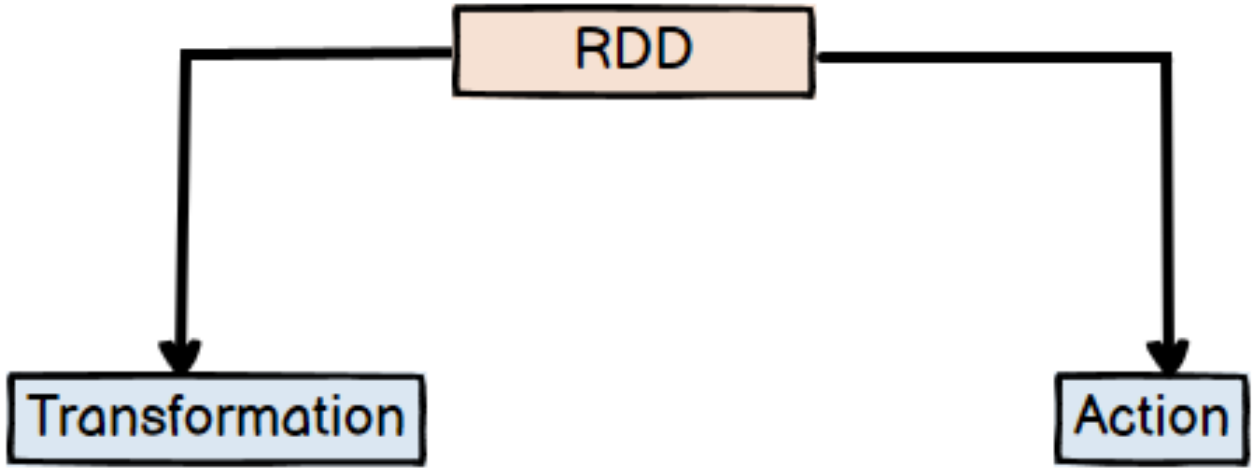
Scheduling, distributing and monitoring jobs on a cluster

Interacting with storage systems

Resilient Distributed Dataset (RDD)

Spark core is embedded with RDDs (Resilient Distributed Dataset), an immutable fault-tolerant, distributed collection of objects that can be operated on in parallel

Spark Core



These are operations (such as *map, filter, union*) that are performed on an RDD that **yields a new RDD** containing the result

These are operations (such as *reduce, first, count*) that **return a value** after running a computation on an RDD

Components of Apache Spark –

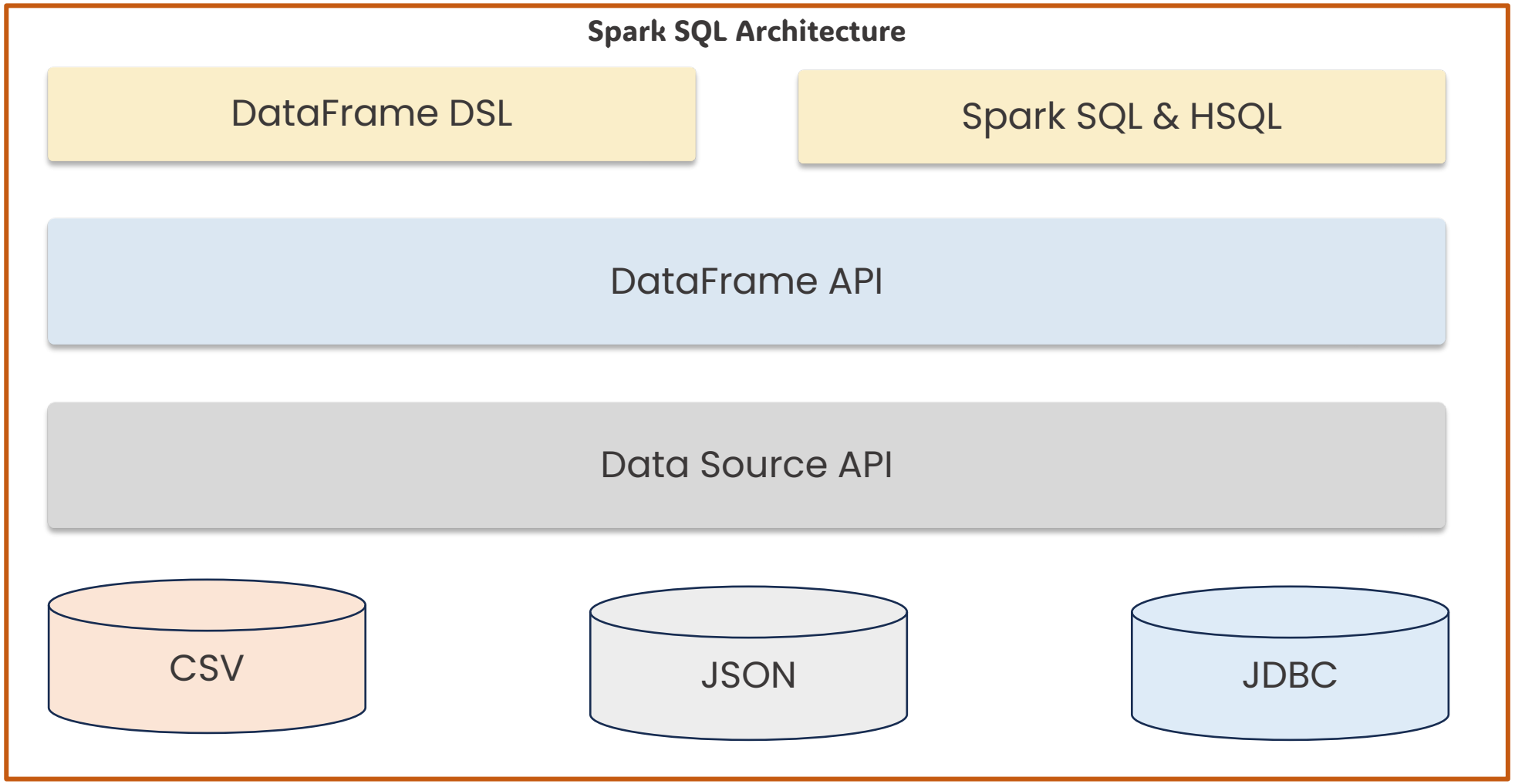
Spark SQL

Spark SQL

Spark SQL framework component is used for structured and semi-structured data processing

Spark SQL

Spark SQL Architecture

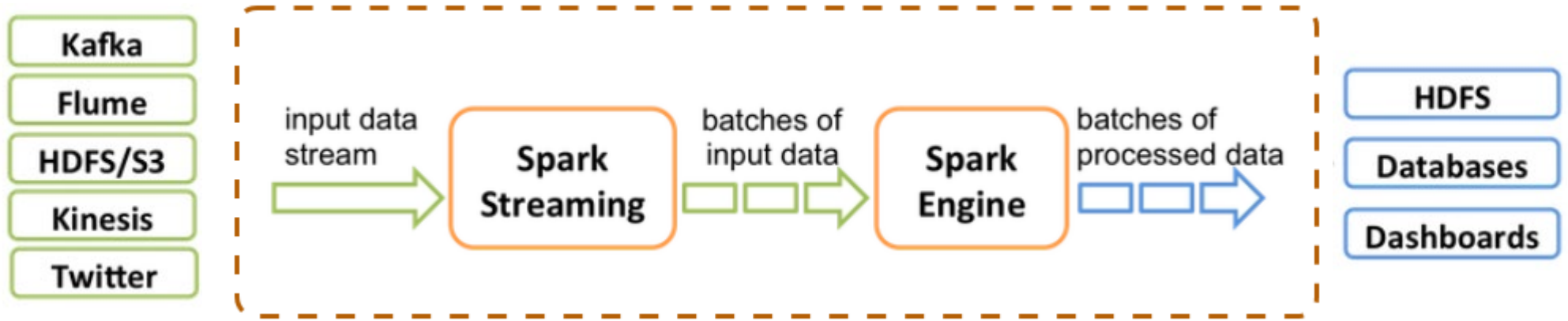


Spark Streaming

Spark Streaming is a lightweight API that allows developers to perform batch processing and real-time streaming of data with ease

Provides secure, reliable, and fast processing of live data streams

Spark Streaming



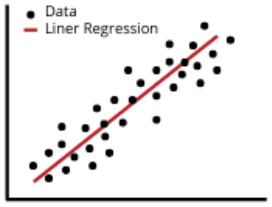
Spark MLlib

Spark Mllib is low-level machine learning library that is simple to use, is scalable, and compativle with various programming languages

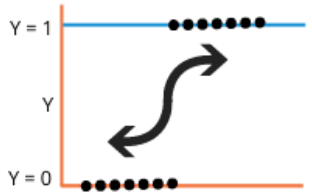
It contains machine learning libraries that have an implementation of various machine learning algorithms

MLlib

Linear Regression

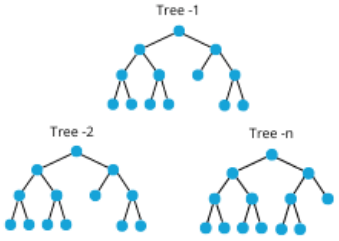


Logistic Regression



Decision Trees

Random Forest



GraphX

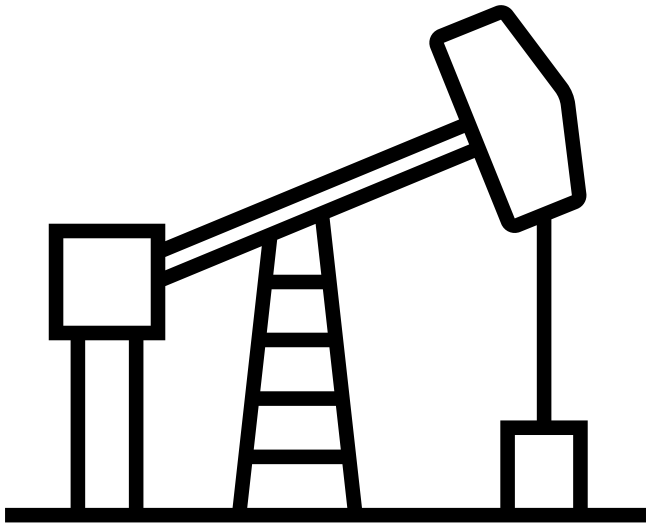
GraphX is Spark's own Graph Computation Engine and data store

Provides a uniform tool for ETL

Exploratory data analysis

Interactive graph computations

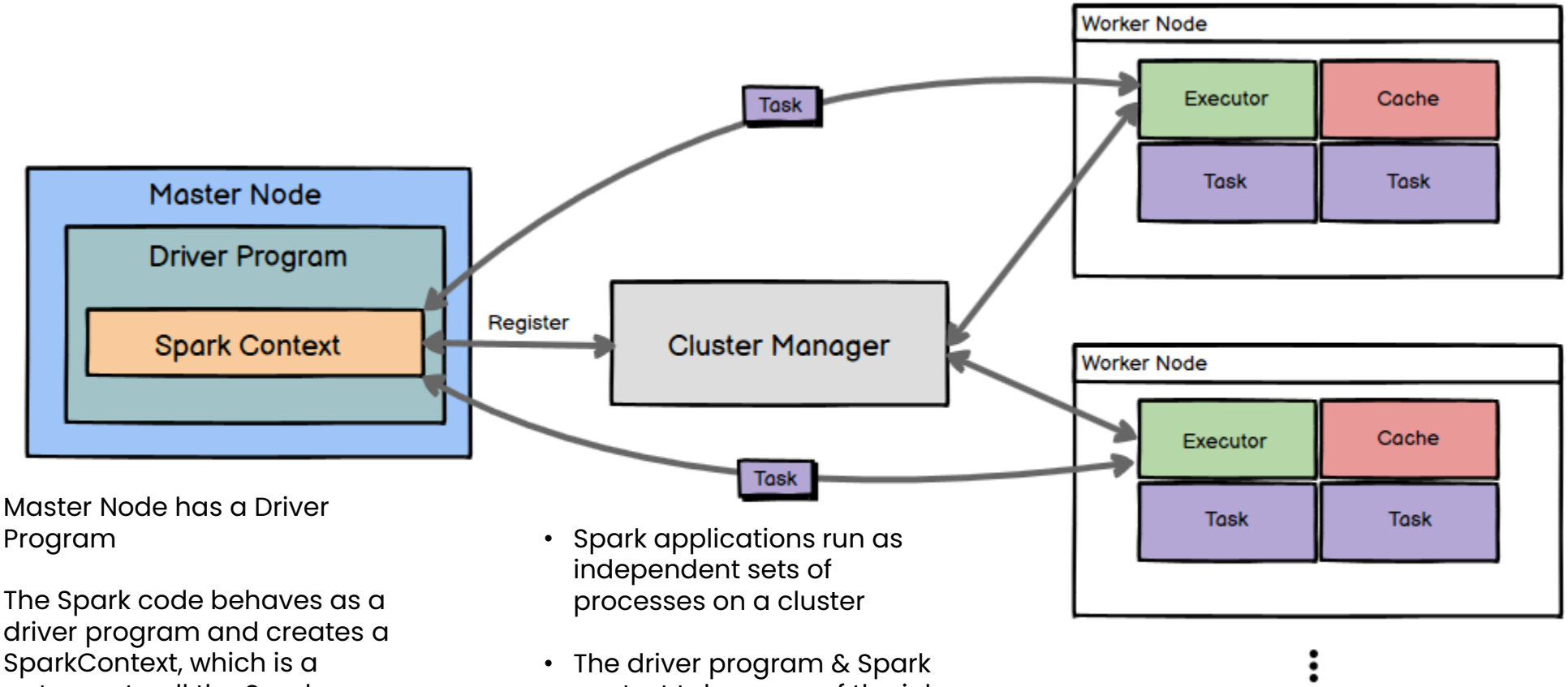
GraphX



Spark Architecture

Spark Architecture

Apache Spark uses a master-slave architecture that consists of a driver, that runs on a master node, and multiple executors which run across the worker nodes in the clester



- Master Node has a Driver Program
- The Spark code behaves as a driver program and creates a SparkContext, which is a gateway to all the Spark functionalities

- Spark applications run as independent sets of processes on a cluster
- The driver program & Spark context takes care of the job execution within the cluster

Spark Cluster Managers

Standalone mode

1

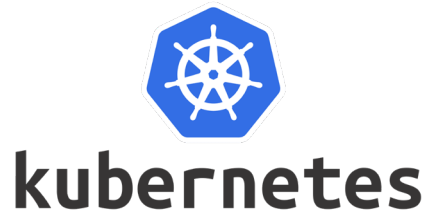
By default, applications submitted to the cluster will run in FIFO order, and each application will try to use all available nodes.

2

Apache Mesos is an open-source project to manage computer clusters, and can also run Hadoop applications

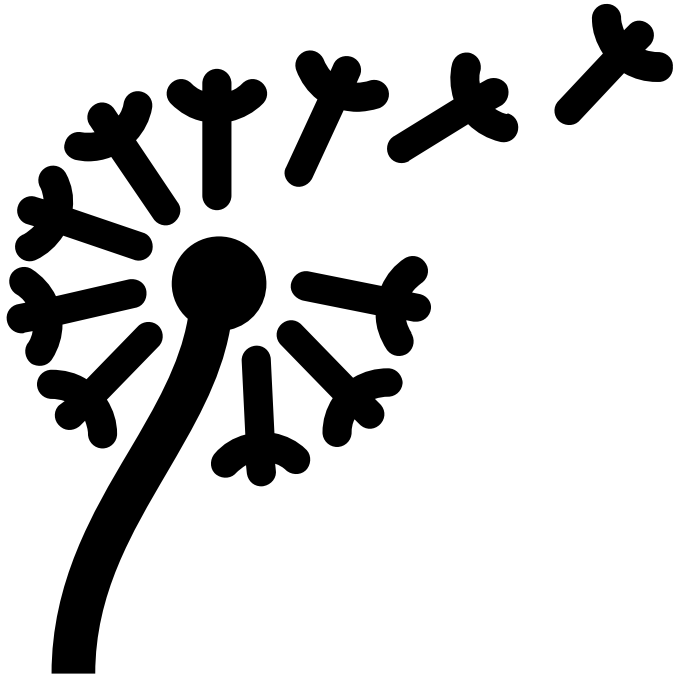
3

Apache YARN is the cluster resource manager of Hadoop 2. Spark can be run on YARN



4

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications



Applications of Spark

Applications of Spark

JPMorgan uses Spark to detect fraudulent transactions, analyze the business spends of an individual to suggest offers, and identify patterns to decide how much to invest and where to invest

JPMorgan Chase & Co.

Banking

Alibaba uses Spark to analyze large sets of data such as real-time transaction details, browsing history, etc. in the form of Spark jobs and provides recommendations to its users.

Alibaba.com

E-commerce

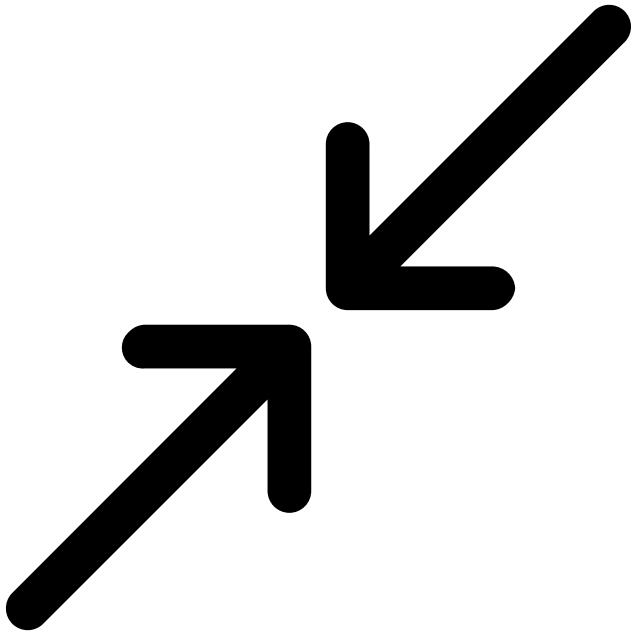
Entertainment and gaming companies like Netflix and Riot games use Apache Spark to showcase relevant advertisements to their users based on the videos that they watch, share and like

NETFLIX

RIOT GAMES

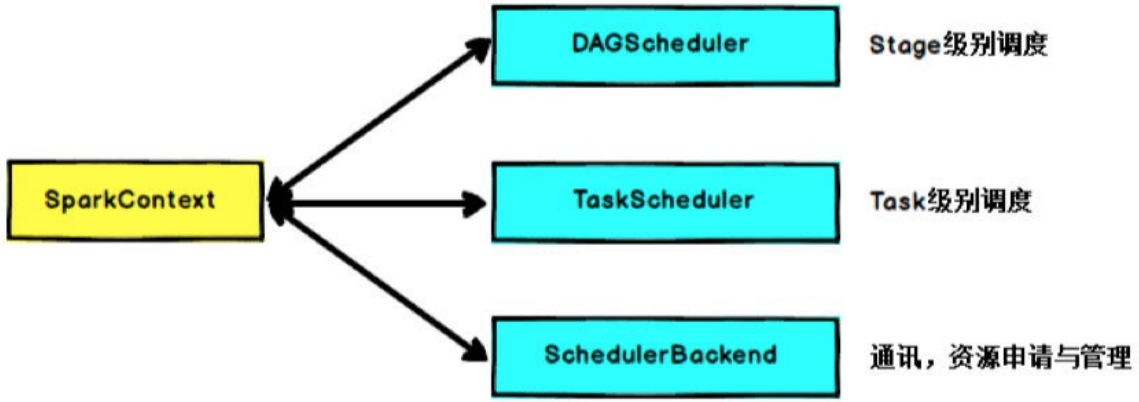
Entertainment

THANK YOU for your precious attention :)

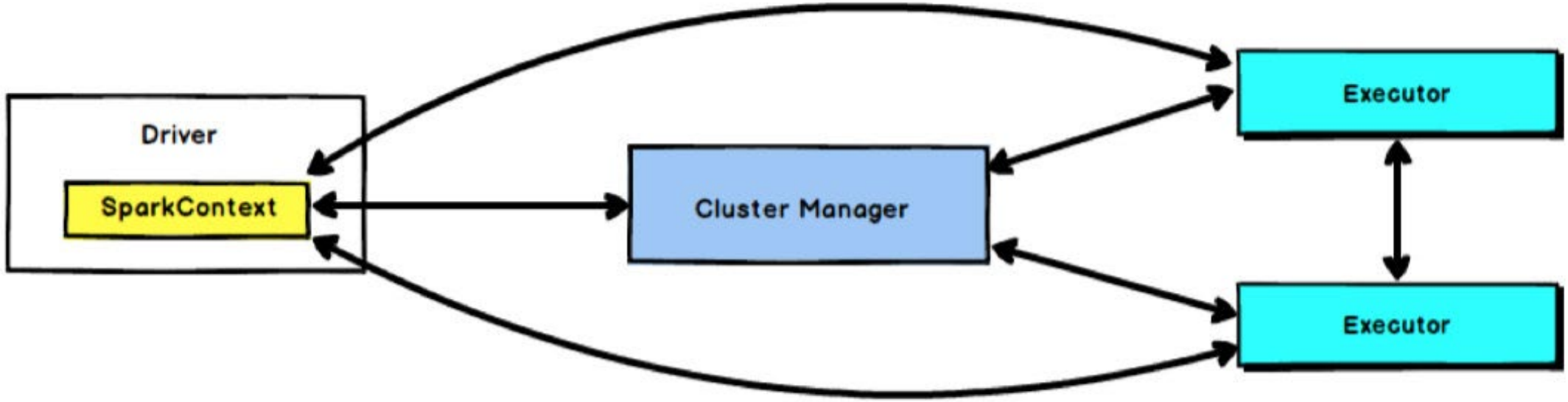


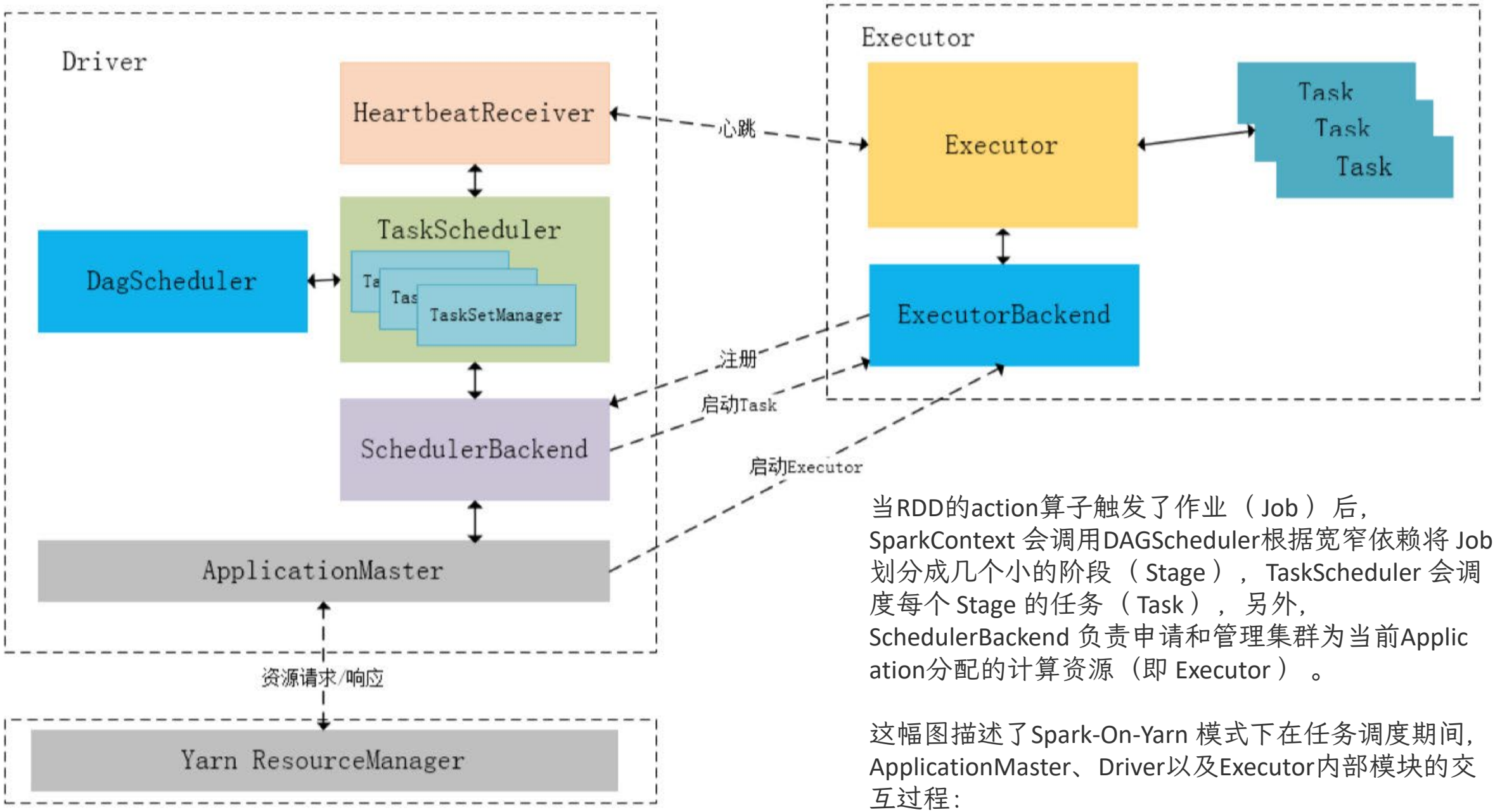
Hadoop vs Spark

SparkContext



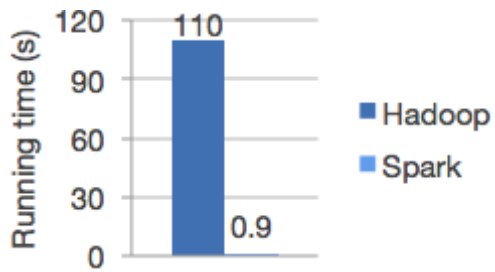
SparkContext的核心作用是初始化 Spark 应用程序运行所需的核心组件，包括高层调度器（DAGScheduler）、底层调度器（TaskScheduler）和调度器的通信终端（SchedulerBackend），同时还会负责Spark程序向ClusterManager的注册等。





当RDD的action算子触发了作业（Job）后，SparkContext 会调用DAGScheduler根据宽窄依赖将 Job 划分成几个小的阶段（Stage），TaskScheduler 会调度每个 Stage 的任务（Task），另外，SchedulerBackend 负责申请和管理集群为当前Application分配的计算资源（即 Executor）。

这幅图描述了Spark-On-Yarn 模式下在任务调度期间，ApplicationMaster、Driver以及Executor内部模块的交互过程：


Running time (s)
120
90
60
30
0
110
0.9
Hadoop
Spark

MapReduce vs Spark (WordCount)

```java
public class WordCount {
    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
                while (itr.hasMoreTokens()) {
                    word.set(itr.nextToken());
                    context.write(word, one);
                }
        }
    }

    public static class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();
        public void reduce(Text key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }
}
```

```scala
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
    .map(word => (word, 1))
    .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

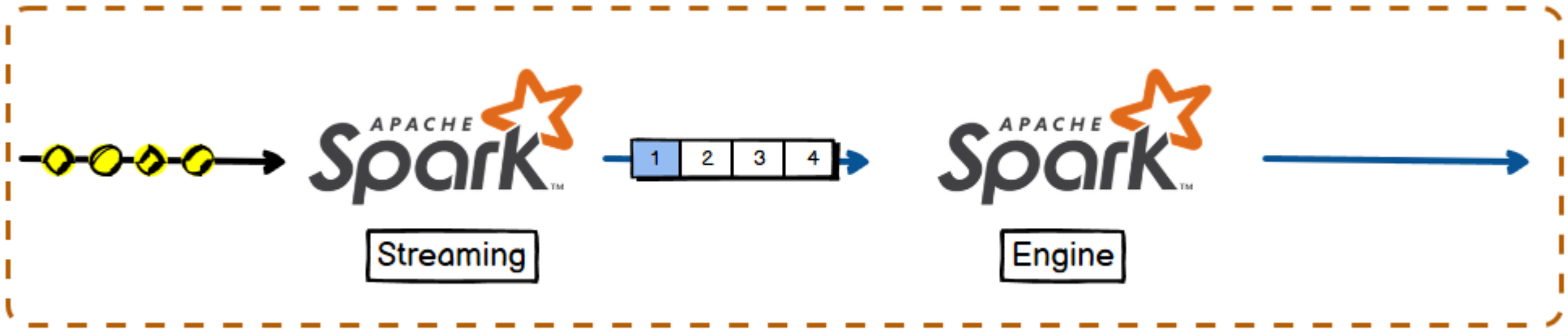
Spark Use Case

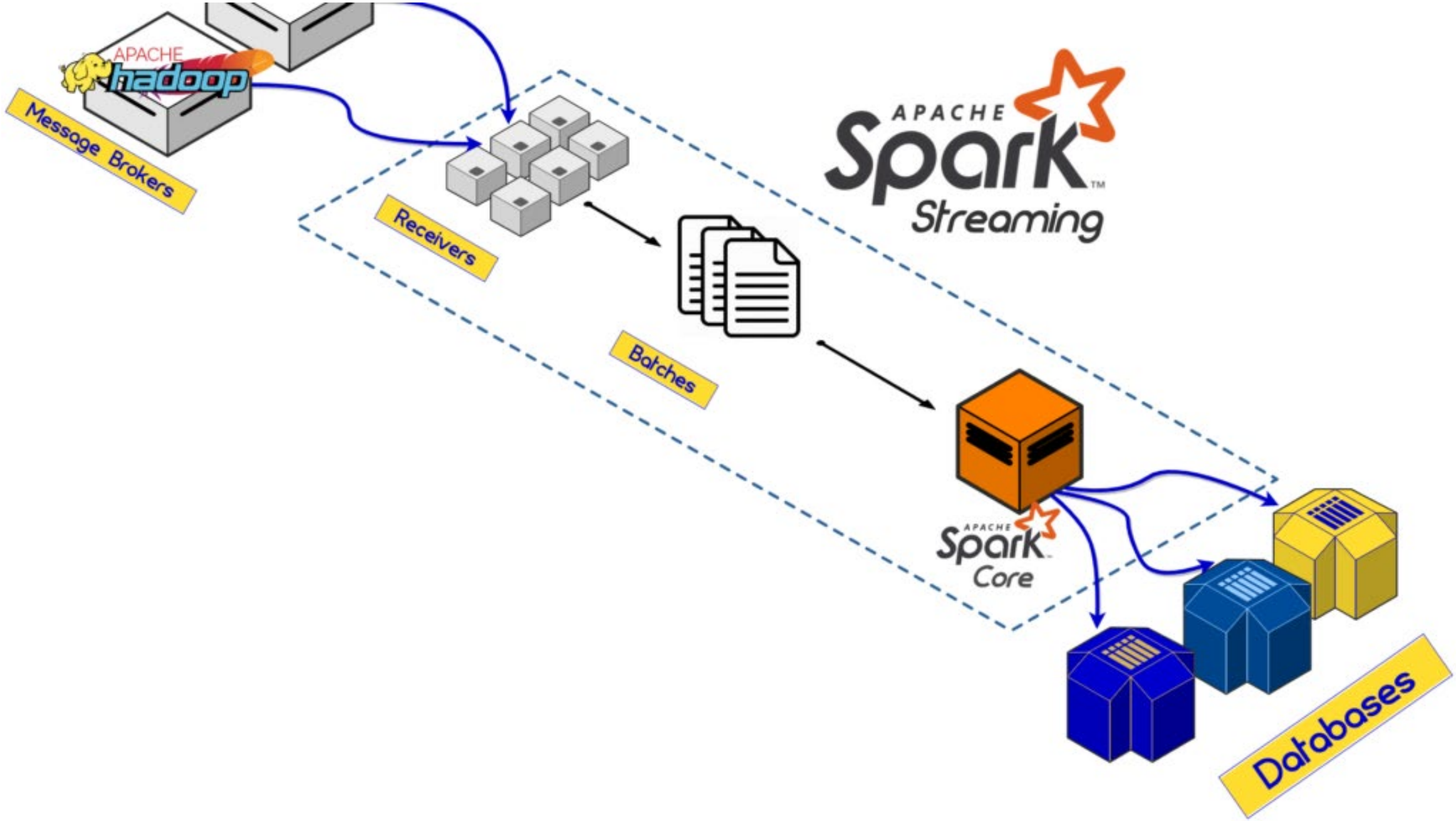
Spark Streaming

Spark Streaming is a lightweight API that allows developers to perform batch processing and real-time streaming of data with ease

Provides secure, reliable, and fast processing
of live data streams

Spark Streaming



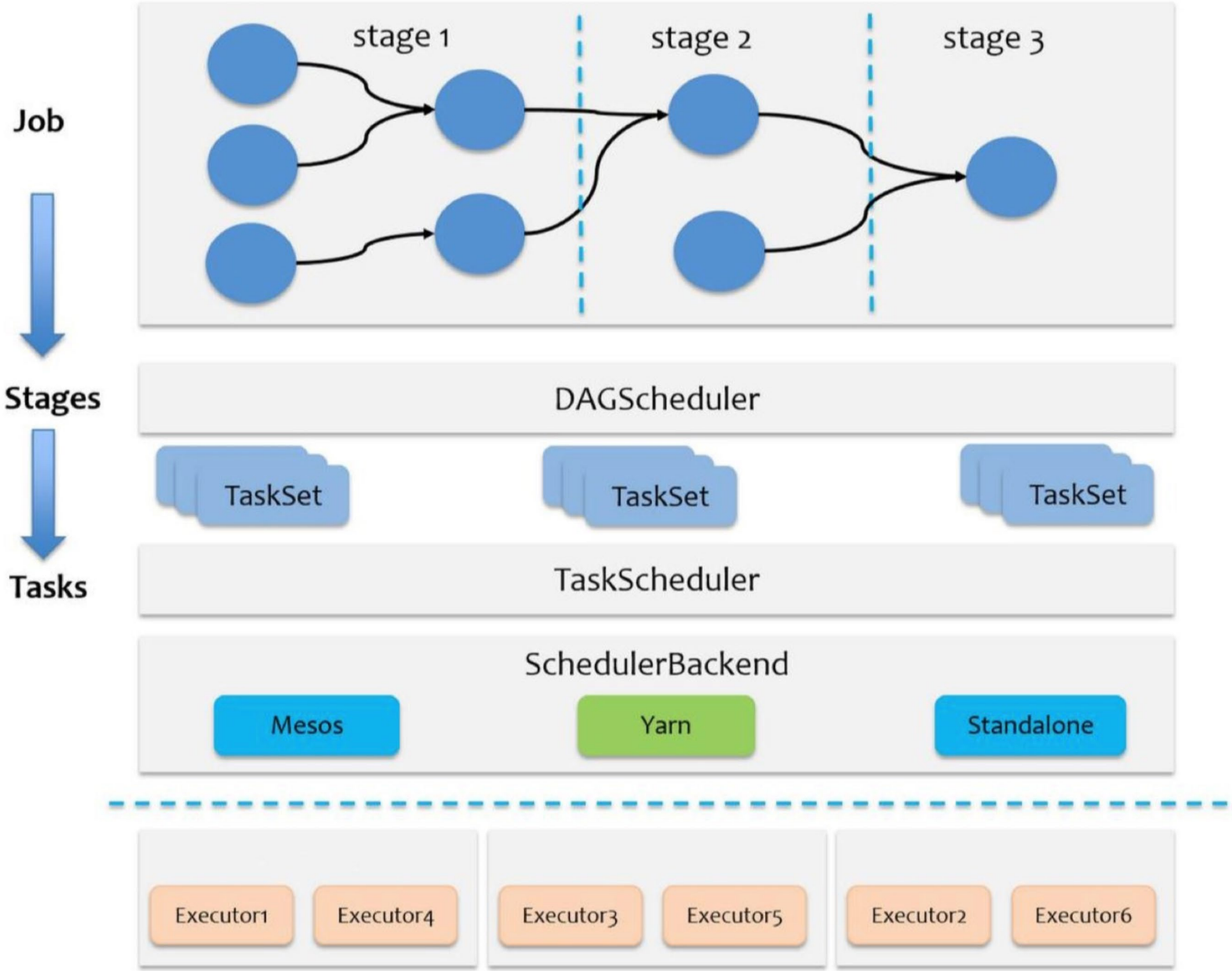

APACHE
hadoop
Message Brokers
Receivers
Batches
APACHE
Spark
Streaming
APACHE
Spark
Core
Databases

[Spark任务调度概述]

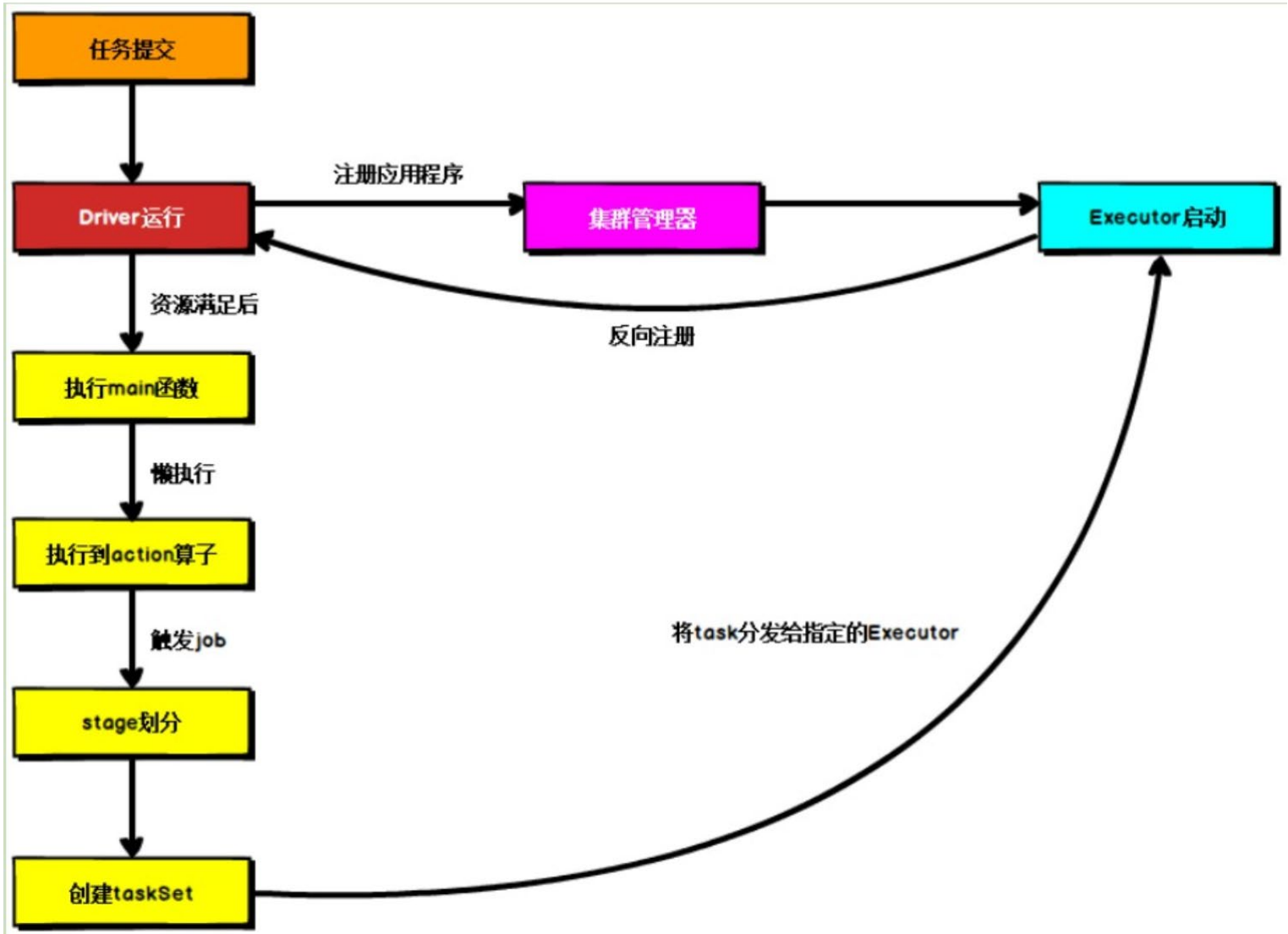
Job	Stage	Task

一个Spark程序包括Job、Stage以及Task三个概念：

Job是以Action方法为界，遇到一个Action方法则触发一个Job；
Stage是Job的子集，以RDD宽依赖（即Shuffle）为界，遇到Shuffle做一次划分；
Task是Stage的子集，以并行度（分区数）来衡量，分区数是多少，则有多少个task。
Spark任务的调度总体上分两路进行，一路是Stage级的调度，一路是Task级的调度，总体的调度流程如下：



提交流程


任务提交
Driver运行
注册应用程序
集群管理器
Executor启动
资源满足后
反向注册
执行main函数
懒执行
执行到action算子
触发job
将task分发给指定的Executor
stage划分
创建taskSet