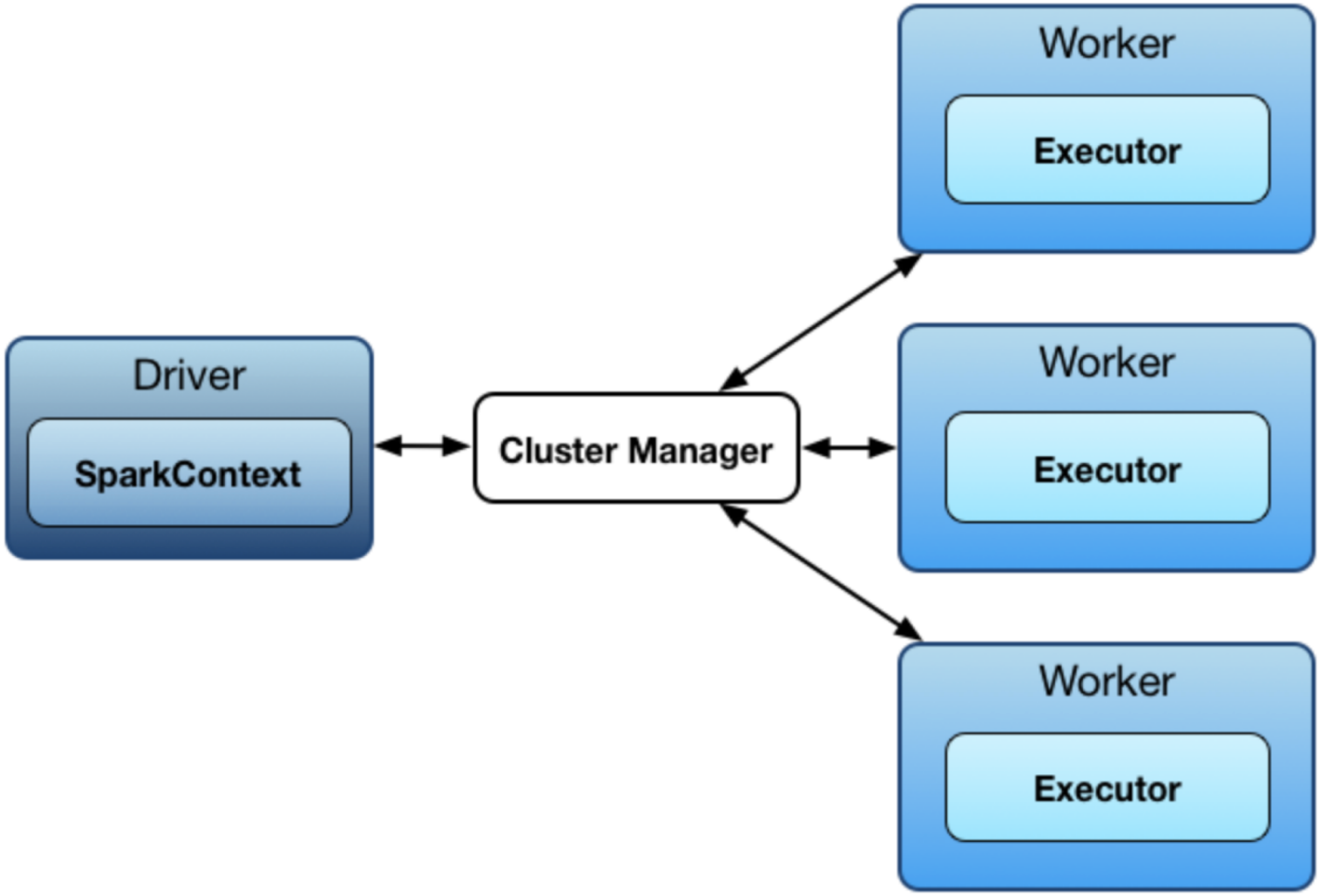
Master-Worker Architecture

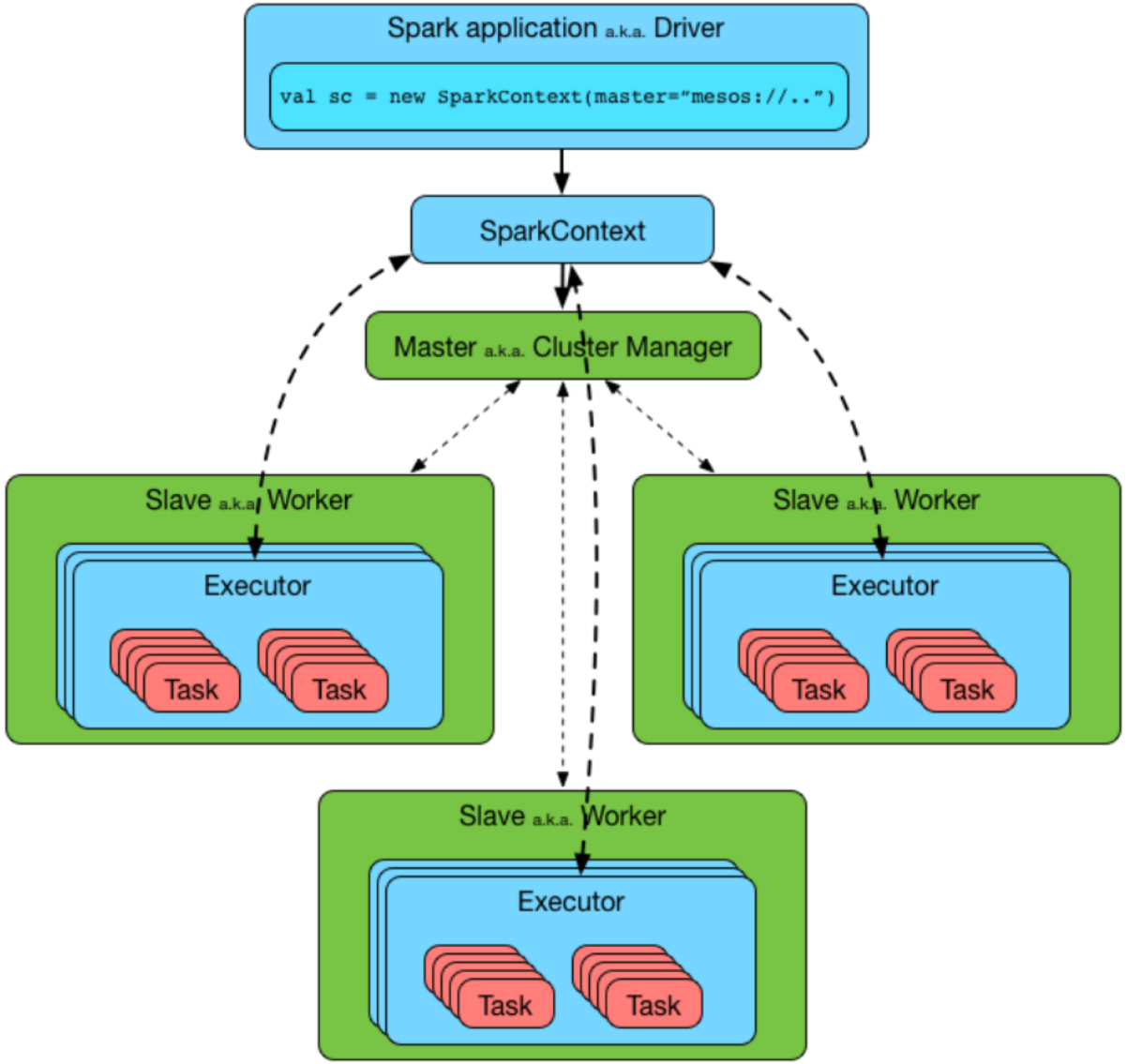
- Spark uses a ***master-worker*** **architecture**.
- The ***driver*** that talks to a coordinator called the ***master*** that manages ***worker*** where ***executors*** run.

Credit

Master-Worker Architecture

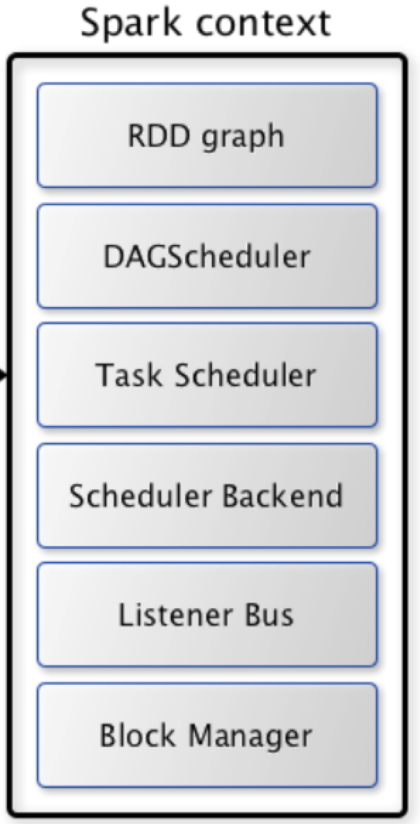


The driver and the executors run in their own Java processes. You can run them all on the same machne or separate machines



Spark Architecture

- An **executor** is a distributed agent that is responsible for executing tasks
- A **master** is a running Spark instance that connects to a cluster manager for resources.
- **Workers** are running Spark instances where executors live to execute tasks. They are the compute nodes in Spark.
- Every Spark application starts by creating **SparkContext**
- **SparkConf** is the Spark Configuration

SparkContext — Entry Point to Spark Core

```
val sc = new SparkContext(master="local[*]",
           appName="SparkMe App", new SparkConf)

val lines = sc.textFile(...).cache()

val c = lines.count()
println(s"There are $c lines in $fileName")
```