

LONDON: CHAPMAN & HALL, 193, PICCADILLY.

Advertisements to be sent to the Publishers, and ADAMS & FRANCIS, 59, Fleet Street, E.C.
[The right of Translation is reserved.]

Part 1

Disclaimer


- These are my notes to answer the question of “what is spark”
- My goal is to create a very very easy-to-digest introduction to Spark
- The information and diagrams are gathered & summarized or copied verbatim from the set of books and tutorials listed in the References slide.

What is Spark?

It is a cluster computing framework for parallel processing

What is cluster computing?

*aka : nodes,
hosts,
machines*



- Use of a group of computers that are linked together on a network and that work together to perform tasks.
- Using multiple computers like this, instead of a single computer, allows us to process more data using many smaller, cheap computers instead of one large, expensive super-computer by leveraging **parallel processing**

* Note: High-performance computing (above) is only one reason that computers may be clustered. Computers may also be clustered for:

- High-availability
- Load-balancing (this is different than parallel processing in that you have many computers each taking on self-contained units of work. In parallel processing the computers are working together to accomplish a single task)

So what does Spark do?

- Manages and coordinates execution of tasks across the multiple computers in the cluster
 - Divides and distributes the data used for the work
 - Divides and distributes the work that needs to be done
 - Brings everything back together to get results

Same Same but Different

- map-reduce is an analogous framework for parallel computing
 - There is a slide for comparing map reduce and spark later on

3 Parts to this

1. The physical infrastructure - the set of computers on which spark is run
2. Spark Core - the framework that distributes and co-ordinates work across the physical infrastructure
3. Spark Application - the work we ultimately want to do

Part 2

The Cluster

...

The physical infrastructure

Physical Spark Cluster Infrastructure

There are a couple options for how you run Spark

- Local Mode:
 - Fake cluster essentially. All the same spark parts exist but they all run on the same physical machine
 - Good for testing, demos, dev envs etc
- Distributed Mode
 - Where computation and data are spread across multiple machines
 - Within distributed mode there are 4 further choices that use different “cluster managers”
 - More on what a cluster manager is later

Part 3

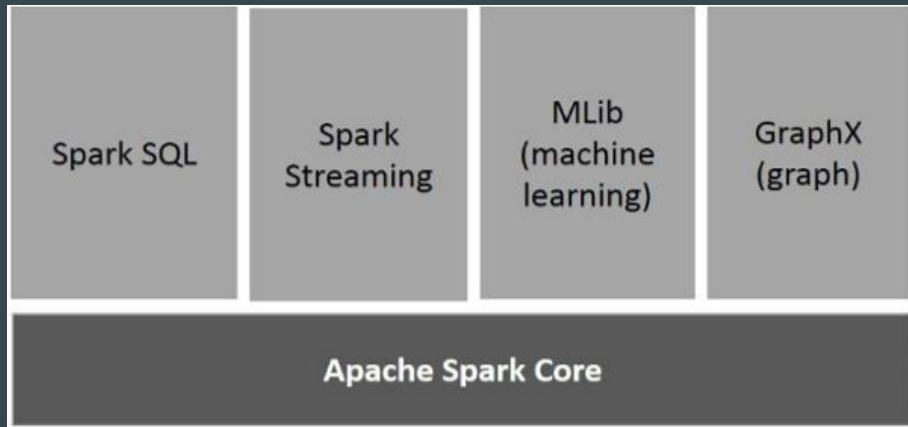
Spark Core

...

The engine

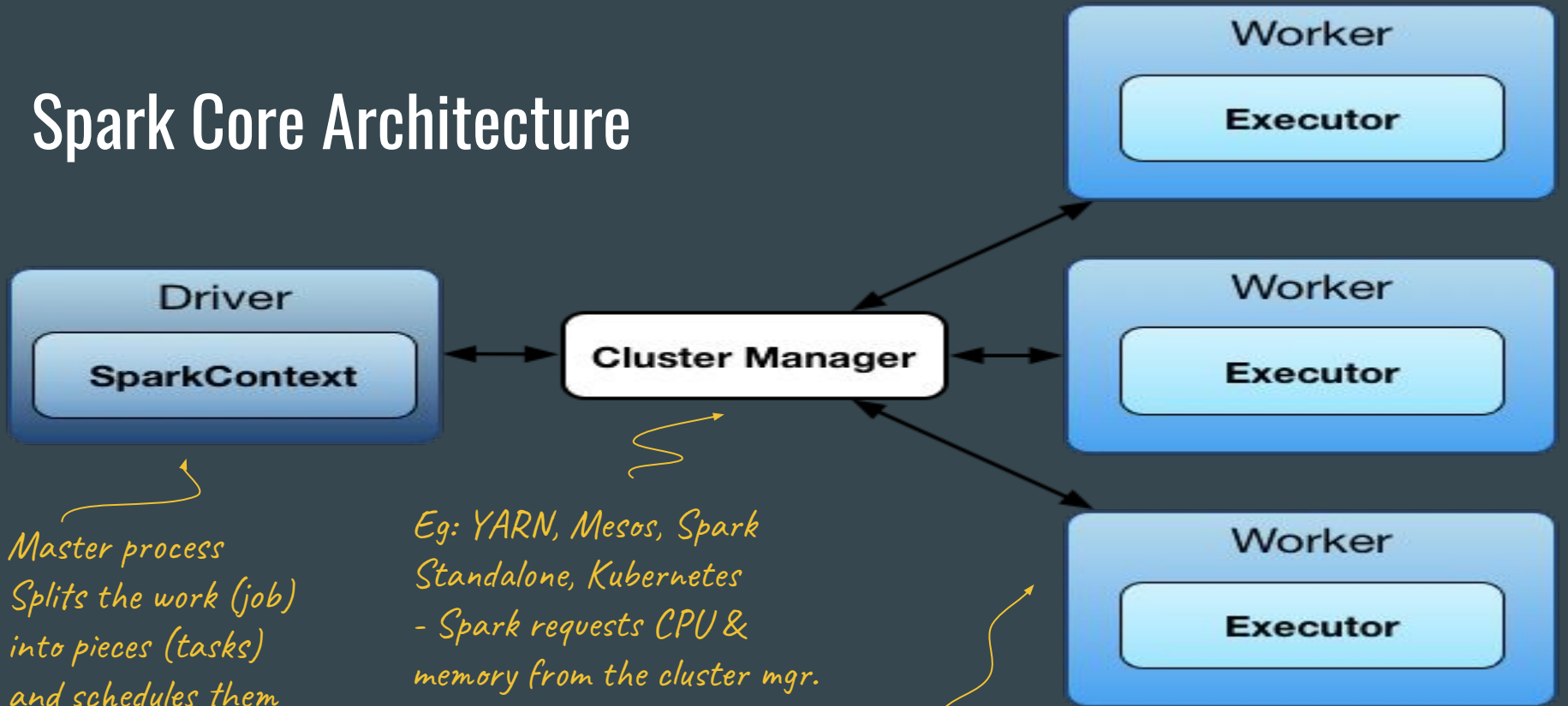
What is Spark Core?

- When people say “**Spark**” they may be referring to either “Spark Core” or to the “Apache Spark Project”



- Spark Core is the piece that does the distribution and management of processing
- Apache Spark Project is an umbrella for an assortment of projects built on top of Spark Core:
 - Spark SQL - <http://spark.apache.org/sql/>
 - Spark Streaming - <http://spark.apache.org/streaming/>
 - Spark Machine Learning Pipeline - <http://spark.apache.org/mllib/>
 - Spark Graph Processing Engine - <http://spark.apache.org/graphx/>

Spark Core Architecture



*Master process
Splits the work (job)
into pieces (tasks)
and schedules them
to run on the
executors*

*Eg: YARN, Mesos, Spark
Standalone, Kubernetes
- Spark requests CPU &
memory from the cluster mgr.
- The cluster mgr is
responsible on spawning
executor processes on worker
nodes that meet the reqs*

*Workers aka Slaves are machine(s) that host the
Executor process
These guys run the computations and store the
data*

Part 4

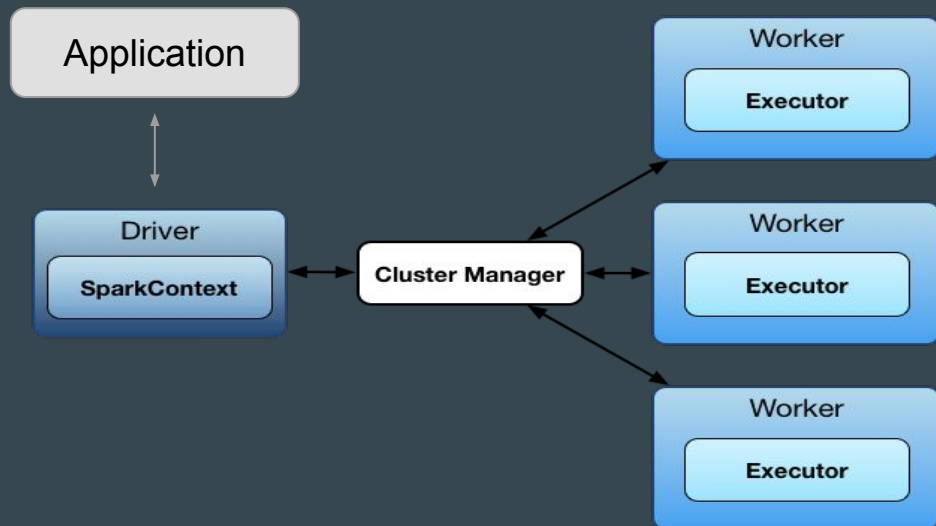
Spark Applications

...

The thing that we give the engine to run

What is a Spark Application?

- A top level computation work item that is executed using the spark execution engine
- Coded by developers to define what computation logic needs to happen
- This is the thing that is given to Spark Core to run
- Spark Core will take this and execute it across the set of worker machines



Sequence of events when you run a Spark Application

Defined in detail in upcoming slides

When you submit a Spark application to the cluster this is what happens

- The Spark driver is launched to invoke the main method of the Spark application.
- The driver asks the cluster manager for resources (CPU & memory) to run the application, i.e. to launch executors that run tasks.
- The cluster manager launches executors.
- The driver runs the Spark application and sends tasks to the executors.
- Executors run the tasks and save the results.
- Right after `SparkContext.stop()` is executed from the driver or the main method has exited all the executors are terminated and the cluster resources are released by the cluster manager.

Will go into what this is in detail too

Units of work - How Spark Core divides up the computation

Application

- Top level unit of computation
- Is a single instance of a Spark Context
- This is the code that the user wants to run in parallel using Spark

Job

- An application can be:
 - a single job or
 - an interactive session with > 1 jobs

Stage

- A job is broken into multiple stages
- A stage is a single set of parallel tasks
- Stages depend on each other and run in sequence

Task

- The smallest unit of execution
- Runs on a single partition
- “a task is a computation on the records in a RDD partition in a stage of a RDD in a Spark job”

How work is divided - part 2

TODO: Example or diagram. Perhaps copy from notebook explain? Or from DAG visualization

Part 5

All the big pieces have been presented.

Feel free to stop here if you just want a highlevel view for now

Next up will go into some of the above pieces (spark execution engine, spark applications etc) in more detail

Spark Application Internals

...

What is a Spark Context?

What it is:

- The spark context is a client of the Spark Core Execution Environment
- First thing a Spark application does when it is submitted to Spark engine is create & initialize the spark context
- Application completes by stopping the spark context
- Heart of a Spark application - main entry point for coding spark functionality in the app

What it does:

It is the interface for the application to perform the following actions:

- Configure spark job
- Create distributed entities - data entities (RDDs), vars etc
- Get current status of the spark app
- Access Spark services like schedulers, managers etc

RDD - Resilient Distributed Dataset

- Oldest and lowest level data abstraction in Spark
- It is a single logical entity that abstracts data that is physically distributed across the various executor machines
- TODO

References

- What is spark video (7:14) -
<https://www.youtube.com/watch?v=SxAxAhn-BDU&feature=youtu.be>
- Book: <https://jaceklaskowski.gitbooks.io/mastering-apache-spark>
- <https://www.slideshare.net/bosswebtech/cluster-computing-11382951>
- https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm
- On cluster managers:
<http://www.agildata.com/apache-spark-cluster-managers-yarn-mesos-or-standalone/>