Introduction to SPARK

Agenda

- What is SPARK
- Spark architecture
- Job execution in SPARK
- Spark deployment modes
- Demo

What is SPARK ?

- A parallel processing framework
- Its open source
- Developed at the university of California, Berkeley
- Team that created SPARK went on to form a company DATABRICKS in 2013
- Initial release was in May 30th 2014
- Developed predominantly using Scala
- High throughput applications deployed on spark enjoys the best performance if developed in SCALA

What is SCALA ?

- A high level programming language
- Supports functional style of programming
- Supports OO style of programming
- It’s actually a multi-paradigm language

What is functional programming ?

Consider the following code snippet

```
int a = 10; //A new variable “a” is getting created
a++ ;  // We are incrementing the variable a
Print(a)   // will generate 11 as the o/p
```

Consider another way of doing the same thing

```
int a= 10;
int a1 = a + 1 ;  // We are creating a new variable a1
Print(a) ; // a will still contain the value 10, state of the
  //variable a is not changed.
```

Functional programming

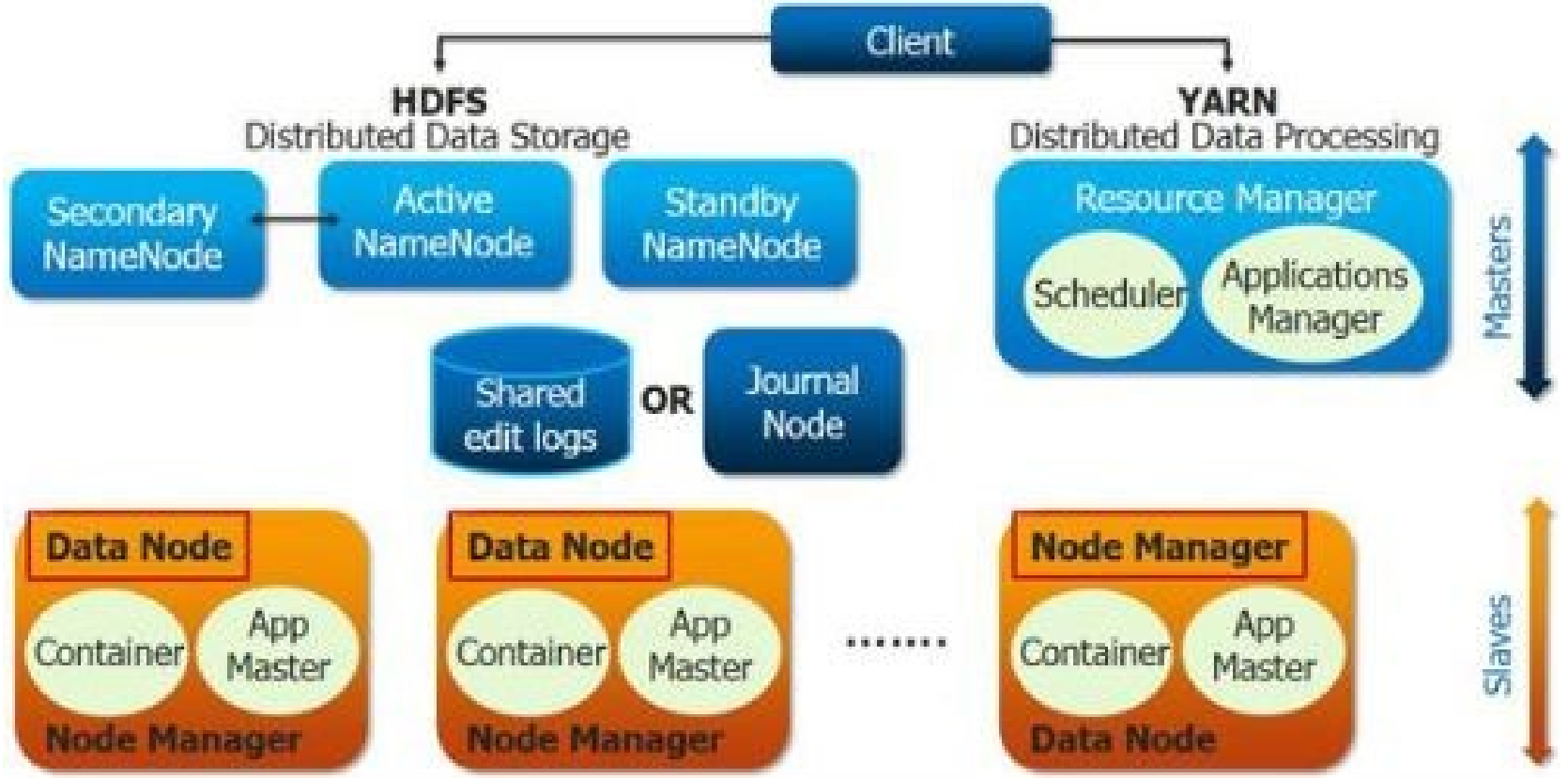
- Functional style of programming inherently supports parallelism
- Functional style of programming is not restricted to a particular programming language
- It can be implemented in any language, just like the way OO concepts can also be implemented in a programming language like C
- Certain programming languages inherently support functional style of programming reducing the programmer's burden
- Scala is one of the functional programming languages

SPARK in comparison to HADOOP

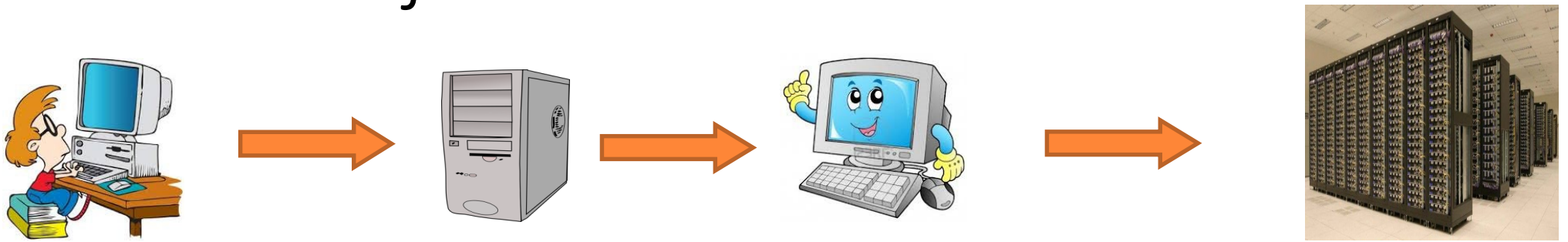
- The high level architecture of SPARK is very similar to that of HADOOP
- A quick way of understanding SPARK is by comparing it with HADOOP
- Several drawbacks of Hadoop are addressed in Spark which gives a 10x-100x performance improvement over Hadoop
- Its well suited for interactive and real time analytics of big data and streaming data

- Its highly popular among machine learning users since it outperforms Hadoop in iterative workloads

A QUICK RECAP OF HADOOP



Overview of job execution in HADOOP



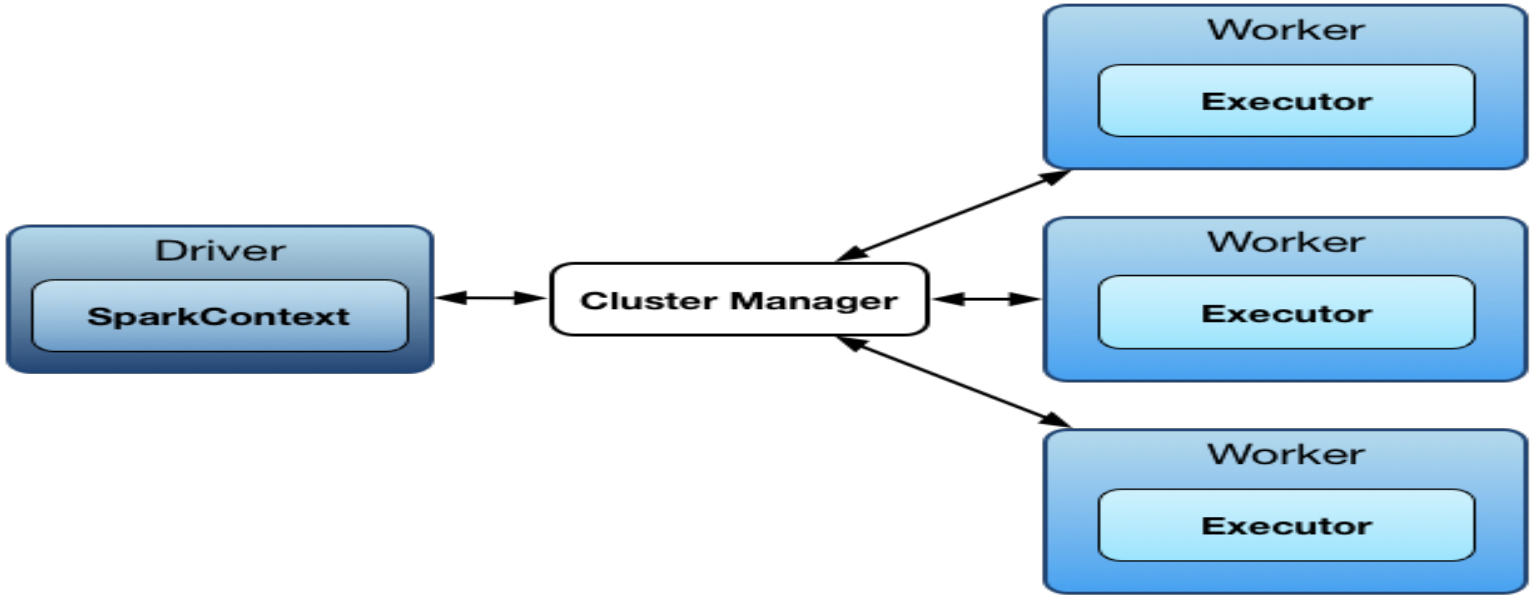
Job client **Name node** **Resource manager/YARN** **Data nodes**

A job submitted by the user is picked up by the name node and the resource manager

Job gets submitted to the name node and eventually resource manager is responsible for scheduling the execution of the job on the data nodes in the cluster

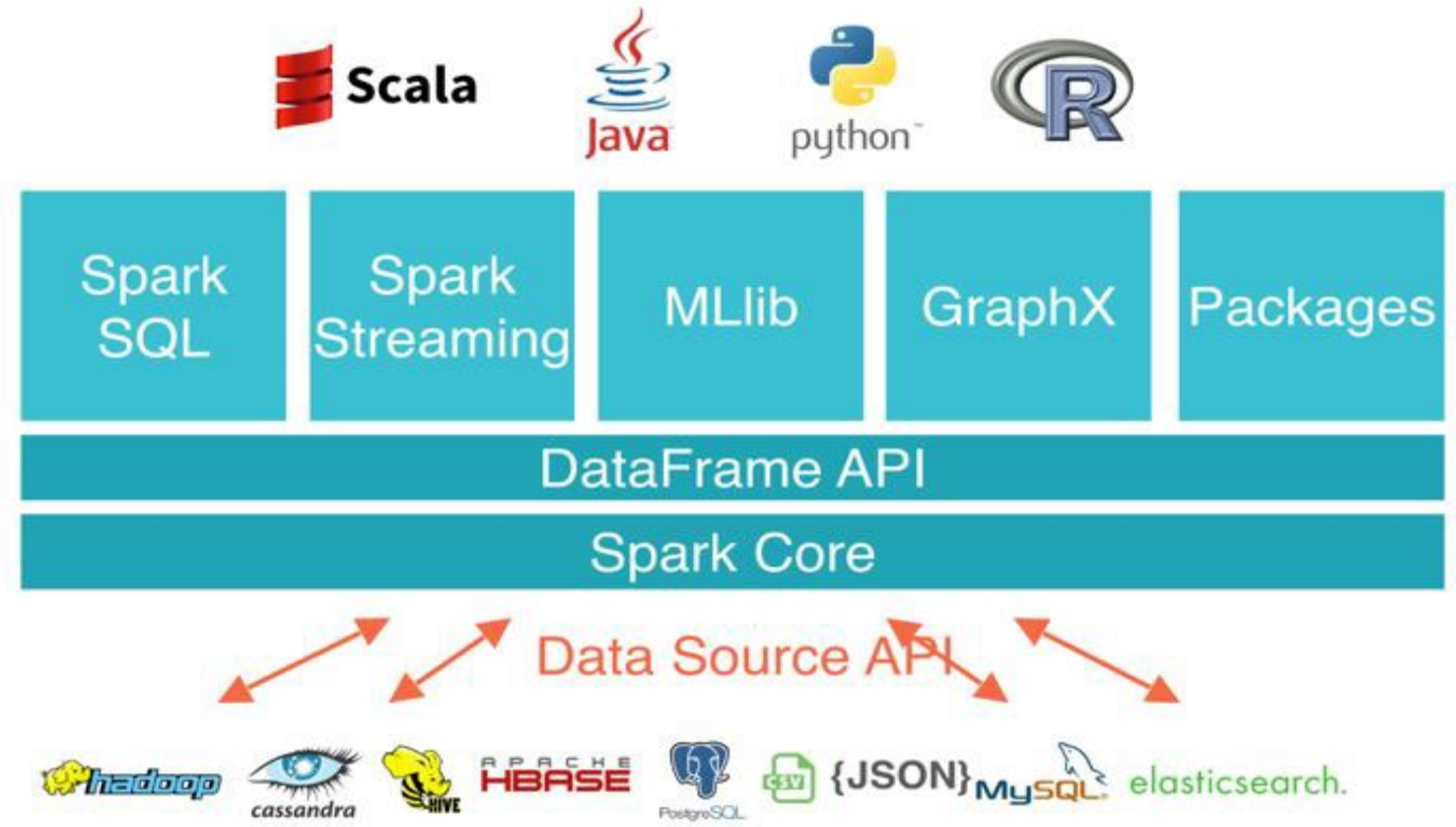
The data nodes in cluster consists the data blocks on which the user's program will be executed in parallel

Overview of SPARK architecture

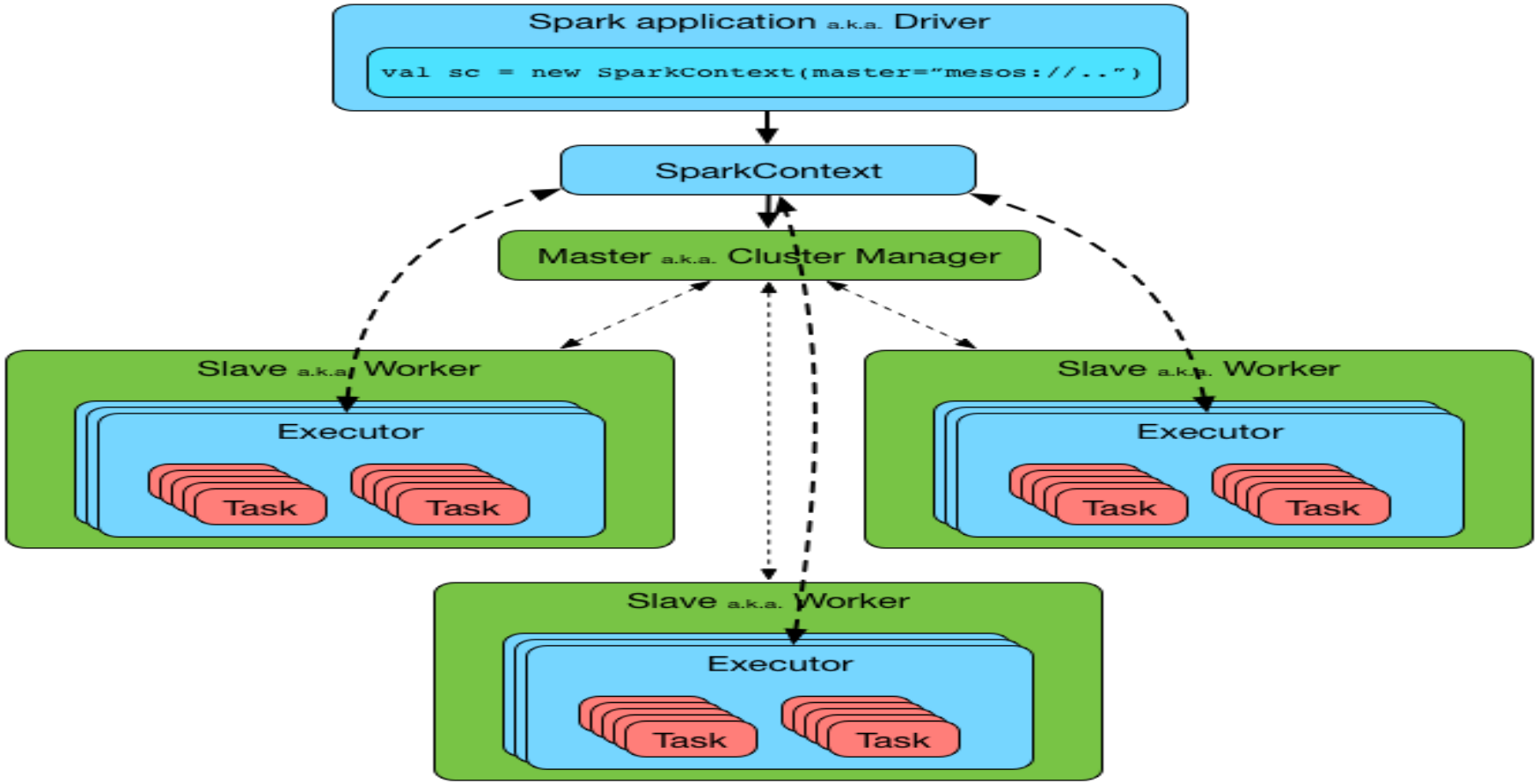


- **Driver is the starting point of a job submission (this can be compared to the driver code in Java MR)**
- **Cluster Manager can be compared to the Resource Manager in Hadoop**
- **Worker nodes are the data nodes in a HADOOP cluster**
- **The executor can be compared to that of a node manager in Hadoop 2 or task tracker in Hadoop 1**

SPARK architecture



A closer look into the SPARK architecture



Spark deployment modes

- Standalone (used for learning & development)
  - This is very similar to the pseudo distributed mode of Hadoop
  - Spark services run on multiple JVM's
  - This is also known as standalone cluster

- Local mode (used for learning & development)
  - There is a single JVM (no need of HDFS)

- Cluster mode (can work with MESOS or YARN)
  - Used for production environment & it's a fully distributed mode

Summary

- What is spark
- Functional programming
- Overview of SPARK architecture in comparison to Hadoop
- Deployment modes of SPARK