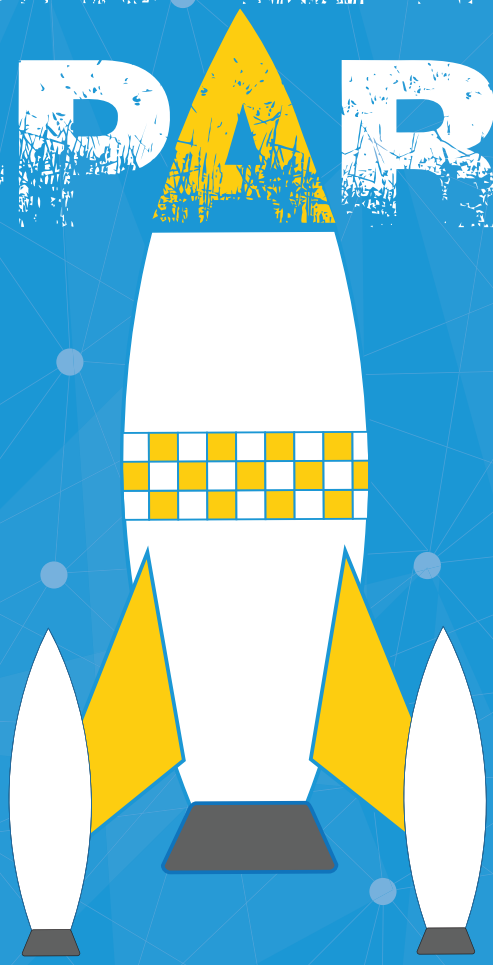
ACADGILD

BEGINNER GUIDE FOR SPARK



LEARN. DO. EARN

About ACADGILD

ACADGILD is a technology education startup that aims to create an ecosystem for skill development in which people can learn from mentors and from each other. We believe that software development requires highly specialized skills that are best learned with guidance from experienced practitioners. Online videos or classroom formats are poor substitutes for building real projects with help from a dedicated mentor. Our mission is to teach hands-on, job-ready software programming skills, globally, in small batches of 8 to 10 students, using industry experts.

ACADGILD offers courses in

Enroll in our programming course
& Boost your career

- ANDROID DEVELOPMENT
- DIGITAL MARKETING
- MACHINE LEARNING WITH R
- BIG DATA ANALYSIS
- JAVA FOR FRESHER
- BIG DATA & HADOOP ADMINISTRATION
- FULL STACK WEB DEVELOPMENT
- NODE JS
- CLOUD COMPUTING
- FRONT END DEVELOPMENT (WITH ANGULARJS)

Watch this short video to know more about ACADGILD.

Disclaimer

This material is intended only for the learners and is not intended for any commercial purpose. If you are not the intended recipient, then you should not distribute or copy this material. Please notify the sender immediately or click here to contact us.

Published by

ACADGILD,

support@acadgild.com

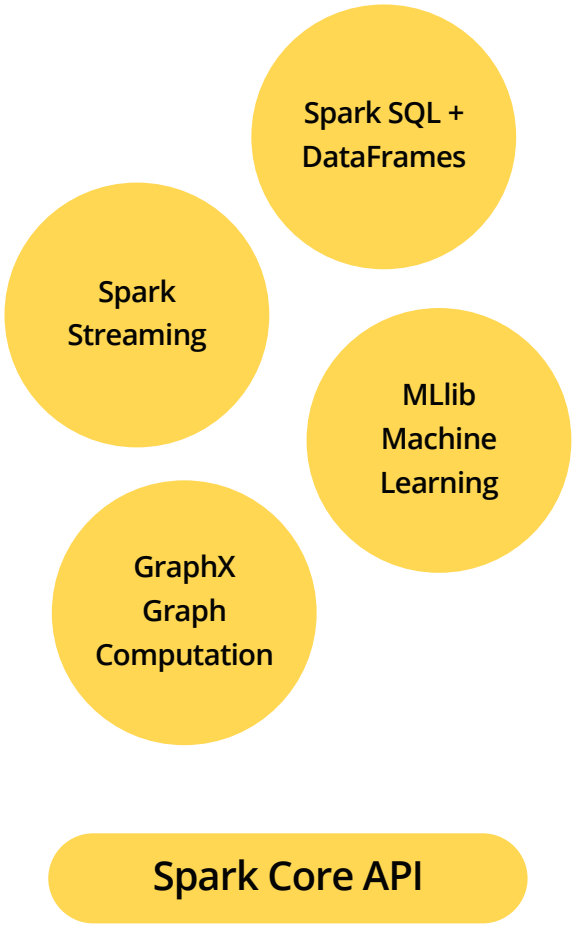
In this EBook we will be discussing the basics of Spark's functionality and its installation.

Spark

What is Spark?

Apache spark is a cluster computing framework which runs on Hadoop and handles different types of data. It is a one stop solution to many problems. Spark has rich resources for handling the data and most importantly, it is 10-20x faster than Hadoop's MapReduce. It attains this speed of computation by its in-memory primitives. The data is cached and is present in the memory (RAM) and performs all the computations in-memory.

Spark's rich resources has almost all the components of Hadoop. For example we can perform batch processing in Spark and real time data processing, without using any additional tools like kafka/flume of Hadoop. It has its own streaming engine called spark streaming.



We can perform various functions with Spark

SQL operations	Machine Learning	Graph processing
It has its own SQL engine called Spark SQL. It covers the features of both SQL and Hive.	It has Machine Learning Library , MLib. It can perform Machine Learning without the help of MAHOUT.	It performs Graph processing by using GraphX component.

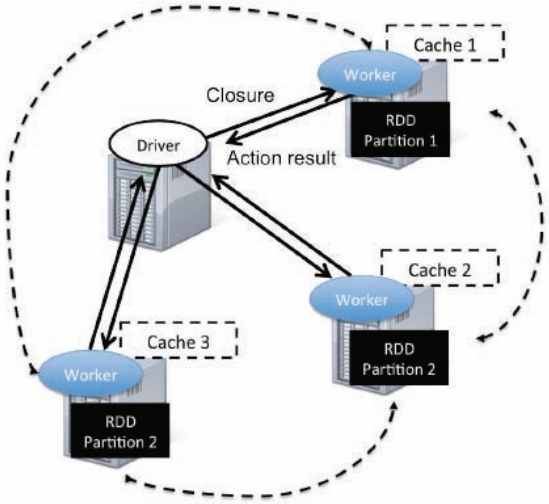
All the above features are in-built in Spark.

It can be run on different types of cluster managers such as Hadoop, YARN framework and Apache Mesos framework. It has its own standalone scheduler to get started, if other frame-works are not available.Spark provides the access and ease of storing the data,it can be run on many file systems. For example, HDFS, Hbase, MongoDB, Cassandra and can store the data in its local files system.

Resilient Distributed Datasets

Resilient Distributed Datasets (RDD) is a simple and immutable distributed collection of objects. Each RDD is split into multiple partitions which may be computed on different nodes of the cluster. In spark all function are performed on RDDs only.

Spark revolves around the concept of a resilient distributed dataset (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel.



Let's see now the features of **Resilient Distributed Datasets** in the below explanation:

01

In Hadoop we store the data as blocks and store them in different data nodes. In Spark, instead of following the above approach, we make partitions of the RDDs and store in worker nodes (datanodes) which are computed in parallel across all the nodes.

02

In Hadoop we need to replicate the data for fault recovery, but in case of Spark, replication is not required as this is performed by RDDs.

03

RDDs load the data for us and are resilient which means they can be recomputed.

04

RDDs perform two types of operations: transformations which creates a new dataset from the previous RDD and actions which return a value to the driver program after performing the computation on the dataset.

05

RDDs keeps a track of transformations and checks them periodically. If a node fails, it can rebuild the lost RDD partition on the other nodes, in parallel.

RDDs can be created in two different ways:

1. Referencing an external dataset in an external storage system, such as a shared file system, HDFS, HBase, or any data source offering a Hadoop Input Format.

2. By parallelizing a collection of objects (a list or a set) in the driver program.

Step by step process to Install Spark

Before installing spark Scala needs to be installed in the system. We need to follow the below steps to install scala.

1. Open the Terminal in your CentOS

- To download Scala type the below command:
- Type: **Wget http://downloads.typesafe.com/scala/2.11.1/scala-2.11.1.tgz**

```
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Desktop]$ cd
[acadgild@localhost ~]$ gedit .bashrc
[acadgild@localhost ~]$ source .bashrc
[acadgild@localhost ~]$ wget http://downloads.typesafe.com/scala/2.11.1/scala-2.
11.1.tgz
--2015-11-03 09:00:49--  http://downloads.typesafe.com/scala/2.11.1/scala-2.11.1
.tgz
Resolving downloads.typesafe.com... 54.230.173.102, 54.192.173.227, 54.239.190.4
3, ...
Connecting to downloads.typesafe.com|54.230.173.102|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25685521 (24M) [application/octet-stream]
Saving to: “scala-2.11.1.tgz”

100%[===================================>] 25,685,521  3.58M/s   in 28s

2015-11-03 09:01:19 (883 KB/s) - “scala-2.11.1.tgz” saved [25685521/25685521]
```

2. Extract the downloaded tar file by using the below command

Extract the downloaded tar file by using the command, **tar –xvfscala-2.11.1.tgz**

```
File  Edit  View  Search  Terminal  Help
Documents                          myip~
Downloads                          Oozie
dummy_ip~                          p
eclipse                            Pictures
gen-php                            pig-0.14.0
gen-py                             Public
hadoop                             satyam
hadoop-2.6.0                       scala-2.11.1.tgz
hadoop-2.6.0.tar.gz                spark-1.5.1-bin-hadoop2.6
hbase-0.98.14-hadoop2              student_data~
hbase-0.98.14-hadoop2-bin.tar.gz   student_test~
hdfs:                              temp_data~
hi.php~                            Templates
hive_folder                        test~
input~                             test1~
input_file~                        test_emp~
input_file1~                       test_input~
java                               thaseen
jdk1.8.0_60                        thrift
jdk-8u60-linux-x64.tar.gz          thrift-0.9.0
listener_data~                     Titanic
list.py~                           Videos
max_min.jar                        workspace
[acadgild@localhost ~]$ tar xvf scala-2.11.1.tgz
```

3. After extracting specify the path of scala in .bashrc file.

```
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ gedit .bashrc
```

```
#export scala path
export SCALA_HOME=$HOME/scala-2.11.1
export PATH=$SCALA_HOME/bin:$PATH
```

After setting the path we need to save the file and type the below command to save all the configurations.:

source .bashrc

The above command will sum up the scala installation.
we need to then install spark after that.

To install spark in centos we need to follow the below steps to download and install **Single Node cluster of Spark in CentOS.**

1. Open the browser and go the link

- Download **spark-1.5.1-bin-hadoop2.6.tgz**
- File will be downloaded into Downloads folder
- Go to the Downloads folder and untar the Downloaded file using the below command:

  tar –xvf spark-1.5.1-bin-hadoop2.6.tgz

```
[acadgild@localhost ~]$ cd Downloads
[acadgild@localhost Downloads]$ ls
Crimes_-_2001_to_present.csv~  hbase-0.90.0.tar.gz
firefox                        hbase-0.98.14-src.tar.gz
firefox-40.0.3.tar.bz2         hbase-1.1.2-bin.tar.gz
hadoop-core-0.20.2-cdh3u0.jar  jdk1.8.0_60
hbase-0.90.0                   spark-1.5.1-bin-hadoop2.6.tgz
[acadgild@localhost Downloads]$
```

2. After untaring the file we need to move the file to the Home Folder using the below command:

sudo mv spark-1.5.1-bin-hadoop2.6 /home/acadgild

The above command moves the file to the Home folder.
We need to update the path for spark in the .bashrc in the same way as we did for scala.

3. Refer the given screenshot for updating the path for .bashrc

```
.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export SPARK_HOME=/home/acadgild/spark-1.5.1-bin-hadoop2.6
export PATH=$PATH:$SPARK_HOME/bin
```

4. After adding the path for SPARK, type the command source .bashrc, refer the screenshot for the same.

```
[acadgild@localhost ~]$ source .bashrc
[acadgild@localhost ~]$
```

5. Make a folder by Name 'work' in HOME using the below command:

```
mkdir work
```

6. Inside the work folder we need to make another folder by name 'sparkdata' using the command

```
chmod 777 $HOME/work/sparkdata
```

We need to give the permissions to the sparkdata folder as 777 using the below command.

```
[acadgild@localhost ~]$ chmod 777 $HOME/work/sparkdata
[acadgild@localhost ~]$
```

7. Now move into the conf directory of spark folder using the below command

```
cd spark-1.5.1-bin-hadoop2.6

cd conf
```

Type the command ls to see the files inside conf folder:

- There will be a file by name spark-env.sh.template, we need to copy that file by name spark-env.sh using the below command:

```
1 cp spark-env.sh.template spark-env.sh
```

```
[acadgild@localhost ~]$ cd spark-1.5.1-bin-hadoop2.6/
[acadgild@localhost spark-1.5.1-bin-hadoop2.6]$ cd conf
[acadgild@localhost conf]$ ls
docker.properties.template  metrics.properties.template   spark-env.sh.template
fairscheduler.xml.template  slaves.template
log4j.properties.template   spark-defaults.conf.template
[acadgild@localhost conf]$ cp spark-env.sh.template spark-env.sh
```

Edit the spark-env.sh file using the below command:

```
1 gedit spark-env.sh
```

and make the configuration as follows

```
#Setting Java Path
export JAVA_HOME=/usr/lib/jvm/java-1.7.0
export PATH=$PATH:$JAVA_HOME/bin

#scala path
export SCALA_HOME=$HOME/scala-2.11.1
export SPARK_WORKER_MEMORY=1g
export SPARK_WORKER_INSTANCES=2
export SPARK_WORKER_DIR=$HOME/work/sparkdata
```

Note: *Make sure that you are giving the paths of Java and Scala correctly. After editing save the file and close the file.*

Let's follow the below steps to start the spark single node cluster.Move to the sbin directory of spark folder using the below command:

```
1 cd spark-1.5.1-bin-hadoop2.6/sbin
```

File Edit View Search Terminal Help

```
[acadgild@localhost ~]$ cd spark-1.5.1-bin-hadoop2.6/sbin
[acadgild@localhost sbin]$
```

Inside sbin type the below command to start the Master and Slave daemons.

```
1 | ./start-all.sh

[acadgild@localhost sbin]$ ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /home/acadgild/spark-
1.5.1-bin-hadoop2.6/sbin/../logs/spark-acadgild-org.apache.spark.deploy.master.M
aster-1-localhost.localdomain.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/acad
gild/spark-1.5.1-bin-hadoop2.6/sbin/../logs/spark-acadgild-org.apache.spark.depl
oy.worker.Worker-1-localhost.localdomain.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/acad
gild/spark-1.5.1-bin-hadoop2.6/sbin/../logs/spark-acadgild-org.apache.spark.depl
oy.worker.Worker-2-localhost.localdomain.out
[acadgild@localhost sbin]$ jps
8057 Jps
7808 Master
8018 Worker
7954 Worker
[acadgild@localhost sbin]$
```

- Now the spark Single Node cluster will start with One Master and Two Workers.
- You can check that the cluster is running or not by using the below command **'jps'**

If the Master and Worker Nodes are running then it means you have successfully started the spark single node cluster.