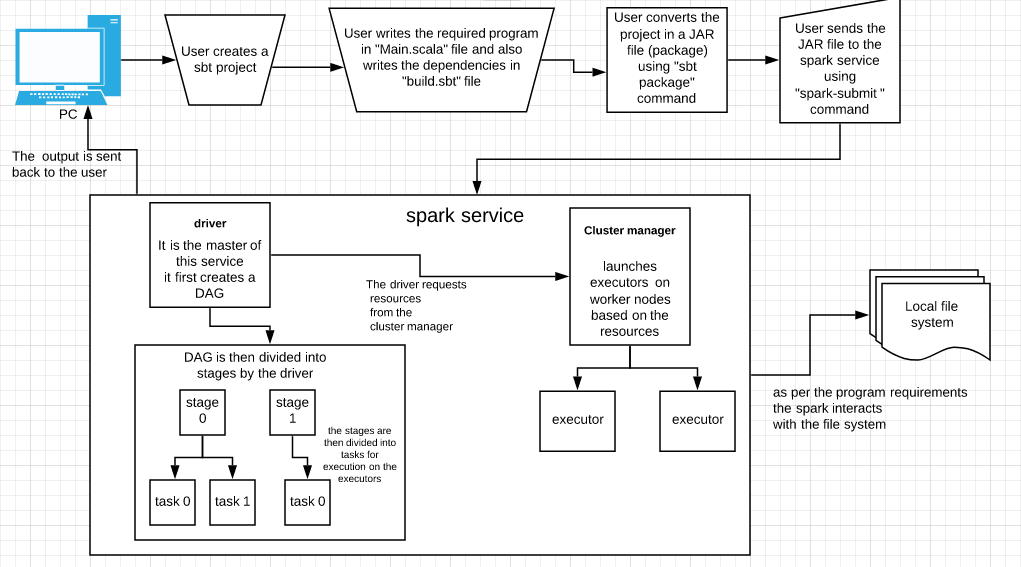
**PoC spark with scala**

* Why to use Spark ?

We can use various methods to read or to process a file. But spark gives us the functionality of processing the file more quickly.

In spark the data is stored in RDD’s which logically divides the data into partitions and this partitions are then sent to different nodes where they use the resources of the nodes for the processing.

* How it works ?



* Steps for installation

1.check whether the system has java if not follow the steps below.

a. sudo apt-add-repository ppa:openjdk-r/ppa

b.sudo apt-get update

c.sudo apt-get install openjdk-8-jdk

d.java -version

2.download spark file

1. wget\ http://apachemirror.wuchna.com/spark/spark-3.0.0-preview/spark-3.0.0-preview-bin-hadoop2.7.tgz

3.extract " tar -xzvf spark-3.0.0-preview-bin-hadoop2.7.tgz"

4.nano ~/.bashrc

5.go to the end of file and write

export SPARK\_HOME=/home/administrator/spark

export PATH=$PATH:$SPARK\_HOME/bin

6.source ~/.bashrc

7.cd /home/administrator/spark/conf

8.cp spark.env.sh.template spark.env.sh

9.nano spark.env.sh

10.go to end of file and write

export JAVA\_HOME =/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

11. Install sbt

1. echo "deb https://dl.bintray.com/sbt/debian /" | sudo tee -a /etc/apt/sources.list.d/sbt.list
2. curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64E40A89B84B2DF73499E82A75642AC823" | sudo apt-key add
3. sudo apt-get update
4. sudo apt-get install sbt

* Create sbt project

1.mkdir trial

2. cd trial

3.mkdir -p src/main/scala/

4.mkdir project

5.mkdir target

6.touch build.sbt

7.cd src/main/scala/

8.touch Main.scala

9.cd ~/trial

10.sbt

Now we can write the required program inside “Main.scala” file and also add dependencies required to run that program in “build.sbt”file

* Program inside “Main.scala” file

package og.sharad.examples

import org.apache.spark.sql.Row

import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.types.\_

object Sales extends App

{

val spark = SparkSession

.builder()

.appName("Spark SQL basic example")

.getOrCreate()

val df = spark.read.format("csv").option("header","True").option("inferschema","True").load("/home/administrator/Sales.csv")

df.createOrReplaceTempView("data")

val sqlDF = spark.sql("SELECT \* FROM data").show(5)

spark.sql("select Country,to\_date(`Order Date`,'M/d/yyyy')as date,`Total Profit` from data ").filter("date>'2014-01-01' and date<'2014-01-31'").show

}

* Code inside “build.sbt” file

scalaVersion := "2.12.10"

name := "trial"

organization := "og.sharad"

version := "1.0"

libraryDependencies++=Seq("org.apache.spark"%%"spark-core"%"3.0.0-preview",

"org.apache.spark"%%"spark-sql"%"3.0.0-preview")

* Steps for execution

1.cd ~/trial

2.sbt package

3.cd ~/spark/bin

4. ./spark-submit --class “name of the package given inside Main.scala followed by the name of the entry point ” /home/administrator/trial/target/scala-2.12.10/\*.jar

* Advantages

The processing in spark is 100 times better than in hadoop mapreduce.

Spark can also be used without hadoop by using different resource managers like mesos,kubernetes etc.

Spark RDD are lazy in nature i.e it does not give the result on transformation until an action has occurred.

* Disadvantages

The machines used in processing with spark require high amount of RAM.

Spark does not support real-time processing rather it is near real-time processing.

It does not have its own file management system rather it relies on other systems like hadoop or other cloud based platform.