Big Data Summer Training

BigData Analytics-BigData Platforms

Prepared and Presenting By: Amrit Chhetri (Certified BigData Analyst),
Principal Techno-Functional Consultant and Principal IT Security Consultant

Pig:BigData Scripting Language

Agendas/Modules:

- Pig Introduction
- Pig Commands
- Pig Installation

Pig Introduction :

- Pig is a Scripting Language
- Supports UDF(User Defined Functions)with Java, Python, Scala, etc.
- It is simpler than SQL Statements
- Pig's Scripting file is saved as .sql

Pig Commands:

- The steps to Pig Commands are:
  - Type $usr/local/pig/bin pig, gives grunt>
- Word Count Sample:
  - grunt> linesString=LOAD '/amritchhetrib/hadoop/data/words.txt' AS (line:chararray);
  - grunt> words=FOREACH linesString GENERATE FALTTERN(TOKENIZE(line)) as word;
  - grunt> grouped=GROUPED words BY word;
  - grunt> wordcount= FOREACH grouped GENERATE group, COUNT(words);
  - grunt> DUMP wordcount
- User Enumeration:
  - $ pig -x localgrunt> data = load '/etc/passwd' using PigStorage(':');
  - grunt> loop= foreach data generate $0 as id;
  - grunt> output = limit loop 5;
  - grunt> dump output;

Pig Installation:

- Requirements Pig with Hadoop 2.7.2:
  - JDK 1.7 or JDK 1.8
  - Hadoop 2.7.2
- Typical steps of Pig configurations:
  - Extract pig and place it inside /usr/local/ping
  - Open ~/.bashrc and make entries for Pig's PATH and CLASSPATH
- Running Pig:
  - $ /usr/local/pig/bin pig –x local
  - $ /usr/local/pig/bin pig –x mapreduce

Apache Spark: In-Memory Computation

Agendas/Modules:

- Spark Introduction
- Features of Spark
- Components of Spark
- Spark Example

Spark Introduction:

- Fast and general-purpose engine for BigData Processing and Analysis
- Spark is lightening fast BigData Computing Platform
- Spark was originally written in Scala
- Supports Java, Scala, Python and R
- Supports advanced analytical capability
- Supports data-accessibility from HBase, Hive, Cassandra, HBase and Tachyon
- It is faster than Hadoop, at multiples of 100
- It stores data into Hadoop i.e Hadoop HDFS
- It support in-memory processing and computation
- Spark SQL and DataFrames provide similar functionalities
- Spark supports two Context- Spark SQLContext and Spark HiveContext

Features of Spark:

- Spark runs in three modes -Standalone, Hadoop Yarn and Spark on MapReduce
- RDD(Resilient Distributed Database) is fundamental data structure of Spark
- Supports Programming Abstraction as DataFrames
- DataFrames can act as distributed SQL
- RDD is core component of Apache Spark(Resilient Distributed Datasets)

Components of Spark:

- Spark SQL
- Spark Streaming
- MLib (Machine Learning)-Distributed machine learning component
- GraphicX -Distributed graph processing framework

Spark Example:

- Load data file into HDFS:
  - # hadoop dfs -put /usr/spark/DataFile.txt /usr/spark/input1
  - # hadoop dfs -ls /usr/spark/input1
- Install PySpark :
  - # sudo easy_install ipython==1.2.1 (if not installed
  - # PYSPARK_DRIVER_PYTHON=ipython pyspark
  - spark> txt_DATA=sc.textFile("/amritchhetrib/data/input1/DataFile.txt")
  - spark> txtData.take(1))

Scala Programming on Linux

Agendas/Modules:

- Scala Installation on Ubuntu
- Eclipse and Scala Plugin

Scala Installation on Ubuntu:

- Ubuntu supports Scala in Terminal as well as with IDE(Eclipse)
- Run $ sudo apt-get install Scala , to install Scala on Ubuntu
- Open terminal and type $ Scala , it gives Scala> prompt
- Examples:
  - Scala> var x-34
  - Scala> var y=45
  - Scala> var z=x+y
- Script Execution:
  - $ scala <file.scala> arguments

Eclipse and Scala Plugin:

- Follow the steps below to install Scala Plugin using archive(.jar) file
  - Download the .zip file of Scala Plugin for Eclipse
  - Unzip it and start Eclipse
  - Click on Help-> New Software Installation-> select ‘Local ‘ folder and select contents.jar from there
  - Select the module you want to install
  - Click on Preference and select scala installation folder
  - Create Scala Project and write Scala Programs