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

knowledgelab SM
Incubated @ STEP IIT (Kgp)

About Me :

- Me:
  - I'm Amrit Chhetri from Bara Mangwa, West Bengal, India, a beautiful Village/Place in Darjeeling.
  - I am CSCU, CEH, CHFI,CPT, CAD, CPD, IOT & BigData Analyst(University of California), Information Security Specialist(Open University, UK) and Machine Learning Enthusiast (University of California[USA] and Open University[UK]), Certified Cyber Physical System Exert(Open University[UK]) and Certified Smart City Expert.
- Current Position:
  - Principal IT Security Consultant/Instructor, Principal Forensics Investigator and Principal Techno-Functional Consultant/Instructor
  - BigData Consultant to KnowledgeLab
- Experiences:
  - I was J2EE Developer and BI System Architect/Designer of DSS for APL and Disney World
  - I have played the role of BI Evangelist and Pre-Sales Head for BI System* from OST
  - I have worked as Business Intelligence Consultant for national and multi-national companies including HSBC, APL, Disney, Fidedality , LG(India) , Fidelity, BOR(currently ICICI), Reliance Power. * *Top 5 Indian BI System (by NASSCOM)*

BigData Analytics Platforms Configurations

Agendas/Modules:

- BigData Advanced Analytics Tools
- BigData Platforms Preparations
- Introduction to Hive-Syncfusion
- Introduction to Hive-Qubole
- Hadoop Configurations Requirements

BigData Advanced Analytics Tools:

- Hadoop Analytics is pointed to extract data from heterogeneous sources in Hadoop System
- The common Data Storage systems in Hadoop are HDFS, Hive, HBase, Logs, No-SQL(Mongo, Cassandra, Couch).
- The most effective tools for Statistical Analysis of MapReduce Data are
  - MATLAB
  - Octave
  - R
  - Spark
- The tools which are used to move data in this ecosystem is called ETL , Extraction, Transformation and Load and they
  - Scoop
  - Pentahoo ETL, Talend Studio

BigData Platform Preparations:

- The common distributions of Hadoop are :
  - Installer and VM: Cloudera ,MapR , Hortonworks , Syncfusion
  - Hadoop-As-A-Service : Qubole, MS Azur, Amazon AWZ EC2
  - Self-Made Quick-Start Hadoop(VM/Standard) : Apache Hadoop 2.7.2 on Ubuntu 15.10
- Hadoop 2.7.2 can be configured on Ubuntu 15.10 for self-made Hadoop Stack or 'Self-Made Quick-Start Hadoop(VM/Standard)'
- Advantages of Self-Made Hadoop QuickStart VM are:
  - Completely Open Source, no Licensing issues
  - Standard configurations for additional or newer components
  - Availability of tons of Free and Open Sources resources and tools or frameworks
- Disadvantages of Self-Made Hadoop QuickStart VM are:
  - Compliances, Compatibility
  - Unavailability of Professional Services

Introduction to Hive-Syncfusion:

- Syncfusion’s Hadoop distribution is available as Single-Node or Cluster-environment. It is the most easiest distribution for Windows Platform.
- Steps to run Hive on Syncfusion Platforms
  - Install MS .Net Framework and install Syncfusion Studio on Windows Machine
  - Open Syncfusion Studio and run ‘Command Shell’ available at top of Syncfusion Studio
  - Type hive to start the HIVE prompt, hive>
  - Table Create HiveQL : hive> CREATE TABLE PRD(id int, category int);
  - Data Insert HiveQL : hive>INSERT INTO PRD VALUES (1, 100);
  - Select HiveQL : hive>SELECT id, category FROM PRD;
- The common HiveQL statement
  - hive> show databases ;
  - hive> use <database name>;
  - hive> show tables;
  - Hive> describe <table>

Introduction to Hive-Qubole:

- Qubole is a HAAS (Hadoop-AS-A-Service) platform and it is also available for free educational use too.
- Simple Signup or logging in with Google account allows to access all BigData Components/Tools/Frameworks available inside it.
- HiveQL Examples on Qubole:
  - Open http://qubole.com and log in using Google account
  - hive> show databases ; hive> use <database name>;
  - hive> show tables;Hive> describe <table>
- The common HiveQL statement:
  - CREATE : CREATE logs(ip string, size string, time string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ‘
  - SELECT : SELECT * FROM logs;
  - UPDATE : UPDATE logs SET ip='192.168.2.10' WHERE time='2:30';
  - ALTER : ALTER TABLE logs COLUMN MODIFY (time STRING);
  - WHERE : select * from logs WHERE Ip='192.168.2.10';
  - GROUP BY : SELECT COUNT(*), logs.ip,count(*) FROM logs logs GROUP BY logs.ip :

Hadoop Configuration Requirements:

- OS Requirement: Ubuntu 15.10 and Hadoop Version: 2.7.2
- Generic Steps are below and ‘Hadoop Configuration Guide-2.7.2-15.10.txt’ detailed steps by me(Amrit Chhetri):
  - Install Ubuntu 15.10 either on Dual boot or as Virtual
  - Install Java
  - Add a Hadoop user
  - Install SSH and configure SSH certificates
  - Check whether SSH works or not
  - Install Hadoop 2.7.2
  - Modify Hadoop configurations files
  - Format Hadoop Files system(HDFS- Hadoop Distributed File System)
  - Start Hadoop
  - Check Hadoop using Web Interface
  - Run Word Count's jar file
  - Stop Hadoop

Programming With Scala

Agendas/Modules:

- Scala Programming Platforms
- Programming Scala on Eclipse
- Pig UDF using Scala
- Scala Programming Fundamentals-I

Scala Programming Platforms:

- Scala is a high-level Programming and Scripting Language
- Spark, In- Memory System was initially developed using Scala
- On Windows, Scala is installed inside C:\Program Files (x86)\scala\bin folder
- Running scala.exe gives Scala prompt which is be used to execute Scala commands
  - scala> var1=20
  - scala> var2=30
  - scala> var sum=var1+var2, unlike in Octave, MATLAB and R, Scala requires var keyword
- Scala is also used to call Pig Scripts, Hive Scripts and Spark RDD(Resilient Distributed Datasets)
- Scala follows the programming paradigm of Java with its own programming constructs
- Scala scripts or programs is saved with .scala extension

Programming Scala on Eclipse:

- Steps to write Scala Scripts using Eclipse are:
  - Download and install Scala Compiler or Environment
  - Install Eclipse and add Scala Plugin using this URL http://download.scala-ide.org/sdk/lithium/e44/scala211/stable/site
  - On Scala Prospective, right-click on your project and select "Add Scala Nature" in "Configure" menu
  - Create Scala Project and start developing codes on Scala
- Scala Example (MaxValue.scala), it can be executed as bin\scala.exe MaxValue.scala

```
 val arrayObj= new Array[String] (5)
arrayObj(0) = "Android\n"
arrayObj(1) = "Blackberry\n"
arrayObj(2) = "Windows\n"
arrayObj(3) = "Tizen!\n"
arrayObj(4) = "Firefox!\n"
for (i <- 0 to 4)
   print(arrayObj(i)
```

Pig UDF using Scala:

- UDF(User Defined Function) are customized mechanism of extending features of Pig, Hive and Sqoop .
- Pig UDF is supported by Scala and it can be written on Eclipse IDE
- Pig's jar file, Pig.jar is needed in build-path to compile Pig UDF on Scala
- The jar file containing code extending Pig feature is registered using REGISTER call
- The functions written inside the registered jar are involved inside the ping script or inside another UDF

Scala Programming Fundamentals:

- Scala Programs or Scripts are saved as .scala file
- Variables are specified using var keyword no data types assigned while declaring a variable
- Steps to run Scala Scripts:
  - Install Scala using scala installer
  - Write Scala scripts
  - Save it with .scala extension
  - Run as ..bin\scala <script name>.scala

Advanced Python Programming

Agendas/Modules:

- MapReduce using MRJob-Advanced
- Python Regular Expressions
- Advanced Web Data Programming
- Web Data Streaming using Tweepy
- Pig UDF using Python

MapReduce using MRJob-Advanced:

- Marjob performs MapReduce using mapper(), reducer() and combiner()
- Example:

```
from mrjob.job import MRJob ; import re
expression= re.compile(r"[\w']+")
class WordCount(MRJob):
    def mapper(self, _, line):
        for word in expression.findall(line):
            yield (word.lower(), 1)
    def combiner(self, word, counts):
        yield (word, sum(counts))
    def reducer(self, word, counts):
        yield (word, sum(counts))
if __name__ == '__main__':
    WordCount.run()
```

Python Regular Expressions:

- Regular expression is UNIX-style expression using character sequence
- Regular expression is achieved by importing re module
- The common function are:
  - re.match(pattern, string, flag=0) ; re.search(pattern, string)
  - re.findall(pattern, string)
- Example:

```
 import re
fo = open("data.txt", "r") ;    line = fo. readline();    words=line.split(" ")
for word in words:
    if re.search("Data", word):
         print("Data is there")
    else:
        print("Data Not Found")
```

Advanced Web Data Programming:

- Streaming of Twitter feeds is performed using Tweepy
- The steps to perform Twitter streaming are:
  - Create a Twitter Account , if does not have
  - Get Consumer Key and API Key by accessing
    - https://dev.twitter.com/oauth/overview and https://apps.twitter.com/app
  - Write Python code to accessing Twitter feeds using Tweepy
  - Save the tweets or feeds into Database capable of storage larger volume of data – MongoDB or Cassandra, etc
- The data populated by tweepy is loaded either loaded directly into HDFS folders for MapReduce jobs or to Hive, Hbase or saved to HDFS as intermediary or final outcomes
- Twitter Sentimental Analysis is a common application possible with Tweepy
- Tweepy-Python code using Twitter keys and REST API to get live feed

Web Data Streaming using Tweepy:

- Tweepy Example:

```
import tweepyimport jsonc

onsumer_key = 'consumer_key from twitter account‘

consumer_secret = ' Consumer secret key consumer_secret‘

access_token = 'access_token‘

access_token_secret = 'access_token_secret‘


class TweetStreaming(tweepy.StreamListener):

         // Code is inside Tweepy Example Code

if __name__ == '__main__':

      l = TweetStreaming() ; auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
      auth.set_access_token(access_token, access_token_secret)

      stream = tweepy.Stream(auth, l)

       stream.filter(track=['WhatsApp'])
```

Pig UDF using Python:

- Python is also used to write or develop UDF- User Defined Function for
  - Pig
  - Hive
  - Spark (pySpark)
- Hadoop supports Pig UDF using Python till 2.0 or lower version
- Example of Pig UDF using Python (Demostrated during Training Session)

BigData Analytics Fundamentals

Agendas/Modules:

- BigData Analytics Tools
- Building Report using BIRT
- Machine Data Analysis using Splunk

BigData Analytics Tools:

- BigData Analytics Tools are used to generate Analytics or Reports from BigData
- The common BigData Analytics Tools are
  - BIRT - Open Source Analytics supported by Actuate
  - Kognitio -In-memory Analytics (Industrial scale) : http://kognitio.com/
  - Spartk - Apache Open Source Project
  - SAP BO(Business Objects), Actuate One, etc
- BigData Analytics comprises 4 category of components
  - BigData Data Processing Platforms : Hadoop, Tez
  - BigData ETL : Sqoop, Pentahoo, Informatica, Talend Studio
  - Statistical or Machine Learning Platforms : R, Octave, MATLAB, Spark's Mllib
  - Visualization Platforms : BIRT, Kognitio, Actuate One, Custom Apps(Mobiles Apps, Web App, Standalone/Desktop Apps)

Building Report using BIRT:

- BIRT is one of the Report Designer for BigData
- BIRT support accessing data from Hive, HDFS and Hbase
- BIRT also supports JDBC Connectivity to various database
- BIRT works fine with Open Source ETL Tools like Pentahoo and Talend Studio
- Designing Report using BIRT:
  - Install BIRT Plugin or get BIRT Report Designer
  - Create BIRT Project and create data-sources
  - Create Result-Set using BIRT’s Query Editor
  - Select the type of Report and put the columns of your interest on Report
- In Pentahoo ETL, all drivers programs including JDBC are loaded from /lib folder, new driver’s jar are installed inside that folder

Machine Data Analysis using Splunk:

- Splunk is Open Source Tool for machine generated data
- It support Search and it has it own search engine and syntaxes and it works on Windows and Linux
- It is also used on Analyzing different types of logs generated by BigData System , including Apache Web Server, Weblogic Application Server
- Splunk's search commands are piped or joined together using | symbol and Splunk commands examples:
  - search | command1 argument1 | command2 argument 1
- Common Splunk Functions:
  - len(x) : Returns length of given string, x
  - Max(args1, args2…): Returns maximum value
  - count(x) : Returns the counts of occurrence of a given characters, x

THANK YOU ALL

knowledgelab℠

Incubated @ STEP IIT (Kgp)