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)*

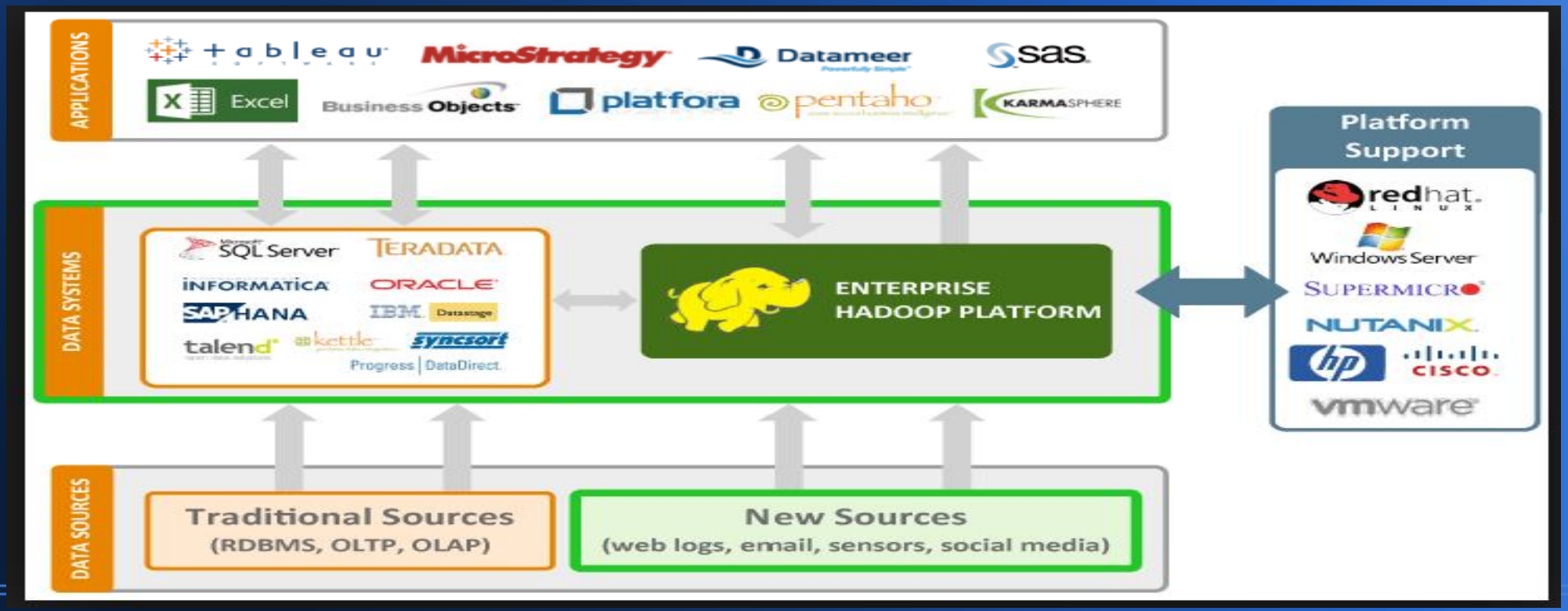
Hadoop-BigData Core Engine

Agendas/Modules:

- BigData Analytics Architecture
- Hadoop Introduction
- Hadoop Components
- Hadoop Administration
- Hadoop MapReduce
- Hadoop MapReduce Programming on Eclipse
- MapReduce -mrjob

BigData Analytics Architecture:

- BigData Analytics is a distributed and clustered Data Analysis Platform
- Apache Hadoop supports MapReduce and leverages MapReduce services for Hive, HBase, Pig and Spark
- A typical BigData Architecture:



Hadoop Introduction:

- Open Source, scalable, reliable and robust BigData processing Platform and it is a Framework for storing and processing huge volume of data from heterogeneous sources
- It supports Distributed Processing of data across Hadoop Clusters or Standalone Node
- It is written in Java and supports simple Programming Paradigm
- Supports Distributed Processing and limitless MapReduce Jobs or Tasks
- It supports 3Vs(Volume, Variety and Velocity) and also supports Verasity
- Open source Software Framework for storing data and running applications on clusters of Commodity Hardware
- Top Hadoop Distributions are available from Cloudera, MapR, Syncfusion, Apache and Hortornworks .The original distribution is from Apache .
- Supports both single node(Standalone) and Cluster of Computers to run jobs parrallely
- It leverages HDFS(Hadoop Distributed File System) services to Spark, HBase, Hive and Tez
- It gives connectivity support for external tools like Toad for BigData, Pentahoo and Talend Studio for Data Analysis

Hadoop Components:

- Hadoop keeps or stores large-volume of data using a distributed file system known as HDFS (Hadoop Distributed File System) and it is based on Google GFS.
- The components of Hadoop are:
  - Hadoop Common-Core Engines
  - HDFS(Hadoop Distributed File System)
  - YARN
  - Hadoop MapReduce
- Hadoop is used for :
  - Massive un-structured and structured Data Processing
  - BigData Storage using HDFS
  - Batch Processing of Data and MapReduce Functionality-Java, Python , C/C++, PHP
  - Distributed Data Storage(HDFS) for Pig, Tez, Hive, HBase, Spark and Cloudera Impala

Hadoop Administration:

- Machine generated Hadoop logs or data are important sources of data in Hadoop Administration
- Hadoop HDFS and Logs are accessed in two ways- HDFS's Web UI and CLI
- Analysis of those logs is crucial in Hadoop Administration and monitoring
- Splunk is used to analyze the machine generated data or logs of Hadoop System
- Hadoop YARN and Apache Spark's logs two primary sources information in Hadoop Information
- Apache Flume is another great Frameworks for BigData Processing

Hadoop MapReduce:

- Hadoop MapReduce is a process or mechanism or way of producing a final result from intermediate results generated by Reducer using different mappings.
- The intermediate results generated by MapReduce jobs are stored inside HDFS(Hadoop Distributed Files System)
- MapReduce are generally written in Java but it supports other languages including
  - C/C++, C#, Scala
  - Python, Ruby etc
- A MapReduce Program reads data from Hadoop's HDFS and same should be updated intelligently using scripts or programs in real applications or real-time BigData Analytics solution

Hadoop MapReduce Programming on Eclipse:

- Hadoop-Eclipse module which available in Github is used to write MapReduce on java using Eclipse
- Steps for Running MapReduce on Eclipse
  - Install JDK on Windows or Ubuntu Linux (14.04 LTS, 15.10 and 16.10)
  - Copy Hadoop-eclipse Plugin inside /pugins directory and re-start Eclipse
  - Click on New Project and select MapReduce and give the project name
  - Writes all three classes- Map class, Reducer class and Driver class
  - It can be run in two – by creating .jar file and secondly directly from Eclipse
- MapReduce on Hadoop 2.7.2 supports :
  - Python ,Java ,Scala
  - C/C++, Scala

MapReduce on Cloudera:

- The steps to MapReduce on Java using Cloudera Quick Start VM:
  - Open a Linux Terminal on Cloudera QuickStart VM
  - Change Directory to /usr/lib/hadoop-mapreduce/
  - Type jsp localhost to check Hadoop
  - Execute, # hadoop jar WordCount.jar WordCountDriver , (Reports missing files)
  - Create two text files:
  - echo "Senssor is important elements of IoT"> /home/cloudera/testfile1
  - echo "IoT is the future of intelligent computation"> /home/cloudera/testfile2
  - Create input directory on HDFS,# hdfs dfs -mkdir /user/cloudera/input
  - Copy the files to Hadoop input directory
  - # hdfs dfs -put /home/cloudera/testfile1 /user/cloudera/input
  - # hdfs dfs -put /home/cloudera/testfile2 /user/cloudera/input
  - Execute,#hadoop jar WordCount.jar WordCountDriver /user/cloudera/input /user/cloudera/output
  - Check output directory, # hdfs dfs -ls /user/cloudera/output

MapReduce using Mrjob:

- mapper(), reducer() and combiner() are common functions of mrjobs and 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()
```

Advanced Python

Agendas/Modules:

- Advanced Python- Tweepy
- Python In-Built Functions

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 one
  - 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

Python In-Built Functions:

- Python has huge set of built-in Functions for mathematical and statistical analysis and common of them :
  - pow(x,y) : It returns the power of the first number raised to the second number
  - random.random() : Generates values between 0.0 to 1.0
  - Random.randint(min, max) : Generates integer number randomly between minimum and maximum values
  - math.sqrt(number): Return the square root of the given function
  - Math.pi : Return the value of PI
  - Os.getcwd() : Gives current working directory
  - Os.system() : Allows to run OS commands

BigData, ETL and Analytics Designing

Agendas/Modules:

- ETL with Pentathoo
- ETL with Sqoop
- BIRT Report Designing -MySQL

ETL with Pentathoo:

- Pentahoo is an Open Source ETL Tool for BigData
- It is used to move data to BigData Warehouse designed using HBase, Hive and other technology
- The JDBC drivers are placed inside lib folder and all standard calls are from there.
- Steps to transform data using Pentahoo:
  - Install JDK on Windows or Linux (or Ubuntu 14.04, 15.10, etc)
  - Install Pentahoo ETL to and run it
  - Create a transformation and create sources for targeted source like HBase, Hive and others
  - Draw the transformation-mapping on main screen
  - Now, either schedule the job to run in future and to run immediately

ETL with Sqoop:

- Sqoop is Data Integration Service developed on Open Source Hadoop Technology
- Sqoop is is transform data between Hadoop Cluster and Database using JDBC ,in bi-direction
- Scoop- Data Integration Steps- Moving to HDFS:
  - import --connect jdbc:mysql://<DB IP>/database --table orders --username <DB User> –P
- Data Integration Steps- Moving to Hive:
  - #sqoop import --hive-import --create-hive-table --hive-table orders –connect jdbc:mysql://<DB IP>/database --table orders --username <DB User> -P <password>

Hive-Query Processing Engine

ETL with Talend Studio:

- Talend is the world-class ETL tool available for BigData and the Data Systems
- It is used to move data to BigData Warehouse designed using HBase, Hive and other technology
- Steps to transform data using Talend Studio:
  - Install JDK on Windows or Linux (or Ubuntu)
  - Install Talend Studio to and run it
  - Create a transformation and create sources for targeted source like HBase, Hive and others
  - Draw the transformation mapping on main screen
  - Now, either schedule the job to run in future and to run immediately

Agendas/Modules:

- Introduction to Hive
- Introduction to HiveQL
- Writing HiveQL Statements

Introduction to Hive:

- DW System for processing un-structured data
- DW Software for querying and managing large datasets
- DW Infrastructure built on top of Hadoop
- High Level Data Processing/Query Language
- Works on top of HDFS(Hadoop Distributed File System) of Hadoop
- Supports HDFS as Storage, MapReduce, Execution and HiveQL for Query execution
- Stores Schemas or Meta-Data on Database
- Derby is default database for Hive but it supports others too(Oracle MySQL)
- Support Apache Spark(In-Memory, Machine Learning and Graphing API)
- Hive Components are:
  - Driver ,Query Compiler , Optimizer
  - NameNode
  - HiveServer2 Engine

Introduction to HiveQL:

- HiveQL is SQLI-like database querying language
- HiveQL runs slowly using MapReduce of Hadoop
- HiveQL is similar to SQL but functions and architectural flow is completely different
- Driver of Hive sends the Hive Query to optimizer before actually it runs
-

Writing HiveQL Statements:

- Sample Examples:
  - CREATE : CREATE logs(ip string, size string, time string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ‘
  - LOADING : LOAD DATA LOCAL INPATH 'log.log' OVERWRITE INTO TABLE logs;
  - SELECTING : 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;
- HiveQL execution uses Hadoop MapReduce Services

THANK YOU ALL