Principles & Practice to Apache Hive :

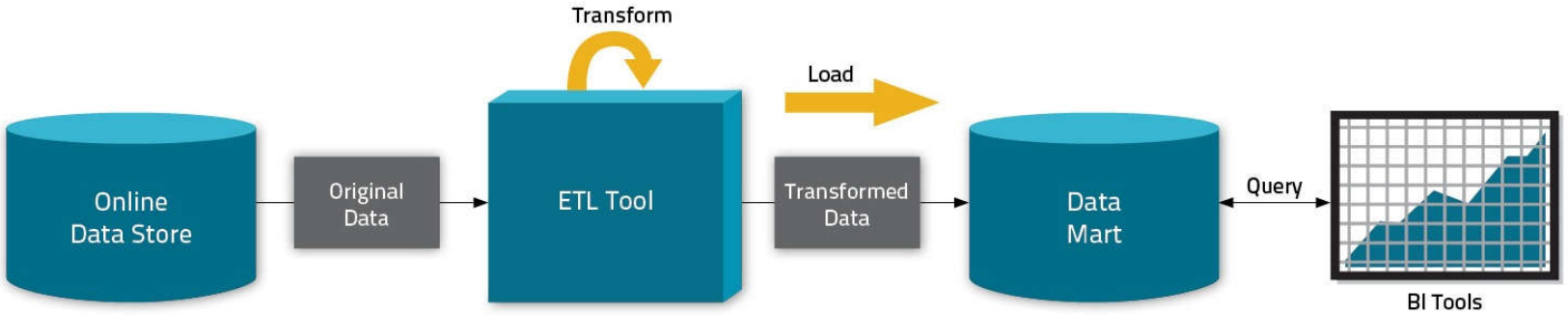
Name: อวยชัย ภิรมย์รื่น

Tel. : 086-813-5354

e-mail : p.Auoychai@gmail.com

ทำความรู้จัก Apache Hive :

Hadoop data Challenge with data processing :



http://blog.cloudera.com/blog/2013/02/big-datas-new-use-cases-transformation-active-archive-and-exploration/

ตัวอย่าง Application สำหรับ Hive :

- Summarization
  - Eg: Daily/Weekly aggregations of impression/click counts
  - Complex measures of user engagement
- Ad hoc Analysis
  - Eg: how many group admins broken down by state/country
- Data Mining (Assembling training data)
  - Eg: User Engagement as a function of user attributes
- Ad Optimization

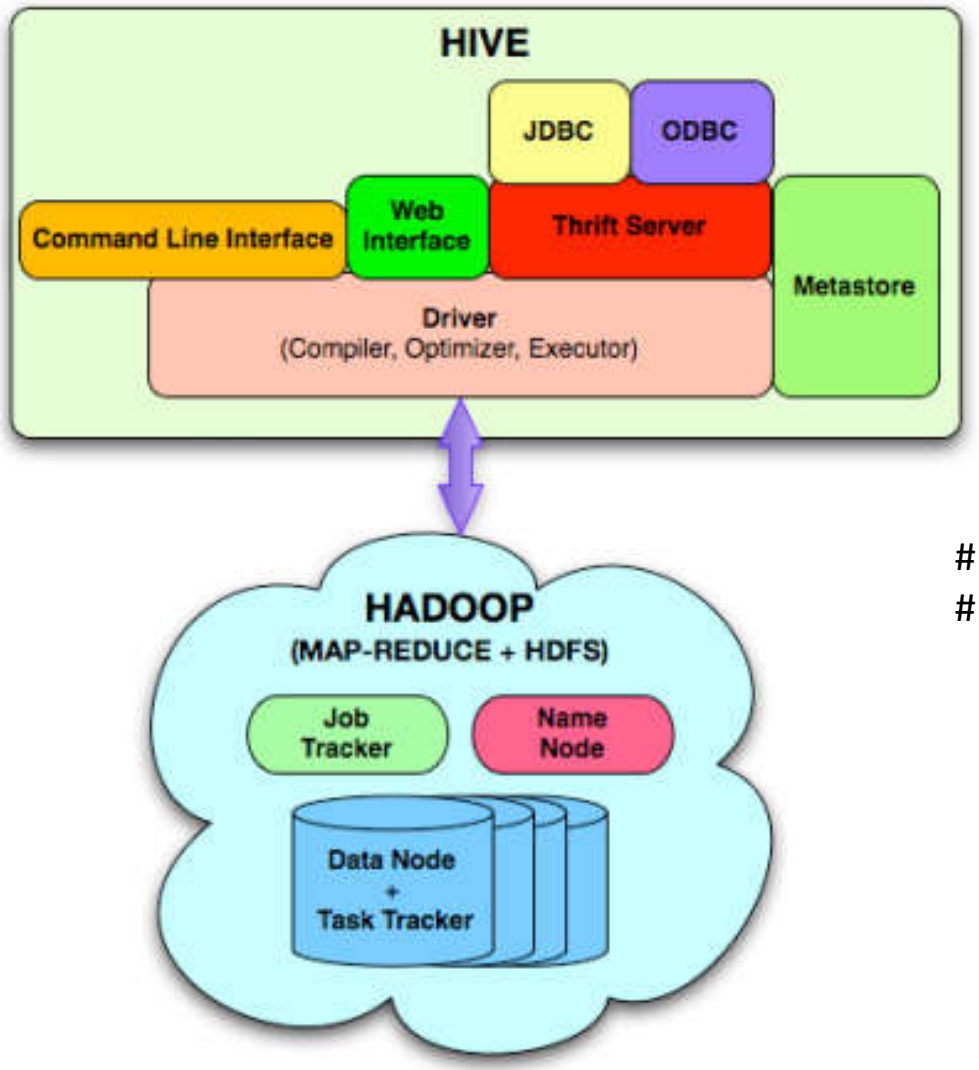
 # ตัวอย่าง Application สำหรับ Hive :

Hadoop Usage @ Facebook

- Data statistics:
  - Total Data: 180TB (mostly compressed)
  - Net Data added/day: 2+TB (compressed)
    - 6TB of uncompressed source logs
    - 4TB of uncompressed dimension data reloaded daily
- Usage statistics:
  - 3200 jobs/day with 800K tasks(map-reduce tasks)/day
  - 55TB of compressed data scanned daily
  - 15TB of compressed output data written to hdfs
  - 80 MM compute minutes/day

Hive Component :

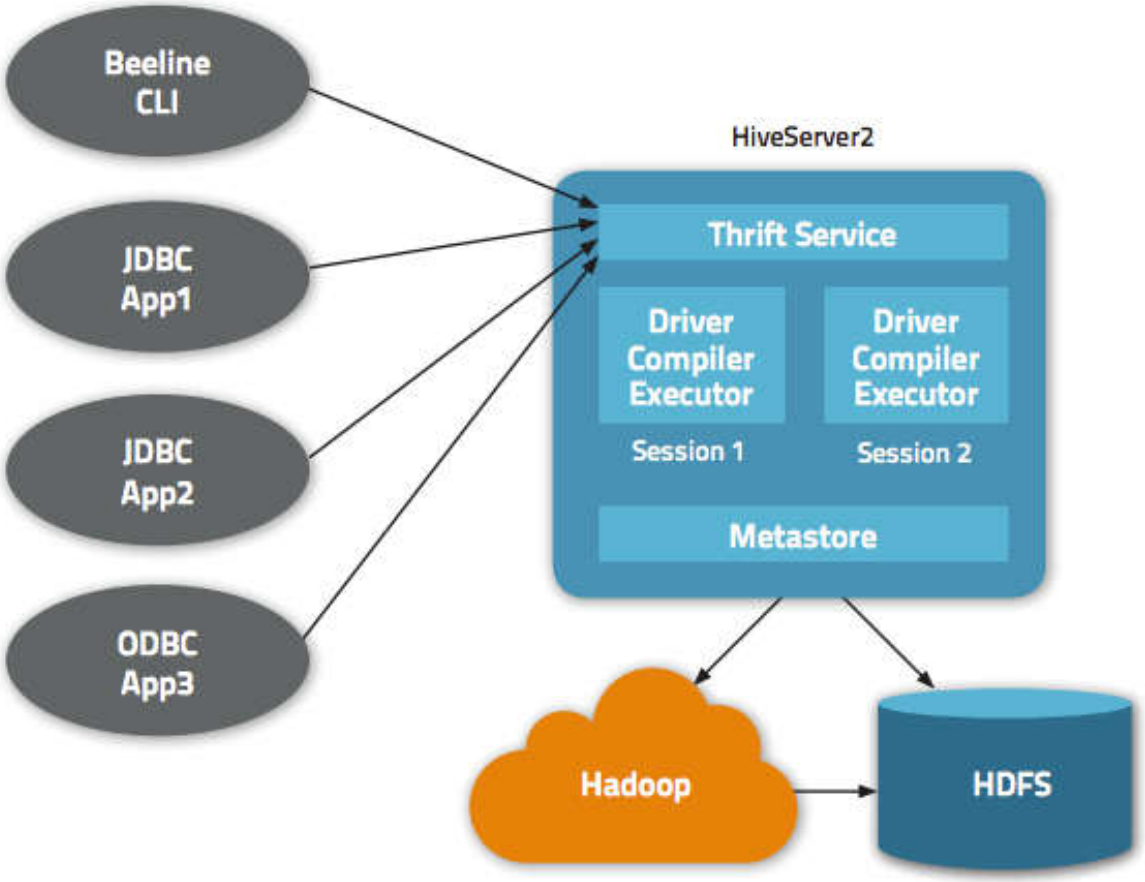
Apache Hive Architecture :



#Hive Core Component
#Hive Interoperability

Hive Component :

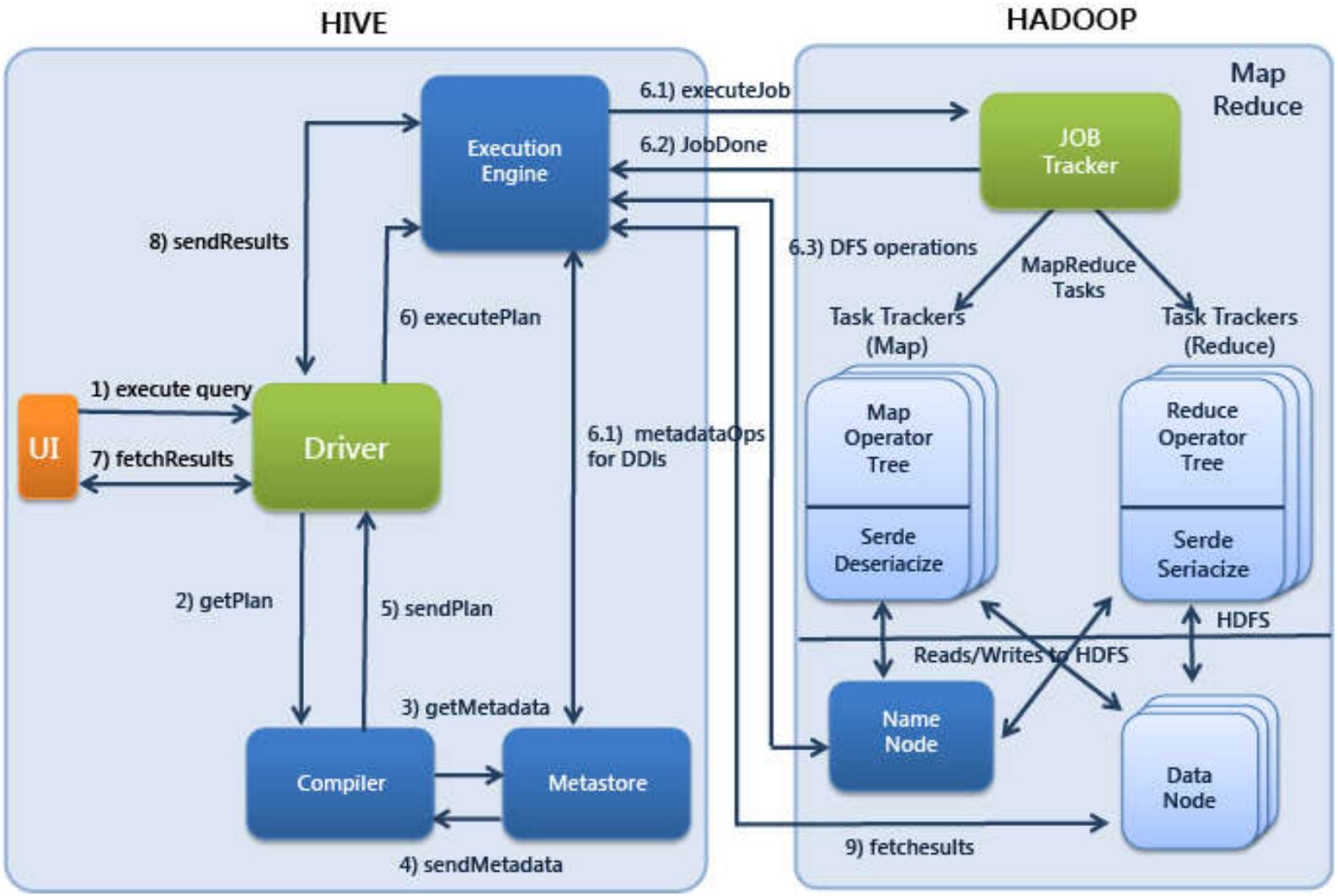
Apache Hive Architecture :



http://blog.cloudera.com/wp-content/uploads/2013/07/hiveserver1.png

Hive Operation Flow :

Apache Hive Architecture :



https://www.edureka.co/blog/introduction-to-apache-hive/

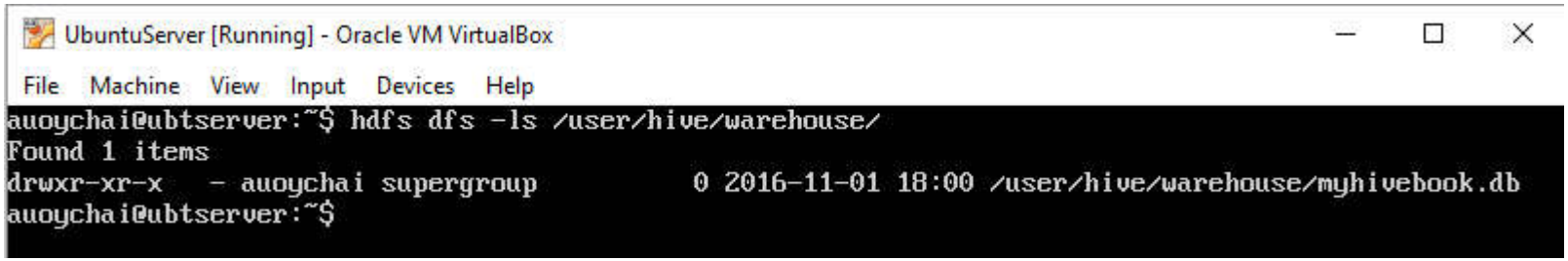
Apache Hive Architecture :

Hive Physical Model :

- Warehouse directory in HDFS

  e.g. , /user/hive/warehouse

- DB is form of subdirectories



- Table row data stored in DB subdirectory
- Partition form subdirectories of table directory
- Actual data store in flat files ,
  - TEXTFILE , SEQUENCEFILE , RCFILE , RC

Apache Hive Architecture :

Hive Fie Format :

- TEXTFILE , SEQUENCEFILE , RCFILE , ORC

การจัดการโครงสร้างระบบข้อมูล Hive :

ประเภทของ Object ทั้งหมด :

- Database
- Table
- Column Data Type
- Partition
- Buckets
- View
- Index

การจัดการโครงสร้างระบบข้อมูล Hive :

- Create DB :

```
CREATE DATABASE myhivebook;

CREATE DATABASE IF NOT EXISTS myhivebook;

CREATE DATABASE IF NOT EXISTS myhivebook;
        COMMENT 'hive database demo'
        LOCATION '/hdfs/directory'
        WITH DBPROPERTIES ('creator'='dayongd','date'='2015-01-01');
```

- Drop DB : **DROP DATABASE IF EXISTS myhivebook;**
- Use DB : **USE myhivebook;**
- Show DB : **SHOW DATABASES;** || **DESCRIBE DATABASE default;**

#Hive Table :

- Concept:
  - Configuration : ${HIVE_HOME}/conf/hive-site.xml
  - Table location: Default /user/hive/warehouse

    *** internal table
- Table => { Internal | External }
- Other :

  !table || jdbc:hive2://> !table

  DESC [Table Name];

Column Data Type :

Primitive Data Type:

TINYINT , SMALLINT , INT , BIGINT , FLOAT , DOUBLE , DECIMAL ,

BINARY , BOOLEAN , STRING , CHAR , VARCHAR , DATE , TIMESTAMP

Complex Data Type:

ARRAY => ['apple','orange','mango'] | array_name[index] | fruit[0]='apple'

MAP => {1: "apple",2: "orange"} | map_name[key] | fruit[1]="apple"

STRUCT => {1, "apple"} | structs_name.column_name | fruit.col1=1

#การสร้าง Table :

```
CREATE TABLE IF NOT EXISTS employee_internal
        (
        name string,
        work_place ARRAY<string>,
        sex_age STRUCT<sex:string,age:int>,
        skills_score MAP<string,int>,
        depart_title MAP<STRING,ARRAY<STRING>>
        )
        COMMENT 'This is an internal table'
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY '|'
        COLLECTION ITEMS TERMINATED BY ','
        MAP KEYS TERMINATED BY ':'
        STORED AS TEXTFILE;
```

#การสร้าง Table :

```
CREATE EXTERNAL TABLE employee_external
        (
        name string,
        work_place ARRAY<string>,
        sex_age STRUCT<sex:string,age:int>,
        skills_score MAP<string,int>,
        depart_title MAP<STRING,ARRAY<STRING>>
        )
        COMMENT 'This is an external table'
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY '|'
        COLLECTION ITEMS TERMINATED BY ','
        MAP KEYS TERMINATED BY ':'
        STORED AS TEXTFILE
        LOCATION '/user/dayongd/employee';
```

#การสร้าง Table :

```
CREATE TABLE ctas_employee
        AS SELECT * FROM employee_external;
```

#การสร้าง Table :

```
CREATE TABLE cte_employee AS
        WITH r1 AS
        (SELECT name FROM r2
        WHERE name = 'Michael'),
        r2 AS
        (SELECT name FROM employee
        WHERE sex_age.sex= 'Male'),
        r3 AS
        (SELECT name FROM employee
        WHERE sex_age.sex= 'Female')
        SELECT * FROM r1 UNION ALL select * FROM r3;
```

#การสร้าง Table :

```
CREATE TABLE empty_ctas_employee AS
        SELECT * FROM employee_internal WHERE 1=2;

CREATE TABLE empty_like_employee
        LIKE employee_internal;
```

#การลบข้อมูล Table :

TRUNCATE TABLE cte_employee;

DROP TABLE IF EXISTS empty_ctas_employee;

#การสร้าง Index Table :

```
CREATE INDEX idx_id_employee_id
        ON TABLE employee_id (employee_id)
        AS 'COMPACT'
        WITH DEFERRED REBUILD;
```

#การสร้าง View/Table :

```
CREATE VIEW employee_skills
        AS
        SELECT name, skills_score['DB'] AS DB,
        skills_score['Perl'] AS Perl,
        skills_score['Python'] AS Python,
        skills_score['Sales'] as Sales,
        skills_score['HR'] as HR
        FROM employee;
```

#การแก้ไข Scheme Table :

```
ALTER TABLE cte_employee RENAME TO c_employee;
ALTER TABLE c_employee
        SET TBLPROPERTIES ('comment'='New name, comments');
ALTER TABLE employee_internal SET
        SERDEPROPERTIES ('field.delim' = '$');
ALTER TABLE c_employee SET FILEFORMAT RCFILE;
ALTER TABLE c_employee
        SET LOCATION
        'hdfs://localhost:8020/user/dayongd/employee';
 ALTER TABLE employee_internal
        CHANGE name employee_name string AFTER sex_age;
ALTER TABLE c_employee ADD COLUMNS (work string);
```

#การโหลดข้อมูลเข้า Hive Table :

```
LOAD DATA LOCAL INPATH
        '/home/dayongd/Downloads/employee_hr.txt'
        OVERWRITE INTO TABLE employee_hr;
LOAD DATA LOCAL INPATH
        '/home/dayongd/Downloads/employee.txt'
        OVERWRITE INTO TABLE employee_partitioned
        PARTITION (year=2014, month=12);
LOAD DATA INPATH
        '/user/dayongd/employee/employee.txt'
        OVERWRITE INTO TABLE employee;
```

#การโหลดข้อมูลเข้า Hive Table :

```
INSERT INTO TABLE employee
        SELECT * FROM ctas_employee;
FROM ctas_employee
        INSERT OVERWRITE TABLE employee
        SELECT *
        INSERT OVERWRITE TABLE employee_internal
        SELECT * ;
IMPORT TABLE empolyee_imported FROM
        '/user/dayongd/output3';
IMPORT EXTERNAL TABLE empolyee_imported_external
        FROM '/user/dayongd/output3'
        LOCATION '/user/dayongd/output4' ;
IMPORT TABLE employee_partitioned_imported
        FROM '/user/dayongd/output5';
```

#การเรียกข้อมูลจาก Hive Table :

```
SELECT * FROM employee;
SELECT DISTINCT name FROM employee LIMIT 2;
SELECT name, work_place FROM employee
        WHERE name = 'Michael';
WITH t1 AS (
        SELECT * FROM employee
        WHERE sex_age.sex = 'Male')
        SELECT name, sex_age.sex AS sex FROM t1;
SELECT name, sex_age.sex AS sex
        FROM
        (
        SELECT * FROM employee
        WHERE sex_age.sex = 'Male'
        ) t1;
```

#การเรียกข้อมูลจาก Hive Table :

```
SELECT name, sex_age.sex AS sex
        FROM employee a
        WHERE a.name IN
        (SELECT name FROM employee
        WHERE sex_age.sex = 'Male'
        );
SELECT name, sex_age.sex AS sex
        FROM employee a
        WHERE EXISTS
        (SELECT * FROM employee b
        WHERE a.sex_age.sex = b.sex_age.sex
        AND b.sex_age.sex = 'Male'
        );
```

#การเรียกข้อมูลจาก Hive Table :

```
SELECT emp.name, emph.sin_number
        FROM employee emp
        JOIN employee_hr emph ON emp.name = emph.name;
SELECT emp.name, empi.employee_id, emph.sin_number
        FROM employee emp
        JOIN employee_hr emph ON emp.name = emph.name
        JOIN employee_id empi ON emp.name = empi.name;
```

#Advance Topic:

Table Data

- Partition
- Buckets

 # Hive Installation :

#Pre-Requisite:

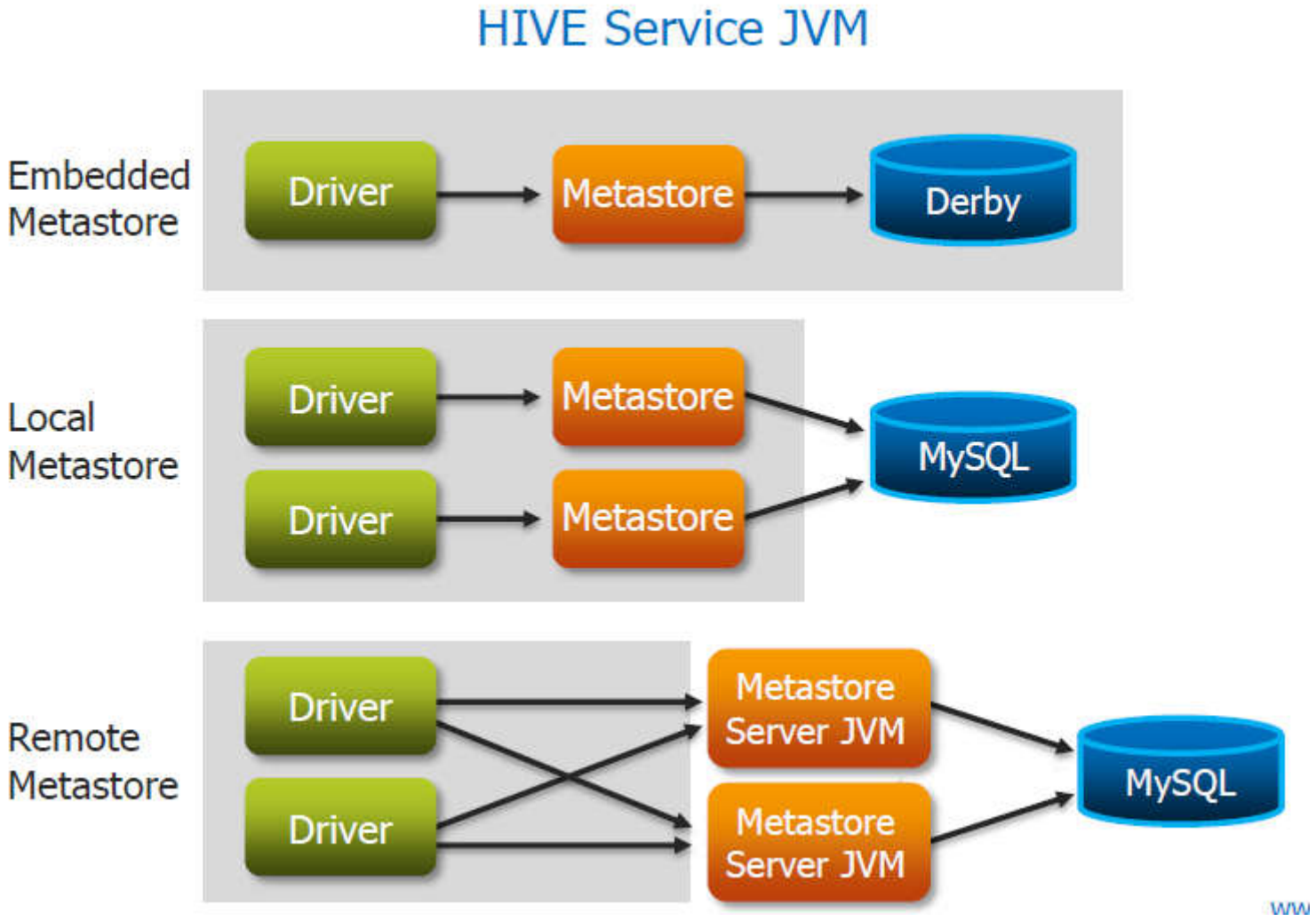
- Hadoop
- Java 7 หรือ 8

MySQL:

Hive:

Hive Installation :

Hive Deployment Model :



http://www.stratapps.net/images/HiveArchitecture.JPG

Hive Installation Directory :

- Main Package :

  /usr/local/hive

- Log file :

  /var/log/hive

Hive Confugration :

hive-env.sh

```
export  JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export  HIVE_HOME=/usr/local/hive
export  HIVE_CONF_LOG=/var/log/hive
```

Hive Confugration :

hive-site.xml

```
<configuration>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</valu
        <description>xx</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>com.mysql.jdbc.Driver</value>
        <description>xx</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>hiveuser</value>
        <description>xx</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>hivepassword</value>
        <description>xxx</description>
    </property>
        <property>
        <name>hive.metastore.warehouse.dir</name>
        <value>hdfs://localhost:9000/user/hive/warehouse</value>
        <description>location of default database for the warehouse</description>
    </property>
    <property>
        <name>hive.metastore.uris</name>
        <value>thrift://localhost:9083</value>
        <description>hello</description>
    </property>
    <property>
        <name>hive.metastore.sasl.enabled</name>
        <value>false</value>
        <description>xxx</description>
    </property>
    <property>
        <name>hive.server2.enable.doAs</name>
        <value>false</value>
    </property>
    <property>
        <name>hive.server2.authentication</name>
        <value>NONE</value>
    </property>
</configuration>
```

Hive Installation :

MySQL:

$ sudo apt-get install mysql-server

$ sudo apt-get install libmysql-java

-- สร้าง Metastore DB สำหรับ Hive Metastore

$ mysql – u root –p

Enter password:123456

mysql> CREATE DATABASE metastore;

mysql> SOURCE /usr/local/hive/scripts/metastore/upgrade/mysql/hive-txt-scheme-2.1.0.mysql.sql

mysql > CREATE USER ‘hiveuser’@‘%’ IDENTIFIED BY ‘hivepassword’;

mysql> GRANT all on *.* to ‘hiveuser’@localhost identified by ‘hivepassword’;

mysql> flush priviledges;

Hive Installation :

Hive:

$wget http://www-eu.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.gz

$tar –vxf apache-hive-2.1.0-bin.tar.gz

-- Create Directory , กำหนดสิทธิ ให้Hive และ Move Package ไปไว้ใน directory ที่ต้องการ

$ sudo mkdir /usr/local/hive

$sudo makdir /var/log/hive

$ sudo chown auoychai:auoychai –R /usr/local/hive

$ sudo chown auoychai:auoychai –R /var/log/hive

$ sudo mv ./ apache-hive-2.1.0-bin/* /usr/local/hive

Hive Installation :

Hive:

-- กำหนด Environment Variable ให้กับ Hive

$ nano /home/auoychai/.bashrc

** เพิ่ม

```
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export PATH=$PATH:$HIVE_HOME/sbin
```

-- Refresh new environment variable => $source .bashrc

-- กำหนด Environment ให้ Hive ที่ไฟล์ /usr/local/hive/conf/sive-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HIVE_HOME=/usr/local/hive
export HIVE_CONF_LOG=/var/log/hive
```

-- เตรียม .jar MySQL ให้กับ Hive

$ cp /usr/local/hive/jdbc/hive-jdbc-2.1.0-standalone.jar /usr/local/hive/lib

Hive Installation :

-- กำหนดค่า Configuration ห้กับ Hive

$ nano /usr/local/hive/conf/hive-site.xml

```
<configuration>
        <property>
                <name>javax.jdo.option.ConnectionURL</name>
                <value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</value>
                <description>xx</description>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionDriverName</name>
                <value>com.mysql.jdbc.Driver</value>
                <description>xx</description>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionUserName</name>
                <value>hiveuser</value>
                <description>xx</description>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionPassword</name>
                <value>hivepassword</value>
                <description>xxx</description>
        </property>
        <property>
                <name>hive.metastore.warehouse.dir</name>
                <value>hdfs://localhost:9000/user/hive/warehouse</value>
                <description>location of default database for the warehouse</description>
        </property>
        <property>
                <name>hive.metastore.uris</name>
                <value>thrift://localhost:9083</value>
                <description>hello</description>
        </property>
        <property>
                <name>hive.metastore.sasl.enabled</name>
                <value>false</value>
                <description>xxx</description>
        </property>
        <property>
                <name>hive.server2.enable.doAs</name>
                <value>false</value>
        </property>
        <property>
                <name>hive.server2.authentication</name>
                <value>NONE</value>
        </property>
</configuration>
```

Hive Start-Stop :

1) Start Hadoop :

- start-dfs.sh , start-yarn.hs

2). Start Metastore :

hive --service metastore&

*** netstat –ln | grep 9083

3). Start Hive Service || HiveServer2 Service

- hive || - hive --service hiveserver2&

4). Connect Hive ด้วย Beeline

-beeline

-beeline>!connect jdbc:hive2//localhost:10000

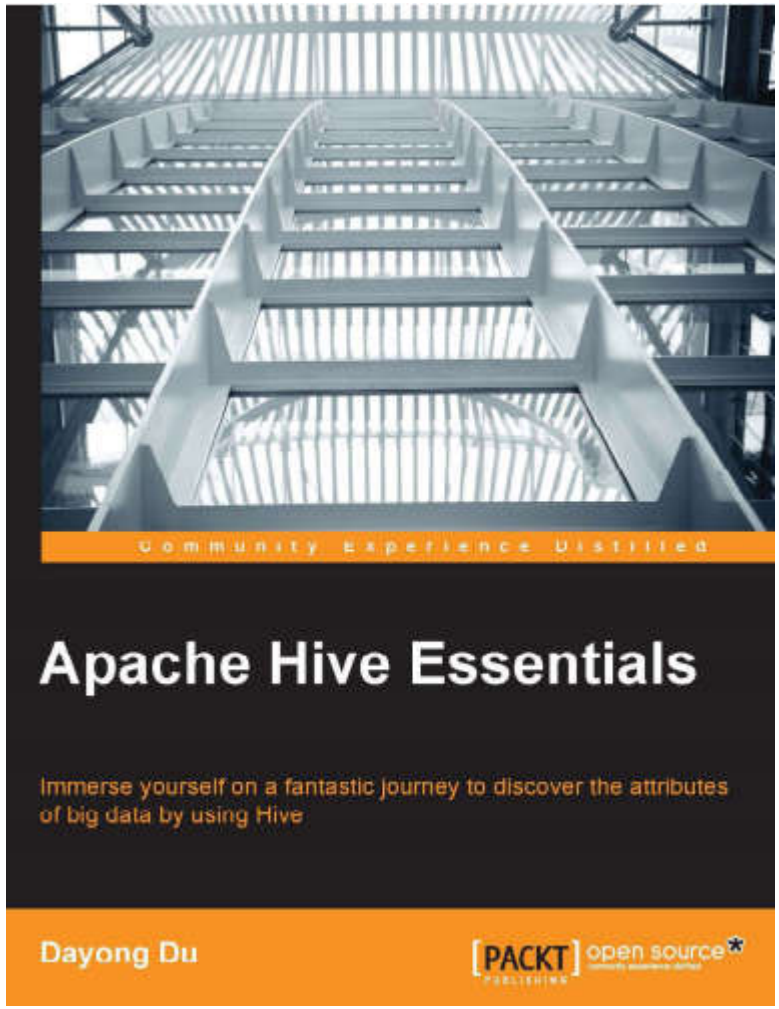
***UserName:auoychai , Password:123456

#Check Server Status :

-- netstat –ln | grep 9083

-- jps

Apache Hive Book :



Hand-On :

1). Apache Hive Installation

2). Create HiveDB & Data Manipulate on Hive Table

- Chapter_03 , Chapter_04


The End
Big
data
Shift