Outline

Module 1：大數據簡介

Module 2：Hadoop Ecosystem介紹

Module 3：Hadoop 平台安裝

Module 4：Hadoop 分散式檔案系統（HDFS）

Module 5：Hadoop MapReduce

Module 6：Apache Hive

Module 7：Sqoop與Flume

Module 8：Apache Spark

Module 9：Spark 平台安裝

Module 10：RDD － Resilient distributed dataset

Module 11：Scala 程式開發基礎

Module 12：Spark SQL 及 DataFrame

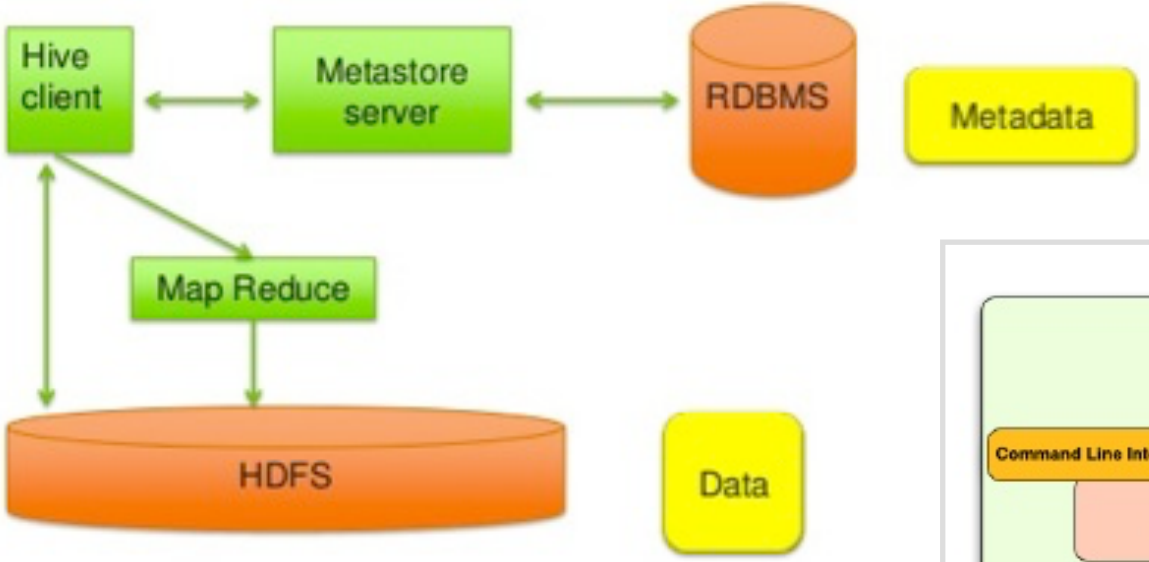
Module 13：Spark 機器學習函式庫(MLlib)

Apache Hive 介紹

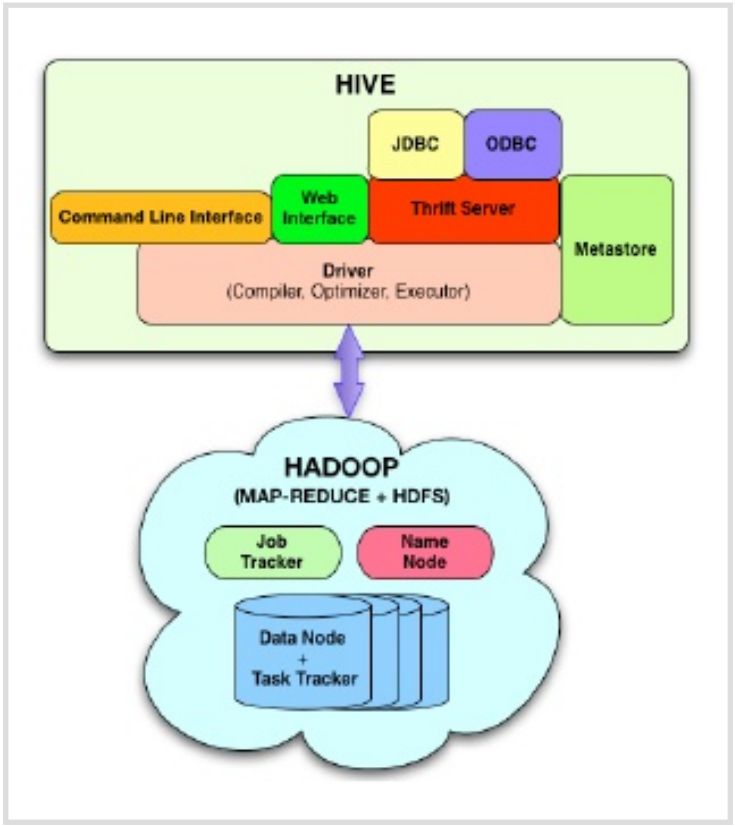
- **Apache Hive**是一個建立在Hadoop架構之上的**數據倉庫(data warehouse)**。它能夠提供數據的精煉，查詢和分析。
- 最初由**Facebook**開發，也有其他公司投入開發及使用，如Netflix、Amazon等
- 將結構化的數據文件映射為一張資料庫**表格(table)**，並提供簡單的**SQL**操作功能
- 可以**將SQL語句轉換為MapReduce任務**進行運行，降低學習及開發成本

Hive architecture

What are we trying to protect here ?

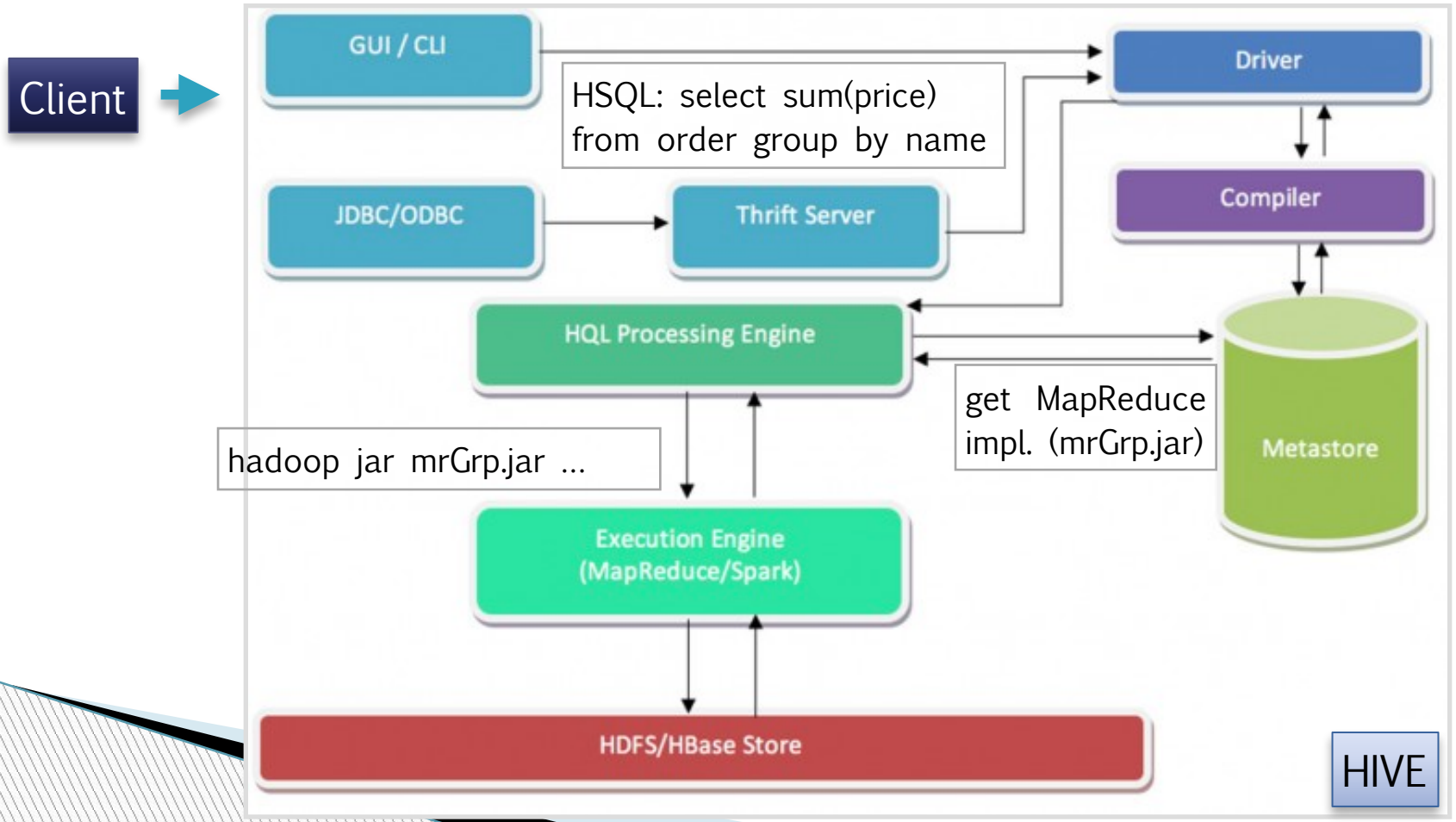


提供Hive對HDSF檔案的對映



Hive運作流程

create table order (id int, name string, price double) path hdfs://cust.txt



Hive與傳統資料庫比較

特徵	Hive	RDBMS
Schema	Schema on READ	Schema on WRITE
更新(Update)	支援 增 / 刪 / 修(刪 / 修在0.14後才支援)	支援 增 / 刪 / 修
交易(Transaction)	部份支援	支援
索引(Indexes)	支援(0.7後才支援)	支援
延遲(Latency)	數分鐘	秒以內
函數(Function)	數十個內建函數	上百個內建函數
SELECT	FROM 子句限用單一資料表	SQL-92 標準
JOIN	INNER, OUTER, SEMI, MAP JOINS	SQL-92 或其他變形
次查詢(Subqueries)	只能在 FROM 子句中使用	在任何子句
擴展性	高	低
數據規模	大	小

Hive的優點

- 簡潔方便，門檻低(相較MapReduce)
- 可透過Partition提升查詢效能及彈性
- DBA可重複使用部份SQL(HiveSQL類似MySQL語法)
- 透過建立VIEW節省表格建立時間成本
  - 處理相同資料來源但不同欄位的情境可不必重覆建立表格

Hive的缺點

- 無法應付即時查詢的情境
- 不支援交易(Transaction)機制
- 不是ETL工具
- 無法精細控制資料流程(IF…ELSE)
- 不易處理非結構化(沒有明確schema)資料

Hive的安裝及設定

- 參考Apache-Hive-Installation.pdf
  - Hive 1.2.1 及 2.2.0 都可與 mysql 5.7 搭配，但 Hive 2.1.1 會有問題
  - Hive 預設使用Derby作為metastore儲存庫，無法多人使用；以mysql取代是最常見方案
- 安裝完成後可連線至mysql檢視hive_metadata資料庫內容
  - DBS：儲存hive內的database
  - TBLS：儲存hive的table(Managed和External)
  - COLUMNS_V2：儲存各table的column屬性

HIVE SQL 介紹

- Ref：https://cwiki.apache.org/confluence/display/Hive/LanguageManual
- 語法與MySQL類似，開發者透過HiveSQL執行MapReduce作業
  - 不會產生Java程式碼
- 基本資料型態
  - 數值
  - 日期 / 時間
  - 字串
  - 布林 / binary / 複合型態

Hive SQL資料型態－數值

Type	Size	Range	Examples
TINYINT	1 Byte signed integer	-128 to 127	100
SMALLINT	2 Bytes signed integer	-32,768 to 32,767	100, 1000
INT	4 Bytes signed integer	-2,147,483,648 to 2,147,483,647	100, 1000, 50000
BIGINT	8-byte signed integer	$-9.2*10^{18}$ to $9.2*10^{18}$	100, $1000*10^{10}$
FLOAT	4-byte single precision float	$1.4*e^{-45}$ to $3.4*e^{+38}$	1500.00
DOUBLE	8-byte double precision float	$4.94e^{-324}$ to $1.79e^{+308}$	750000.00
DECIMAL	17 Bytes Precision upto 38 digits	$-10^{38}+1$ to $10^{38}-1$	DECIMAL(5,2)

Ref：http://hadooptutorial.info/hive-data-types-examples/

Hive SQL資料型態－字串

Type	Description	Examples
STRING	Sequence of characters. Either single quotes (') or double quotes (") can be used to enclose characters	'Welcome to Hadooptutorial.info'
VARCHAR	Max length is specified in braces. Similar to SQL's VARCHAR. Max length allowed is 65355 bytes	'Welcome to Hadooptutorial.info tutorials'
CHAR	Similar to SQL's CHAR with fixed-length. i.e values shorter than the specified length are padded with spaces	'Hadooptutorial.info'

Ref：http://hadooptutorial.info/hive-data-types-examples/

Hive SQL資料型態－日期時間

- DATE
  - 格式YYYY-MM-DD的字串，範圍0000-01-01～9999-12-31
- TIMESTAMP
  - 用整數、浮點數及字串表示時間
    - 整數 / 浮點數：自1970.01.01秒數
    - 字串：YYYY-MM-DD HH:MM:SS.fffffffff格式字串
- 字串及日期型態間可用cast函式作轉換
  - ex：cast(string as date)、cast(date as string)

Hive SQL資料型態－複合型態

- arrays: ARRAY<data_type>
- maps: MAP<primitive_type, data_type>
- structs: STRUCT<col_name : data_type [COMMENT col_comment], …>
- union: UNIONTYPE<data_type, data_type, ...>

```
CREATE TABLE union_test(foo UNIONTYPE<int, double, array<string>, struct<a:int,b:string>>);
SELECT foo FROM union_test;

{0:1}
{1:2.0}
{2:["three","four"]}
{3:{"a":5,"b":"five"}}
{2:["six","seven"]}
{3:{"a":8,"b":"eight"}}
{0:9}
{1:10.0}
```

Hive SQL - 資料庫操作

- 查看目前系統內的資料庫
  - show databases;
- 建立資料庫：
  - CREATE database db_name [COMMENT database_comment] [LOCATION hdfs_path];
- 切換目前使用的資料庫
  - USE db_name;
- 刪除資料庫
  - DROP db_name;

Hive SQL - 資料表操作

- 查看目前資料庫內的表格
  - show tables;
- 建立內部資料表：
  - create table tb_name(field1 type1, field2 type2, ...) [ROW FORMAT row_format];
  - row_format: ROW FORMAT DELIMITED FIELDS TERMINATED BY '-'
- 將資料由file中讀入表格
  - LOAD DATA LOCAL INPATH 'file_path' OVERWRITE INTO TABLE tb_name;
- 查看資料表Schema
  - desc tb_name;
- 刪除資料表
  - drop table tb_name;

[提示]：操作過程中可注意HDSF中/user/hive/的內容變化

Hive SQL - 資料操作

- 查詢－支援join、where、order、group by、having
  - **SELECT** * **FROM** sales **WHERE** amount > 10 AND region = "US" order by amount Limit 5;
  - SELECT col1 FROM t1 GROUP BY col1 HAVING SUM(col2) > 10;
  - SELECT a.* FROM a JOIN b ON (a.id = b.id);
- 新增
  - **INSERT INTO TABLE** students **VALUES** ('fred flintstone', 35, 1.28), ('barney rubble', 32, 2.32);
- 修改
  - UPDATE students SET age=40 WHERE name='smith';
- 刪除
  - DELETE FROM students WHERE name='smith';

[練習]WordCount的HIVE實作

- 建立t_wc table：
  - CREATE TABLE t_wc (sentence String)
- 載入本機檔案到hive table中：
  - LOAD DATA LOCAL INPATH '/home/hduser/Downloads/gettysburg.txt' OVERWRITE INTO TABLE t_wc;
- 執行WordCount（依出現次數由大到小排序）
  - SELECT word, COUNT(*) as cnt FROM t_wc **LATERAL VIEW explode(split(sentence, ' ')) lTable** as word **GROUP BY** word **order by cnt desc**;
- 執行WordCount（依出現次數由大到小排序，取前五筆）
  - SELECT word, COUNT(*) as cnt FROM t_wc LATERAL VIEW explode(split(sentence, ' ')) lTable as word GROUP BY word order by cnt desc **LIMIT 5**;

Lateral view參考：https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView

[練習]建立複雜資料表

- 下載yelp.zip，解壓縮後將Yelp_ALL底下的ratings.txt放在本機上
- 用文字編輯器開啟ratings.txt，將欄位分隔符號由"：："改為"－"
- 在hive shell中，輸入以下指令：
  - create table ratings(userid STRING, itemid INT, rating INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '-' tblproperties ("skip.header.line.count"="1");
  - 用tblproperties來告訴後面的LOAD DATA指令別載入第一行
  - LOAD DATA LOCAL INPATH '/home/hduser/Downloads/ratings.txt' OVERWRITE INTO TABLE ratings;
- 查看匯入資料
  - select * from ratings limit 5;
- 測試GroupBy SQL
  - select userid, avg(rating) from ratings group by userid having avg(ratings) > 3.5;

Hive SQL - 外部資料表操作

- 建立外部資料表：
  - create EXTRANAL table tb_name(field1 type1, field2 type2, ...) [ROW FORMAT row_format] [LOCATION hdfs_path];
- 將指定檔案上傳至hdfs_path
- 查看資料表Schema
  - desc tb_name;
- 刪除資料表
  - drop table tb_name;

[提示]：Hive內外部資料表的差別在那裡？可試著觀察drop table後HDSF的變化看看

[練習]建立外部資料表

- 下載yelp.zip，解壓縮後將Yelp_ALL底下的items.txt上傳至HDFS的/yelp路徑下
- 在hive shell中，輸入以下指令：
  - create external table items(itemid INT, category String) ROW FORMAT DELIMITED FIELDS TERMINATED BY '-' LOCATION '/yelp';
- 查看匯入筆數
  - select * from items;
- 與剛才建立的ratings資料表作join查詢
  - select a.userid, a.itemid, b.category, a.ratings from ratings a join items b on a.itemid = b.itemid;
- 刪除items資料表
  - drop table items;
  - 觀察HDFS的/yelp路徑下，items.txt是否仍存在

[提示1]：所有在/yelp下的檔案都會進到items中
[提示2]：可由網頁查看MR Job執行狀況

[補充]Hive Local Mode

- 使用本機資源執行Hive SQL，而非使用Hadoop
  - 仍使用MapReduce為基礎
  - 資料量限制：預設128MB
    - hive.exec.mode.local.auto.inputbytes.max
  - Mapper的數量：預設為4
    - hive.exec.mode.local.auto.tasks.max
  - Reducer的數量：0或1
- 指令
  - set hive.exec.mode.local.auto=true;

[補充]不使用Cli執行Hive SQL

- Hive SQL 可不用進入Hive Command介面即可執行
  - hive -e 'show tables;' > tables.txt => 在Terminal 中執行 Hive SQL並將結果輸出
  - hive -f rating_avg.sql > ratings_avg.txt => 在Terminal 中執行sql檔內Hive SQL並將結果輸出

```
select userid, avg(ratings) as r_avg from ratings group by userid having avg(ratings) > 3.5 order
by r_avg desc limit 200 ;
```

```
hduser@spark-single:~/Downloads/material$ hive -f hive_sql_sample.sql > hive_res
.txt
```

建立Partition資料表

- Hive Table可指定表內欄位或不存在之欄位作為分區依據，產生分區資料表
- 分區資料表因資料依分區欄位拆分，檔案較小，搜尋速度較快
- 建立語法：create [external] table tb_name(field1 type1, field2 type2, ...) PARTITIONED BY (fieldA typeA) [ROW FORMAT row_format];
  - EX: create table items_pt(itemid INT, category String) PARTITIONED BY (part STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '-';
- 資料載入語法：LOAD DATA LOCAL INPATH 'file_path' INTO TABLE tb_name PARTITION(part=...);
  - EX: LOAD DATA LOCAL INPATH '/itemsA.txt' INTO TABLE items_pt PARTITION(part='A');

[提示]：查看HDFS的hive warehouse下分區資料表的儲存方式