

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)



大象的成長歷程



Doug Cutting
等人開發Nutch

Doug Cutting實作
GFS & MapReduce
to Nutch

Yahoo!
用910個Node，創下
在3.5分鐘完成1TB資

創下62秒Sort 1TB
資料記錄(1,460
Nodes)
16.25小時Sort 1PB
資料(3,658 Nodes)

2002

2003

2004

2005

2006

2007

2008

2009

Google發表GFS
及MapReduce
相關論文

Doug Cutting至
Yahoo!任職
Hadoop由Nutch分
立出來



Cloudera
成立

Doug Cutting
加入Cloudera

Facebook開發
Hive



Apache Hadoop

- ▶ Apache Hadoop是一款支援大量資料處理的分散式應用並以Apache 2.0許可協議發布的開源軟體框架(Open Source)
- ▶ 支援在量產硬體(Commodity Hardware)構建的大型叢集(Cluster)上運行的應用程式
- ▶ 目前最新版本為3.0.0 (alpha1)
 - 2.6.5 - stable
 - 2.7.3 - stable



由分散式的挑戰來看Hadoop的特色

- ▶ 分散式應用開發複雜度極高
 - Hadoop實現了MapReduce的程式框架：應用程式被分割成許多小部分，而每個部分都能在集群中的任意節點上執行或重新執行
- ▶ 節點失效的處理
 - Hadoop提供的分散式檔案系統(HDFS)，資料分散儲存於多個節點、且儲存多份
- ▶ 資料讀取 / 處理的瓶頸、有效利用頻寬
 - 區塊(Block)儲存、區塊讀取
 - Process data where the data is

Hadoop Ecosystem 概觀

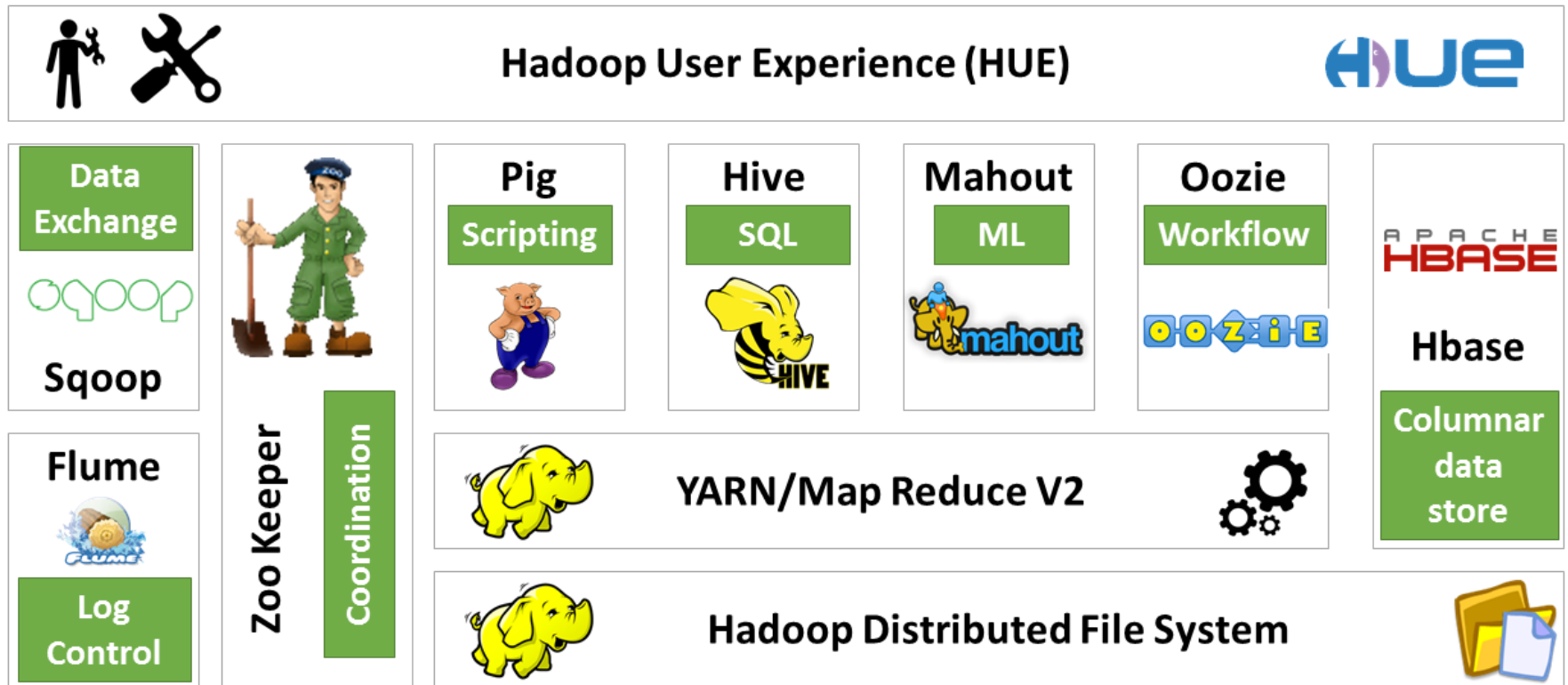
- ▶ 為降低Hadoop使用門檻、強化Hadoop核心功能管理而生的子專案或元件
- ▶ Hadoop的核心
 - Hadoop Distributed File System(HDFS)
 - MapReduce
- ▶ 來源
 - Apache Foundation
 - Cloudera, MapR, Hontonworks

Hadoop Ecosystem 成員列表

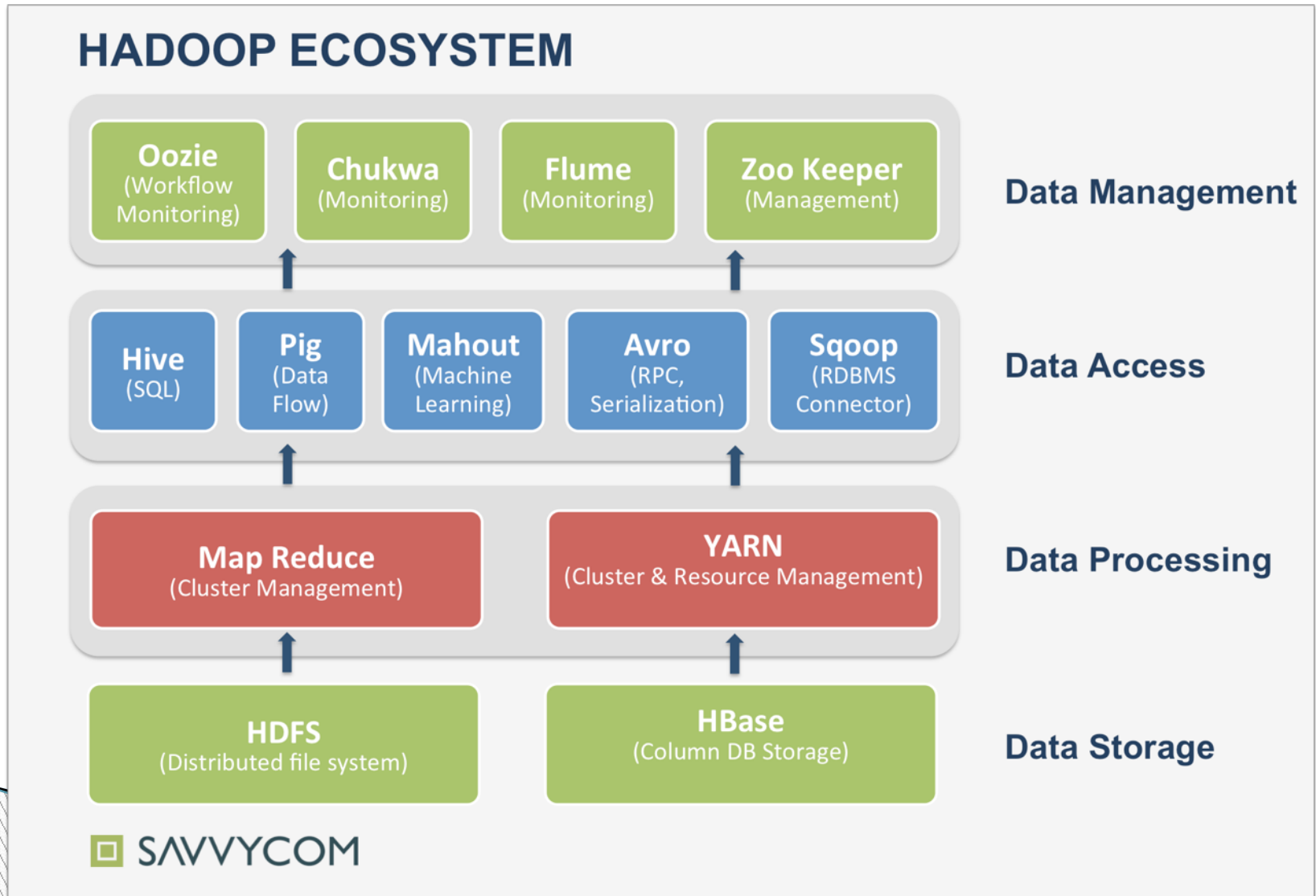
專案	類型	用途
Hive	資料處理	透過SQL語法實作MapReduce
Pig	資料處理	透過腳本語言(Script)實作MapReduce
HBase	NoSQL	NoSQL資料庫
Hue	Web介面	透過Web介面操作Hadoop
Sqoop	資料蒐集	讓Hadoop與資料庫進行資料交換
Flume	資料蒐集	蒐集串流資料
Oozie	工作流程管理	管理MapReduce工作與資料轉移Workflow
Avro	資料序列化與交換	資料序列化與反序列化
Impala	即時SQL查詢	即時查詢資料

Hadoop的生態系統(By Stack)

The Apache Hadoop Stack



Hadoop的生態系統(By Function)



Hadoop核心 — HDFS簡介

- ▶ Hadoop Distributed File System的縮寫
- ▶ 依據Google GFS實作
 - Ghemawat, S.; Gobioff, H.; Leung, S. T. (2003). "The Google file system". Proceedings of the nineteenth ACM Symposium on Operating Systems Principles - SOSP '03. p. 29.
- ▶ 可建置於OS的原生檔案系統
- ▶ 分散式儲存檔案
 - 將檔案切割為多個Block作儲存
 - 每個Block預設為128MB
 - Block複製多份，分散儲存，達到備源效果

HDFS特色

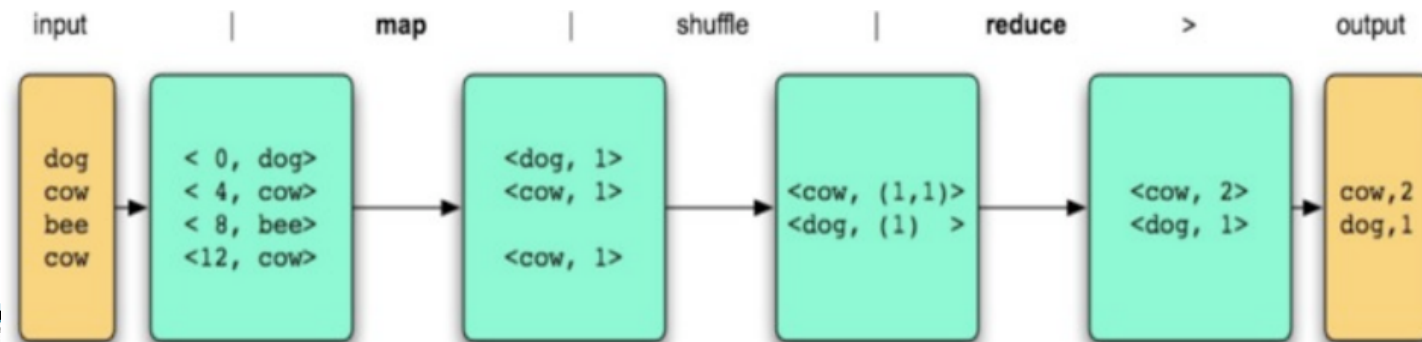
- ▶ 適用於資料以**大檔案方式**儲存
 - 每個檔案約100MB上下
- ▶ 適用**讀取導向的應用**，採用一次寫入，多次讀取的策略
 - 沒有附加(Append)及編輯(Edit)功能
- ▶ 採用**循序的寫入**方式
 - 不同於資料庫採隨機寫入策略
- ▶ 可自我修復(系統設定的儲存複本要大於**1**)
 - 允許存放資料的節點(Node)毀損
 - 單一Block毀損亦可回復

Hadoop核心－MapReduce簡介

- ▶ HDFS是分散式儲存、MapReduce是分散式處理作業
- ▶ 為分而治之(Divide and Conquer)的概念
- ▶ 將作業分散至不同節點(Node)執行
- ▶ 採本地化優先策略
 - 盡量處理作業所在節點上之Data
- ▶ 分Map和Reduce兩階段
 - Map：產生多個Mapper分散處理作業
 - Reduce：聚合Map階段產生之結果
- ▶ 至少會有一個Mapper，但未必會有Reducer

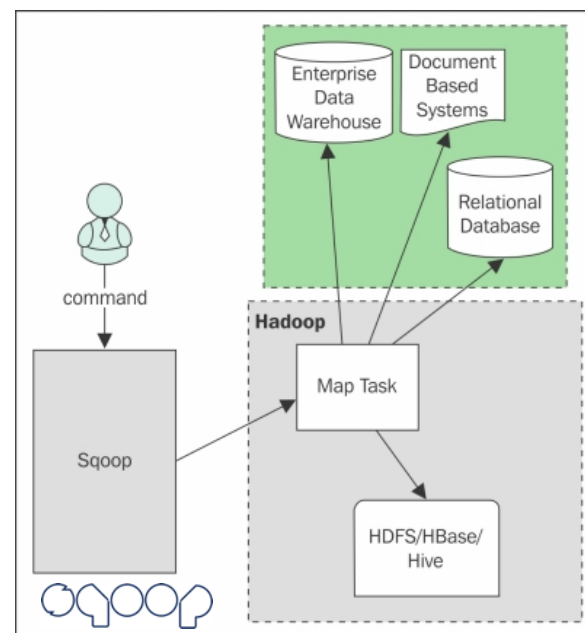
MapReduce特色

- ▶ 可容錯
 - 系統自動監控作業執行狀態
 - 作業失敗時，自動將工作分派至其它節點
- ▶ 開發人員不需知道分散式處理的細節
 - 專注於Map及Reduce邏輯的開發



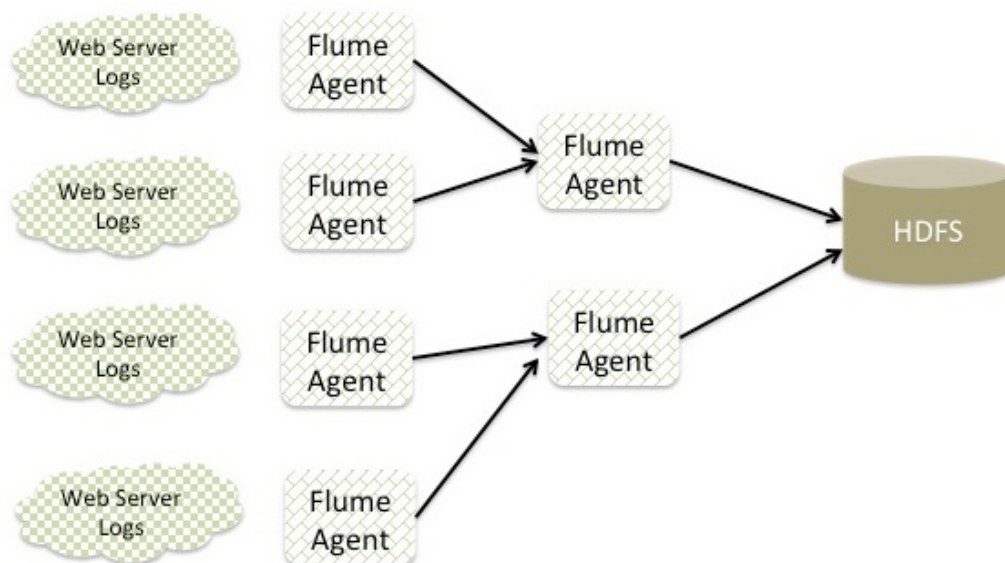
資料擷取: Sqoop 概述

- ▶ SQL to Hadoop 的簡稱
- ▶ 將結構化資料匯入及匯出資料庫 / HDFS
- ▶ 透過JDBC跟資料庫進行溝通
- ▶ 可與Oozie搭配作定期的資料維護



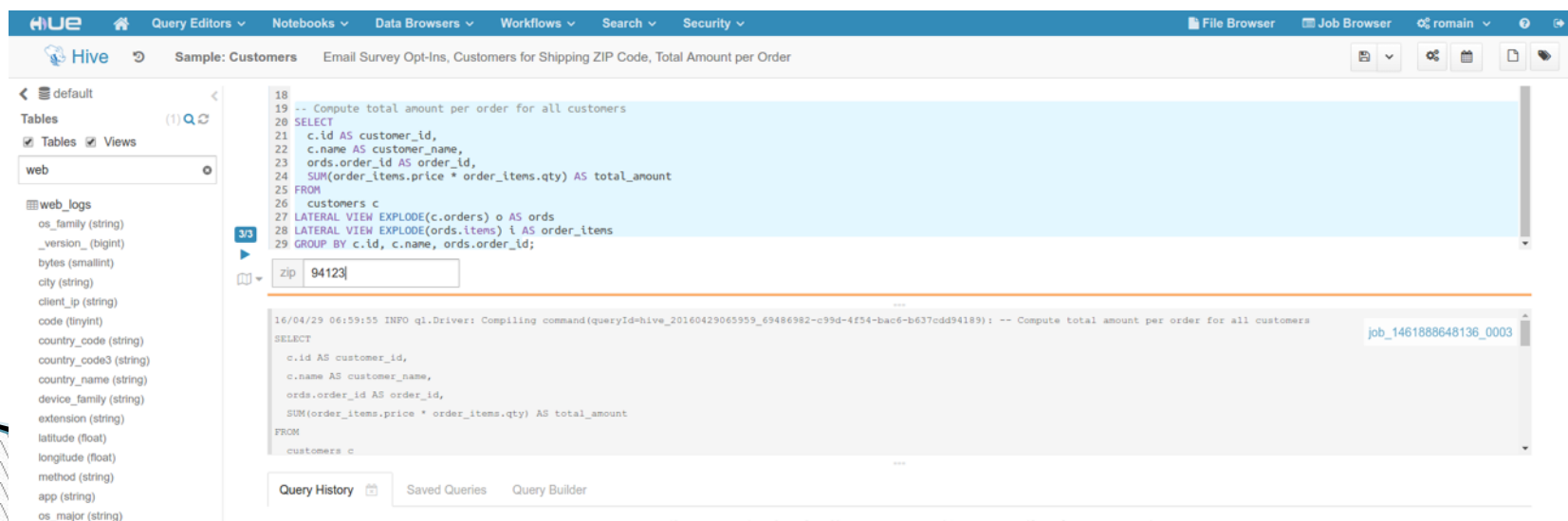
資料擷取: Flume概述

- ▶ 透過Agent傳輸即時串流資料(如Log)
 - 每台機器上通常只有一個Agent
- ▶ 使用者可自訂函式去除重覆或篩選資料
- ▶ 架構可水平延展(Horizontal Scalable)

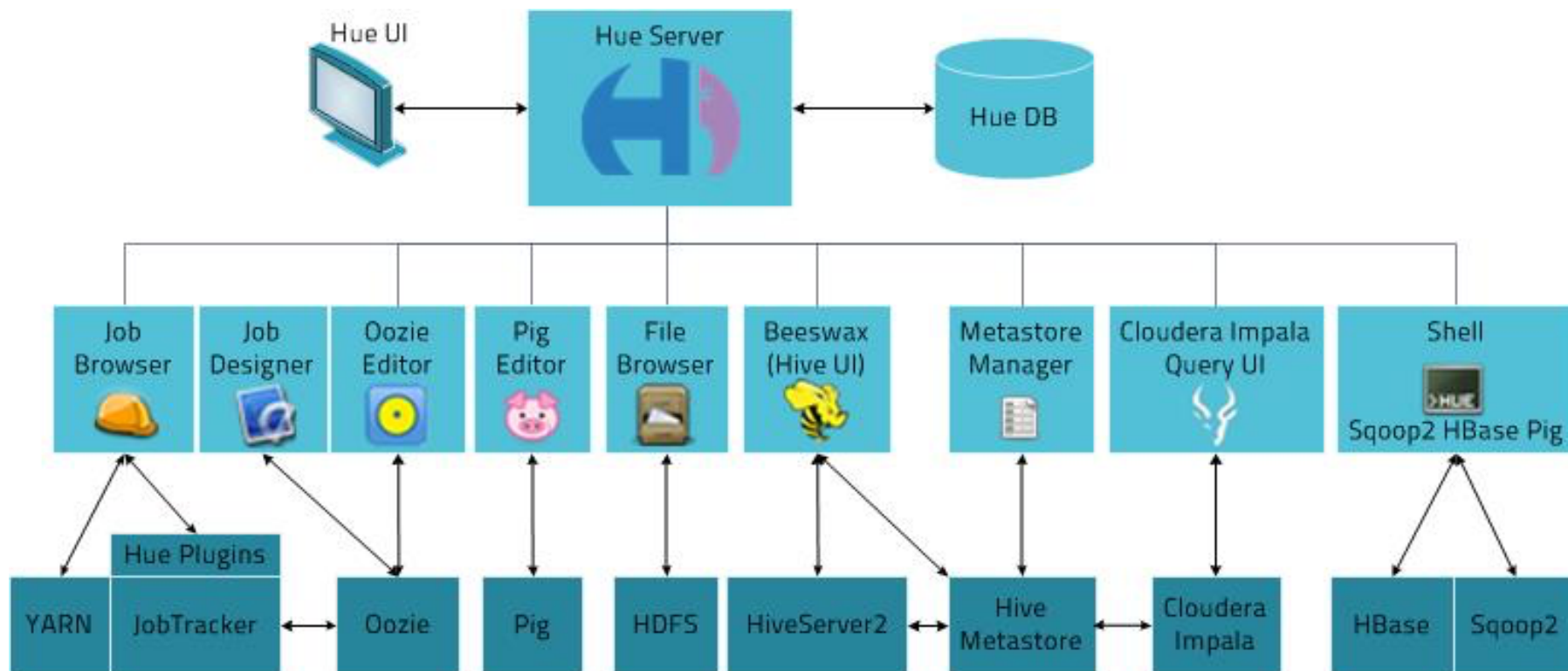


管理介面 HUE 概述

- ▶ Hadoop User Experience的縮寫
- ▶ 提供整合式的UI介面以操作所有Hadoop生態系統的成員
- ▶ 提供API供開發人員在APP上實作HUE的功能



HUE支援的管理模組



ref: http://www.cloudera.com/documentation/archive/cdh/4-x/4-3-0/CDH4-Installation-Guide/cdh4ig_topic_15.html

資料分析模組 Pig和Hive簡介

- ▶ 簡化MapReduce的開發介面
 - 讓非開發者也能操作Hadoop
 - Hive – SQL Like Language(Hive SQL)
 - Pig – Data Flow Language
- ▶ 運作效能較使用Java開發之MapReduce差
 - 效能約下降10~15%
 - 但可大幅提昇開發生產力



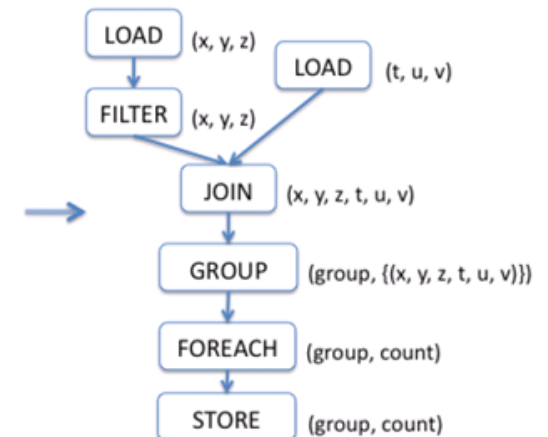
Hive SQL vs Pig Latin

```
CREATE TABLE u_data_new (  
  userid INT,  
  movieid INT,  
  rating INT,  
  weekday INT)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';  
  
add FILE weekday_mapper.py;  
  
INSERT OVERWRITE TABLE u_data_new  
SELECT  
  TRANSFORM (userid, movieid, rating, unixtime)  
  USING 'python weekday_mapper.py'  
  AS (userid, movieid, rating, weekday)  
FROM u_data;  
  
SELECT weekday, COUNT(*)  
FROM u_data_new  
GROUP BY weekday;
```

Pig Latin

```
A = LOAD 'file1' AS (x, y, z);  
B = LOAD 'file2' AS (t, u, v);  
C = FILTER A by y > 0;  
D = JOIN C BY x, B BY u;  
E = GROUP D BY z;  
F = FOREACH E GENERATE  
  group, COUNT(D);  
STORE F INTO 'output';
```

Logical Plan



Hive vs Pig

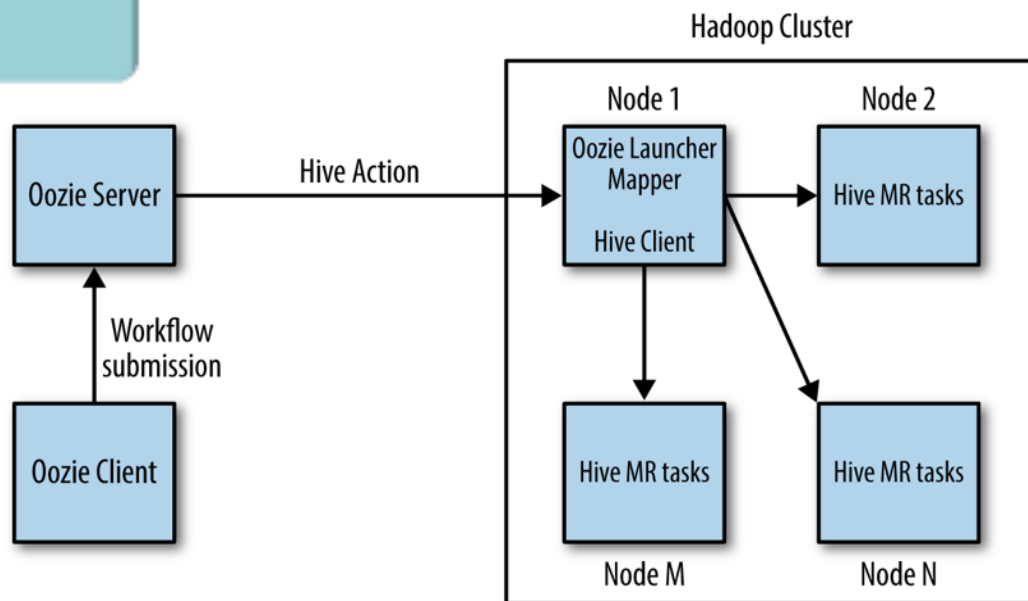
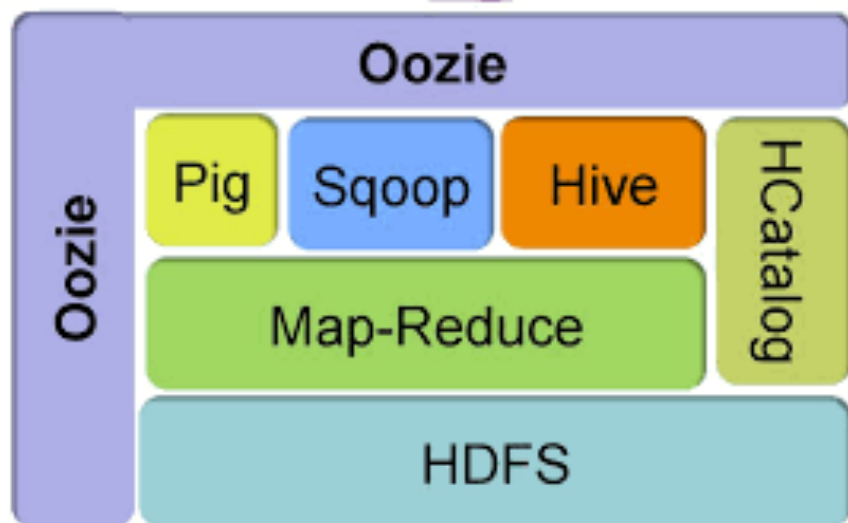
Hive	Pig
可透過客戶端程式或網頁介面存取 (Hive2, HUE)	可透過客戶端程式或網頁介面存取(HUE)
Facebook開發	Yahoo開發
SQL Like語言	Script Language – 適合實作ETL
需要Metastore(如MySQL)儲存與HDFS之對應關係	使用Pig Interpreter
資料有明確Schema、適合使用SQL語法存取時適用	適用於實作複雜的ETL功能 (Pig提供更多資料控制功能)

流程管理模組－Oozie簡介

- ▶ Oozie獨立的Java Web應用程式，可以接受Request而執行流程
- ▶ 流程(Workflow)是以XML定義(或可透過HUE編寫)
 - 流程步驟的動作由Sqoop、Pig、Hive、Java等程式實作
- ▶ Workflow啟動
 - 按需求
 - 定期
 - 透過API呼叫
 - 監控特定目錄是否有資料而執行



Oozie架構及運作示意



機器學習模組－Mahout簡介

- ▶ 支援使用MapReduce進行機器學習(Machine Learning)
- ▶ 使用Java語言實作ML演算法
- ▶ 支援多種機器學習演算法
 - 降維演算法(Dimension Reduction)
 - 推薦演算法(Collaborative Filtering)
 - 分群演算法
 - 分類演算法
 - 迴歸演算法



NoSQL資料庫－HBase簡介

- ▶ 為提供Hadoop隨機存取(Random Reads and Writes)的功能而生
- ▶ HBase的特點
 - NoSQL
 - 分散式系統平台上的資料庫
 - 多維度儲存
 - Key-Value的方式存放，多層的Map資料結構
 - 高效能
 - 各分散式節點同步處理Request，達到高效能的運作
 - 高可靠性(High-Availability)



即時SQL查詢模組－Impala簡介

- ▶ 由Cloudera發佈的Open Source項目
- ▶ 解決MapReduce及Hive處理較久之缺點，提供即時(Low-Latency)查詢功能
- ▶ Impala特色
 - In memory的處理，即時回應
 - 但無容錯功能
 - 相容HiveSQL，無縫接軌
- ▶ 適用情境
 - 有明確資料結構(Schema)
 - 查詢需即時得到結果
 - 無容錯性需求的作業



Impala's Architecture

Common Hive SQL and interface

Unified metadata

