hive学习笔记

# Hive简介

由facebook 开源, 最初用于解决海量结构化的日志数据统计问题;

ETL(Extraction-Transformation-Loading) 工具

构建于hadoop 的hdfs 和mapred 之上, 用于管理和查询结构化/ 非结构化数据的数据仓库

2008 年facebook 把hive 项目贡献给Apache

## Hive的优势

Hive 的出现, 完美的解决了传统数据分析人员所面临的问题：

1. Hive 使用类SQL 查询语法, 最大限度的实现了和SQL 标准的兼容。
2. JDBC 接口/ODBC 接口也使开发人员更易开发应用

为超大数据集设计的计算/ 扩展能力

1. MR 作为计算引擎
2. HDFS 作为存储系统

统一的元数据管理

1. 可与Pig 、Presto 等共享

## hive 的缺点

1. Hive 的HQL 表达的能力有限
   * 有些复杂运算用HQL 不易表达
2. Hive 效率较低
   * Hive 自动生成MapReduce 作业, 通常不够智能;
   * HQL 调优困难, 粒度较粗
   * 可控性差

# hive 的运行架构





# hive 系统架构

1. 元数据存储（ Metastore ）
2. 驱动（ Driver ）

* 编译器
* 优化器
* 执行器

1. 接口

* CLI
* HWI
* ThriftServer

1. Hadoop

* 用 MapReduce 进行计算
* 用 HDFS 进行存储

## 元数据存储（ MetaStore ）



### 存储数据库

1. Derby
2. MySQL



### Hive元数据表数据字典



## 驱动（ Driver ）

### 编译器（ hive 的核心）



1. 语义解析器（ ParseDriver ）：将查询字符串转换成解析树表达式
2. 语法分析器（ SemanticAnalyzer ）：将解析树转换成基于语句块的内部查询表达式。
3. 逻辑计划生成器（ logical plan generator ）

* 将内部查询表达式转换为逻辑计划，这些计划由逻辑操作树组成。
* 操作符是hive 的最小处理单元, 每个操作符处理代表一道HDFS 操作或MR 作业

1. 查询计划生成器（ query plan generator ）

* 将逻辑计划转化成物理计划（ MR JOB ）

### 优化器

优化器是一个演化组件。当前，它的规则是：列修剪，谓词下压。

### 执行器

编译器将操作树切分为一个JOB 链（ DAG ），执行器会顺序执行其中所有JOB ；如

果Task 链（ DAG ）不存在依赖关系时，可采用并发执行的方式进行Job 的执行。

## 接口

1. CLI ：命令行工具，为默认服务，启动方式为：
   * bin/hive
   * bin/hive --service cli
2. hwi ：为web 接口，可以通过浏览器访问hive ，默认端口9999，启动方式为：
   * bin/hive --service hwi
3. ThriftServer ：通过Thrift 对外提供服务，默认端口10000，启动方式为：
   * bin/hive --service hiveserver

## 其他服务（ bin/hive --service –help ）

1. metastore （ bin/hive --service metastore ）
2. hiveserver2 （ bin/hive --service hiveserver2 ）
   * HiveServer2 是HiveServer 的改进版本，它提供新的Thrift API 来处理JDBC 或者ODBC 客户端， Kerberos 身份验证，多个客户端并发
   * HiveServer2 还提供了新的CLI ： BeeLine ， Beeline 是hive 0.11 引入的新的交互式CLI ，它基于SQLLine ，可以作为Hive JDBC Client 端访问Hive Server 2 ，启动一个beeline就是维护了一个session 。

# hive 的数据模型

## Database

## Table

## Partition

## Bucket

## File