Google在开源的三篇论文GFS、MapReduce和BigTable后又开源了另外三篇论文Percolator（项目上线名称为Caffeine）、Pregel（图计算）和Dremel，针对Dremel的开源实现有Apache的Drill（较晚于Impala）和Cloudera的Impala

Impala是Cloudera公司推出提供对HDFS、Hbase数据的高性能、低延迟的交互式SQL查询，基于（依赖于）Hive使用内存计算，兼顾数据仓库、具有实时、批处理、多并发，CDH平台首选的PB级大数据实时查询分析引擎（Spark SQL也是PB级大数据实时查询分析引擎，但CDH不兼容Spark SQL）

Impala的Cloudera地址http://www.cloudera.com/products/apache-hadoop/impala.html Impala的Apache官方地址：<http://www.impala.io/index.html>

Impala的特点：

1、基于内存进行计算，能够对PB级数据进行交互式实时查询、分析

2、无需转换为MR，直接读取HDFS数据

3、C++编写，LLVM统一编译运行

4、兼容HiveSQL，Impala不支持Hive的复杂数据结构类型，不支持Hive的自定义函数UDF

5、具有数据仓库的特性， 可对hive数据直接做数据分析

6、支持Data Local

7、支持列式存储

8、支持JDBC/ODBC远程访问

Impala的缺点：对内存依赖大（官方建议内存128G），C++编写（源代码不易修改），完全依赖于hive，实践过程中分区超过1w性能严重下下降，稳定性不如hive（内存稳定性不如部分数据存放于硬盘的MapReduce）

一个Statestore Daemon（实例名statestored）：负责收集分布在集群中各个impalad进程的资源信息、各节点健康状况，同步节点信息，负责query的调度

一个Catalog Daemon（实例名catalogd，Impala1.2版本之后具有的服务）：自动同步Hive或HBase中的数据变化（只能同步元数据中存储过的数据变化，新创建的数据无法自动同步），分发表的元数据信息到各个impalad中，接收来自statestore的所有请求

多个Impala Daemon（实例名impalad）：接收client、hue、jdbc或者odbc请求、Query执行并返回给中心协调节点，子节点上的守护进程，负责向statestore保持通信，汇报工作

由于Impala是基于内存的计算，不排除Impala Daemon会占用Statestore Daemon和Catalog Daemon的内存导致进程挂掉，一般分配角色时将Impala Daemon与Statestore Daemon和Catalog Daemon分配到不同机器主机上

客户端发送SQL到指定impalad，收到SQL的impalad作为中心此次任务的中心协调节点（负责按数据所在节点分发其他impalad节点任务和收集计算结果），中心协调节点的Query Planner将SQL解析为执行计划后交由Query Coordinator，Query Coordinator根据数据所在节点并将任务分发给指定的impalad的Query Executor执行，Query Executor在执行过程中可以互相通信执行任务完成后将结果返回给中心协调节点，中心协调节点的Query Coordinator汇总数据结果后返回给客户端

Impala不能像MapReduce自己实现负载均衡需要手动控制

impala-shell命令默认连接本机Impala Shell客户端

在Impala Shell客户端外使用的Shell命令为Impala外部Shell命令（一般作用于整个Impala）格式为impala-shell 选项，选项可以为：

-h（--help）帮助

-v （--version）查询版本信息，内部Shell命令version会显示Impala Shell的版本和Server的版本（即Impala版本），更新Impala需要将Impala与Impala同时更新到相对应版本

-V（--verbose）启用详细输出

--quiet关闭详细输出

-p 显示执行计划

-i hostname（--impalad=hostname）指定连接主机格式 hostname:port默认端口21000

-r（--refresh\_after\_connect）刷新所有元数据，手动同步Hive或HBase中的数据变化

-q query（--query=query）从命令行执行查询，不进入impala-shell

-d default\_db（--database=default\_db）指定数据库

-B（--delimited）去格式化输出

--output\_delimiter=character 指定分隔符

--print\_header 打印列名

-f query\_file（--query\_file=query\_file）执行查询文件，以分号分隔

-o filename（--output\_file filename ）结果输出到指定文件

-c 查询执行失败时继续执行

-k （--kerberos）使用kerberos安全加密方式运行impala-shell

-l 启用LDAP认证

-u 启用LDAP时，指定用户名

在Impala Shell客户端内使用的Shell命令为Impala内部Shell命令：

help 帮助

connect hostname:port 连接主机，默认端口21000

refresh 表名 增量刷新元数据库，同步数据

invalidate metadata 全量刷新元数据库

explain SQL语句 显示查询执行计划、步骤信息

set explain\_level=数值 设置显示级别（0,1,2,3）默认2

shell Shell语句 不退出impala-shell执行Linux命令

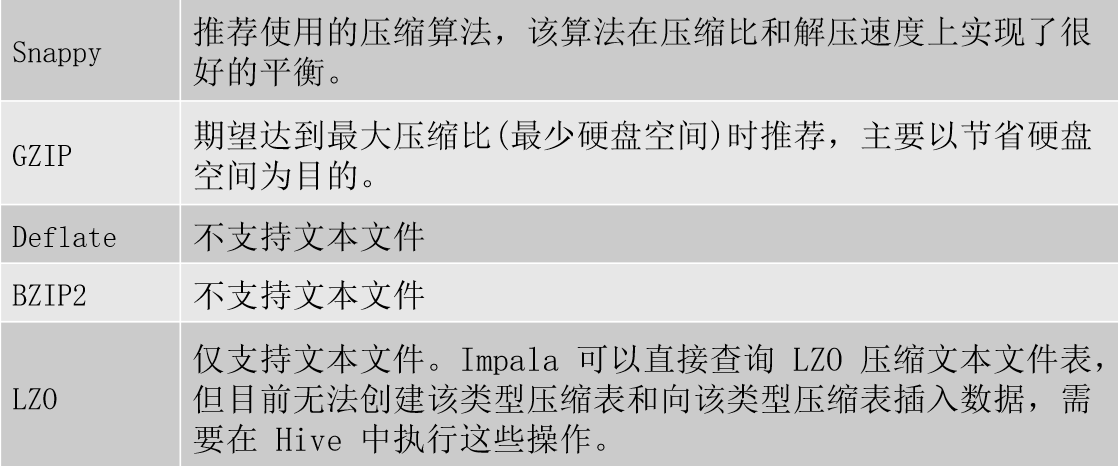
profile（查询完成后执行）查询最近一次查询的底层信息

查看StateStore的地址：http://hadoop1:25020/

查看Catalog的地址：<http://hadoop1:25010/>



压缩方式：



Impala操作分区方式与Hive类似，但只能在创建表时指定分区不支持对已经创建的表添加分区

Impala支持数据类型：INT、TINYINT、SMALLINT、BIGINT、BOOLEAN、CHAR、VARCHAR、STRING、FLOAT、DOUBLE、REAL、DECIMAL、TIMESTAMP

CDH5.5版本之后增加支持类型：ARRAY、MAP、STRUCT、Complex

Impala不支持HiveQL的特性：

可扩展机制，例如：TRANSFORM、自定义文件格式、自定义SerDes

XML、JSON函数

某些聚合函数：covar\_pop、covar\_samp、corr、percentile、percentile\_approx、 histogram\_numeric、collect\_set，Impala仅支持：AVG、COUNT、MAX、MIN、SUM

多Distinct查询

HDF、UDAF

语句： ANALYZE TABLE (Impala：COMPUTE STATS)、DESCRIBE COLUMN、 DESCRIBE DATABASE、EXPORT TABLE、IMPORT TABLE、SHOW TABLE EXTENDED、SHOW INDEXES、SHOW COLUMNS

Impala SQL的insert方式插入数据虽然速度很快可以使用，但由于insert方式插入的数据会产生小文件降低效率一般不使用

describe formatted 视图名 查看视图定义

Impala的视图不允许插入数据操作