一页纸精华 | Impala

原创 2016-03-28 牛家浩 中兴大数据

这是**中兴大数据**第**218**篇原创文章

要入门大数据，最好的办法就是理清Hadoop的生态系统。中兴大数据公众号将推出“一页纸精华”栏目，将用最精炼的语言，陆续为你介绍Hadoop生态系统的各个组件。本期为你介绍Hadoop交互式查询引擎Impala。

Hive在查询数据的时候，采用了MapReduce执行框架。由于MapReduce本身具有较高的延迟，因此在利用MapReduce执行Hive查询时，延时比较长。为了提升查询速度解决Hadoop批处理延迟问题，Cloudera公司发布了Impala实时查询引擎。

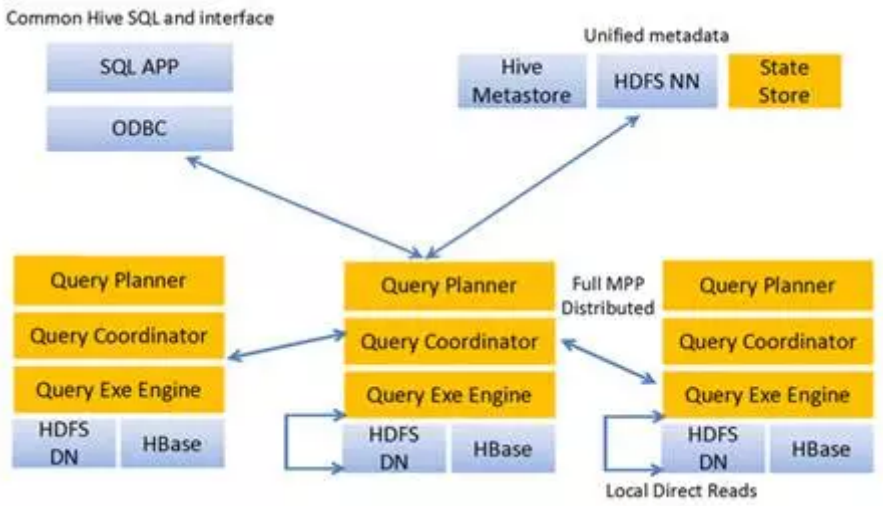
Impala是基于MPP的SQL查询系统，可以直接为存储在HDFS或HBase中的Hadoop数据提供快速、交互式的SQL查询。Impala和Hive一样也使用了相同的元数据、SQL语法（Hive SQL）、ODBC驱动和用户接口（Hue Beeswax），这就很方便的为用户提供了一个相似并且统一的平台来进行批量或实时查询。

Impala设计目标：

- 分布式环境下通用SQL引擎：既支持OLTP也支持OLAP
- SQL查询的规模和粒度：从毫秒级到小时级
- 底层存储依赖HDFS和HBase
- 使用更加高效的C++编写
- SQL的执行引擎借鉴了分布式数据库MPP的思想而不再依赖MapReduce

Impala体系结构

Impala系统架构图下图所示：

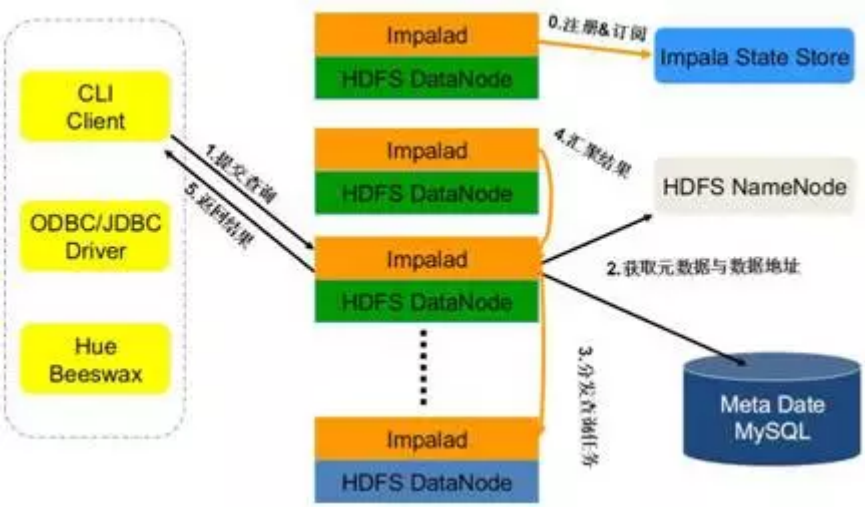


Impala主要包括以下组成部分：

- **Impala shell**：客户端工具，提供一个交互接口ODBC，供使用者连接到Impalad发起数据查询或管理任务等。
- **Impalad**：分布式查询引擎，由QueryPlanner、Query Coordinator和QueryExec Engine三部分组成，可以直接从HDFS或者HBase中用SELECT、JOIN和统计函数查询数据。
- **StateStore**：主要为跟踪各个Impalad实例的位置和状态，让各个Impalad实例以集群的方式运行起来。
- **CatalogService**：主要跟踪各个节点上对元数据的变更操作，并且通知到每个节点。

Impala查询处理基本流程

Impala查询处理流程如下图所示：



Impalad分为Java前端（Frontend）与C++处理后端（Backend），接受客户端连接的Impalad即作为这次查询的Coordinator，Coordinator通过JNI调用Java前端对用户的查询SQL进行分析生成执行计划树，不同的操作对应不同的PlanNode，如：SelectNode， ScanNode， SortNode， AggregationNode，HashJoinNode等等。

执行计划树的每个原子操作由一个PlanFragment表示，通常一条查询语句由多个PlanFragment组成， PlanFragment 0表示执行树的根，汇聚结果返回给用户，执行树的叶子结点一般是Scan操作，分布式并行执行。

Java前端产生的执行计划树以Thrift数据格式返回给ImpalaC++后端（Coordinator）（执行计划分为多个阶段，每一个阶段叫做一个PlanFragment，每一个PlanFragment在执行时可以由多个Impalad实例并行执行(有些PlanFragment只能由一个Impalad实例执行，如聚合操作)，整个执行计划为一执行计划树），由Coordinator根据执行计划，数据存储信息（Impala通过libhdfs与HDFS进行交互。通过hdfsGetHosts方法获得文件数据块所在节点的位置信息），通过调度器（现在只有simple-scheduler，使用round-robin算法）Coordinator：：Exec对生成的执行计划树分配给相应的后端执行器Impalad执行（查询会使用LLVM进行代码生成，编译，执行），通过调用GetNext()方法获取计算结果，如果是insert语句，则将计算结果通过libhdfs写回HDFS，当所有输入数据被消耗光，执行结束，之后注销此次查询服务。

Impala与Hive的关系

Impala与Hive都是构建在Hadoop之上的数据查询工具。从客户端使用来看Impala与Hive有很多的共同之处，如数据表元数据、ODBC/JDBC驱动、SQL语法、灵活的文件格式、存储资源池等。Hive适合于长时间的批处理查询分析，而Impala适合于实时交互式SQL查询。Impala给数据分析人员提供了快速实验、验证想法的大数据分析工具。

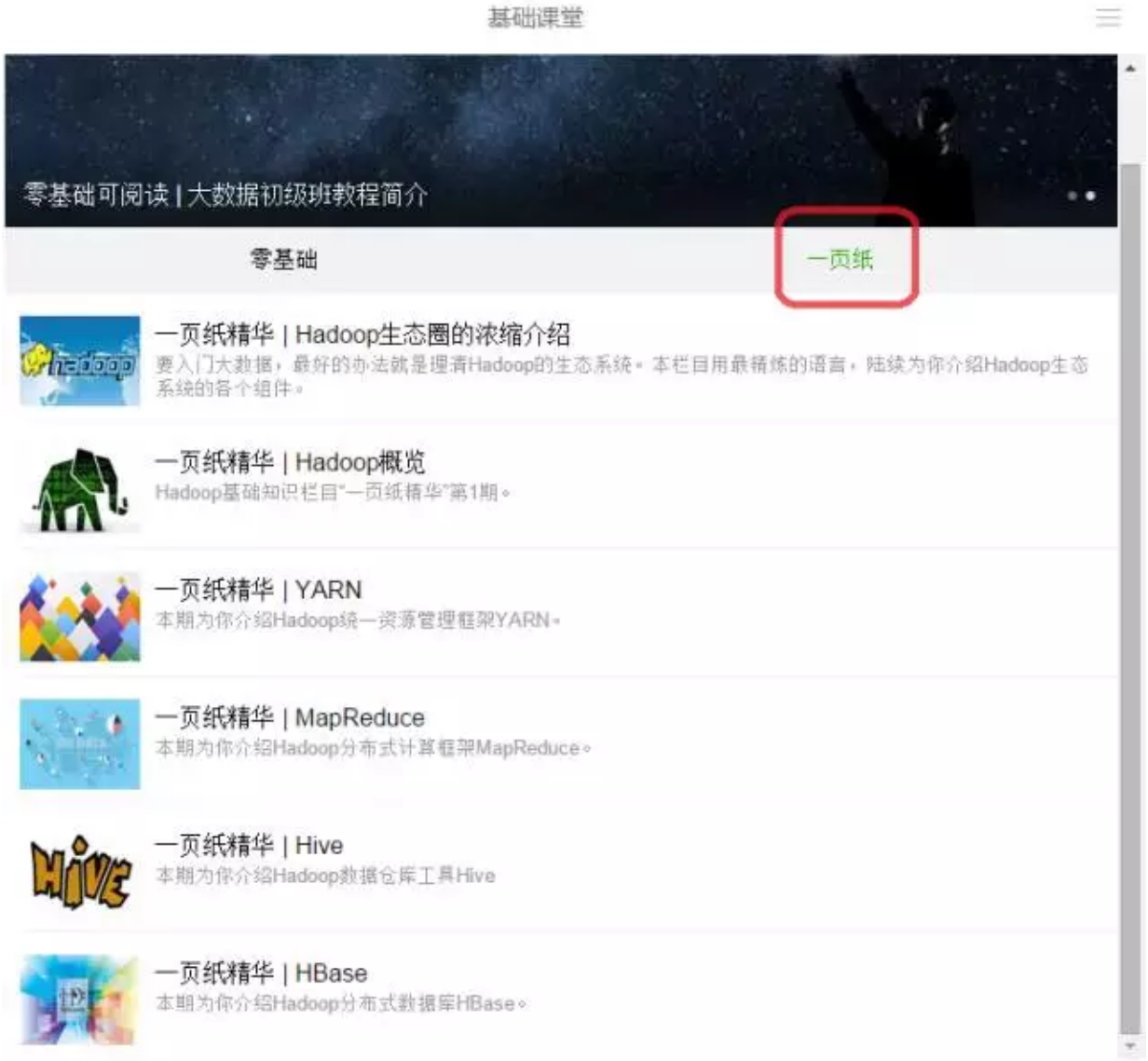
Impala相对于Hive的优势：

- Impala没有使用MapReduce进行并行计算，把整个查询分成一执行计划树，Impala使用拉式获取数据的方式获取结果，把结果数据组成按执行树流式传递汇集，减少的了把中间结果写入磁盘的步骤，再从磁盘读取数据的开销。Impala使用服务的方式避免每次执行查询都需要启动的开销，即相比Hive没了MapReduce启动时间。
- 使用LLVM产生运行代码，针对特定查询生成特定代码，同时使用Inline的方式减少函数调用的开销，加快执行效率。
- 充分利用可用的硬件指令。
- 更好的IO调度，Impala知道数据块所在的磁盘位置能够更好的利用多磁盘的优势，同时Impala支持直接数据块读取。
- 通过选择合适的数据存储格式可以得到最好的性能（Impala支持多种存储格式）。
- 最大使用内存，中间结果不写磁盘，及时通过网络以stream的方式传递。

如何阅读往期“一页纸精华”？

1 **进入公众号对话框界面，选择“干货专区”—“基础课堂”子菜单**

2 **在弹出页面选择“一页纸”栏目**



再来一篇？

长按二维码关注