技术剖析

分析型数据仓库中读写分离的实现

神策数据
SENSORS Data

和以 MySQL 为代表的传统事务型数据库相比，数据仓库有一个很大的特点，就是主要面向批量写和查询进行优化，可以不支持更新、事务这些高级特性。一些商用的数据仓库分析系统，例如 Vertica，已经可以做到千亿级数据的秒级导入和秒级查询。

神策数据一直致力于帮助企业搭建数据仓库，实现数据的秒级响应，积累数据资产。本文主要通过神策数据在技术上的探索与实践，探讨如何利用现有的开源组件实现分析型数据仓库当中的读写分离。

为什么要进行读写分离

分析型数据仓库一般有如下几个特点：

（1）面临着复杂的多维分析需求，能够进行任意维度的上卷下钻。

（2）存储的数据维度一般较多，所以是宽表，而且一般比较稀疏。

（3）数据量比较大，一次写入，多次查询。

针对这样特点，分析型数据库一般选择列存储数据格式，例如 Parquet 等。优点是对于统计分析效率很高，而且对于稀疏的宽表具有很高的存储压缩比。所以我们可以认为列存储格式是一种面向读进行优化的存储格式，我们称为 **Read-Optimized Store（ROS）**。

但是列存储格式也有一个缺点：这种格式的数据一旦生成，就很难进行修改，也很难往已有的数据文件当中插入新数据，只能增加新的数据文件。像 MySQL 这种传统的数据库，使用的行存储文件格式是一种适合修改和插入的存储格式，我们

可以认为这种行存储格式是面向写进行优化的存储格式，称为 **Write-Optimized Store（WOS）**。

综上所述，要实现一个可以秒级导入、秒级查询的分析型数据库，如果只选用 ROS，则很难支持大数据量的秒级导入。如果只选用 WOS，则很难实现任意维度的秒级查询，所以我们需要进行读写分离。

读写分离的实现原理

数据仓库当中需要同时存在 WOS 和 ROS，这样对于所有的写操作我们都生成 WOS 型文件；同时所有的读操作，则主要依赖于 ROS 文件，但也要查询少量的 WOS 文件。整体示意图如下：

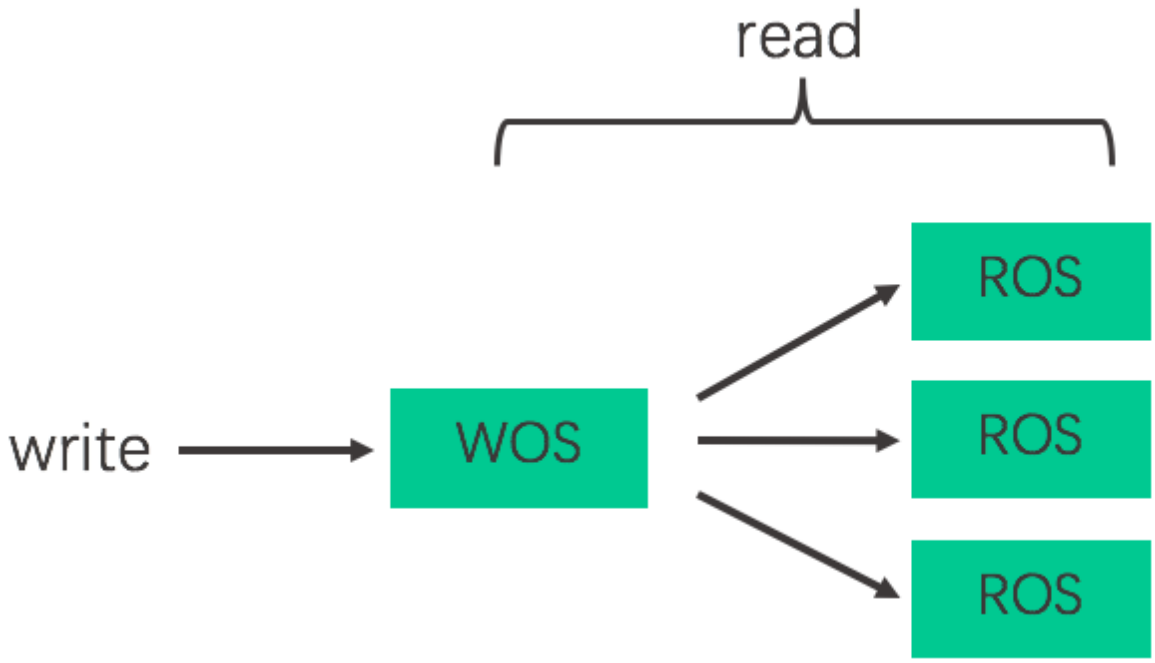

图 1 读写分离原理图

如图所示，WOS 文件需要定期转换为 ROS 文件，同时因为 ROS 在数据仓库当中一般是分为多个 Partition 存在，所以一个 WOS 可能转化为多个 ROS。转化的过程需要是原子操作，因为对上层查询引擎来说，同一时刻，同样的数据只能有一份。

开源方案的操作

前面简单介绍了读写分离方案的原理，具体的工程实践过程中，神策数据的工程师还面临着很多方案的选择和实践难点。下面简单介绍一下神策数据在搭建数据仓库的实践中啃过的“硬骨头”。

ROS 的选择比较简单，我们的工程师选择了 Parquet + Impala 的查询方案，同时结合我们的业务特点做了很多代码级别的优化。（*相关链接：付力力：基于 Impala 构建实时用户行为分析引擎*）WOS 的选择可能会比较多，我们可以选择常用的 HDFS 行存储文件格式，例如 TextFile、SequenceFile、Avro 等。

以 SequenceFile 为例，我们在定义自己的 Impala 表的时候，可以指定一个特殊的 Partition 文件的存储格式为 SequenceFile，同时其它的 Partition 作为正常的按照日期 Partition 的数据，指定格式为 Parquet，这种方式的优势体现在始终只有一个表。

后来基于查询效率和未来架构升级方面的考虑，我们最终选择了 Kudu 作为 WOS，架构实现示意图如下：

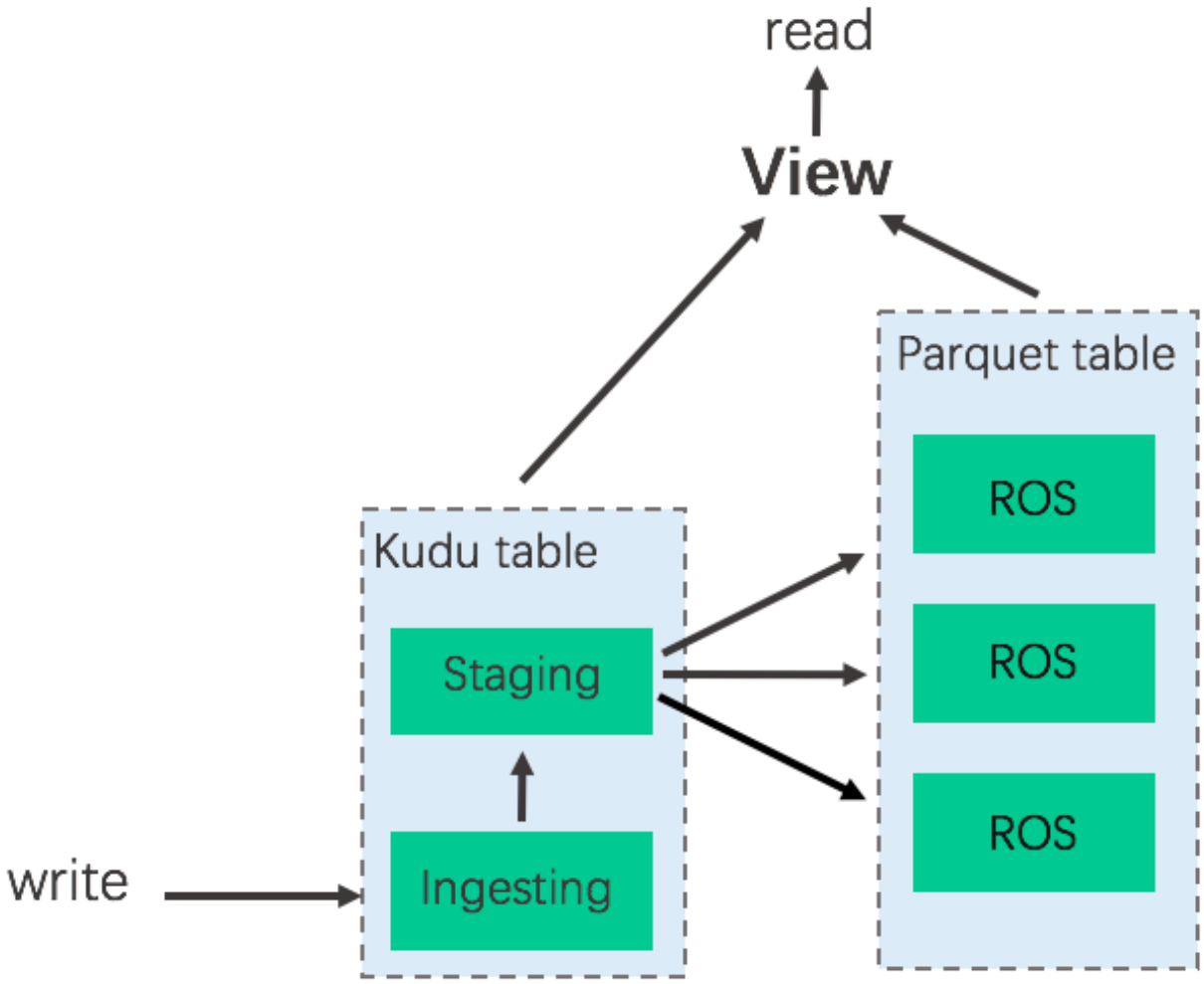

图 2 读写分离的实现图

如图所示，我们会建立三张物理表，其中两张 Kudu 表作为 WOS，一张 Parquet 表作为 ROS。所有的写操作都会写入到 Ingesting 状态的 Kudu 表中，当 Ingesting 表写到一定大小之后，会自动转换为 Staging 状态。

这时我们一方面生成一张新的 Kudu 表作为 Ingesting 表，另一方面开始 WOS 到 ROS 的转换，通过一个叫做 Mover 的任务执行这个操作。将 Staging 状态的 Kudu 表中的数据全部转换到对应 Partition 的 Parquet 表当中。

Staging 状态的表转换完成且 Ingesting 状态的表写满时，会触发一个切表操作，需要更新元数据，告诉 Impala 使用新的数据进行查询，整个切表的操作是原子的。而且已经转化的 Staging 表还需要保留一段时间，避免切表之前发起的查询操作没有及时执行完成。

对于查询请求来说我们会建立一个包含 Staging 表、Ingesting 表和 ROS 表的虚拟表，即一个 View。用户的查询始终指向一个 View，但是下面的物理表会经常发生变化。这样就兼顾查询数据的不断更新及查询性能的优化两方面了。

在实现的过程中还有很多具体的工作，例如如何对表进行加列操作，保证各个表的结构一致；Parquet 表中碎文件较多影响查询效率，如何定期合并等。限于篇幅，这里不再具体介绍。神策数据最终的技术架构如下图：

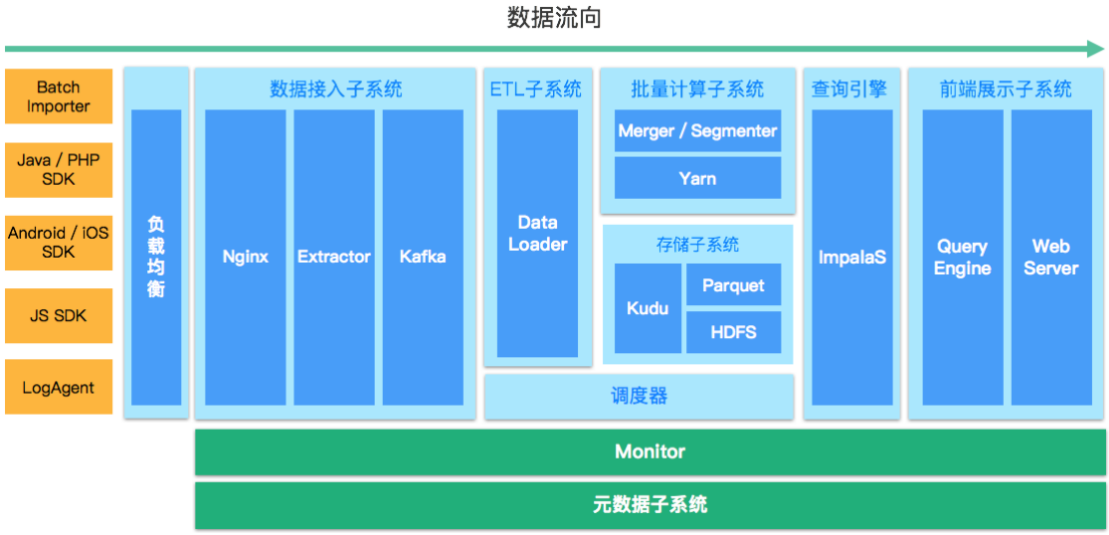

图 3 神策数据技术架构图

综上所述，神策数据为了实现数据驱动，在数据仓库的读写效率方面做了比较深入的探索，也参考了众多优秀的开源项目，做了适配产品的优化，累计十万行代码以上，大数据行业技术才是企业的核心竞争力，也希望大家在技术和业务层面进行开放性的探讨。

——本文作者为神策数据资深研发工程师张广强，版权归神策数据所有。

关于神策数据：

神策数据(https://www.sensorsdata.cn),一家专业的大数据分析服务公司，致力于帮助客户实现数据驱动。公司推出深度用户行为分析产品神策分析(Sensors Analytics ），支持私有化部署、全端数据接入，并作为 PaaS 平台支持二次开发。此外，还提供大数据相关咨询和完整解决方案。目前已赢得中国银联、中国电信、百度视频、百联、万达、人民日报、中邮消费金融、广发证券、四川航空、链家、聚美优品、中商惠民、趣店、纷享销客、Keep、秒拍、36 氪等数百家行业领先企业认可。希望更深入了解神策数据或有数据驱动相关问题咨询，请咨询 4006509827，由专业的工作人员为您解答。

神策数据
SENSORS Data

关注我们

试用产品

帮助企业实现数据驱动

邮箱：contact@sensorsdata.cn
电话：400-650-9827
网址：www.sensorsdata.cn