Apache Hudi 常见问题汇总

Apache Hudi 对个人和组织何时有用

如果你希望将数据快速提取到HDFS或云存储中，Hudi可以提供帮助。另外，如果你的ETL /hive/spark作业很慢或占用大量资源，那么Hudi可以通过提供一种增量式读取和写入数据的方法来提供帮助。

作为一个组织，Hudi可以帮助你构建高效的数据湖，解决一些最复杂的底层存储管理问题，同时将数据更快地交给数据分析师，工程师和科学家。

如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

Hudi不打算达成的目标

Hudi不是针对任何OLTP案例而设计的，在这些情况下，通常你使用的是现有的NoSQL / RDBMS数据存储。Hudi无法替代你的内存分析数据库（至少现在还没有！）。Hudi支持在几分钟内实现近乎实时的摄取，从而权衡了延迟以进行有效的批处理。如果确实希望亚-
分钟处理延迟，请使用你最喜欢的流处理解决方案。

什么是增量处理？为什么Hudi一直在谈论它

增量处理是由Vinoth Chandar在O'reilly博客中首次引入的，博客中阐述了大部分工作。用纯粹的技术术语来说，增量处理仅是指以流处理方式编写微型批处理程序。典型的批处理作业每隔几个小时就会消费所有输入并重新计算所有输出。典型的流处理作业会连续/每隔几秒钟消费一些新的输入并重新计算新的/更改以输出。尽管以批处理方式重新计算所有输出可能会更简单，但这很浪

费并且耗费昂贵的资源。Hudi具有以流方式编写相同批处理管道的能力，每隔几分钟运行一次。

虽然可将其称为流处理，但我们更愿意称其为增量处理，以区别于使用Apache Flink，Apache Apex或Apache Kafka Streams构建的纯流处理管道。

写时复制（COW）与读时合并（MOR）存储类型之间有什么区别

写时复制（Copy On Write）：此存储类型使客户端能够以列式文件格式（当前为parquet）摄取数据。使用COW存储类型时，任何写入Hudi数据集的新数据都将写入新的parquet文件。更新现有的行将导致重写整个parquet文件（这些parquet文件包含要更新的受影响的行）。因此，所有对此类数据集的写入都受parquet写性能的限制，parquet文件越大，摄取数据所花费的时间就越长。

读时合并（Merge On Read）：此存储类型使客户端可以快速将数据摄取为基于行（如avro）的数据格式。使用MOR存储类型时，任何写入Hudi数据集的新数据都将写入新的日志/增量文件，这些文件在内部将数据以avro进行编码。压缩（Compaction）过程（配置为嵌入式或异步）将日志文件格式转换为列式文件格式（parquet）。

两种不同的格式提供了两种不同视图（读优化视图和实时视图），读优化视图取决于列式parquet文件的读取性能，而实时视图取决于列式和/或日志文件的读取性能。

更新现有的行将导致：

- 写入从以前通过压缩（Compaction）生成的基础parquet文件对应的日志/增量文件更新；
- 在未进行压缩的情况下写入日志/增量文件的更新。因此，对此类数据集的所有写入均受avro /日志文件写入性能的限制，其速度比parquet快得多（写入时需要复制）。虽然，与列式（parquet）文件相比，读取日志/增量文件需要更高的成本（读取时需要合并）。

如何为工作负载选择存储类型

Hudi的主要目标是提供更新功能，该功能比重写整个表或分区要快几个数量级。

如果满足以下条件，则选择写时复制（COW）存储：

- 寻找一种简单的替换现有的parquet表的方法，而无需实时数据。
- 当前的工作流是重写整个表/分区以处理更新，而每个分区中实际上只有几个文件发生更改。
- 想使操作更为简单（无需压缩等），并且摄取/写入性能仅受parquet文件大小以及受更新影响文件数量限制
- 工作流很简单，并且不会突然爆发大量更新或插入到较旧的分区。COW写入时付出了合并成本，因此，这些突然的更改可能会阻塞摄取，并干扰正常摄取延迟目标。

如果满足以下条件，则选择读时合并（MOR）存储：

- 希望数据尽快被摄取并尽可能快地可被查询。
- 工作负载可能会突然出现模式的峰值/变化（例如，对上游数据库中较旧事务的批量更新导致对DFS上旧分区的大量更新）。异步压缩（Compaction）有助于缓解由这种情况引起的写放大，而正常的提取则需跟上上游流的变化。

不管选择何种存储，Hudi都将提供：

- 快照隔离和原子写入批量记录
- 增量拉取
- 重复数据删除能力

Hudi是分析型数据库吗

典型的数据库有一些长时间运行的服务器，以便提供读写服务。Hudi的体系结构与之不同，它高度解耦读写，为对应扩容挑战可以独立扩展写入和查询/读取。因此，它可能并不总是像数据库一样。

尽管如此，Hudi的设计非常像数据库，并提供类似的功能（更新，更改捕获）和语义（事务性写入，快照隔离读取）。

如何对存储在Hudi中的数据建模

在将数据写入Hudi时，可以像在键-值存储上那样对记录进行建模：指定键字段（对于单个分区/整个数据集是唯一的），分区字段（表示要放置键的分区）和preCombine/combine逻辑（用于指定如何处理一批写入记录中的重复记录）。该模型使Hudi可以强制执行主键约束，就像在数据库表上一样。请参阅此处的示例。

当查询/读取数据时，Hudi只是将自己显示为一个类似于json的层次表，每个人都习惯于使用Hive/Spark/Presto 来对Parquet/Json/Avro进行查询。

Hudi是否支持云存储/对象存储

一般来说，Hudi能够在任何Hadoop文件系统实现上提供该功能，因此可以在Cloud Store（Amazon S3或Microsoft Azure或Google Cloud Storage）上读写数据集。Hudi还进行了特定的设计，使在云上构建Hudi数据集变得非常容易，例如S3的一致性检查，数据文件涉及的零移动/重命名。

Hudi支持Hive/Spark/Hadoop的哪些版本

从2019年9月开始，Hudi可以支持Spark 2.1 +，Hive 2.x，Hadoop 2.7+（非Hadoop 3）。

Hudi如何在数据集中实际存储数据

从更高层次上讲，Hudi基于MVCC设计，将数据写入parquet/基本文件以及包含对基本文件所做

更改的日志文件的不同版本。所有文件都以数据集的分区模式存储，这与Apache Hive表在DFS上的布局方式非常相似。请参考这里了解更多详情。

欢迎Star&Fork：https://github.com/apache/incubator-hudi

本文原文：ApacheHudi常见问题汇总