Apache Hudi 0.7.0 版本发布，新特性介绍

本版本迁移指南

- If migrating from release older than 0.5.3, please also check the upgrade instructions for each subsequent release below.
- Specifically check upgrade instructions for 0.6.0. This release does not introduce any new table versions.
- The HoodieRecordPayload interface deprecated existing methods, in favor of new ones that also lets us pass properties at runtime. Users are
  encouraged to migrate out of the deprecated methods, since they will be removed in 0.9.0.

重点特性

Clustering

0.7.0版本中支持了对Hudi表数据进行Clustering（对数据按照数据特征进行聚簇，以便优化文件大小和数据布局），Clustering提供了更灵活地方式增加文件大小，有了Clustering特性，便可更快速地摄取数据，然后聚簇为更大的文件，实验数据表明查询性能可以提升34倍，文件数可以减少1020倍；另外Clustering对于查询侧优化也很明显，在查询时通常会基于字段进行Clustering，通过完全跳过一些文件来极大提升查询性能，这与云数仓Snowflake提供的Clustering功能非常类似，我们非常高兴地宣称这个特性在0.7.0版本中完全开源免费。

想要了解更多Clustering细节，可以参考RFC-19，可以查阅这些配置来在你的数据管道中启用Clustering，现在Hudi支持同步和异步的Clustering模式。

Metadata Table

Hudi项目始于Uber，开始是基于HDFS实现的数据湖，对于云上对象存储的数据湖性能不如HDFS。在0.7.0版本，我们解决了该问题，即支持了内部Metadata表，此表可存储索引数据，其他元数据信息等。

Metadata表的实现使用了Hudi MOR表，这意味着像其他任何Hudi表一样，可以被压缩（Compaction）、清理（Clean）、增量更新（incrementally updated）。 而且与其他项目中的类似实现不同，我们选择将文件列表等信息索引为HFile格式（格式可插拔），HFile提供了很好的点查性能，可以高效获取分区文件列表等信息。

在0.7.0版本中，在写入端配置hoodie.metadata.enable=true即可构建Metadata表，这样后续操作将不再调用fs.listStatus()接口，我们引入了一种同步机制来保证对数据timeline中进行的文件新增/删除操作都会同步到Metadata表。

测试有25W个文件的表，Metadata表相比使用Spark并发Listing要快2~3倍，更多设计细节可查阅

RFC-15，其他Metadata表相关配置可参考这里，提供了参数以便在生产环境中安全使用该特性。

Java/Flink Writers

Hudi最开始设计时依赖Spark，但随着项目成为Apache顶级项目，我们意识到需要抽象内部表格式、表服务、写入层的代码以支持更多的引擎。在0.7.0版本，我们完成了写入层的解耦，添加了Flink和Java客户端，现在你可以使用HoodieFlinkStreamer来消费Kafka中的数据，以写入Hudi的COW表中。

写入端优化

- Spark3支持：0.7.0版本支持使用Spark3进行写入和查询，请注意使用scala 2.12版本的hudi-spark-bundle包；
- 并行Listing：我们已将所有List操作移至HoodieTableMetadata接口下，该接口可以多线程/Spark并行执行，该优化可以在未开启Metadata表时提升清理、压缩性能。
- Kafka Commit Callbacks：0.7.0添加了HoodieWriteCommitKafkaCallback接口，当每次进行commit后可以向Kafka中发送事件，以此来触发派生/ETL数据管道，类似Apache Airflow中的Sensors
- Insert Overwrite/Insert Overwrite Table：0.7.0版本中新增了这两种操作类型，主要用于批处理ETL作业，该作业通常会在每次运行时覆盖整个表/分区。考虑到这些操作可以批量替换目标表，因此这些操作比upsert更合适，请查看[示例](/docs/ quick-start-guide.html # insert-overwrite-table)。
- 删除分区支持：对于使用WriteClient/RDD级别API的用户，Hudi提供了一个新的API来删除整个分区，而不是采用记录级别删除方式。
- 新增 DefaultHoodieRecordPayload 解决乱序问题： 当前默认的OverwriteWithLatestAvroPayload将覆盖存储中已有的值，即使使用较旧值进行upsert。0.7.0版本添加了一个新的DefaultHoodieRecordPayload和一个有效负载配置hoodie.payload.ordering.field来指定一个字段，可以将传入的upsert记录与已存储的记录进行比较，以决定是否覆盖。推荐用户使用这种更新、更灵活的Payload模型。
- Hive同步：支持使用 SlashEncodedHourPartitionValueExtractor 同步小时分区至 Hive 中。
- 支持IBM云对象存储 以及 Open Java 9版本。

查询端优化

- Spark Datasource 支持 MOR 增量查询：0.7.0版本支持使用 Spark datasource 增量查询 MOR 表，在后续版本中会继续加强和重构该特性。
- Metadata表支持 File Listings：用户还可以将元数据表用于以下查询端，对于Hive，设置hoodie.metadata.enable=true会话

属性，对于使用SparkSQL查询注册的Hive表，请使用参数 --conf spark.hadoop.hoodie.metadata.enable = true 来允许从元数据中获取分区的文件列表，而非使用File Listing。

更多关于 Apache Hudi Release Note 参见 这里