

[Apache Spark 社区期待的 Delta Lake 开源了](#)

本文英文原文：[Open Sourcing Delta Lake](#)

2019年4月24日在美国旧金山召开的 Spark+AI Summit 2019 会上，Databricks 的联合创始人及 CEO Ali Ghodsi 宣布将 Databricks Runtime 里面的 Delta Lake 基于 Apache License 2.0 协议开源。Delta Lake 是一个存储层，为 Apache Spark 和大数据 workloads 提供 ACID 事务能力，其通过写和快照隔离之间的乐观并发控制（optimistic concurrency control），在写入数据期间提供一致性的读取，从而为构建在 HDFS 和云存储上的数据湖（data lakes）带来可靠性。Delta Lake 还提供内置数据版本控制，以便轻松回滚。目前 Delta Lake 项目地址为 <https://delta.io/>，代码维护地址 <https://github.com/delta-io/delta>。



如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：iteblog_hadoop

为什么需要 Delta Lake

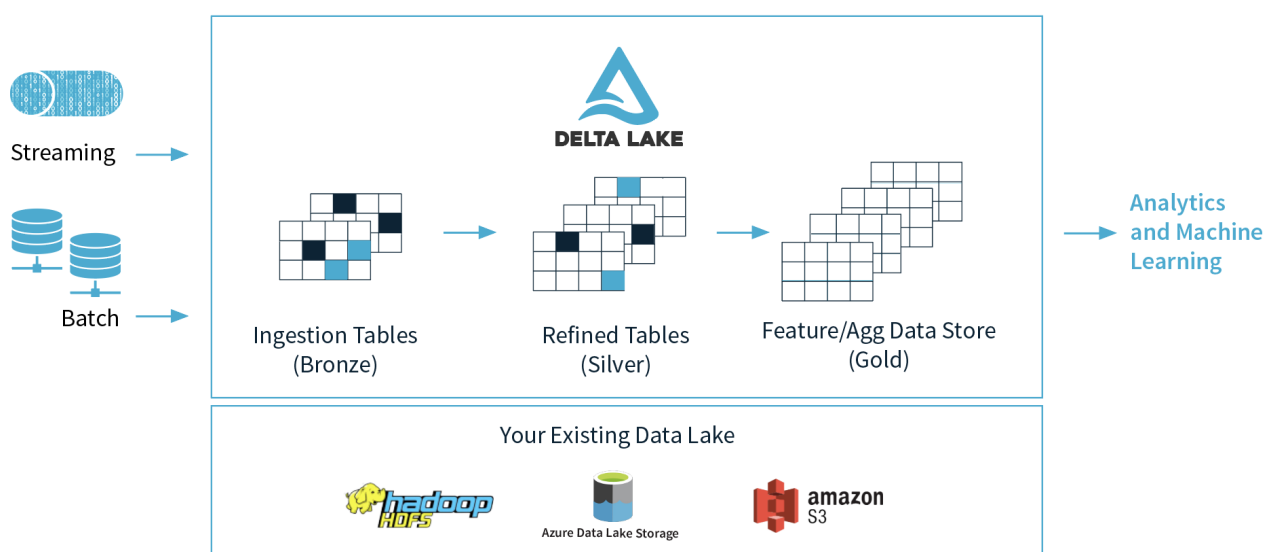
现在很多公司内部数据架构中都存在数据湖，数据湖是一种大型数据存储库和处理引擎。它能够存储大量各种类型的数据，拥有强大的信息处理能力和处理几乎无限的并发任务或工作的能力，最早由 Pentaho 首席技术官詹姆斯迪克森在2011年的时候提出。虽然数据湖在数据范围方面迈出了一大步，但是也面临了很多问题，主要概括如下：

- 数据湖的读写是不可靠的。
数据工程师经常遇到不安全写入数据湖的问题，导致读者在写入期间看到垃圾数据。他们必须构建方法以确保读者在写入期间始终看到一致的数据。
- 数据湖中的数据质量很低。
将非结构化数据转储到数据湖中是非常容易的。但这是以数据质量为代价的。没有任何验证模式和数据的机制，导致数据湖的数据质量很差。因此，努力挖掘这些数据的分析项目也会失败。

- 随着数据的增加，处理性能很差。
随着数据湖中存储的数据量增加，文件和目录的数量也会增加。处理数据的作业和查询引擎在处理元数据操作上花费大量时间。在有流作业的情况下，这个问题更加明显。
- 数据湖中数据的更新非常困难。
工程师需要构建复杂的管道来读取整个分区或表，修改数据并将其写回。这种模式效率低，并且难以维护。

由于存在这些挑战，许多大数据项目无法实现其愿景，有时甚至完全失败。我们需要一种解决方案，使数据从业者能够利用他们现有的数据湖，同时确保数据质量。这就是 Delta Lake 产生的背景。

Delta Lake 开源项目介绍



如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：iteblog_hadoop

Delta Lake 很好地解决了上述问题，以简化我们构建数据湖的方式。Delta Lake 提供以下主要功能：

支持 ACID 事务

Delta Lake 在多并发写入之间提供 ACID 事务保证。每次写入都是一个事务，并且在事务日志中记录了写入的序列顺序。事务日志跟踪文件级别的写入并使用乐观并发控制，这非常适合数据湖，因为多次写入/修改相同的文件很少发生。在存在冲突的情况下，Delta Lake 会抛出并发修改异常以使用户能够处理它们并重试其作业。Delta Lake 还提供强大的可序列化隔离级别，允许工程师持续写入目录或表，并允许消费者继续从同一目录或表中读取。读者将看到阅读开始时存在的最新快照。

模式管理 (Schema management)

Delta Lake 自动验证正在被写的 DataFrame 模式是否与表的模式兼容。表中存在但不在 DataFrame 中的列设置为 null。如果 DataFrame 有表中不存在的列，则此操作会引发异常。Delta Lake 具有显式添加新列的 DDL 以及自动更新模式的能力。

可扩展元数据处理

Delta Lake 将表或目录的元数据信息存储在事务日志中，而不是 Metastore 中。这允许 Delta Lake 在恒定时间内列出大型目录中的文件，同时在读取数据时非常高效。

数据版本

Delta Lake 允许用户读取表或目录之前的快照。当文件被修改文件时，Delta Lake 会创建较新版本的文件并保留旧版本的文件。当用户想要读取旧版本的表或目录时，他们可以在 Apache Spark 的读取 API 中提供时间戳或版本号，Delta Lake 根据事务日志中的信息构建该时间戳或版本的完整快照。这允许用户重现之前的数据，并在需要时将表还原为旧版本的数据。

统一流和批处理 Sink

除批量写入外，Delta Lake 还可用作 Apache Spark structured streaming 的高效流式 sink。结合 ACID 事务和可扩展的元数据处理，高效的流式 sink 现在可以实现大量近实时分析用例，而无需同时维护复杂的流式传输和批处理管道。

数据存储格式采用开源的

Delta Lake 中的所有数据都是使用 Apache Parquet 格式存储，使 Delta Lake 能够利用 Parquet 原生的高效压缩和编码方案。

记录更新和删除

这个功能马上可以使用。Delta Lake 将支持 merge, update 和 delete 等 DML 命令。这使得数据工程师可以轻松地在数据湖中插入/更新和删除记录。由于 Delta Lake 以文件级粒度跟踪和修改数据，因此它比读取和覆盖整个分区或表更有效。

数据异常处理

Delta Lake 还将支持新的 API 来设置表或目录的数据异常。工程师能够设置一个布尔条件并调整报警阈值以处理数据异常。当 Apache Spark 作业写入表或目录时，Delta Lake 将自动验证记录，当数据存在异常时，它将根据提供的设置来处理记录。

100% 兼容 Apache Spark API

这点非常重要。开发人员可以将 Delta Lake 与他们现有的数据管道一起使用，仅需要做一些细微的修改。比如我们之前将处理结果保存成

Parquet 文件，如果想使用 Delta Lake 仅仅需要做如下修改：

```
dataframe  
.write  
.format("parquet")  
.save("/data")
```

修改成

```
dataframe  
.write  
.format("delta")  
.save("/data")
```

本博客文章除特别声明，全部都是原创！

转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)

本文链接: 【】 ()