

盘点2020年晋升为Apache TLP的大数据相关项目

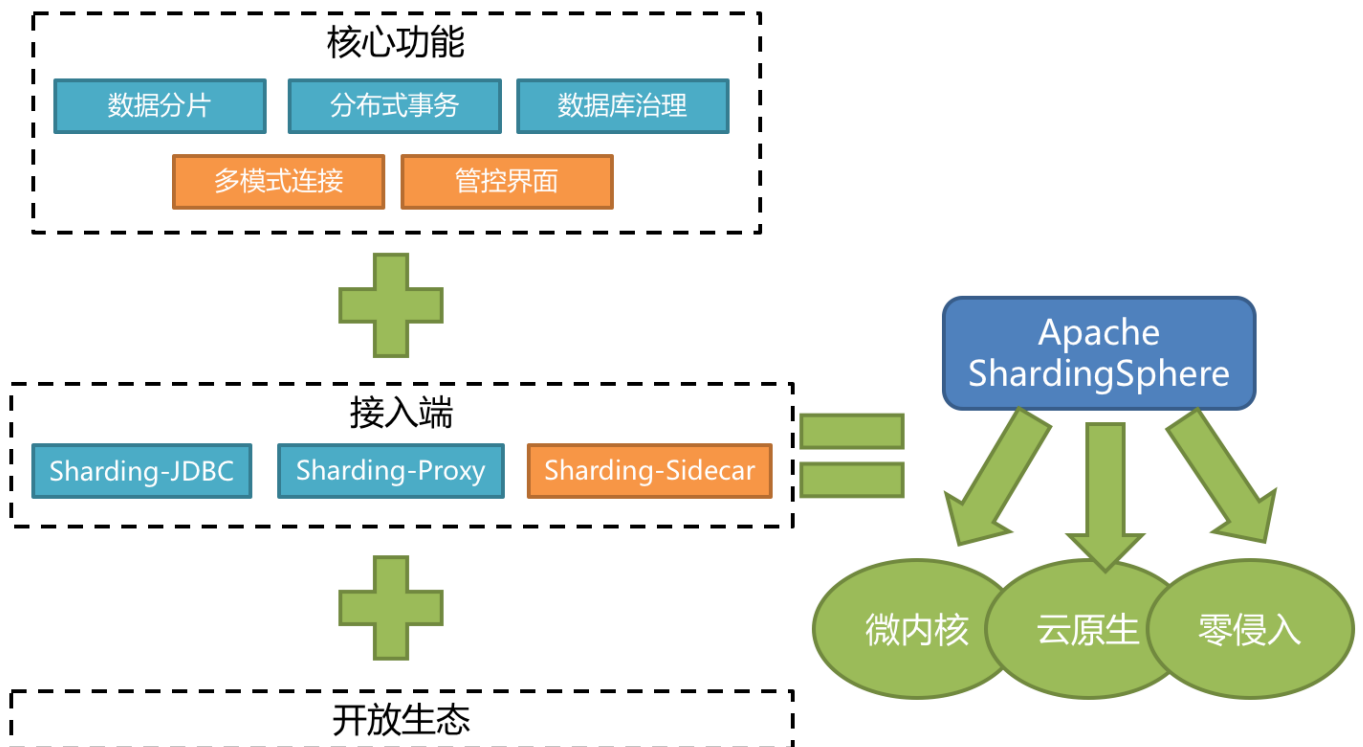
在过去一年有很多 Apache 孵化项目顺利毕业成顶级项目（Top-Level Project，简称 TLP），在这里我将给大家盘点 2020 年晋升为 Apache TLP 的大数据相关项目。在2020年一共有四个大数据相关项目顺利毕业成顶级项目，主要是 Apache® ShardingSphere™、Apache® Hudi™、Apache® Iceberg™ 以及 Apache® IoTDB™，这里以毕业的时间顺序依次介绍。

关于过去几年毕业成 TLP 的大数据项目可以参见[《盘点2017年晋升为Apache TLP的大数据相关项目》](#)、[《盘点2018年晋升为Apache TLP的大数据相关项目》](#) 以及 [《盘点2019年晋升为Apache TLP的大数据相关项目》](#)。

Apache

ShardingSphere：开源的分布式数据库中间件解决方案组成的生态圈

Apache ShardingSphere 是一套开源的分布式数据库中间件解决方案组成的生态圈，它由 JDBC、Proxy 和 Sidecar（规划中）这三款相互独立，却又能够混合部署配合使用的产品组成。它们均提供标准化的数据分片、分布式事务和数据库治理功能，可适用于如 Java 同构、异构语言、云原生等各种多样化的应用场景。其架构如下：



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

Apache ShardingSphere 定位为关系型数据库中间件，旨在充分合理地在分布式的场景下利用关系型数据库的计算和存储能力，而并非实现一个全新的关系型数据库。它通过关注不变，进而抓住事物本质。关系型数据库当今依然占有巨大市场，是各个公司核心业务的基石，未来也难于撼动，我们目前阶段更加关注在原有基础上的增量，而非颠覆。

Apache ShardingSphere 是由京东主导，多个公司共同贡献的项目。它是京东集团首个进入 Apache 基金会的开源项目，也是 Apache 基金会首个分布式数据库中间件项目。该项目于2018年11月进入 Apache 孵化器，并已于2020年4月16日成为 Apache 软件基金会的顶级项目（参见 [The Apache Software Foundation Announces Apache® ShardingSphere™ as a Top-Level Project](#)）。关于 Apache ShardingSphere 更多资料可以参见官方网站：<https://shardingsphere.apache.org/>。

Apache Iceberg：跟踪超大规模表的数据湖解决方案

Apache Iceberg 最初由 Netflix 设计开发，主要为了解决列出大量的分区耗时以及元数据和 HDFS 数据不一致等问题设计的轻量级数据湖解决方案。是一种用于跟踪超大规模表的新格式，是专门为对象存储（如S3）而设计的，其核心思想是在时间轴上跟踪表的所有变化。Iceberg 中有个比较重要的概念就是快照（snapshot），快照表示表数据文件的一个完整集合，每次更新操作会生成一个新的快照。

Apache Iceberg 主要有以下几个特点：

- 优化数据入库流程：Iceberg 提供 ACID 事务能力，上游数据写入即可见，不影响当前数据处理任务，这大大简化了 ETL；Iceberg 提供了 upsert、merge into 能力，可以极大地缩小数据入库延迟；
- 支持更多的分析引擎：优秀的内核抽象使之不绑定特定的计算引擎，目前 Iceberg 支持的计算引擎有 Spark、Flink、Presto 以及 Hive。
- 统一数据存储和灵活的文件组织：提供了基于流式的增量计算模型和基于批处理的全量表计算模型。批处理和流任务可以使用相同的存储模型，数据不再孤立；Iceberg 支持隐藏分区和分区进化，方便业务进行数据分区策略更新。支持 Parquet、Avro 以及 ORC 等存储格式。
- 增量读取处理能力：Iceberg 支持通过流式方式读取增量数据，支持 Structed Streaming 以及 Flink table Source。

Apache Iceberg 于2018年11月16日进入 Apache 孵化器，并于2020年05月20日成功毕业成 Apache 顶级项目。比较奇怪的是，并没有看到 Apache 官方宣布其成为顶级项目，本文主要参考：<https://incubator.apache.org/projects/iceberg.html> 以及 <https://incubator.apache.org/projects/iceberg.html>。

关于 Apache Iceberg 的更详细信息可以参见其官方网站：<https://iceberg.apache.org/>

Apache Hudi：大数据增量处理框架

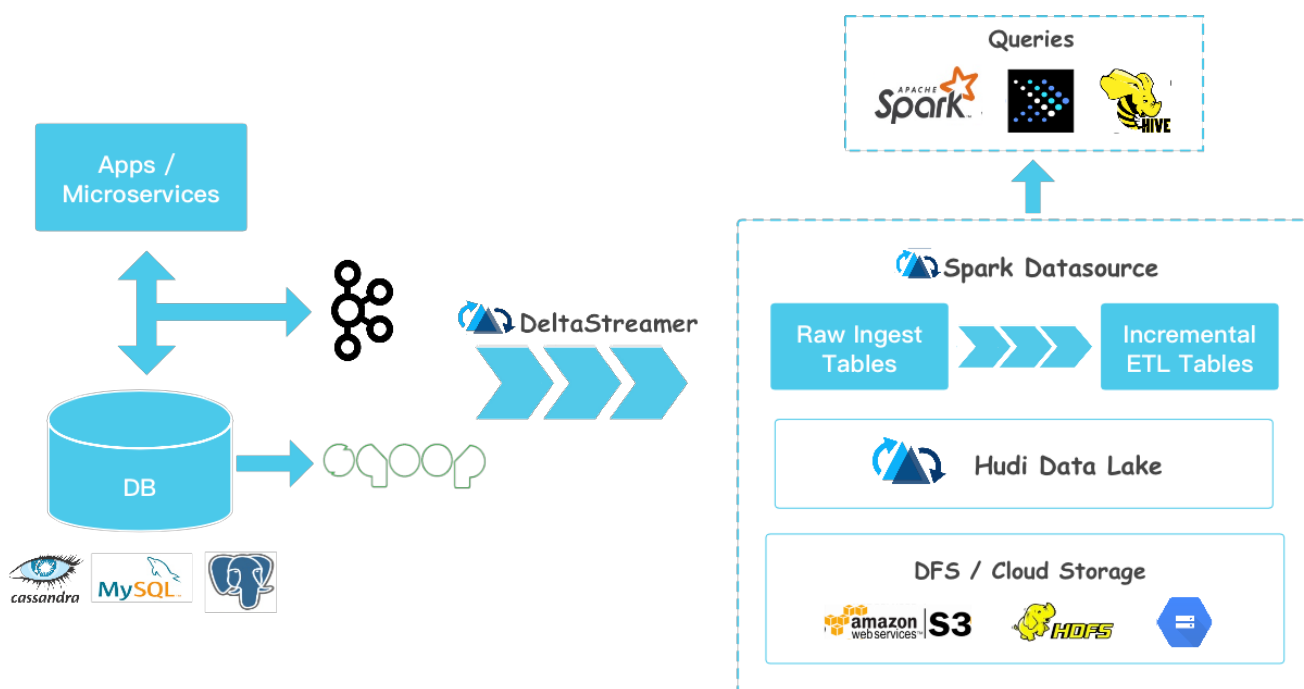
Apache Hudi (Hadoop Upsert Delete and Incremental) 是 Uber

为了解决大数据生态系统中需要插入更新及增量消费原语的摄取管道和 ETL 管道的低效问题。它是一种分析和扫描优化的数据存储抽象，可在几分钟之内将变更应用于 HDFS 中的数据集中，并支持多个增量处理系统处理数据。通过自定义的 InputFormat 与当前 Hadoop 生态系统(包括 Apache Hive、Apache Parquet、Presto 和 Apache Spark)集成，使得该框架对最终用户来说是无缝的。

Hudi 的设计目标就是为了快速增量更新 HDFS

上的数据集，它提供了两种更新数据的方式：Copy On Write 和 Merge On Read。Copy On Write 模式就是我们更新数据的时候需要通过索引获取更新的数据所涉及的文件，然后把这些数据读出来和更新的数据进行合并，这种模式更新数据比较简单，但是当更新涉及到的数据比较大时，效率非常低；而 Merge On Read 就是将更新写到单独的新文件里面，然后我们可以选择同步或异步将更新的数据和原来的数据进行合并（可以称为 combination），因为更新的时候只写新的文件，所以这种模式更新的速度会比较快。

有了 Hudi 之后，我们可以实时采集 MySQL、HBase、Cassandra 里面的增量数据然后写到 Hudi 中，然后 Presto、Spark、Hive 可以很快地读取到这些增量更新的数据，如下：



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

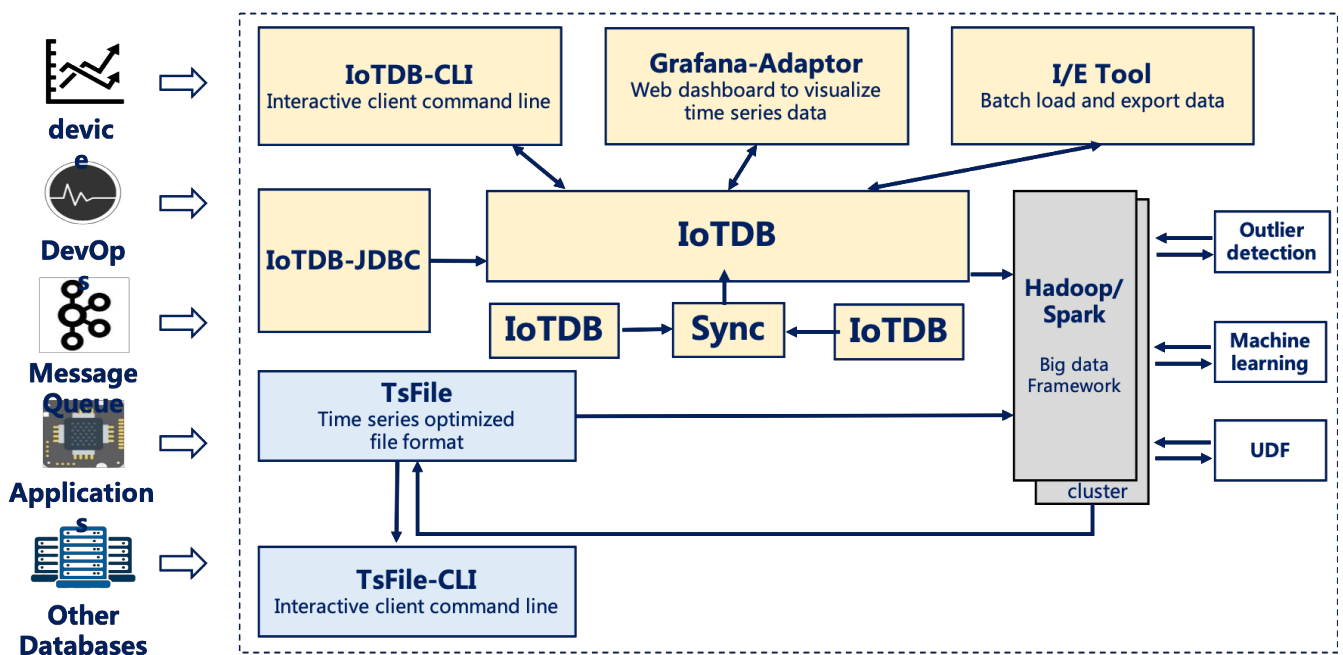
Apache Hudi 项目在2016年开始开发，当时在 Uber 的内部代号为 Hoodie，其于2017年开源，2019年1月进入 Apache 孵化器，[2020年6月4日正式成为顶级项目](#)。

更多关于 Apache Hudi 的介绍可以参见过往记忆大数据的 [《Apache Hudi: Uber 开源的大数据增量处理框架》](#) 以及 [《Uber 大数据平台的演进（2014~2019）》](#) 的介绍，以及 Apache Hudi 的官方文档：<http://hudi.apache.org/>

Apache IoTDB : 物联网数据库

Apache IoTDB (物联网数据库) 是一体化收集、存储、管理与分析物联网时序数据的软件系统。Apache IoTDB 采用轻量式架构, 具有高性能和丰富的功能, 并与 Apache Hadoop、Spark 和 Flink 等进行了深度集成, 可以满足工业物联网领域的海量数据存储、高速数据读取和复杂数据分析需求。

Apache IoTDB 套件由若干个组件构成, 共同形成“数据收集-数据写入-数据存储-数据查询-数据可视化-数据分析”等一系列功能。其架构如下:



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

用户可以通过 JDBC 将来自设备上传感器采集的时序数据、服务器负载和 CPU 内存等系统状态数据、消息队列中的时序数据、应用程序的时序数据或者其他数据库中的时序数据导入到本地或者远程的 IoTDB 中。用户还可以将上述数据直接写成本地 (或位于 HDFS 上) 的 TsFile 文件。可以将 TsFile 文件写入到 HDFS 上, 进而实现在 Hadoop 或 Spark 的数据处理平台上的诸如异常检测、机器学习等数据处理任务。对于写入到 HDFS 或者本地的 TsFile 文件, 可以利用 TsFile-Hadoop 或 TsFile-Spark 连接器允许 Hadoop 或 Spark 进行数据处理。对于分析的结果, 可以写回成 TsFile 文件。IoTDB 和 TsFile 还提供了相应的客户端工具, 满足用户查看和写入数据的 SQL 形式、脚本形式和图形化形式等多种需求。

Apache IoTDB

是清华大学自研的时间序列数据库, 其于2014年项目启动, 2018年11月18日正式进入 Apache 孵化器, 是中国高校首个进入 Apache 孵化器的项目, [2020年9月23日正式毕业成为 Apache 顶级项目](#)。

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: **【】**（**）**