

Apache Kafka 2.5.0 稳定版正式发布

Apache Kafka 2.5.0 稳定版于美国当地时间2020年4月15日正式发布，这个版本包含了一系列的重要功能发布，比较重要的可以特性重要包括：

- 支持 TLS 1.3 (目前默认是用 1.2)
- Kafka Streams DSL 中支持 Co-groups ;
- Kafka Consumer 支持增量再平衡 (Incremental rebalance)
- 为更好地洞察算子运行，引入了新的指标；
- Apache Zookeeper 升级到 3.5.7
- 不再支持 Scala 2.11
- KIP-500 有了新的进展。



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

如果需要下载最新版的 Apache Kafka 2.5.0，可以到 [这里](#) 下载。由于时间和篇幅的关系，下面文章列出本次比较重要的特性和进展。

Kafka broker, producer, 和 consumer 的更新

KIP-500 有了新的进展

用过 Kafka 的用户应该清楚，Kafka 的 Broker、topic 以及分区相关的信息是存放在 Apache Zookeeper 中的，Apache Zookeeper 是一个独立的系统，有其独立的存储，为了 Kafka

集群能够更好地运行，开发者们必须对 Kafka 和 Zookeeper 两个系统都很熟悉，这在一定程度上是增加了我们的负担。为了减少使用者的负担，KIP-500 提议将之前存储在 Zookeeper 里面的东西全部移到 Kafka 集群内部维护，也就是说未来 Kafka 不需要依赖 Zookeeper 即可运行。

Apache Kafka 2.5.0 版本对这项工作有了新的进展，比如 KIP-555 中解决管理员工具中不再需要请求 Zookeeper，KIP-543 中使得动态配置不再需要请求 Zookeeper。

KIP-500 是一个大功能，它的最后解决需要在未来几个版本才能体现。

仅且一次语义 (Exactly once semantics) 改进

KIP-447: Producer 可伸缩性支持仅且一次语义

这个 KIP 简化了以事务方式从 Kafka 读写的应用程序的 API。以前，这个用例通常需要为每个输入分区提供单独的生产者实例，现在这个限制已经被取消了。这使得构建消费大量分区的仅且一次语义的应用程序变得更加容易。详情可以参见 KIP-447。

KIP-360: 提高幂等/事务性生产者的可靠性

这个 KIP 解决了生产者状态保留在 broker 上问题，使得幂等性保证成为可能。在此之前，当日志被截断以强制保留或删除记录的调用中截断时，broker 将删除生产者的状态，这将导致 UnknownProducerId 异常。通过这个改进，broker 将保留 producer 状态直到过期。这个 KIP 还为生产者提供了一种从意外错误中恢复的强大方法。

Kafka Streams

KIP-150: Add Cogroup to the DSL

过去，将多个流聚合成一个流可能很复杂，而且容易出错。通常我们需要将所有流分组 (group) 并聚合 (aggregate) 到表中，然后进行多个 outer join 调用。新的 co-group 算子简化了这个过程，其减少了状态存储调用的数量，总体上提高了性能。具体使用如下：

```
KTable<K, CG> cogrouped =  
  grouped1  
    .cogroup(agggregator1)  
    .cogroup(grouped2, aggregator2)  
    .cogroup(grouped3, aggregator3)  
    .aggregate(initializer1, materialized1);
```

KIP-523: Add toTable() to the DSL

将事件流解释为一个变更日志并将其物化成一张表表是一种强大的方法。KIP-523 引入了 toTable() 函数的可以将其应用到流中，并能够将每个 key 的最新值进行物化。需要注意的是，任何空值 (null) 都将被解释为对给定键的删除。

KIP-535: 允许状态存储在重新平衡期间提供过时的数据

以前，在进行重新平衡期间，针对状态存储的交互式查询 (IQ) 将失败。这降低了依赖于查询 Kafka Streams 状态表功能的应用程序的正常运行时间。KIP-535 使应用程序能够查询状态存储的任何副本，并能够观察到每个副本落后于主副本多远。

Dropped support and deprecations

我们已经在 Apache Kafka 2.5 中删除了对 Scala 2.11 的支持，现在 Scala 2.12 和 2.13 是唯一受支持的版本。

TLS 1.2 现在是默认的 SSL 协议，不过仍然支持 TLS 1.0 和 1.1。

更多请查看这篇文章：[What's New in Apache Kafka 2.5](#)

Apache Kafka 2.5.0 Release

Notes：https://downloads.apache.org/kafka/2.5.0/RELEASE_NOTES.html

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: 【】（）