

## Apache Kafka 2.7.0 稳定版发布，有哪些值得关心的变化？

Apache Kafka 2.7.0 于2020年12月21日正式发布，这个版本是目前 Kafka 最新稳定版本，大家可以根据需要自行决定是否升级到次版本，关于各个版本升级到 Apache Kafka 2.7.0 请参见 [《Upgrading to 2.7.0 from any version 0.8.x through 2.6.x》](#)。

在这个版本中，社区仍然在推进从 Kafka 移除对 ZooKeeper 的依赖，比如这个版本在 KIP-497 里面添加了可以修改 ISR 的 Broker 内部 API；在 KIP-500 里面增加了自元数管理（Self-Managed Metadata Quorum）的 Raft 核心实现，这个也是去掉 Zookeeper 的一部分工作。现在在 Kafka 的源代码里面有一个名为 raft 的模块专门用于实现共识协议（consensus protocol）。在与控制器（Kafka 集群中负责管理分区和副本状态的 broker）的集成完成之前，有一个独立的服务器可以用来测试 Raft 实现的性能。

当然，为了取代 Zookeeper，还有更多的工作正在努力进行中，很多 KIP 正在积极开发中，以使得每个集群能够支持更多的分区、更简单的操作和更严格的安全性。

分级存储（Tiered Storage）工作也正在继续进行中，这个可以参见 KIP-405。



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 Apache Kafka 2.7.0，可以到 [这里](#) 下载。由于时间和篇幅的关系，下面我们只介绍 Kafka 2.7.0 版本部分变化，更多内容请参见 [Kafka 2.7.0 Release Notes](#)。

### Kafka broker, producer 和 consumer 的更新

## KIP-654: 终止带有任何未刷新数据的事务时抛出一个非致命异常

在 Kafka 2.7.0 之前，当 Java client producer 程序终止带有任何未刷新（挂起）数据的事务时，将抛出一个致命异常。但是，终止一个有挂起数据的事务实际上被认为是一种正常的情况。抛出的异常应该是通知我们没有发送记录，而不是通知应用程序处于不可恢复状态。所以，在 KIP-654 中引入了一个名为 TransactionAbortedException 的异常，允许我们在需要时重试。

## KIP-651: 私钥和 SSL证书支持 PEM 格式

目前，Kafka 在使用 SSL 时只支持基于 JKS 或 PKCS12 文件的密钥和信任存储。虽然 Privacy-Enhanced Mail

(PEM) 不再是电子邮件的标准，但它是存储和分发加密密钥和证书的标准格式。在 KIP-651 中为密钥和信任存储增加了对 PEM 文件的支持，允许使用依赖于 PEM 格式的第三方提供商。

## KIP-612: 能够限制 broker 上的连接创建速率

创建连接会给 broker 增加 CPU 开销。连接风暴 (Connection storms) 可能来自表面上表现良好的客户机，并可能阻止 broker 执行其他有用的工作。这个版本添加了一种方法可以强制限制每个 broker 中每个监听器创建连接的速率。2.7.0 版本包含了 KIP-612 的第一部分，每个 IP 的连接限制预计会在 2.8.0 版本中出现。

## KIP-599: 支持对创建主题、创建分区和删除主题的操作进行节流

创建主题、创建分区和删除主题的 API 是直接影响 Kafka 控制器总体负载的操作。为了防止由于高并发主题和分区创建或主题删除而导致集群不堪重负，KIP-599 中引入了一个新的配额来限制这些操作。

## KIP-584: Versioning scheme for features

除了 broker-client 之间的兼容性，其他方面的兼容性比较差，比如 Kafka 中添加了新的特性，会带来两个主要的问题：

- Kafka 客户端如何知道 Broker 这个新功能？
- Broker 端如何决定启用哪些功能？

KIP-584 为客户端发现、功能识别和柔性升级提供了灵活和操作友好的解决方案。

## KIP-554: 添加 Broker 端的 SCRAM 配置 API

通过 KIP-554, SCRAM 凭据可以通过 Kafka 协议进行管理，kafka-configs 工具使用了这个新的协议 API 以便在 Broker 端对 SCRAM 进行配置，这个也是 Kafka 移除 Zookeeper 项目的一部分。

## KIP-497: 添加可以修改 ISR 的 Broker 内部 API

目前，Kafka partition leader 和 ISR 信息存储在 ZooKeeper 中。控制器或分区 leader 都可以更新这个信息。因为两者都可以更新此状态，所以需要有一种机制来共享此信息，这可能会出现一方出现 ISR 信息延迟，这些延迟的影响意味着元数据请求可能接收到旧的信息。

在 2.7.0 版本中，添加了一个名为 AlterIsr 新的 API，它赋予 controller 独家更新分区 leader 和 ISR 状态的能力。这个新 API 的主要好处是元数据请求将始终获取到最新的状态。更多的信息可以参见 KIP-497。

## KIP-431: 使用 ConsoleConsumer 打印 Record 中的其他字段

现在我们可以使用 ConsoleConsumer 打印 ConsumerRecord 中的头信息，具体细节可以参见 KIP-431。

## Kafka Connect 的更新

### KIP-632: 添加 DirectoryConfigProvider 类

KIP-632 添加了 DirectoryConfigProvider 类，以支持需要为存储在容器文件系统（比如 Kubernetes 环境）中的密钥提供保护的用户。

## Kafka Streams 的更新

### KIP-662: 当 Kafka Streams 应用程序的源主题被删除时抛出异常

现在，如果用户删除正在运行的 Kafka Streams 应用程序的源主题，其中的消费者客户端会正常关闭。这个客户端关闭触发重新平衡，直到 Streams 应用程序的所有流线程优雅退出，使应用程序完全关闭，没有任何方法来处理这个错误。这个版本添加了 KIP-662，当用户从正在运行的 Streams 应用程序中删除源主题时，应用程序将抛出 MissingSourceTopicException，允许我们对错误作出反应。

### KIP-648: 为交互式查询重命名 getter 方法

KIP-648 改变了交互式查询对象的 getter 方法，使其遵循不使用 get 前缀的 Kafka 格式。

### KIP-617: 允许 Kafka Stream 状态存储向后迭代

目前，在 Kafka Streams 状态存储上使用迭代器时，只能从最老的元素遍历到最新的元素。当迭代一个有窗口的状态存储时，如果用户希望返回最新的 N 条记录，那么别无选择，只能使用低效的方法，即遍历所有最旧的记录，然后再获得所需的新记录。KIP-617 增加了对反向状态存储迭代的支持，反向迭代使最新的 N 条记录检索更加有效。

### KIP-613: 添加了 Kafka Stream 端到端的延迟度量

目前，Kafka Stream 中消息的实际端到端延迟是很难估计的，Kafka Stream 现在公开了端到端的指标，可以使得用户在选型是做出更好的选择，具体细节参见 KIP-613。

## KIP-607: 在 Kafka Stream 中添加了 RocksDB 属性相关的度量

当前 Kafka Stream 中 RocksDB 公开的指标不包括内存或磁盘使用情况。在 2.7.0 中，Kafka Stream 将 RocksDB 相关属性也暴露出来了，具体细节参见 KIP-607。

## KIP-450: DSL 中的滑动窗口支持聚合操作

Kafka Streams 实现了会话窗口（session windows），滚动窗口（tumbling windows）和跳跃窗口（hopping windows）作为窗口聚合方法。虽然提前时间较短的跳转窗口可以模仿滑动窗口的行为，但这种实现的性能很差，因为它会导致许多重叠且经常是冗余的窗口，这需要昂贵的计算。不过，在 KIP-450 中，Kafka Stream 现在提供了一种有效的方式来世行滑动聚合。

本文翻译自：[What's New in Apache Kafka 2.7.0](#)

**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: 【】（）**