重磅：Kafka 迎来 1.0.0 版本，正式告别四位数版本号！

Kafka 从首次发布之日起，已经走过了七个年头。从最开始的大规模消息系统，发展成为功能完善的分布式流式处理平台，用于发布和订阅、存储及实时地处理大规模流数据。来自世界各地的数千家公司在使用 Kafka，包括三分之一的 500 强公司。Kafka 以稳健的步伐向前迈进，首先加入了复制功能和无边界的键值数据存储，接着推出了用于集成外部存储系统的 Connect API，后又推出了为实时应用和事件驱动应用提供原生流式处理能力的 Streams API，并于今年春季开始支持仅一次处理语义。如此广泛的应用和完备的功能以及如此悠久的历史，无一不在说明 Kafka 已经成为一款稳定的企业级产品。而更为激动人心的是，Kafka 现在正式迎来了 1.0.0 版本！

Kafka 1.0.0 主要更新

- 0.10.0 版本里开始引入的 Streams API 在 1.0.0 版本里继续演进，改进了 builder API（KIP-120），新增了用于查看运行时活跃任务的 API（KIP-130）和用于聚合分区的 cogroup API（KIP-150）。增强的 print() 和 writeAsText() 方法让调试变得更容易（KIP-160）。其他更多信息可以参考 Streams 文档。
- 改进了 Connect 的度量指标（KIP-196），新增了大量用于健康监测的度量指标（KIP-188），并提供了集群的 GloabalTopicCount 和 GlobalPartitionCount 度量指标（KIP-168）。
- 支持 Java 9，实现更快的 TLS 和 CRC32C，加快了加密速度，降低了计算开销。
- 调整了 SASL 认证模块的错误处理逻辑（KIP-152），原先的认证错误信息现在被清晰地记录到日志当中。
- 更好地支持磁盘容错（KIP-112），更优雅地处理磁盘错误，单个 JBOD 上的磁盘错误不会导致整个集群崩溃。
- 0.11.0 版本中引入的幂等性生产者需要将 max.in.flight.requests.per.connection 参数设置为 1，这对吞吐量造成了一定的限制。而在 1.0.0 版本里，这个参数最大可以被设置为 5（KAFKA-5949），极大提升了吞吐量范围。

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 Kafka 1.0.0 RELEASE NOTES下载Apache Kafka 1.0.0

崛起的 Kafka

Kafka 起初是由 LinkedIn 公司开发的一个分布式的消息系统，后成为 Apache 的一部分，它使用 Scala 编写，以可水平扩展和高吞吐率而被广泛使用。目前越来越多的开源分布式处理系统如 Cloudera、Apache Storm、Spark 等都支持与 Kafka 集成。

随着微服务的流行，很多公司都在尝试将现有的系统进行架构升级。促成 Movio 公司架构改造的一项关键技术就是 Kafka 消息队列。

Kafka 作为分布式消息队列，在可靠性和可扩展性方面有非常大的优势。它不仅成为了 Movio 公司基础架构的关键组成部分，还为正在创建的系统架构提供了依据。

Kafka 全面解析

Kafka 数据可靠性深度解读

Kafka 作为一个商业级消息中间件，消息可靠性的重要性可想而知。如何确保消息的精确传输？如何确保消息的准确存储？如何确保消息的正确消费？这些都是需要考虑的问题。

唯品会消息中间件团队首先从 Kafka 的架构着手，解释了 Kafka 的基本原理，然后通过对 kakfa 的存储机制、复制原理、同步原理、可靠性和持久性保证等等一步步对其可靠性进行分析，最后

通过 benchmark 来增强对 Kafka 高可靠性的认知。

Kafka Stream 设计详解

本文介绍了 Kafka Stream 的背景，如 Kafka Stream 是什么，什么是流式计算，以及为什么要有 Kafka Stream。接着介绍了 Kafka Stream
的整体架构、并行模型、状态存储以及主要的两种数据集 KStream 和 KTable。然后分析了 Kafka Stream 如何解决流式系统中的关键问题，如时间定义、窗口操作、Join
操作、聚合操作，以及如何处理乱序和提供容错能力。最后结合示例讲解了如何使用 Kafka Stream。

Kafka 不只是个消息系统

Confluent 联合创始人兼 CEO Jay Kreps 发表了一篇博文，指出了 Kafka 的真正定位——它不只是个消息系统，它还是个存储系统，而它的终极目标是要让流式处理成为现代企业的主流开发范式。

人们更多的是把 Kafka 当成了消息队列系统。消息队列有一些不成文的规则，比如“不要在消息队列里保存消息”。传统的消息系统在设计上存在很多不足。从根本上讲，任何一个异步消息系统都会保存消息，只是时间很短，有时候只有几秒钟，直到消息被消费为止。

实际上，Kafka 并非传统意义上的消息队列，它与 RabbitMQ
等消息系统并不一样。它更像是一个分布式的文件系统或数据库。Kafka
与传统消息系统之间有三个关键区别。

- Kafka 持久化日志，这些日志可以被重复读取和无限期保留
- Kafka 是一个分布式系统：它以集群的方式运行，可以灵活伸缩，在内部通过复制数据提升容错能力和高可用性
- Kafka 支持实时的流式处理

以上三点足以将 Kafka
与传统的消息队列区别开，我们甚至可以把它看成是流式处理平台。因此，在 Kafka
里存储数据并不是什么疯狂事，甚至可以说 Kafka 本来就是设计用来存储数据的。数据经过校验后被持久化在磁盘上，并通过复制副本提升容错能力。再多的数据都不会拖慢
Kafka，在生产环境中，有些 Kafka 集群甚至已经保存超过 1 TB 的数据。