HZ BOOKS
华章IT

资深架构师多年工作经验结晶

包含Kafka源代码分析与内部的实现原理，以及外部的维护工具、客户端编程、与第三方集成方式，书中穿插了大量的图片，讲解细致、便于理解

大数据
技术丛书

Source Code Analysis and Application of Kafka

Kafka源码解析与实战

王亮◎编著

机械工业出版社
China Machine Press

内容简介

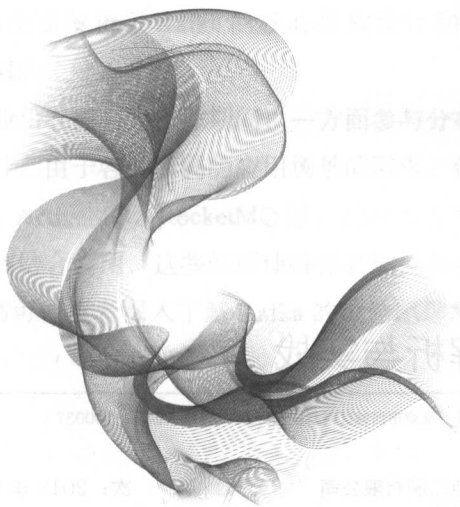
本书系统介绍Kafka的实现原理和应用方法，并介绍Kafka的运维工具、客户端编程方法和第三方集成方式，深入浅出、图文并茂、分析透彻。本书共10章，主要内容包括：第1章介绍Kafka诞生的背景和主要设计目标。第2章介绍Kafka的基本组成、拓扑结构以及内部的通信协议。第3章介绍Broker Server及内部的模块组成。第4章介绍Broker Server内部的九大基本模块。第5章介绍Broker的控制管理模块。第6章介绍Topic的管理工具。第7章从设计原则、示例代码、模块组成和发送模式四个方面介绍有关消息生产者的相关知识。第8章介绍两种消费者：简单消费者和高级消费者。第9章介绍Kafka的典型应用，包括与Storm、ELK、Hadoop、Spark典型大数据系统的集成。第10章介绍了一个综合实例，描述Kafka作为数据总线在安防整体解决方案中的作用。

大数据技术丛书

Source Code Analysis and Application of Kafka

Kafka源码解析与实战

王亮◎编著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Kafka 源码解析与实战 / 王亮编著 . —北京：机械工业出版社，2017.10
（大数据技术丛书）

ISBN 978-7-111-58401-8

I. K…　II. 王…　III. 分布式操作系统　IV. TP316.4

中国版本图书馆 CIP 数据核字（2017）第 268376 号

Kafka 源码解析与实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：吴　怡
印　　刷：北京市荣盛彩色印刷有限公司　　版　　次：2018 年 1 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：17
书　　号：ISBN 978-7-111-58401-8　　定　　价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

Preface 序

近些年来，大数据技术蓬勃发展，各种围绕大数据处理的平台技术，包括组件、工具、框架越来越丰富；相关的开源工具和实践资料也越来越多，其中消息队列便是一个重要的组成部分。对于一个大型系统而言，我们通常需要围绕消息来构建整个系统的逻辑，Kafka 便是目前最主流的消息系统之一。网络上有很多关于 Kafka 使用的文章，但是始终没有一本全面的从源码和设计上展开阐述的书籍。

值得庆幸的是，本书全面解析了 Kafka 的核心架构设计和源码，是国内少有的针对 Kafka 进行系统性讲解的书籍。

作者在浙江大华技术股份有限公司工作期间，一方面参与分布式数据库平台开发，一方面参与整体的系统架构设计。由于各种不同的应用场景的需求，作者所在公司内部用过多种不同消息队列，如 Kafka、ActiveMQ、RocketMQ 等，同时也实操了大量的 Hadoop、Spark 等大数据技术和消息队列的结合应用，这些经历使得作者能比较全面地从理论和实践两个视角去看待 Kafka。阅读本书可使读者深入了解 Kafka 的设计原理和使用技巧，相信读者一定会有所收获。

许焰

大华股份，研发副总经理

前　言 *Preface*

我开始接触分布式计算的时候，正好需要利用 Spark 结合 Kafka 进行流式处理。恰巧的是 Kafka 和 Spark 底层都是利用 Scala 语言编写的，并且当时市面上有关 Kafka 的中文书籍几乎没有，因此正好利用这个机会学习了 Scala 语言，并且通读了 Kafka 和 Spark 的源码，随后把日常的积累通过博客的形式慢慢记录下来。在这一年多的积累过程中，发现有关 Kafka 的中文书籍还是很缺乏，便有了总结出书的想法，而恰在这个时候吴怡编辑通过博客联系上了我，希望我把日常的积累总结成 Kafka 的专业性书籍，分享给更广大的从事大数据相关工作的人群。

本书将从初学者的角度出发，循序渐进地讲解 Kafka 内部的实现原理，但是由于 Kafka 是基于 Scala 语言编写的，因此为了更好地阅读本书，希望读者对于 Scala 语言有大致的了解。

阅读指南

本书将从 Kafka 的内部实现原理、运维工具、客户端编程以及实际应用这四个方面出发，系统阐述有关 Kafka 的各方面知识，全书共 10 章，每章的大致内容如下。

第 1 章介绍 Kafka 诞生的背景、Kafka 在 LinkedIn 内部的应用、Kafka 的主要设计目标以及为什么使用消息系统。

第 2 章介绍 Kafka 的基本组成、拓扑结构及其内部的通信协议。

第 3 章描述 Kafka 集群组成的基本元素 Broker Server 的启动以及内部的模块组成。通过阅读这一章，读者能对 Broker Server 有整体上的印象，为之后章节的阅读打下基础。

第 4 章描述 Broker Server 内部的九大基本模块：SocketServer、KafkaRequestHandlerPool、LogManager、ReplicaManager、OffsetManager、KafkaScheduler、KafkaApis、KafkaHealthcheck 和 TopicConfigManager。

第 5 章介绍 Broker Server 的控制管理模块 KafkaController，这个模块负责整个 Kafka 集群的管理，例如：Topic 的新建和删除、分区状态和副本状态的转换、集群的负载均衡管理等。

第 6 章介绍三个维护脚本：kafka-topics.sh、kafka-reassign-partitions.sh 和 kafka-preferred-replica-election.sh，它们分别涉及 Topic 的生命周期管理、Topic 分区的重分配和分区首选副本的选择。

第 7 章从设计原则、示例代码、模块组成和发送模式四个部分介绍有关消息生产者的相关知识，从设计原则至客户端编程，从客户端编程到内部实现原理，由浅入深，循序渐进地讲解。

第 8 章分别介绍两种消费者：简单消费者和高级消费者。针对每种消费者都将依次从设计原则、消费者流程、示例代码以及原理解析四个部分介绍消费者的相关知识。

第 9 章介绍 Kafka 与典型大数据系统的集成，包括：Kafka 和 Storm 的集成、Kafka 和 ELK 的集成、Kafka 和 Hadoop 的集成以及 Kafka 和 Spark 的集成。希望通过本章使读者对 Kafka 和第三方大数据平台集成有大致的了解。

第 10 章用综合实例描述了 Kafka 的应用，案例描述 Kafka 作为数据总线在安防整体解决方案中的作用，通过车辆人脸图片数据的入库、视频数据的入库、数据延时的监控、数据质量的监控、布控统计和容灾备份 6 个业务，简要阐述内部的实现原理。

本书是基于 0.8.2 版本的 Kafka 编写的，其相关配套的源码可以从 Kafka 的官方网站上下载，下载地址为 http://kafka.apache.org/downloads，也可以从开源或者私有软件项目托管平台 GitHub 上下载，下载地址为 https://github.com/apache/kafka。为了简化代码流程描述，笔者会将一些日志打印等不影响阅读的代码用“……”代替，如果需要知道“……”代表的实际含义，可以参考源码包中的真实代码。

本书特点

由浅入深，循序渐进：本书从 LinkedIn（领英）公司内部大数据架构讲起，引出消息队列 Kafka，接着讲解 Kafka 的基本架构，然后着重分析 Kafka 内部的各模块实现细节。从诞生背景至架构组成，再到内部实现细节，由浅入深，循序渐进，让读者在阅读时能够逐步了解 Kafka。

由里到外，层层剖析：本书不仅讲解 Kafka 内部的实现原理，而且还详细描述 Kafka 外部的维护工具，对外的客户端编程原理以及和第三方集成的方式。由里到外，层层剖析，让读者在阅读时能够更加全面地掌握 Kafka。

图文并茂，生动形象：本书在讲解 Kafka 的过程中穿插了大量的图片，直观地描述了工作原理，使读者在阅读时能够加深对代码的理解。

读者对象

本书适合以下人群阅读：

- ❑ 想熟悉典型消息系统架构的大数据从业人员。
- ❑ 想了解分布式系统开发的软件工程师。
- ❑ 想掌握 Kafka 内部实现原理的中高级开发人员。
- ❑ 想搭建传统大数据框架的系统分析师。

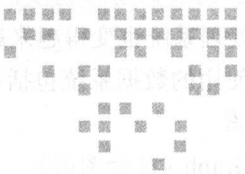
致谢

首先感谢我的夫人在我背后默默的付出，是她给了我动力，陪伴我度过了长达半年之久的枯燥时光，坚定了我完成此书的决心。其次感谢机械工业出版社吴怡编辑的鼓励和支持，是她促成了这本书的出版。接着感谢我的鱼儿们（布隆迪、金头虎、蓝茉莉、三间鼠和反游猫），每当我思绪混乱的时候可以静静地看着它们慢慢梳理。

在本书成书的过程中也得到了许多同事和同学的支持、鼓励，在此一并致谢。

由于作者水平及能力有限，加之时间仓促，本书难免存在错误和不妥之处，恳请广大读者批评指正，邮箱地址为：wangliang168219@126.com。

Contents 目　录



第 1 章 Chapter 1

Kafka 简介

Kafka 是一个高度可扩展的消息系统，它在 LinkedIn 的中央数据管道中扮演着十分重要的角色，因其可水平扩展和高吞吐率而被广泛使用，现在已被多家不同类型的公司作为多种类型的数据管道和消息系统。本章将聚集于 Kafka 诞生的背景、Kafka 在 LinkedIn 内部的应用、Kafka 的主要设计目标和为什么使用消息系统这四个方面，简单介绍典型的消息系统 Kafka，尤其在应用的时候需要多思考后两个方面：Kafka 的设计目标和应用层开发为什么需要使用消息系统。并以此为切入点，为之后的章节作铺垫。

1.1 Kafka 诞生的背景

对于一个高效的组织，所有数据需要对该组织的所有服务和系统是可用的，以便挖掘出数据的最大价值。数据采集和数据使用是一个金字塔的结构，底部为以某种统一的方式捕获数据，这些数据需要以统一的方式建模，以方便读取和处理。捕获数据的工作做扎实后，在这个基础上以不同方法处理这些数据就变得得心应手。

数据捕获的来源主要有两种：一种是记录正在发生的事件数据。比如 Web 系统中的用户活动日志（用户的点击选择等）、交警行业中的违章事件等。随着传统行业业务活动的数字化，事件数据正在不断增长，而且这个趋势没有停止。这种类型的事件数据记录了已经发生的事情，往往比传统数据库应用要大好几个数量级。因此对于数据的捕获、数据的处理提出了重大的挑战；另一种是经过二次分析处理之后的数据。对捕获的数据进行二次分析处理后得到的数据也需要记录保存，这里的处理指的是利用批处理、图分析等专有的数据处理系统进行了处理，这些加工后的数据可以作为数据捕获的第二个来源。

总之，捕获的数据越来越多，如何将这些巨量的数据以可靠的、完整的数据流方式传递给数据分析处理系统也变得越来越困难。

LinkedIn 使用的数据系统包括：

- 全文搜索
- Social Graph（社会图谱）
- Voldemort（键值存储）
- Espresso（文档存储）
- 推荐引擎
- OLAP（查询引擎）
- Hadoop
- Teradata（数据仓库）
- Ingraphs（监控图表和指标服务）

上述专用的分布式系统都需要经过数据源来获取数据，同时有些系统还会产生数据，作为其他系统的数据源。LinkedIn 曾尝试为每个数据源和目标构建自定义的数据加载，很显然这是不可行的。LinkedIn 有几十个数据系统和数据仓库。把这些系统和仓库联系起来，就会导致任意两两系统间构建自定义的管道，如图 1-1 所示。

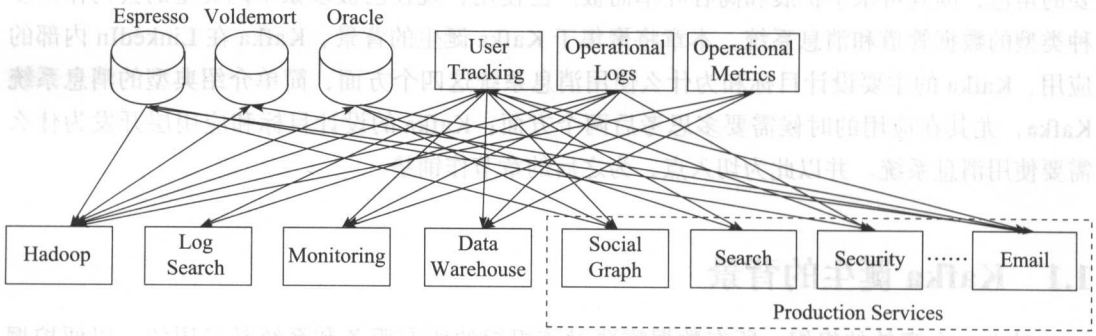


图 1-1　各系统之间的自定义管道

需要注意的是，数据是双向流动的，例如许多系统（数据库、Hadoop）同时是数据传输的来源和目的端。这就意味着我们最后要为那些系统建立两个通道：一个用于数据输入，一个用于数据输出。要避免上面的问题，我们需要如图 1-2 所示的通用方式。

我们需要尽量将每个生产者、消费者与数据源隔离。理想情况下，生产者或消费者应该只与一个数据源单独集成，这样就能访问到所有数据。根据这个思路想到增加一个新的数据系统：

- 作为数据来源或者数据目的地。
- 集成工作只需要连接这个新系统到一个单独的管道，而无须连接到每个数据的生产者和消费者。

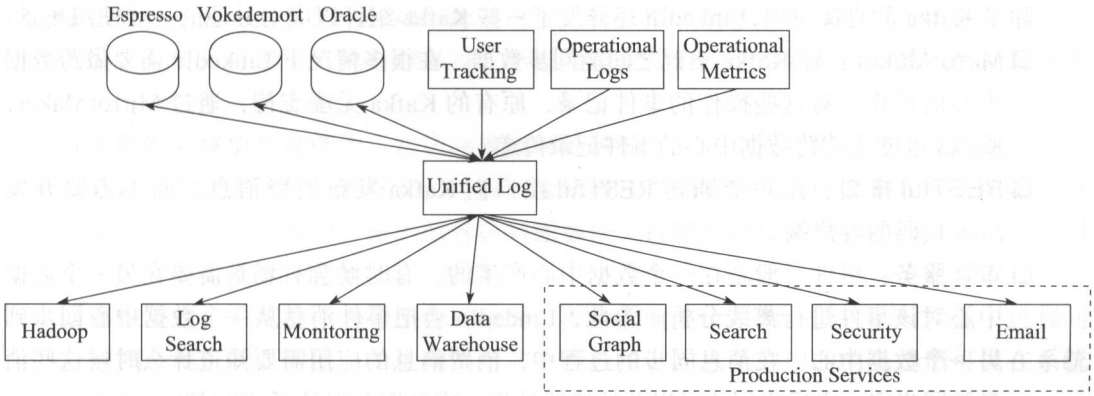


图 1-2　各系统之间的共享管道

这个新的数据系统就是 Kafka。Kafka 作为 LinkedIn 中的“中枢神经系统”，管理从各个应用程序汇聚到此的信息流，这些数据经过处理后再被分发到各处。Kafka 作为一个消息系统，进行消息的传递；同时它也是日志存储系统，以日志的形式存储了数据源的所有数据。

1.2　Kafka 在 LinkedIn 内部的应用

LinkedIn 的工程师团队已经把 Kafka 打造为管理信息流的开源解决方案。他们把 Kafka 作为消息中枢，帮助公司的各个应用以松耦合的方式在一起工作。LinkedIn 已经严重依赖于 Kafka，并且基于 Kafka 的生态系统，LinkedIn 开发出了一些开源组件和公司内部组件。

Kafka 在 LinkedIn 中的使用场景如下所述：

- **系统监控**：LinkedIn 内所有的主机都会往 Kafka 发送系统健康信息和运行信息，负责展示运维信息和报警的系统则从 Kafka 订阅获取这些运维信息，处理后进行业务的展示和告警业务的触发。更进一步，LinkedIn 通过自家的实时流处理系统 Samza 对这些运维数据进行实时的处理分析，生成实时的调用图分析。
- **传统的消息队列**：LinkedIn 内大量的应用系统把 Kafka 作为一个分布式消息队列进行使用。这些应用囊括了搜索、内容相关性反馈等应用，这些应用将处理后的数据通过 Kafka 传递到 Voldemort 分布式存储系统。
- **分析**：LinkedIn 会搜集所有的数据以更好地了解用户是如何使用 LinkedIn 的产品的。哪些网页被浏览，哪些内容被点击这样的信息都会发送到每个数据中心的 Kafka。这些数据被汇总起来并通过 Kafka 发送到 Hadoop 集群进行分析和每日报表生成。
- **作为其他分布式日志系统的组件**：Kafka 也被 LinkedIn 内其他分布式系统作为核心的日志组件，比如大数据仓储解决方案 Pinot。Kafka 也被分布式数据库 Espresso 用于负责数据的复制与修改。

除了 Kafka 的直接应用，LinkedIn 还开发了一些 Kafka 组件以应对其他的一些使用场景：

- MirrorMaker：让 Kafka 集群之间能同步数据。在很多情况下 LinkedIn 需要做跨数据中心的操作，对这些操作的事件记录，原有的 Kafka 无法支持，通过 MirrorMaker，Kafka 也能支持跨数据中心的事件记录传递。
- RESTful 接口：用户能通过 RESTful 接口向 Kafka 发布消费消息，而不需要开发 Java 代码的客户端。
- **审计服务**：事件一般是在一个数据中心产生的，有时候会有场景需要在另一个数据中心对该事件进行离线分析。为此，LindedIn 会把事件消息从一个数据中心同步到另一个数据中心。在消息同步的过程中，消费消息的应用需要知道什么时候这些消息被同步完，之后应用才可以开始离线处理。审计服务保证了这一点。

1.3 Kafka 的主要设计目标

Kafka 作为一种分布式的、基于发布 / 订阅的消息系统，其主要设计目标如下：

- 以时间复杂度为 $O(1)$ 的方式提供消息持久化能力，即使对 TB 级以上的数据也能保证常数时间的访问性能。
- 高吞吐率，即使在非常廉价的商用机器上也能做到单机支持每秒 100K 条消息的传输。
- 支持 Kafka Server 间的消息分区，及分布式消费，同时保证每个分区内的消息顺序传输。
- 支持离线数据处理和实时数据处理。
- 支持在线水平扩展。

1.4 为什么使用消息系统

回过头来看一下，我们为什么需要使用消息系统呢？其大致目的如下：

- **解耦**：在项目启动之初来预测将来项目会碰到什么需求，是极其困难的。消息系统在处理过程中间插入了一个隐含的、基于数据的接口层，两边的处理过程都要实现这一接口。这使得开发人员可以独立地扩展或修改两边的处理过程，只要确保它们遵守同样的接口约束即可。
- **冗余**：有些情况下，处理数据的过程会失败。除非数据被持久化，否则将造成丢失。消息队列把数据进行持久化直到数据完全处理完，通过这一方式规避了数据丢失的风险。许多消息队列所采用的“插入 - 获取 - 删除”范式中，在把一个消息从队列中删除之前，需要处理系统明确指出该消息已经被处理完毕，从而确保数据被安全保存直到使用完毕。

- **扩展性**：因为消息队列解耦了处理过程，所以增大消息入库和处理的频率是很容易的，只要另外增加处理过程即可。不需要改变代码，不需要调节参数，扩展就像调大电力按钮一样简单。
- **灵活性和峰值处理能力**：在访问量剧增的情况下，应用仍然需要继续发挥作用，但是这样的突发流量并不常见，并且以能处理这类峰值访问为标准来投入资源随时待命无疑是巨大的浪费。使用消息队列能够使关键组件顶住突发的访问压力，不会因为突发的超负荷的请求而完全崩溃。
- **可恢复性**：系统的一部分组件失效时，不会影响整个系统。消息队列降低了进程间的耦合度，所以即使一个处理消息的进程挂掉，加入队列中的消息仍然可以在系统恢复后被处理。
- **顺序保证**：在大多使用场景下，数据处理的顺序都很重要。大部分消息队列本来就是排序的，并且能保证数据会按照特定的顺序来处理。Kafka可保证一个分区内的消息是有序的。
- **缓冲**：在任何重要的系统中，都会需要不同的处理时间的元素。例如，加载一张图片比应用过滤器花费更少的时间。消息队列通过一个缓冲层来帮助任务最高效率地执行，写入队列的处理会尽量的快速。该缓冲有助于控制和优化数据流经过系统的速度。
- **异步通信**：很多时候，用户不想也不需要立即处理消息。消息队列提供了异步处理机制，允许用户把一个消息放入队列，但并不立即处理。想向队列中放入多少消息就放多少，然后在需要的时候再去处理。

1.5 本章小结

本章先讲述了Kafka诞生的背景及在LinkedIn公司中的应用，接着讲述了LinkedIn设计Kafka的主要目标，本质上最重要的就是两点：高吞吐量和可水平扩展。最后讲述了为什么需要使用消息系统，也就是应用层使用消息系统可以解决的问题，并且这也是本书的读者需要不断思考的问题。希望通过以上四个方面的介绍使得读者能够对Kafka形成初步的认识，同时对自己的系统进行思考。