异步图书
www.epubit.com.cn

全面讲解 Kafka 核心组件的
实现原理和基本操作
兼顾 Kafka 应用与实战

Kafka 入门与实践

牟大恩 著

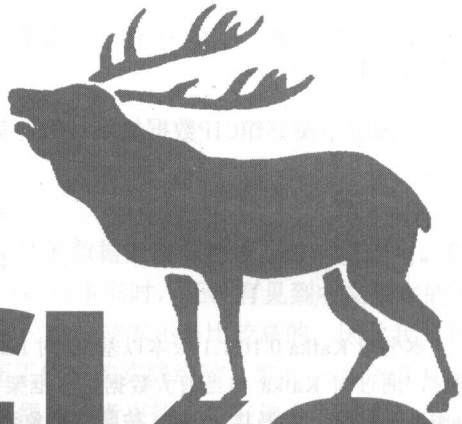
真正**零基础**入门，**深入浅出**全面剖析 Kafka
任务驱动式学习，**轻松掌握** Kafka 实战知识

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

牟大恩

武汉大学硕士，曾先后在网易杭州研究院、掌门科技、优酷土豆集团担任高级开发工程师和资深开发工程师职务，目前就职于海通证券总部。有多年的Java 开发及系统设计经验，专注于互联网金融及大数据应用相关领域。

Kafka



Kafka 入门与实践

牟大恩 著

人民邮电出版社
北京

图书在版编目（CIP）数据

Kafka入门与实践 / 牟大恩著. -- 北京 : 人民邮电出版社, 2017.11
ISBN 978-7-115-46957-1

Ⅰ. ①K… Ⅱ. ①牟… Ⅲ. ①分布式操作系统 Ⅳ. ①TP316.4

中国版本图书馆CIP数据核字(2017)第235944号

内 容 提 要

本书以 Kafka 0.10.1.1 版本以基础，对 Kafka 的基本组件的实现细节及其基本应用进行了详细介绍，同时，通过对 Kafka 与当前大数据主流框架整合应用案例的讲解，进一步展现了 Kafka 在实际业务中的作用和地位。本书共 10 章，按照从抽象到具体、从点到线再到面的学习思维模式，由浅入深，理论与实践相结合，对 Kafka 进行了分析讲解。

本书中的大量实例来源于作者在实际工作中的实践，具有现实指导意义。相信读者阅读完本书之后，能够全面掌握 Kafka 的基本实现原理及其基本操作，能够根据书中的案例举一反三，解决实际工作和学习中的问题。此外，在阅读本书时，读者可以根据本书对 Kafka 理论的分析，再结合 Kafka 源码进行定位学习，了解 Kafka 优秀的设计和思想以及更多的编码技巧。

本书适合应用 Kafka 的专业技术人员阅读，包括但不限于大数据相关应用的开发者、运维者和爱好者，也适合高等院校、培训结构相关专业的师生使用。

◆ 著　　牟大恩
责任编辑　杨海玲
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：22
字数：495 千字　　2017 年 11 月第 1 版
印数：1 – 2 400 册　　2017 年 11 月北京第 1 次印刷

定价：69.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前 言

为什么要写这本书

Kafka 由于高吞吐量、可持久化、分布式、支持流数据处理等特性而被广泛应用。但当前关于 Kafka 原理及应用的相关资料较少，在我打算编写本书时，还没有见到中文版本的 Kafka 相关书籍，对于初学者甚至是一些中高级应用者来说学习成本还是比较高的，因此我打算在对 Kafka 进行深入而系统的研究基础上，结合自己在工作中的实践经验，编写一本介绍 Kafka 原理及其基本应用的书籍，以帮助 Kafka 初、中、高级应用者更快、更好地全面掌握 Kafka 的基础理论及其基本应用，从而解决实际业务中的问题。同时，一直以来我都考虑在技术方面写点什么，将自己所学、所积累的知识沉淀下来。

通过编写本书，我最大收获有如下两点。

第一，凡事不是要尽力而为，而是要全力以赴，持之以恒。写书和阅读源码其实都是很枯燥的事，理工科出身的我，在文字表达能力上还是有所欠缺的，有些知识点可能在脑海里十分清晰，然而当用文字表述出来时，就显得有些“力不从心”了。对于纯技术的东西要用让读者阅读时感觉轻松的文字描述出来更是不易，因此看似简短的几行文字，我在编写时可能斟酌和修改了很久。我真的很钦佩那些大师们，他们写出来的东西总让人很轻松地就能够掌握，“路漫漫其修远兮，吾将上下而求索”，向大师们致敬！虽然有很多客观或主观的因素存在，但我依然没有放弃。还记得 2016 年 10 月的一天，当我决定编写本书时，我告诉妻子：“我要写一本书作为送给我们未来宝宝的见面礼！”带着这份动力我利用下班时间、周末时间，在夜深人静时默默地进行着 Kafka 相关内容的研究、学习、实战，妻子对我的鼓励、陪伴更是激励我要坚持本书的编写。带着这份动力，带着这份爱，我终于完成了本书。

第二，通过对 Kafka 源码的阅读，我除了对很多原来在实践中只知其然而不知其所以然的问题有了更深入的理解以外，还对 Kafka 优秀的设计思想及其编码技巧有所了解。

如何阅读本书

本书共 10 章，各章主要内容具体描述如下。

第 1 章对 Kafka 的基本概念进行简要介绍，方便读者对 Kafka 有一个大致的了解。

第 2 章详细介绍 Kafka 安装环境的配置及 Kafka 源码的编译，这一章为后续各章的 Kafka 原理讲解及基本操作进行准备。

第 3 章对 Kafka 基本组件的实现原理、实现细节进行了分析。如果只想了解 Kafka 的相关应用，而不关注 Kafka 的实现原理，在阅读时可以直接跳过这一章。但我觉得，如果想真正掌握 Kafka 及其实现细节，这一章是值得花时间仔细阅读的。

第 4 章对 Kafka 核心流程进行分析，主要从 Kafka 启动流程到创建一个主题、生产者发送消息、消费者消费消息的过程进行了简要介绍。这一章是 Kafka 运行机制的缩影，如果跳过了第 3 章关于组件实现原理的讲解，那么建议一定要阅读这一章，因为通过阅读这一章可以更进一步地了解 Kafka 运行时的主要角色及其职责，为后面的 Kafka 实战部分打下坚实基础。

第 5 章开始就进入了 Kafka 实战部分。这一章通过 Kafka 自带脚本演示，详细介绍了 Kafka 基本应用的操作步骤，基本覆盖了 Kafka 相关操作，因此请读者在阅读时要跟随本书所讲内容进行实战。

第 6 章对 Kafka 的 API 应用进行了详细介绍。如果读者在实践工作中不会用到调用 Kafka 的相关 API，在阅读时也可以跳过这一章。

第 7 章对 Kafka Streams 进行了介绍。Kafka Streams 是 Kafka 新增的支持流数据处理的 Java 库。如果读者不希望使用此功能，也可以跳过这一章。

第 8 章介绍 Kafka 在数据采集方面的应用，主要包括与 Log4j、Flume 和 HDFS 的整合应用。

第 9 章对 Kafka 与 ELK（Elasticsearch、Logstash 和 Kibana）整合实现日志采集平台相关应用进行介绍。

第 10 章通过两个简单的实例，介绍了 Spark 以及 Kafka 与 Spark 整合在离线计算、实时计算方面的应用。

本书的结构安排上，各章的内容相互独立，因此读者可以首先选择自己最感兴趣的章进行阅读，之后再阅读其他章。例如，读者可以先阅读第 5 章及其之后的几章，先通过实践操作对 Kafka 有一个感性的认识，然后再阅读第 3 章和第 4 章的相关原理及运行机制的内容，逐步加深对 Kafka 实现细节的理解。而第 8 章至第 10 章则是 Kafka 与当前大数据处理主流框架的整合应用，属于 Kafka 高级应用部分，可以帮助读者解决实际业务问题。

我建议读者一定要阅读第 2 章。通过第 2 章介绍的环境配置，读者能自己在本地搭建 Kafka 运行环境，阅读本书时，可跟随本书所讲解的操作进行实践。

读者对象

本书的目标读者定位是应用 Kafka 的初、中、高级开发人员及运维工程师。

从事 Kafka 应用开发的技术人员读完本书，可以学习到 Kafka 原理的分析及相关 API 应用以及结合当前主流大数据框架整合的应用，应该能够全面掌握 Kafka 的基本原理和整体结构，并为实际业务实现提供思路，从而能够更加快速地解决一些问题。

从事 Kafka 或数据运维的技术人员，读完本书详细的 Kafka 基本操作以及 Kafka 与其他大数据框架的整合应用案例，应该可以快速搭建、运维和管理 Kafka 及相应的系统平台。

从事 Kafka 相关应用的资深开发或架构人员，读完本书对 Kafka 原理的分析有助于对 Kafka

性能进行调优，可以更好地开发和设计与 Kafka 相关的应用。

对于初学者，通过阅读本书可以全面掌握 Kafka 的知识，同时可以通过 Kafka 与其他框架整合的案例来拓宽视野，为学习分布式相关知识打下基础。

在阅读本书之前，读者需要具备以下基础。

- 具有一定的 Linux 操作系统基本操作的基础知识。
- 对于分布式系统的基础有所了解，这关系到对集群的理解。
- 如果希望阅读本书第 3 章至第 7 章关于 Kafka 基本组件实现原理及编程实战的内容，需要具有 Java 或 Scala 语言基础，尤其是 Java 语言基础，这有助于阅读 Kafka 源码和调用相应的 API。

参考资料

在写作过程当中，我除阅读了 Kafka 源码之外，还从网络上阅读了大量参考资料，从中获得了很多帮助，在此对这些前辈的无私奉献精神表示由衷的钦佩和衷心的感谢。本书参考的资料如下。

- 书籍
  - 怀特．Hadoop 权威指南［M］．3 版．华东师范大学数据科学与工程学院，译．北京：清华大学出版社，2015：20-156.
  - 霍夫曼，佩雷拉．Flume 日志收集与 MapReduce 模式［M］．张龙，译．北京：机械工业出版社，2015：1-61.
  - 耿嘉安．深入理解 Spark：核心思想与源码分析［M］．北京：机械工业出版社，2016：224-282.
- 网络资源
  - Kafka 官方网站：http://kafka.apache.org/0101/documentation.html。
  - Elasticsearch 官方网站：https://www.elastic.co/guide/en/elasticsearch/reference/index.html。
  - http://orchome.com/网站上关于 Kafka 系列文章。
  - conflument 官方博客：https://www.confluent.io/blog/。
  - zqhxuyuan 的博客：http://zqhxuyuan.github.io/。
  - lizhitao 的博客：http://blog.csdn.net/lizhitao。

读者反馈

非常高兴能将这本书分享给大家，也十分感谢大家购买和阅读本书。在编写本书时，虽然我精益求精，尽了最大的努力，但由于能力有限，加之时间仓促，书中难免存在不足甚至错误，敬请读者给予指正。如果有任何问题和建议，读者可发送邮件至 moudaen@163.com。

致谢

在编写本书时得到了很多人的帮助。

首先我要感谢我的妻子，在我编写本书时你承担了所有家务，让我过着饭来张口、衣来伸手的生活，使我能够全身心投入到写作当中，这本书能够完成有你一半的功劳。也要感谢我的家人，家永远是我心灵的港湾，家人的爱永远是我奋斗的动力。同时也将本书献给我即将出生的宝宝，愿你健康成长，在未来的日子里我会给你更多的惊喜。

然后我特别要感谢人民邮电出版社的杨海玲老师，感谢你一直以来给予我的支持和鼓励，感谢你在本书编写、出版整个过程当中的辛勤付出。也要感谢人民邮电出版社所有参与本书编辑和出版的老师们，正是由于你们的辛勤付出和一丝不苟的工作态度才让本书出版成为可能。

同时要感谢我的工作单位海通证券，公司为我提供了一个非常优越的工作、学习和生活环境。在此要特别感谢部门领导和同事在我编写本书过程中提出很多宝贵的建议，我很荣幸能够与大家成为同事，共同奋斗。

最后我要感谢所有培养过我的老师们，是你们教会了我用知识改变命运，用学习成就未来。

牟大恩

2017 年 9 月于上海

目 录

第 1 章

Kafka 简介

Kafka 是一个高吞吐量、分布式的发布—订阅消息系统。据 Kafka 官方网站介绍，当前的 Kafka 已经定位为一个分布式流式处理平台（a distributed streaming platform），它最初由 LinkedIn 公司开发，后来成为 Apache 项目的一部分。Kafka 核心模块使用 Scala 语言开发，支持多语言（如 Java、C/C++、Python、Go、Erlang、Node.js 等）客户端，它以可水平扩展和具有高吞吐量等特性而被广泛使用。目前越来越多的开源分布式处理系统（如 Flume、Apache Storm、Spark、Flink 等）支持与 Kafka 集成，本书第 8 章至第 10 章将通过具体案例详细介绍 Kafka 与当前一些流行的分布式处理系统的集成应用。接下来我们将对 Kafka 相关知识做进一步深入介绍。

1.1 Kafka 背景

随着信息技术的快速发展及互联网用户规模的急剧增长，计算机所存储的信息量正呈爆炸式增长，目前数据量已进入大规模和超大规模的海量数据时代，如何高效地存储、分析、处理和挖掘海量数据已成为技术研究领域的热点和难点问题。当前出现的云存储、分布式存储系统、NoSQL 数据库及列存储等前沿技术在海量数据的驱使下，正日新月异地向前发展，采用这些技术来处理大数据成为一种发展趋势。而如何采集和运营管理、分析这些数据也是大数据处理中一个至关重要的组成环节，这就需要相应的基础设施对其提供支持。针对这个需求，当前业界已有很多开源的消息系统应运而生，本书介绍的 Kafka 就是当前流行的一款非常优秀的消息系统。

Kafka 是一款开源的、轻量级的、分布式、可分区和具有复制备份的（Replicated）、基于 ZooKeeper 协调管理的分布式流平台的功能强大的消息系统。与传统的消息系统相比，Kafka 能够很好地处理活跃的流数据，使得数据在各个子系统中高性能、低延迟地不停流转。

据 Kafka 官方网站介绍，Kafka 定位就是一个分布式流处理平台。在官方看来，作为一个流式处理平台，必须具备以下 3 个关键特性。

- 能够允许发布和订阅流数据。从这个角度来讲，平台更像一个消息队列或者企业级的

消息系统。

- 存储流数据时提供相应的容错机制。
- 当流数据到达时能够被及时处理。

Kafka 能够很好满足以上 3 个特性，通过 Kafka 能够很好地建立实时流式数据通道，由该通道可靠地获取系统或应用程序的数据，也可以通过 Kafka 方便地构建实时流数据应用来转换或是对流式数据进行响应处理。特别是在 0.10 版本之后，Kafka 推出了 Kafka Streams，这让 Kafka 对流数据处理变得更加方便。

Kafka 已发布多个版本。截止到编写本书时，Kafka 的最新版本为 0.10.1.1，因此本书内容都是基于该版本进行讲解。

1.2　Kafka 基本结构

通过前面对 Kafka 背景知识的简短介绍，我们对 Kafka 是什么有了初步的了解，本节我们将进一步介绍 Kafka 作为消息系统的基本结构。我们知道，作为一个消息系统，其基本结构中至少要有产生消息的组件（消息生产者，Producer）以及消费消息的组件（消费者，Consumer）。虽然消费者并不是必需的，但离开了消费者构建一个消息系统终究是毫无意义的。Kafka 消息系统最基本的体系结构如图 1-1 所示。

生产者负责生产消息，将消息写入 Kafka 集群；消费者从 Kafka 集群中拉取消息。至于生产者如何将生产的消息写入 Kafka，消费者如何从 Kafka 集群消费消息，Kafka 如何存储消息，Kafka 集群如何管理调度，如何进行消息负载均衡，以及各组件间如何进行通信等诸多问题，我们将在后续章节进行详细阐述，在本节我们只需对 Kafka 基本结构轮廓有个清晰认识即可。随着对 Kafka 相关知识的深入学习，我们将逐步对 Kafka 的结构图进行完善。

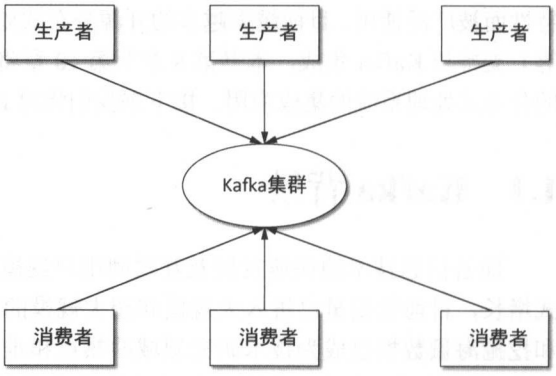


图 1-1　Kafka 消息系统最基本的体系结构

1.3　Kafka 基本概念

在对 Kafka 基本体系结构有了一定了解后，本节我们对 Kafka 的基本概念进行详细阐述。

1. 主题

Kafka 将一组消息抽象归纳为一个主题（Topic），也就是说，一个主题就是对消息的一个分类。生产者将消息发送到特定主题，消费者订阅主题或主题的某些分区进行消费。

2. 消息

消息是 Kafka 通信的基本单位，由一个固定长度的消息头和一个可变长度的消息体构成。在老版本中，每一条消息称为 Message；在由 Java 重新实现的客户端中，每一条消息称为 Record。

3. 分区和副本

Kafka 将一组消息归纳为一个主题，而每个主题又被分成一个或多个分区（Partition）。每个分区由一系列有序、不可变的消息组成，是一个有序队列。

每个分区在物理上对应为一个文件夹，分区的命名规则为主题名称后接“—”连接符，之后再接分区编号，分区编号从 0 开始，编号最大值为分区的总数减 1。每个分区又有一至多个副本（Replica），分区的副本分布在集群的不同代理上，以提高可用性。从存储角度上分析，分区的每个副本在逻辑上抽象为一个日志（Log）对象，即分区的副本与日志对象是一一对应的。每个主题对应的分区数可以在 Kafka 启动时所加载的配置文件中配置，也可以在创建主题时指定。当然，客户端还可以在主题创建后修改主题的分区数。

分区使得 Kafka 在并发处理上变得更加容易，理论上来说，分区数越多吞吐量越高，但这要根据集群实际环境及业务场景而定。同时，分区也是 Kafka 保证消息被顺序消费以及对消息进行负载均衡的基础。

Kafka 只能保证一个分区之内消息的有序性，并不能保证跨分区消息的有序性。每条消息被追加到相应的分区中，是顺序写磁盘，因此效率非常高，这是 Kafka 高吞吐率的一个重要保证。同时与传统消息系统不同的是，Kafka 并不会立即删除已被消费的消息，由于磁盘的限制消息也不会一直被存储（事实上这也是没有必要的），因此 Kafka 提供两种删除老数据的策略，一是基于消息已存储的时间长度，二是基于分区的大小。这两种策略都能通过配置文件进行配置，在这里不展开探讨，在 3.5.4 节将详细介绍。

4. Leader 副本和 Follower 副本

由于 Kafka 副本的存在，就需要保证一个分区的多个副本之间数据的一致性，Kafka 会选择该分区的一个副本作为 Leader 副本，而该分区其他副本即为 Follower 副本，只有 Leader 副本才负责处理客户端读/写请求，Follower 副本从 Leader 副本同步数据。如果没有 Leader 副本，那就需要所有的副本都同时负责读/写请求处理，同时还得保证这些副本之间数据的一致性，假设有 n 个副本则需要有 $n \times n$ 条通路来同步数据，这样数据的一致性和有序性就很难保证。

引入 Leader 副本后客户端只需与 Leader 副本进行交互，这样数据一致性及顺序性就有了保证。Follower 副本从 Leader 副本同步消息，对于 n 个副本只需 $n-1$ 条通路即可，这样就使得系统更加简单而高效。副本 Follower 与 Leader 的角色并不是固定不变的，如果 Leader 失效，通过相应的选举算法将从其他 Follower 副本中选出新的 Leader 副本。

5. 偏移量

任何发布到分区的消息会被直接追加到日志文件（分区目录下以“.log”为文件名后缀的数据文件）的尾部，而每条消息在日志文件中的位置都会对应一个按序递增的偏移量。偏移量

是一个分区下严格有序的逻辑值，它并不表示消息在磁盘上的物理位置。由于Kafka几乎不允许对消息进行随机读写，因此Kafka并没有提供额外索引机制到存储偏移量，也就是说并不会给偏移量再提供索引。消费者可以通过控制消息偏移量来对消息进行消费，如消费者可以指定消费的起始偏移量。为了保证消息被顺序消费，消费者已消费的消息对应的偏移量也需要保存。需要说明的是，消费者对消息偏移量的操作并不会影响消息本身的偏移量。旧版消费者将消费偏移量保存到ZooKeeper当中，而新版消费者是将消费偏移量保存到Kafka内部一个主题当中。当然，消费者也可以自己在外部系统保存消费偏移量，而无需保存到Kafka中。

6. 日志段

一个日志又被划分为多个日志段（LogSegment），日志段是Kafka日志对象分片的最小单位。与日志对象一样，日志段也是一个逻辑概念，一个日志段对应磁盘上一个具体日志文件和两个索引文件。日志文件是以“.log”为文件名后缀的数据文件，用于保存消息实际数据。两个索引文件分别以“.index”和“.timeindex”作为文件名后缀，分别表示消息偏移量索引文件和消息时间戳索引文件。

7. 代理

在Kafka基本体系结构中我们提到了Kafka集群。Kafka集群就是由一个或多个Kafka实例构成，我们将每一个Kafka实例称为代理（Broker），通常也称代理为Kafka服务器（KafkaServer）。在生产环境中Kafka集群一般包括一台或多台服务器，我们可以在一台服务器上配置一个或多个代理。每一个代理都有唯一的标识id，这个id是一个非负整数。在一个Kafka集群中，每增加一个代理就需要为这个代理配置一个与该集群中其他代理不同的id，id值可以选择任意非负整数即可，只要保证它在整个Kafka集群中唯一，这个id就是代理的名字，也就是在启动代理时配置的broker.id对应的值，因此在本书中有时我们也称为brokerId。由于给每个代理分配了不同的brokerId，这样对代理进行迁移就变得更方便，从而对消费者来说是透明的，不会影响消费者对消息的消费。代理有很多个参数配置，由于在本节只是对其概念进行阐述，因此不做深入展开，对于代理相关配置将穿插在本书具体组件实现原理、流程分析及相关实战操作章节进行介绍。

8. 生产者

生产者（Producer）负责将消息发送给代理，也就是向Kafka代理发送消息的客户端。

9. 消费者和消费组

消费者（Comsumer）以拉取（pull）方式拉取数据，它是消费的客户端。在Kafka中每一个消费者都属于一个特定消费组（ConsumerGroup），我们可以为每个消费者指定一个消费组，以groupId代表消费组名称，通过group.id配置设置。如果不指定消费组，则该消费者属于默认消费组test-consumer-group。同时，每个消费者也有一个全局唯一的id，通过配置项client.id指定，如果客户端没有指定消费者的id，Kafka会自动为该消费者生成一个全局唯一的id，格式为${groupId}-${hostName}-${timestamp}-${UUID前8位字符}。同一个主题的一条消息只能