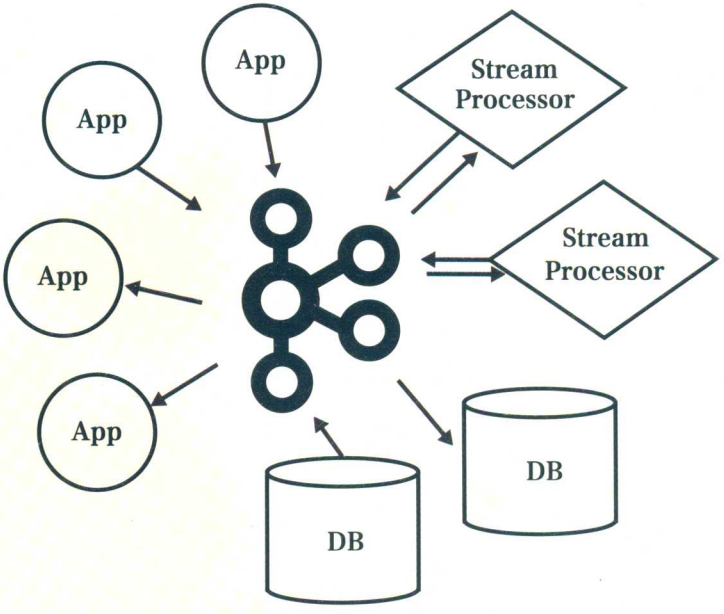
Broadview®
www.broadview.com.cn

从架构设计到实现细节，从应用场景到核心概念，
详细分析Kafka生产者、消费者、Server端源码

Apache Kafka
源码剖析

徐郡明 编著

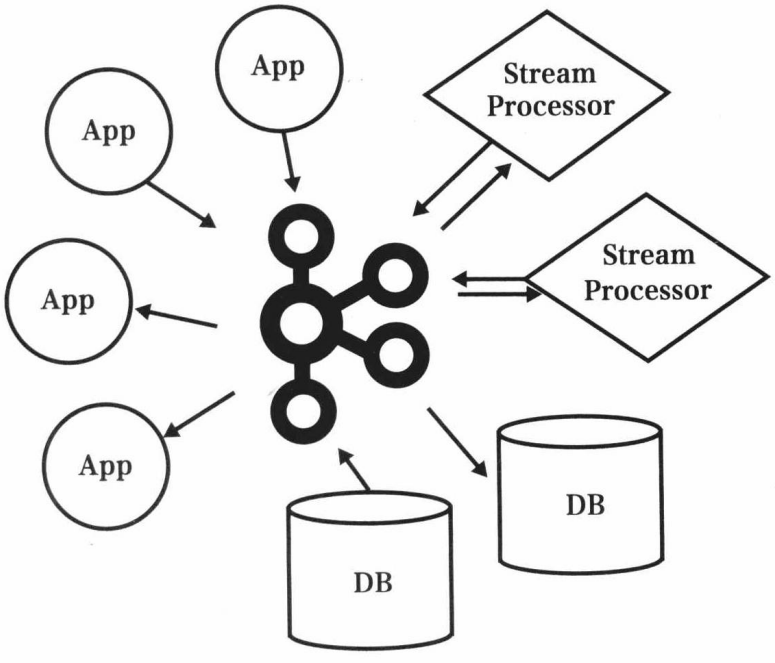


中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Broadview®
www.broadview.com.cn

Apache Kafka
源码剖析

徐郡明 编著



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内容简介

本书以Kafka 0.10.0版本源码为基础，针对Kafka的架构设计到实现细节进行详细阐述。本书共5章，从Kafka的应用场景、源码环境搭建开始逐步深入，不仅介绍Kafka的核心概念，而且对Kafka生产者、消费者、服务端的源码进行深入的剖析，最后介绍Kafka常用的管理脚本实现，让读者不仅从宏观设计上了解Kafka，而且能够深入到Kafka的细节设计之中。在源码分析的过程中，还穿插了笔者工作积累的经验和对Kafka设计的理解，希望读者可以举一反三，不仅知其然，而且知其所以然。

本书旨在为读者阅读Kafka源码提供帮助和指导，让读者更加深入地了解Kafka的运行原理、设计理念，让读者在设计分布式系统时可以参考Kafka的优秀设计。本书的内容对于读者全面提升自己的技术能力有很大帮助。

图书在版编目（CIP）数据

Apache Kafka源码剖析 / 徐郡明编著. —北京：电子工业出版社，2017.5

ISBN 978-7-121-31345-5

Ⅰ. ①A… Ⅱ. ①徐… Ⅲ. ①分布式操作系统—研究 Ⅳ. ①TP316.4

中国版本图书馆CIP数据核字（2017）第076301号

责任编辑：陈晓猛
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×980　1/16　印张：37.75　字数：720千字
版　　次：2017年5月第1版
印　　次：2017年5月第1次印刷
定　　价：89.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
本书咨询联系方：010-51260888-819，faq@phei.com.cn。

前言

这是一个数据大爆炸的时代，互联网成为了数据传播的主要载体。大数据处理平台在现代化的互联网公司进行商业决策、规划发展、市场拓展等方面扮演着越来越重要的角色。Kafka 作为大数据平台的重要组件之一，受到越来越多的设计人员和开发人员的青睐，Kafka 的社区也变得越来越活跃，Kafka 本身的架构设计、应用场景也得到了长足的发展。

Kafka 最开始由 LinkedIn 设计开发，并于 2011 年年初开源，2012 年 10 月成为 Apache 基金会的顶级项目。目前 Kafka 为越来越多的分布式大数据处理系统提供支持，其中也包括著名的 Apache Spark，LinkedIn、Netflix、Uber、Verizon、网易、美团等互联网公司也选择以 Kafka 为基础搭建其大数据处理平台或消息中间件系统。随着 Kafka 的应用场景越来越丰富，用户对 Kafka 的吞吐量、可扩展性、稳定性和可维护性等有了更多的期许，也有很多开发人员参与到 Kafka 的开发建议制定和代码提交中。在 Kafka 0.10.X 版本中出现了很多令人欣喜的新功能，本书深入剖析了 Kafka 0.10.X 的内部设计和实现细节。

本书以 Kafka 0.10.0 版本源码为基础，深入剖析了 Kafka 的各个模块的实现，包括 Kafka 的生产者客户端、消费者客户端、服务端的各个模块以及常用的管理脚本。笔者对 Kafka 设计的理解和经验分享也穿插在了剖析源码的过程中，希望读者能够通过本书理解 Kafka 的设计原理和源码实现，同时也学习到 Kafka 中优秀的设计思想以及 Java 和 Scala 的编程技巧和规范。

如何阅读本书

由于本书的篇幅限制，本书并没有详细介绍 Kafka 源码中涉及的所有基础知识，例如 Java NIO、J.U.C 包中工具类的使用、命令行参数解析器的使用等，为方便读者阅读，笔者仅介绍了一些必须且重要的基础知识。在开始源码分析之前，希望读者按照第 1 章的相关介绍完成 Kafka 源码环境的搭建，并了解 Kafka 的核心概念，这样也可以有更好的学习效果。

本书共五章，它们互相之间的联系并不是很强，读者可以从头开始阅读，也可以选择自己感兴趣的章节进行学习。

第 1 章是 Kafka 的快速入门，其中介绍了 Kafka 的背景、特性以及应用场景。之后介绍了笔者在实践中遇到的一个以 Kafka 为中心的案例，并分析了在此案例中选择使用 Kafka 的具体原因和 Kafka 起到的关键作用。最后介绍了 Kafka 中的核心概念和 Kafka 源码调试环境的搭建。

第 2 章介绍了生产者客户端的设计特点和实现细节，剖析了 KafkaProducer 拦截消息、序列化消息、路由消息等功能的源码实现，介绍了 RecordAccumulator 的结构和实现。最后剖析了 KafkaProducer 中 Sender 线程的源码。

第 3 章介绍了 Kafka 的消息传递保证语义并给出了相关的实践建议，还介绍了 Consumer Group Rebalance 操作各个版本方案的原理和弊端。最后详细剖析了 KafkaConsumer 相关组件的运行原理和实现细节。

第 4 章介绍了构成 Kafka 服务端的各个组件，依次分析了 Kafka 网络层、API 层、日志存储、DelayedOperationPurgatory 组件、Kafka 的副本机制、KafkaController、GroupCoordinator、Kafka 的身份认证与权限控制以及 Kafka 监控相关的实现。本章是 Kafka 的核心内容，涉及较多的设计细节和编程技巧，希望读者阅读之后有所收获。

第 5 章介绍了 Kafka 提供的多个脚本工具的使用以及具体实现原理，了解这些脚本可以帮助管理人员快速完成一些常见的管理、运维、测试功能。

如果读者在阅读本书的过程中，发现任何不妥之处，请将您宝贵的意见和建议发送到邮箱 xxxlxy2008@163.com，也欢迎读者朋友通过此邮箱与笔者进行交流。

致谢

感谢电子工业出版社博文视点的陈晓猛老师，是您的辛勤工作让本书的出版成为可能。同时还要感谢许多我不知道名字的幕后工作人员为本书付出的努力。

感谢张占龙、张亚森、杨威、刘克刚、刘思等朋友在百忙之中抽出时间对本书进行审阅和推荐。感谢林放、米秀明、星亮亮、王松洋、褚洪洋、曾天宁、葛彬、赵美凯、顾聪慧、孙向川、段鑫冬、彭海蛟、赵仁伟等同事，帮助我解决工作中的困难。

感谢冯玉玉、李成伟，是你们让写作的过程变得妙趣横生，是你们让我更加积极、自信，也是你们的鼓励让我完成了本书的写作。

最后，特别感谢我的母亲大人，谢谢您默默为我做出的牺牲和付出，您是我永远的女神。

徐郡明

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务：

- **提交勘误**：您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流**：在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/31345

二维码：

专家推荐

《Apache Kafka 源码剖析》一书深入浅出地分析了 Kafka 的源代码，无论是刚接触 Kafka 的菜鸟，还是已经有多年 Kafka 使用经验的老鸟，这本书都能让你有所收获。

——搜狗高级研发工程师　张亚森

Kafka 是大数据平台中的关键部分之一。《Apache Kafka 源码剖析》全面细致地剖析了 Kafka 的运行原理和架构设计，在带领读者进入 Kafka 源码世界的同时，也分析了许多设计经验，是一本不可多得的好书。

——华为高级研发工程师　张占龙

在阅读《Apache Kafka 源码剖析》时，作者在每一章节中都会给我意外之惊喜。作者对 Kafka 源代码已有相当深刻的理解，此书代码分析过程逻辑清晰，详略得当，实属不易。

——网易游戏高级数据挖掘研究员　杨威

大型分布式系统犹如一个生命，系统中各个服务犹如骨骼，其中的数据犹如血液，而 Kafka 犹如经络，串联整个系统。《Apache Kafka 源码剖析》通过大量的设计图展示、代码分析、示例分享，把 Kafka 的实现脉络展示在读者面前，帮助读者更好地研读 Kafka 代码。

——今日头条高级研发工程师　刘克刚

《Apache Kafka 源码剖析》中汇集了作者多年 Kafka 开发经验，为读者深入学习 Kafka 实现指明了方向。对于想学习 Kafka 的程序员来说，这是一本非常不错的进阶书籍。

——美团高级研发工程师　刘思

目录

第1章
快速入门

1.1 Kafka 简介

Apache Kafka 是一种分布式的、基于发布 / 订阅的消息系统，由 Scala 语言编写而成。它具备快速、可扩展、可持久化的特点。Kafka 最初由 LinkedIn 开发，并于 2011 年初开源，2012 年 10 月从 Apache 孵化器毕业，成为 Apache 基金会的顶级项目。目前，越来越多的开源分布式处理系统支持与 Kafka 集成，例如：Apache Storm、Spark。也有越来越多的公司在 Kafka 的基础上建立了近乎实时的信息处理平台，例如：LinkedIn、Netflix、Uber 和 Verizon。在国内也有很多互联网公司在生产环境中使用 Kafka 作为其消息中间件。

Kafka 之所以受到越来越多开发人员的青睐，主要与其关键特性相关。

- Kafka 具有近乎实时性的消息处理能力，即使面对海量消息也能够高效地存储消息和查询消息。Kafka 将消息保存在磁盘中，在其设计理念中并不惧怕磁盘操作，它以顺序读写的方式访问磁盘，从而避免了随机读写磁盘导致的性能瓶颈。
- Kafka 支持批量读写消息，并且会对消息进行批量压缩，这样既提高了网络的利用率，也提高了压缩效率。
- Kafka 支持消息分区，每个分区中的消息保证顺序传输，而分区之间则可以并发操作，这样就提高了 Kafka 的并发能力。
- Kafka 也支持在线增加分区，支持在线水平扩展。
- Kafka 支持为每个分区创建多个副本，其中只会有一个 Leader 副本负责读写，其

他副本只负责与 Leader 副本进行同步，这种方式提高了数据的容灾能力。Kafka 会将 Leader 副本均匀地分布在集群中的服务器上，实现性能最大化。

随着 Kafka 在各大公司的实践应用，Kafka 的应用场景也变得越来越丰富。

- 在应用系统中可以将 Kafka 作为传统的消息中间件，实现消息队列和消息的发布 / 订阅，在某些场景下，性能会超越 RabbitMQ、ActiveMQ 等传统的消息中间件。
- Kafka 也被用作系统中的数据总线，将其接入多个子系统中，子系统会将产生的数据发送到 Kafka 中保存，之后流转到目的系统中。
- Kafka 还可以用作日志收集中心，多个系统产生的日志统一收集到 Kafka 中，然后由数据分析平台进行统一处理。日志会被 Kafka 持久化到磁盘，所以同时支持离线数据处理和实时数据处理。
- 现在也有开发人员基于 Kafka 设计数据库主从同步的工具。

1.2　以 Kafka 为中心的解决方案

在大数据、高并发的系统中，为了突破瓶颈，会将系统进行水平扩展和垂直拆分，形成独立的服务。每个独立的服务背后，可能是一个集群在对外提供服务。这就会碰到一个问题，整个系统是由多个服务（子系统）组成的，数据需要在各个服务中不停流转。如果数据在各个子系统中传输时，速度过慢，就会形成瓶颈，降低整个系统的性能。

下面介绍的场景是笔者工作中遇到的一个案例，在一个政企信息化的云平台网站上，用户与网站交互的很多操作行为（例如，浏览某些新闻等）都会被记录下来，等待后台的多个子系统进行消费，其中比较重要的几个子系统是利用这些数据进行机器学习或是数据挖掘，产生用户的侧写。这样，网站就可以根据用户的侧写，推送给他们需要的配置和查询信息。图 1-1 就是这个云平台系统的架构图，其中每一个箭头都表示一条数据流。

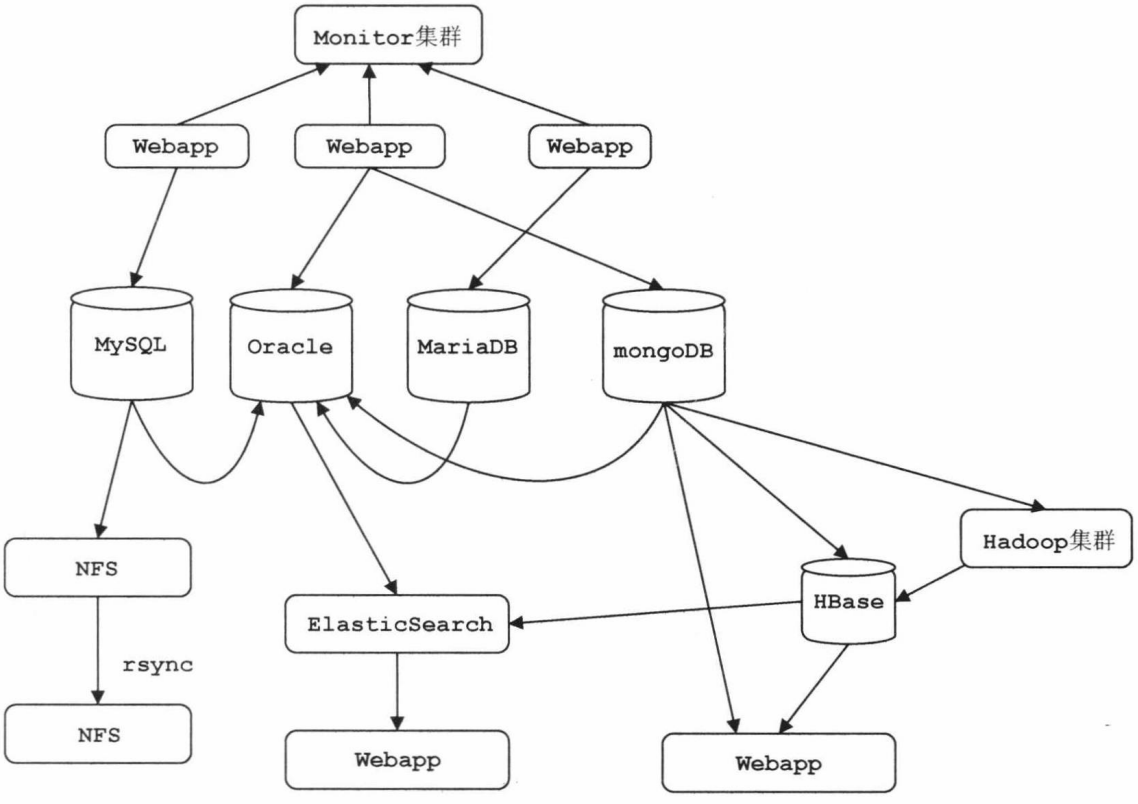


图 1-1

上面的架构中涉及的子系统、存储、服务种类繁多，而且它们之间都存在较强的耦合，会出现下面的问题：

- 由于子系统之间存在的耦合性，两个存储之间要进行数据交换的话，开发人员就必须了解这两个存储系统的 API，不仅是开发成本，就连维护成本也会很高。一旦其中一个子系统发生变化，就可能影响其他多个子系统，这简直就是一场灾难。
- 在某些应用场景中，数据的顺序性尤为重要，一旦数据出现乱序，就会影响最终的计算结果，降低用户体验，这就提高了开发的难度。
- 除了考虑数据顺序性的要求，还要考虑数据重传等提高可靠性的机制，毕竟通过网络进行传输并不可靠，可能出现丢失数据的情况。
- 进行数据交换的两个子系统，无论哪一方宕机，重新上线之后，都应该恢复到之前的传输位置，继续传输。尤其是对于非幂等性的操作，恢复到错误的传输位置，就会导致错误的结果。
- 随着业务量的增长，系统之间交换的数据量会不断地增长，水平可扩展的数据传

输方式就显得尤为重要。

针对这个案例，我们看看 Kafka 是如何有效地解决上面的这些问题的（Kafka 中的相关概念可以参见下文相关内容）。

- 解耦合

将 Kafka 作为整个系统的中枢，负责在任意两个系统之间传递数据。架构如图 1-2 所示，所有的存储只与 Kafka 通信，开发人员不需要再去了解各个子系统、服务、存储的相关接口，只需要面向 Kafka 编程即可。这样，在需要进行数据交换的子系统之间形成了一个基于数据的接口层，只有这两者知道消息存放的 Topic、消息中数据的格式。当需要扩展消息格式时，只需要修改相关子系统的 Kafka 客户端即可。这样，与 Kafka 通信的模块就可以实现复用，Kafka 则承担数据总线的作用。更简单点说，就像是一个生产者—消费者模式，而 Kafka 则扮演其中“队列”的角色。

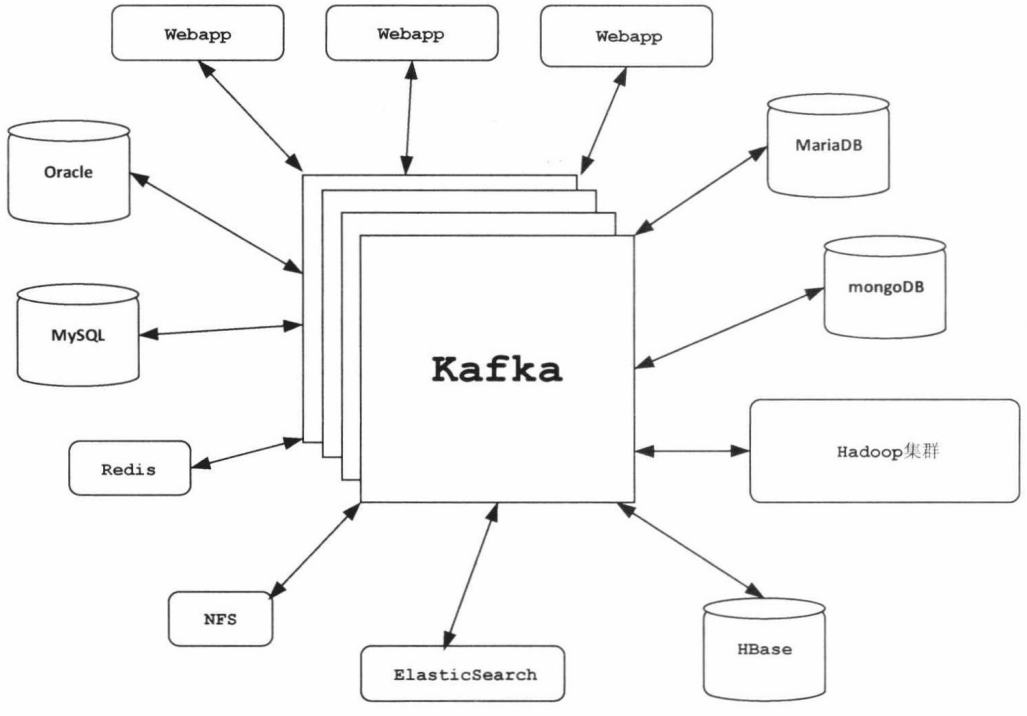


图 1-2

- 数据持久化

在分布式系统中，各个组件是通过网路连接起来的。一般认为网络传输是不可靠的，当数据在两个组件之间进行传递的时候，传输过程可能会失败。除非数据被持久化到磁盘，

否则就可能造成消息的丢失。Kafka 把数据以消息的形式持久化到磁盘，即使 Kafka 出现宕机，也可以保证数据不会丢失，通过这一方式规避了数据丢失风险。为了避免磁盘上的数据不断增长，Kafka 提供了日志清理、日志压缩等功能，对过时的、已经处理完成的数据进行清除。在磁盘操作中，耗时最长的就是寻道时间，这是导致磁盘的随机 I/O 性能很差的主要原因。为了提高消息持久化的性能，Kafka 采用顺序读写的方式访问，实现了高吞吐量。

- 扩展与容灾

Kafka 的每个 Topic（主题）都可以分为多个 Partition（分区），每个分区都有多个 Replica（副本），实现消息冗余备份。每个分区中的消息是不同的，这类似于数据库中水平切分的思想，提高了并发读写的能力。而同一分区的不同副本中保存的是相同的消息，副本之间是一主多从的关系，其中 Leader 副本负责处理读写请求，Follower 副本则只与 Leader 副本进行消息同步，当 Leader 副本出现故障时，则从 Follower 副本中重新选举 Leader 副本对外提供服务。这样，通过提高分区的数量，就可以实现水平扩展；通过提高副本的数量，就可以提高容灾能力。

Kafka 的容灾能力不仅体现在服务端，在 Consumer 端也有相关设计。Consumer 使用 pull 方式从服务端拉取消息，并且在 Consumer 端保存消费的具体位置，当消费者宕机后恢复上线，可以根据自己保存的消费位置重新拉取需要的消息进行消费，这就不会造成消息丢失。也就是说，Kafka 不决定何时、如何消费消息，而是 Consumer 自己决定何时、如何消费消息。

Kafka 还支持 Consumer 的水平扩展能力。我们可以让多个 Consumer 加入一个 Consumer Group（消费组），在一个 Consumer Group 中，每个分区只能分配给一个 Consumer 消费，当 Kafka 服务端通过增加分区数量进行水平扩展后，我们可以向 Consumer Group 中增加新的 Consumer 来提高整个 Consumer Group 的消费能力。当 Consumer Group 中的一个 Consumer 出现故障下线时，会通过 Rebalance 操作将下线 Consumer 负责处理的分区分配给其他 Consumer 继续处理；当下线 Consumer 重新上线加入 Consumer Group 时，会再进行一次 Rebalance 操作，重新分配分区。当然，一个 COnsumer Group 可以订阅很多不同的 Topic，每个 Consumer 可以同时处理多个分区。

- 顺序保证

在很多场景下，数据处理的顺序都很重要，不同的顺序就可能导致不同的计算结果。Kafka 保证一个 Partition 内消息的有序性，但是并不保证多个 partition 之间的数据有顺序。

- 缓冲 & 峰值处理能力