

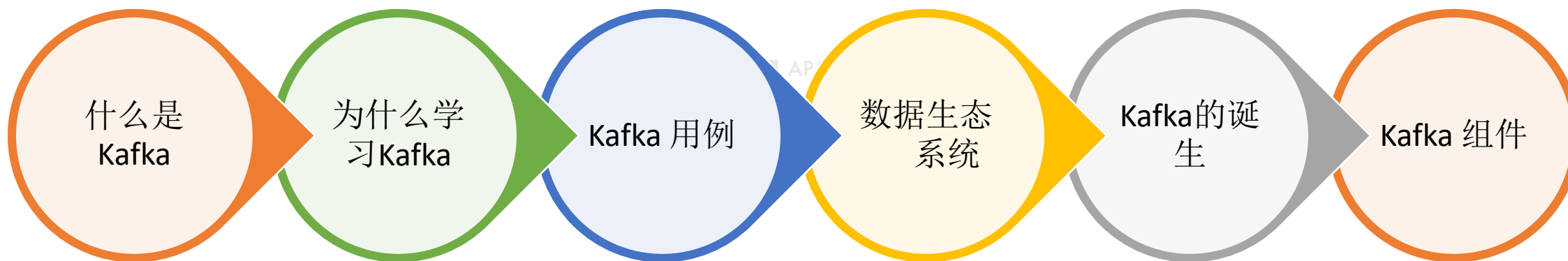
# 利用Kafka实现分布式消息传递系统

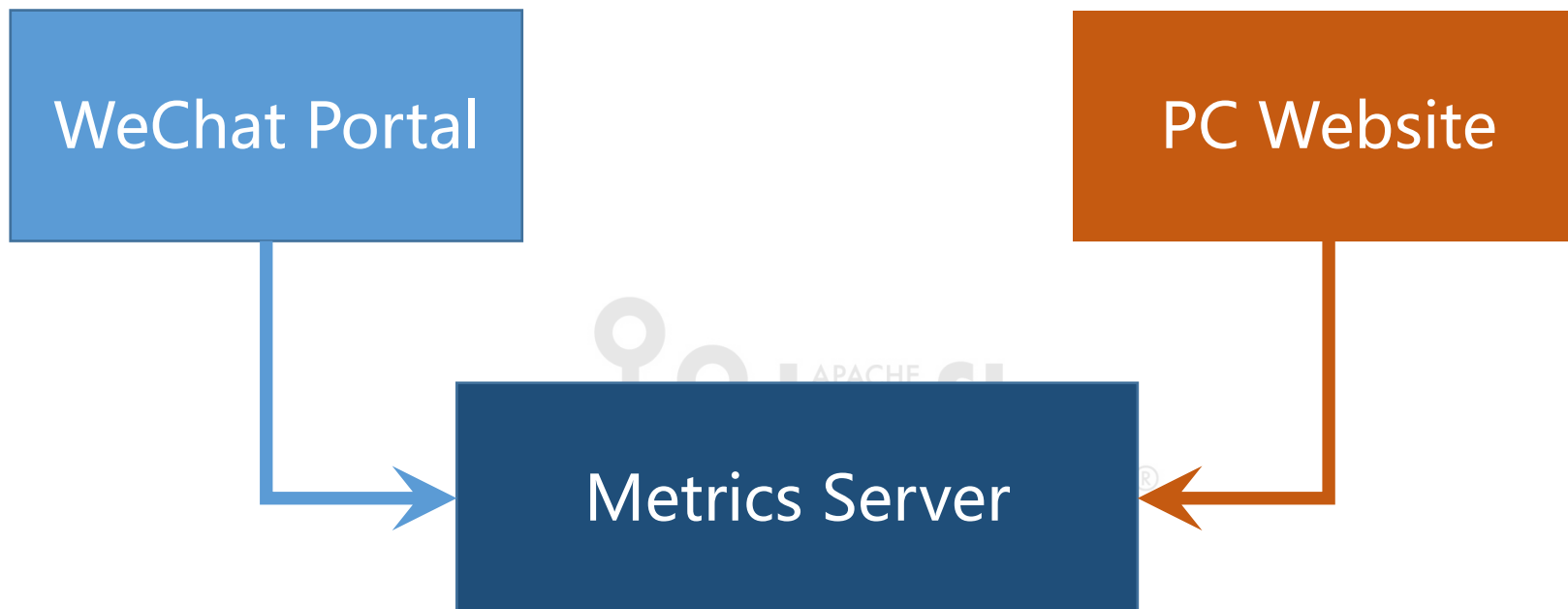
---

## 第一章：Apache Kafka入门



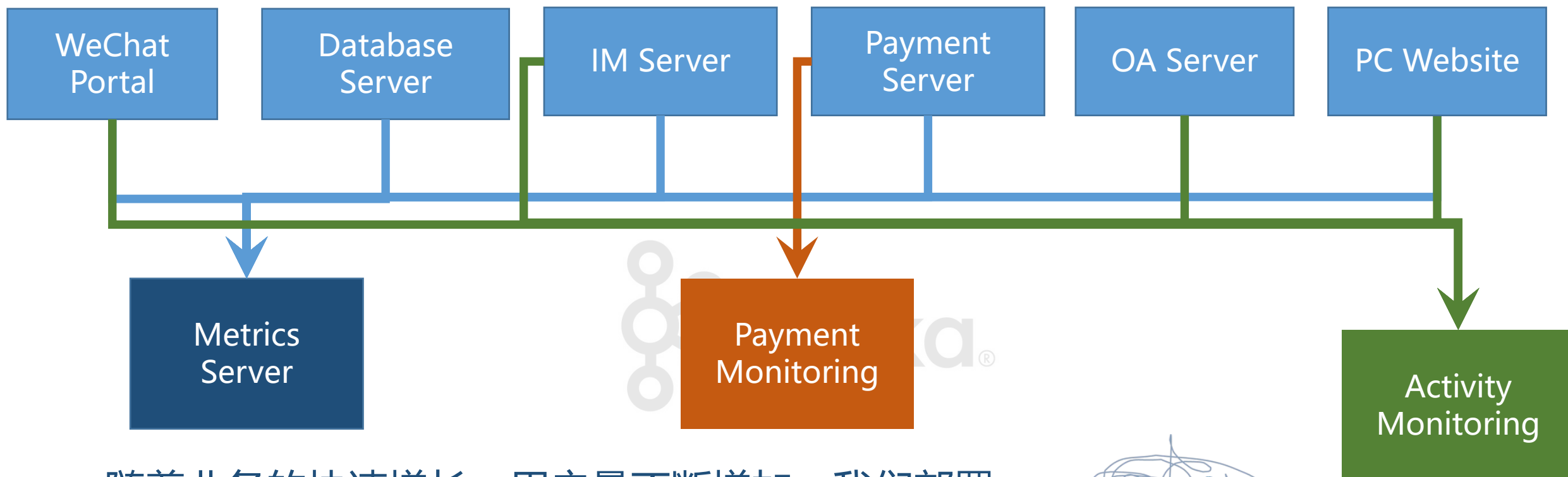
本章你将学到:





最初业务规模并不大，我们的线上业务就只有WeChat公众号和PC网站，我们要收集的指标数据来源有限。

# 一个消息处理从简单到复杂的例子



随着业务的快速增长，用户量不断增加，我们部署了很多其他的业务系统，需要监控的指标和事件越来越多，于是有了上图所示的消息流。

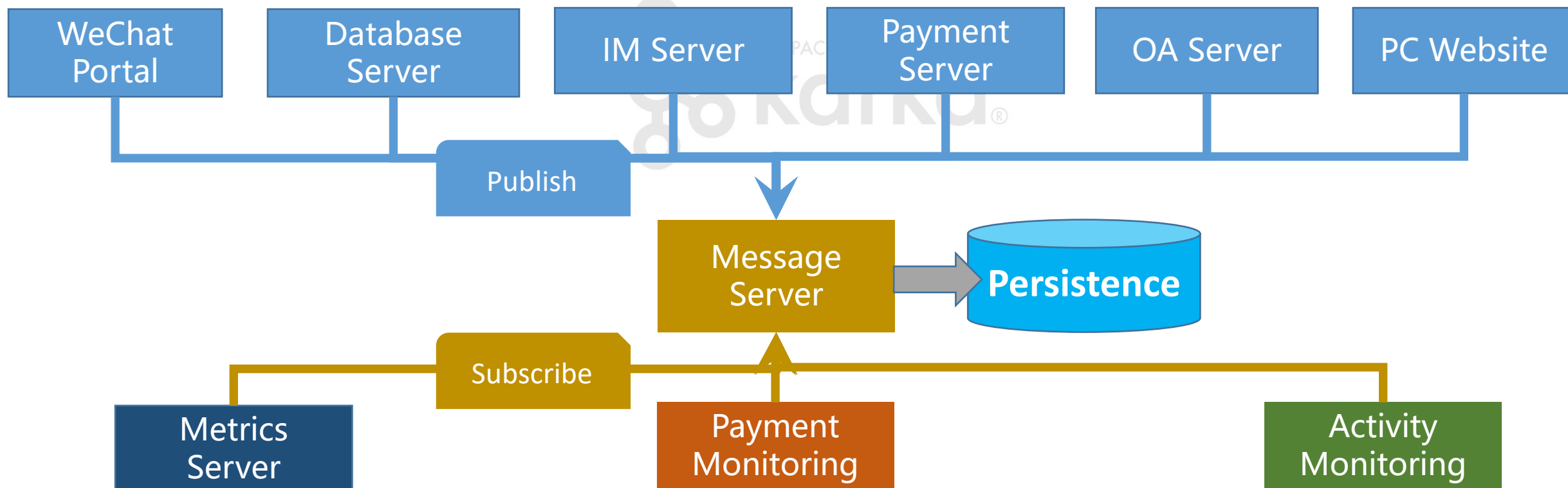
预计未来两年，要支持业务的增长，我们还要再部署2倍以上规模的系统才行，并监控运行指标。



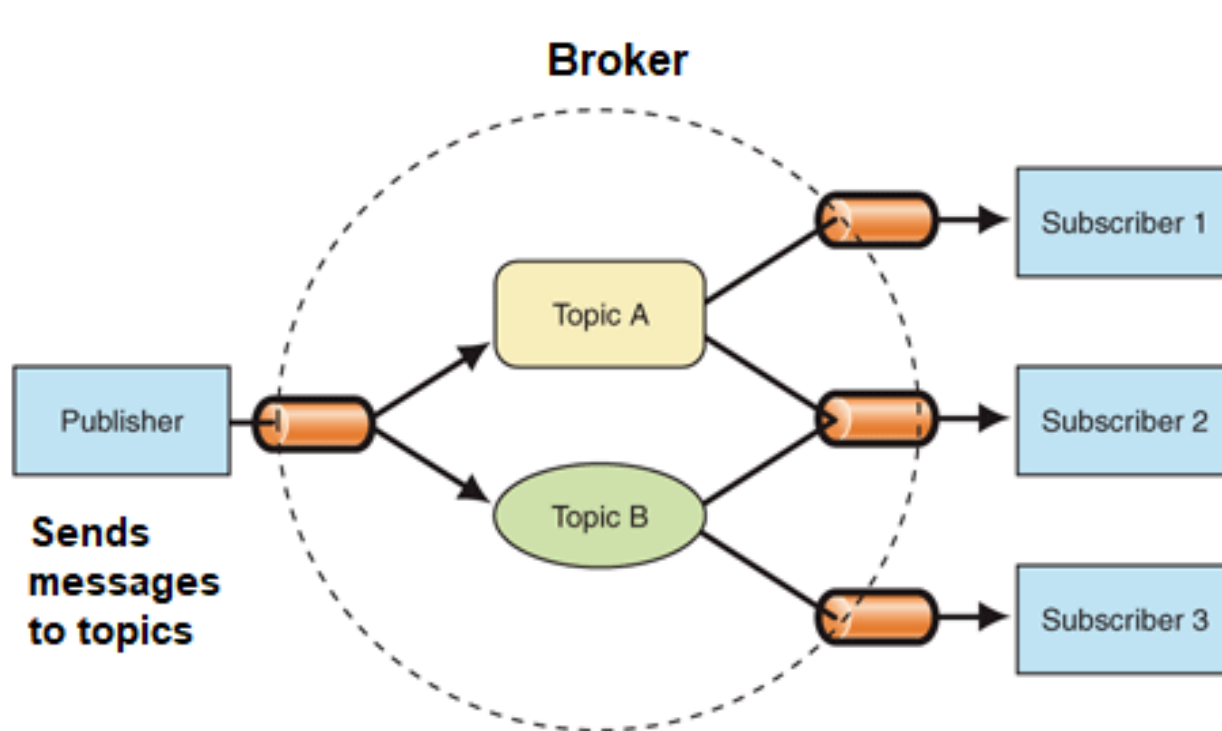
# 一个消息处理从简单到复杂的例子



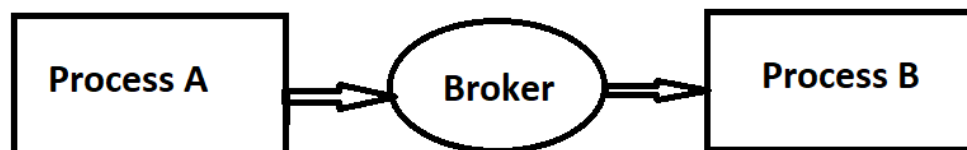
Obviously, if we add new application and monitor it, we need to develop and deploy duplicated components in every application, sending and retrieving the messages by application itself, the complexity is increased, not very easy to maintain and do troubleshooting. The key potential issue is that if monitor application is down because of software/hardware failure, the metrics data is lost.



- 发布/订阅是消息传递的一种模式，其特征是发送者（发布者）发布数据（消息），没有专门将其定向到某个接收方，消息发送端与消息接收端是异步的。
- Apache Kafka 是一个分布式发布-订阅消息系统，它接收来自不同源系统的数据，并实时将数据提供给目标系统。
- Kafka 是用Scala和Java编写的，通常与大数据的实时事件流处理相关。
- Kafka 由以下几个部分组成
  - Topic（主题）
  - Producer（生产者）
  - Consumer（消费者）
  - Partition（分区）
  - Kafka broker（Kafka 代理），核心组件
  - Consumer Groups（消费者组）



- 传统的消息代理系统（如符合 JMS 或 AMQP 标准的系统）往往具有直接连接到代理的流程，以及直接连接到流程的代理。JMS is short for Java Message Service, AMQP is short for Advanced Message Queuing Protocol.



- Apache Kafka 实现了一种通常被称为分布式提交日志（distributed commit log）技术，它的作用与传统的代理消息系统类似。从生产者的角度来看，它将事件记录到通道中，而 Kafka 在消费者处理它们时保留这些消息。
- Kafka 针对不同的用例进行了优化。然而，它们并没有集中在灵活性和交付保证上，而是倾向于关注可伸缩性和吞吐量。这样的设计为大数据的处理带来了先天的优势，随着数据量的变化，可以动态扩容/扩容，以满足对吞吐量的要求。



- 成为传统消息代理的好选择的原因

- **多个生产者**

- Kafka 能够无缝地处理多个生产者，无论这些客户使用的是多个主题还是同一个主题。

- **多个消费者**

- Kafka是为多个用户设计的，可以在不相互干扰的情况下读取任何单一的消息流。

- **基于磁盘的Retention (持久化)**

- 消息将提交到磁盘，并将使用可配置的保留规则进行存储：按时间或者容量配额。

- 可以根据每个主题选择这些选项，从而允许不同的消息流具有不同的保留量，具体取决于消费者的需要。

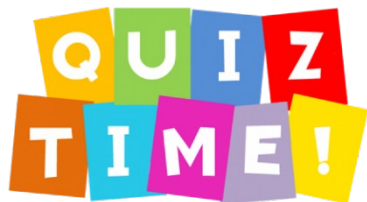
- **可伸缩**

- Kafka 灵活的可伸缩性使处理任何数量的数据变得容易。用户可以从一个代理开始作为概念证明，扩展到由三个代理组成的小型开发集群，然后随着数据规模的扩大，由数十个甚至数百个代理组成的更大集群投入生产。当不再需要大集群时，可以动态缩容，不影响正在处理的消息。

- **高性能**

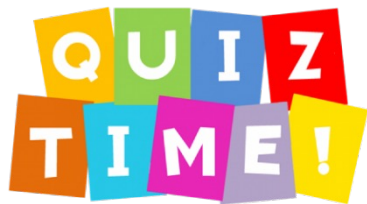
- 生产者、消费者和代理都可以被扩展，以轻松地处理非常大的消息流。





Kafka  
topics推送  
消息的是  
哪一项？

- a. 生产者 producer
- b. 消费者 consumer
- c. Kafka 代理 (broker)
- d. 分区 (partition)

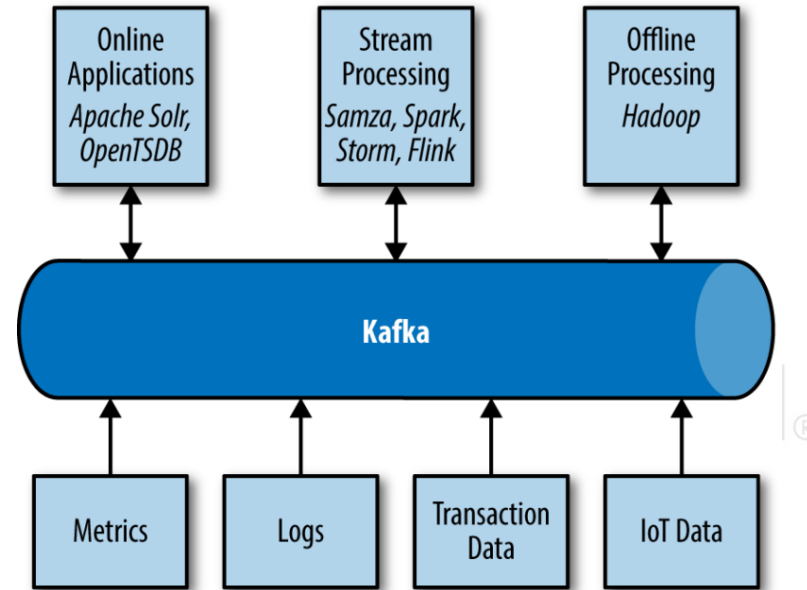


Kafka  
topics处理  
消息的是  
哪一项?

- a. 生产者 producer
- b. 消费者 consumer
- c. Kafka 代理 (broker)
- d. 分区 (partition)

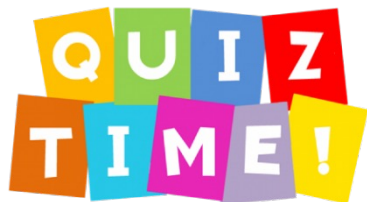
- 在任何体系结构中，Kafka 使用方法很多。本节讨论Apache Kafka的一些流行用例
  - 消息传递
    - 消息代理用于将数据处理与数据生产者分离。
  - 网站活动跟踪
    - Kafka最初的用例是能够将用户活动跟踪管道重建为一组实时发布订阅源。
    - 这意味着站点活动（页面视图、搜索或用户可能采取的其他操作）将发布到中心主题，每个活动类型有一个主题。
  - **Metrics**（监控指标）
    - 应用程序定期向Kafka主题发布**Metrics**。
  - 日志聚合
    - 使用Kafka可以将日志或事件数据抽象为消息流，从而消除对文件细节的依赖。
  - 流式处理
    - Kafka在多个阶段收集数据进行处理——例如，从Topic消耗的原始数据被丰富或转换为新的Kafka Topic以供进一步使用。因此，这种处理也称为流处理。

- 为数据生态系统提供了循环系统，如图所示。



- 它在所有基础结构的各个成员之间传递消息，为所有客户端提供一致的接口。
- 组件可以随着业务案例的创建和分解而添加或删除，而生产者不需要关心谁在使用数据或使用应用程序的数量。

- LinkedIn的开发团队由Jay Kreps领导，他是一位首席软件工程师，他曾负责Voldemort（一种分布式密钥存储系统）的开发和开源发行。
- Kafka设计的主要目标是：
  - 通过使用推拉模型来分离生产者和消费者
  - 为消息传递系统中的消息数据提供持久性，以允许多个使用者
  - 优化消息的高吞吐量
  - 允许系统的水平扩展随着数据流的增长而增长
- 其结果是一个发布/订阅邮件系统，该系统具有典型的消息传递系统的接口，但存储层更像日志聚合系统。
- Kafka是在2010年底作为一个开源项目在GitHub上发布的。随着它开始在开源社区引起关注，它在2011年7月被提议并被接受为Apache软件基金会孵化器项目。



从服务器收集物理日志文件并放在中心位置（文件服务器或 HDFS）进行处理的过程叫什么？

a. 流式处理

b. 日志聚合

c. Metric（指标）

d. 消息传递

## ○ 消息和批处理

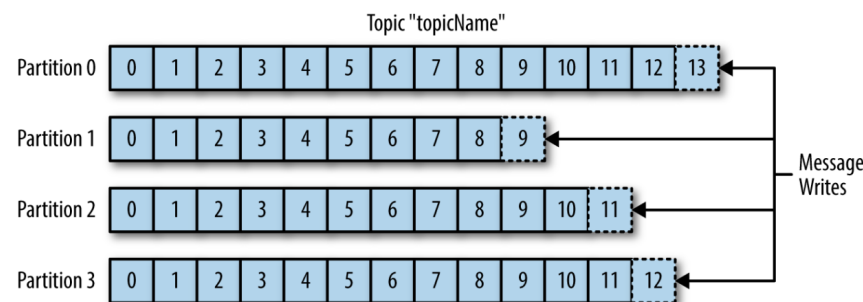
- 就Kafka而言，消息只是一个字节数组，因此其中包含的数据对Kafka来说没有特定的格式或含义。
- 批处理只是消息的集合，所有这些消息都是针对同一主题和分区生成的。

## ○ 模式

- 根据应用程序的个别需要，消息模式有许多可用的选项。简单化的系统，如JavaScript对象表示法（JSON）和可扩展标记语言（XML），易于使用和可读。
- 在Kafka中，一致的数据格式是很重要的，因为它允许消息的读写分离。

## ○ 主题和分区

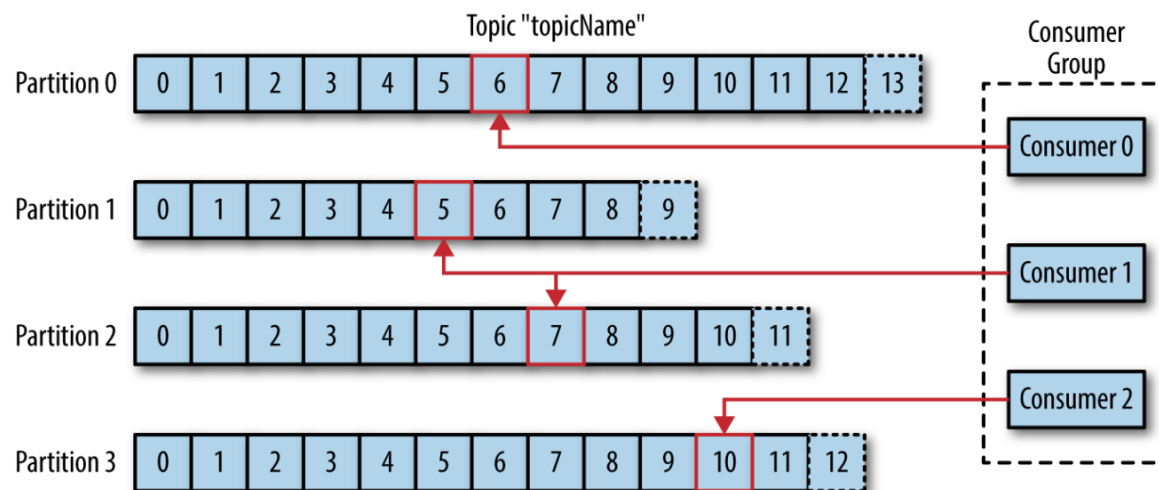
- 主题还被分解为多个分区。回到“提交日志”描述，分区是单个日志。





## ○ 生产者和消费者

- 生产者创造新的信息。在其他发布/订阅系统中，这些可能被称为发布者或写入者。
- 消费者阅读信息。在其他发布/订阅系统中，这些客户端可以称为订阅服务器或读卡器。



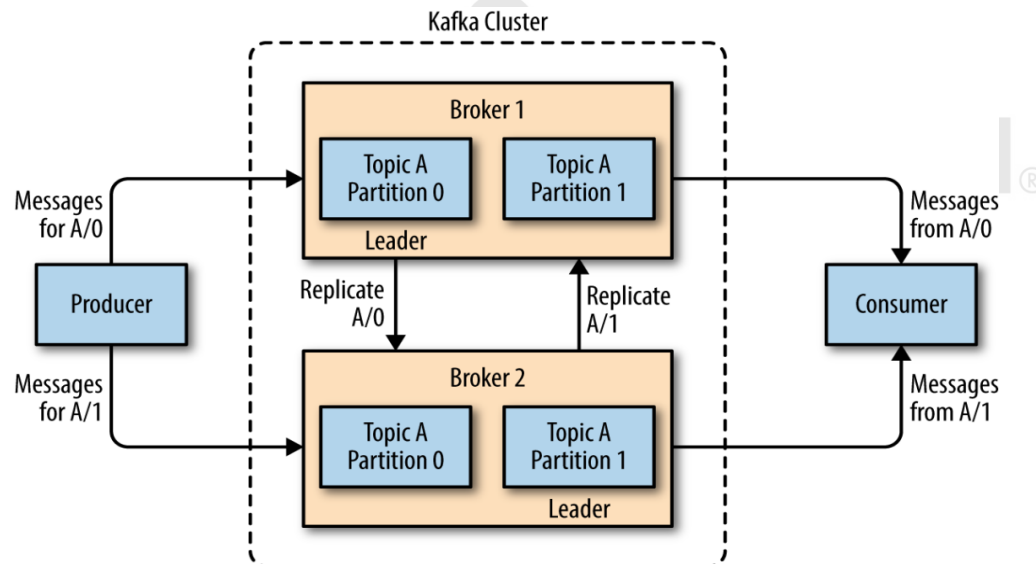
- 消费者可以横向扩展以使用包含大量消息的主题。此外，如果单个使用者失败，组的其余成员将重新平衡正在使用的分区，以接管丢失的成员。

## ○ 代理和和集群

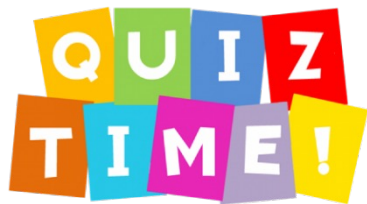
- 单个Kafka服务器称为代理。代理接收来自生产者的消息，为其分配偏移量，并将消息提交到磁盘上的存储器中。
- Kafka 代理是设计成集群的一部分。

## ○ 代理和集群

- Kafka 代理是设计成集群的一部分。
- 控制器负责管理操作，包括为代理分配分区和监视代理故障。控制器Broker从集群所有的Broker中选举出来。
- 一个分区由集群中的一个代理拥有，这个代理称为分区的Leader。



- Apache Kafka 的一个关键功能是保留功能，它是一段时间内消息的持久存储。
- 达到这些限制后，消息将过期并删除，以便保留配置是随时可用的最小数据量。



代理被设计成集群的一部分的是哪一项？

- a. Kafka
- b. Schema
- c. HBase
- d. Producer



## 活动 1.1: Apache Kafka 案例研究



请参阅实验指南了解活动1.1

## 第一章主要词汇

English	Chinese	Pinyin
publish/subscribe	发布/订阅	fā bù/ ding yuè
Producer	生产者	shēng chǎn zhě
Consumer	消费者	xiāo fèi zhě
Partition	分区	fēn qū
Multiple Producers	多个生产商	duō gè shēng chǎn shāng
Multiple Consumers	多个消费者	duō gè xiāo fèi zhě
Disk-Based Retention	基于磁盘的保留	jī yú cí pán de bǎo liú
Scalable	可扩展	kě kuò zhǎn
Stream processing	流处理	liú chǔ lǐ
Log aggregation	日志聚合	rì zhì jù hé
Metrics	度量	dù liàng

Apache Kafka 是一个分布式的发布-订阅消息传递系统，它从不同的源系统接收数据，并使数据实时可用于目标系统。

为什么 Kafka

Kafka 用例

Apache Kafka 为大数据生态系统提供了循环系统。他既可以用于数据的来源，也能用于计算后的数据存储与传输通道。

它在所有基础结构的各个成员之间传递消息，为所有客户端提供一致的接口，生产者与消费者只用关注消息的格式。

Kafka是在2010年底作为一个开源项目在GitHub上发布的。

Kafka 组件: Broker(Core Component), Topic, Partition, Producer, Consumer, Consumer Group, Kafka cluster

在下一章，你将学到

- 配置 Kafka 环境变量
- 安装Java
- 安装Zookeeper
- 安装Kafka 代理
- Broker 配置
- Kafka 控制台工具
- Kafka 代理属性列表





1. 完成学生指南手册上本章节(IK\_SG\_CN\_01.pdf)的课后练习。
2. 在自己的计算机上部署好老师共享在百度网盘的虚拟机，并根据 IK\_LG\_CN\_02.pdf 与 IK\_SG\_CN\_02.pdf)进行预习。  
链接: <https://pan.baidu.com/s/1402YnCah3ZJ7fQX8PYbW6w>  
提取码: 5viv