1-2 kafka核心概念与应用场景

Kafka的主要特点

Kafka是分布式发布-订阅消息系统。它最初由LinkedIn公司开发，之后成为Apache项目的一部分。Kafka是一个分布式的，可划分的，冗余备份的持久性的日志
用于处理活跃的流式数据。

- kafka的主要特点：
  - 同时为发布和订阅提供高吞吐量。据了解，Kafka每秒可以生产约25万消息（50 MB），每秒处理55万消息（110 MB）。
  - 可进行持久化操作。将消息持久化到磁盘，因此可用于批量消费，例如ETL，以及实时应用程序。通过将数据持久化到硬盘以及r
    止数据丢失。
  - 分布式系统，易于向外扩展。所有的producer、broker和consumer都会有多个，均为分布式的。无需停机即可扩展机器。
  - 消息被处理的状态是在consumer端维护，而不是由server端维护。当失败时能自动平衡。
  - 支持online和offline的场景。

Kafka的架构

Kafka的整体架构非常简单，是显式分布式架构，producer、broker（kafka）和consumer都可以有多个。Producer，consumer实现Kafka注册的接口，数据从pr
broker，broker承担一个中间缓存和分发的作用。broker分发注册到系统中的consumer。broker的作用类似于缓存，即活跃的数据和离线处理系统之间的缓存。
器端的通信，是基于简单，高性能，且与编程语言无关的TCP协议。

- 基本概念：
  - Topic：特指Kafka处理的消息源（feeds of messages）的不同分类。
  - Partition：Topic物理上的分组，一个topic可以分为多个partition，每个partition是一个有序的队列。partition中的每条消息都会被分
    序的id（offset）。
  - Message：消息，是通信的基本单位，每个producer可以向一个topic（主题）发布一些消息。
  - Producers：消息和数据生产者，向Kafka的一个topic发布消息的过程叫做producers。
  - Consumers：消息和数据消费者，订阅topics并处理其发布的消息的过程叫做consumers。
  - Broker：缓存代理，Kafka集群中的一台或多台服务器统称为broker。
- 发送消息的流程：
  - Producer根据指定的partition方法（round-robin、hash等），将消息发布到指定topic的partition里面
  - kafka集群接收到Producer发过来的消息后，将其持久化到硬盘，并保留消息指定时长（可配置），而不关注消息是否被消费。
  - Consumer从kafka集群pull数据，并控制获取消息的offset

kafka的优秀设计

接下来我们从kafka的吞吐量、负载均衡、消息拉取、扩展性来说一说kafka的优秀设计。

- 高吞吐是kafka需要实现的核心目标之一，为此kafka做了以下一些设计：
  - 内存访问：直接使用 linux 文件系统的cache，来高效缓存数据，对数据进行读取和写入。
  - 数据磁盘持久化：消息不在内存中cache，直接写入到磁盘，充分利用磁盘的顺序读写性能。
  - zero-copy：减少IO操作步骤
    - 采用linux Zero-Copy提高发送性能。传统的数据发送需要发送4次上下文切换，采用sendfile系统调用之后，数据直接在内核
      统上下文切换减少为2次。根据测试结果，可以提高60%的数据发送性能。Zero-Copy详细的技术细节可以参考：https://ww
      developerworks/linux/library/j-zerocopy/
    - 对消息的处理：
      - 支持数据批量发送

检测到您还没有关注慕课网服务号，无法接收课程更新通知。请扫描二维码即可绑定

下一节

1-2 kafka核心概念与应用场景

播放下一节

重新观看