# Kafka

## 简介

Apache Kafka是分布式发布-订阅系统。

Kafka具有如下特征：

*Fast：*

单机可以支持上千客户端每秒百兆的读写操作。

*Scalable:*

Kafka可以允许集群，并且可以在不down的情况下在线伸缩。数据流被分散到集群的机器上从而提供超越单机的处理能力。

*Durable:*

消息以O(1)的形式持久化到硬盘上，并且在集群中重复多份防止数据丢失。能够支持TB级消息处理。

*Distributed by Design:*

Kafka以现代集群为中心设计，提供了强持久性和容错能力。

# 入门

## 入门简介

Kafka is a distributed, partitioned,replicated commit log service. 它提供了消息系统的功能，但是使用了独特的设计。

*术语：*

Kafka支持Topic，生产者，消费者。

Kafka作为一个集群，有一个或多个broker组成。



生产者、消费者与服务器之间是通过tcp交流的。

## Topic

对于每个Topic，Kafka集群维护一个partitioned log,partition是保证顺序的，不可修改的消息append to a commit log. Partition中的消息被赋予了一个叫offset的id，用来identify本partition中的消息。

Kafka 保存所有published消息一段可配置的时间。例如两天，两天后，消息被抛弃，硬盘空间释放。Kafka可以保存巨量的数据。



实际上对于每个consumer，metadata仅仅保存当前读到的offset。Consumer可以控制offset，从而可以任意顺序的读取消息。

Log中的partition允许消息量超过单机容量，每个独立的partition必须存在于一个服务器，但一个topic可以有多个partition，所以topic可以有任意数量的消息，其次它允许一定程度的平行。

## 分布式

Partition分布到各个服务器上，每个partition在一个可配置的数量上实现备份，从而实现容错。

每个partition有一个leader server，0或多个followers。The leader处理所有的请求读写，follower备份leader。如果leader fails，one of the followers自动变成new leader。每个server为某些partition充当leader，为其它一些partition充当follower，从而实现负载。

## Producer

生产者发布消息到Topic中。生产者负责选择消息应该放到topic的哪个Partition中。通用的方式是使用轮盘法。

## Consumer

消息通常有两种模型：队列和发布-订阅模式。在队列模式中，每个消息仅到达众多consumer中的一个，发布-订阅模式会把消息广播到所有的consumer.Kafka提供了单个consumer group来抽象两种模型。

Consumer可以标记它们自己一个Consumer group name,每个发送到Topic中的消息发送到订阅该消息的Consumer group中的其中一个Consumer。如果所有的Consumer有相同的Consumer group，那么this works like a traditional queue balancing load over the consumer。如果所有的consumer有不同的Consumer group,this works like publish-subscribe模式，消息广播到所有的Consumer中。

一般的用法通常是建立少量的Consumer group，每个group有众多的Consumer。Kafka是强顺序的。

通过将topic中的partition分给consumer，每个partition一个consumer可以实现强顺序。

Kafka提供了如下保证：

* 发送者A按顺序发送M1,M2发送到某个partition，该partition将会按发送的顺序保存。
* A consumer instance sees message in the order they are stored in the log.
* For a topic with replication factor N, we will tolerate up N-1 sever failures without losing any messages committed to the log.