

Kafka的了解与使用

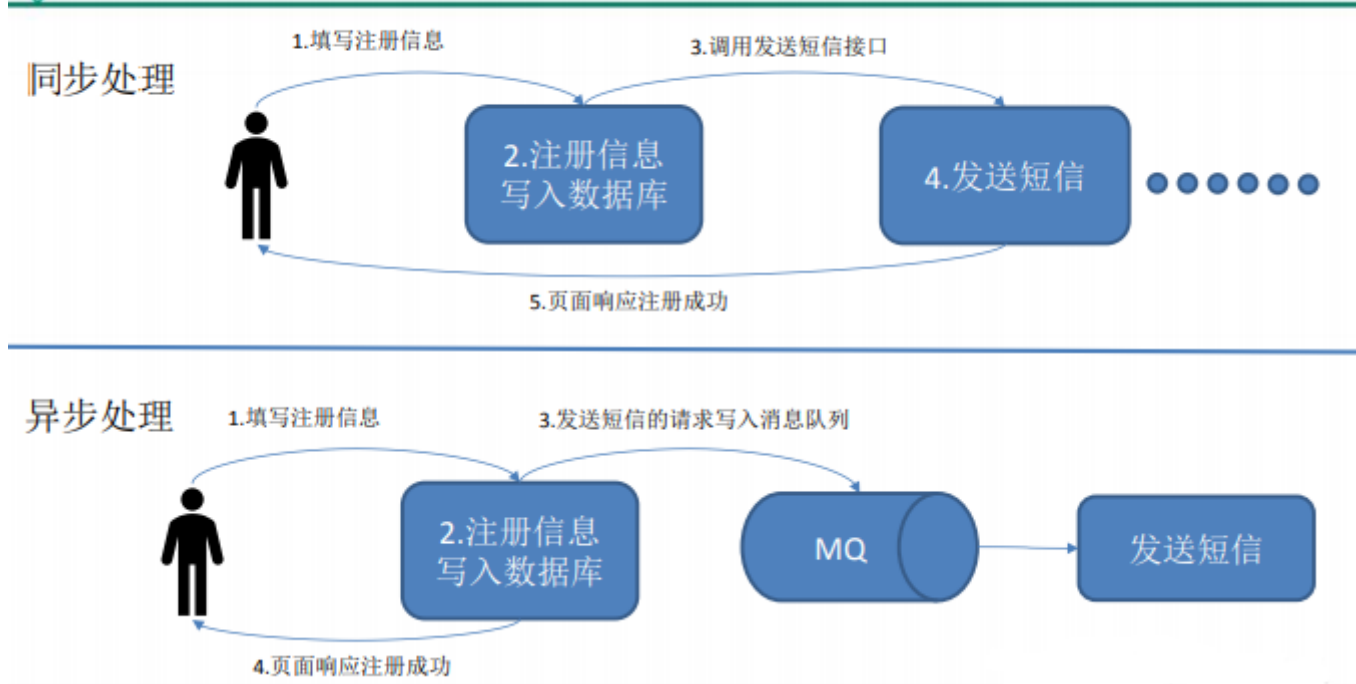
1) Kafka的概述

- Kafka是一个分布式的基于发布/订阅模式的消息队列(Message Queue, 简称MQ), 主要用于大数据实时处理领域。

2) Kafka的相关概念

2.1) 消息队列传统应用

MQ传统应用场景之异步处理



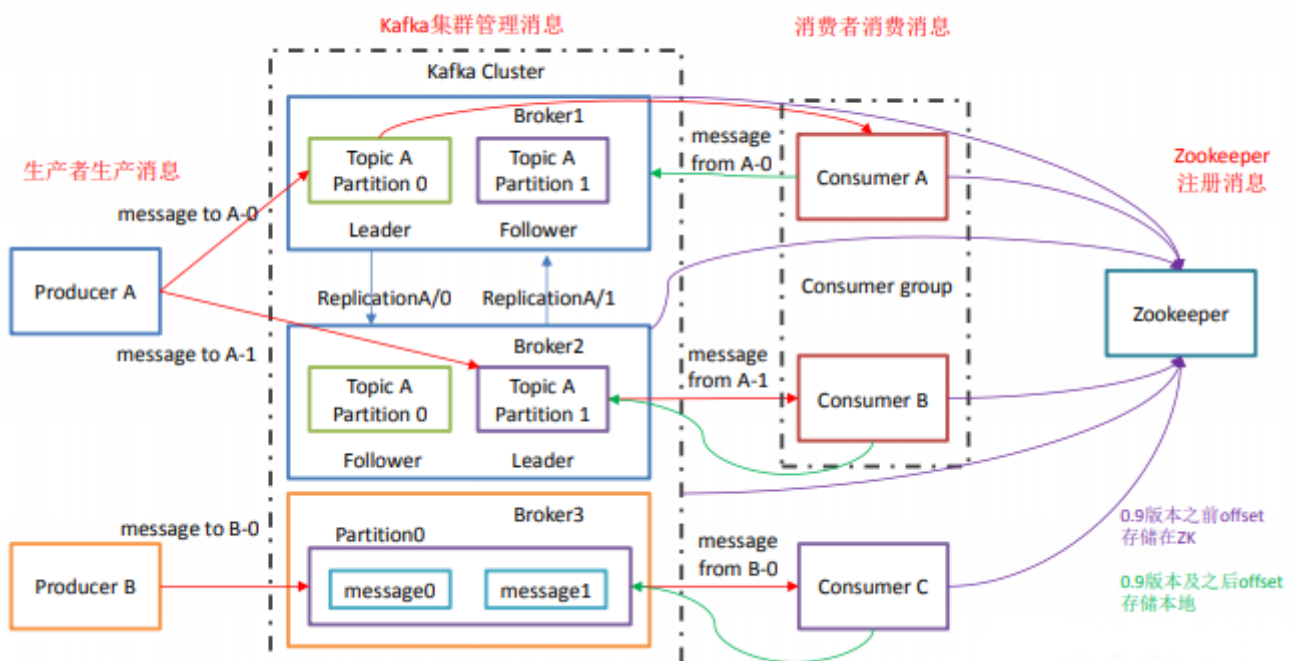
2.2) 消息队列的好处

- 解耦: 允许独立的扩展或修改两边的处理过程, 只需要遵守相同的接口约束
- 可恢复性: 系统一部分组件失效时, 不会影响整个系统, 即使某个进程挂掉, 已进入队列的消息可在系统恢复后被处理
- 缓冲: 解决生产消息和处理消息速度不一致的情况
- 灵活性与峰值处理能力: 削峰以及动态处理资源投入和撤销
- 异步通信: 提供异步处理机制

2.3) 消息队列的两种模式

- 点对点模式：一对一，生产者发布到Queue上，消费者主动拉取，消息收到后清除
- 发布/订阅模式：一对多，生产者发布到topic上，消费者消费后不删除。发布/订阅模式细分为两种，一种是队列主动推送信息，由队列控制发送速度，可能造成部分消费者处理不过来，导致奔溃或部分消费者处理数量太少，资源浪费；另一种是消费者主动拉取信息，由各消费者控制收取速度，消费者需要轮询去查看是否有信息需要拉取，会产生一定的资源浪费。Kafka是属于消费者主动拉取模式。

2.4) Kafka的基础架构



- broker：一台kafka就是一个broker(消息代理)，一个broker可容纳多个topic
- consumer group：消费者组，同一个组内的消费者之间只能处理同一个topic的不同partition分区，一般来说组内消费者数和分区数一致可保证资源不浪费
- partition：为了实现扩展性，一个非常大的topic可以分布到多个broker上，即分为多个partition，每个partition是有序的队列
- replication：副本，保证集群上的某个节点故障时，该节点的partition数据不丢失。一个分区可由一个leader和多个follower组成

2.5) 部署Kafka集群(参考网上部署到本地伪分布式)

3) Kafka的使用

3.1) Kafka命令行操作

- topic相关

查看topic

```
bin/kafka-topics.sh --zookeeper Master:2181 --list
```

创建topic,指定副本数(副本数不可大于服务器数broker)和分区数

```
bin/kafka-topics.sh --zookeeper Master:2181 --create --replication-factor 3 --partitions 1 --topic first
```

删除topic

```
bin/kafka-topics.sh --zookeeper Master:2181 --delete --topic first
```

描述topic

```
bin/kafka-topics.sh --zookeeper Master:2181 --describe --topic first
```

- 生产者消费者相关

控制台生产者(通过消息代理broker向topic发送信息),发送端口9092

```
bin/kafka-console-producer.sh --topic --broker-list Master:9092
```

控制台消费者,接收端口2181(过时方法)

```
bin/kafka-console-consumer.sh --topic --zookeeper Master:2181 --from-beginning
```

控制台消费者,使用端口9092(新方法),与offset有关,50个offset分区组成一个副本

```
bin/kafka-console-consumer.sh --topic --bootstrap-server Master:9092
```

3.2) Kafka日志与数据分离

server.properties里的log.dirs其实是数据目录, 建议修改成专有数据目录