

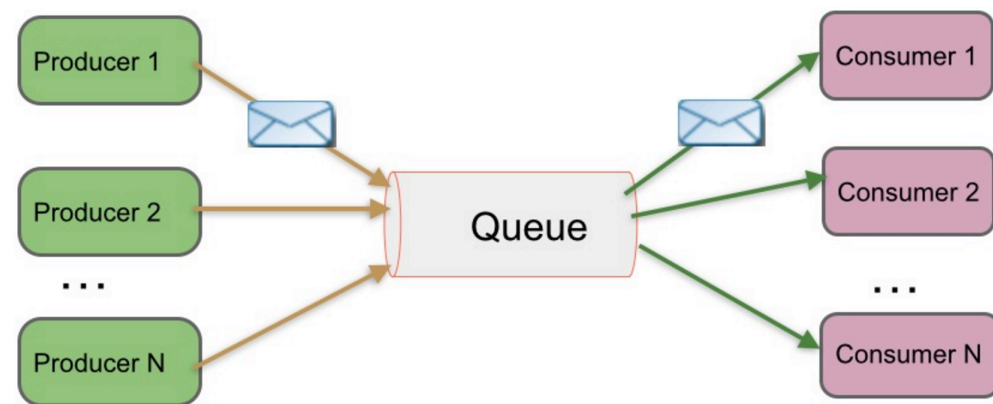


《Kafka原理剖析及实战演练》 第1课



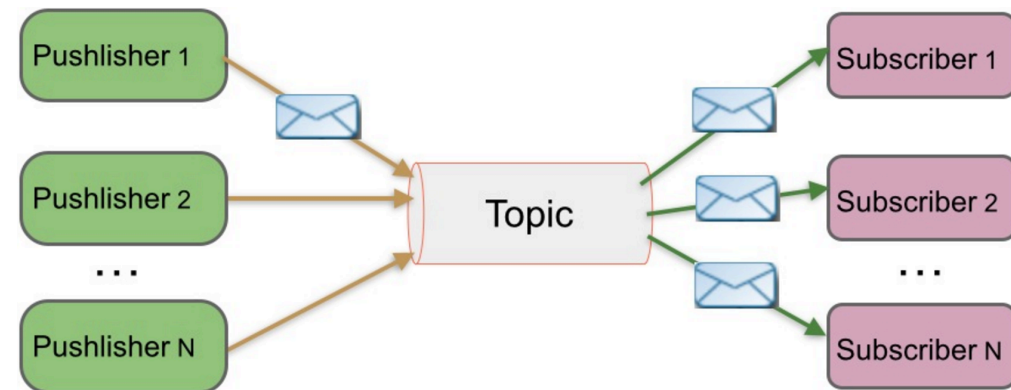
■ Peer-to-Peer

- 一般基于Pull或者Polling接收消息
- 发送到队列中的消息被一个而且仅仅一个接收者所接收，即使有多个接收者在同一个队列中侦听同一消息
- 即支持异步“即发即弃”的消息传送方式，也支持同步请求/应答传送方式



■ 发布/订阅

- 发布到一个主题的消息，可被多个订阅者所接收
- 发布/订阅即可基于Push消费数据，也可基于Pull或者Polling消费数据
- 解耦能力比P2P模型更强

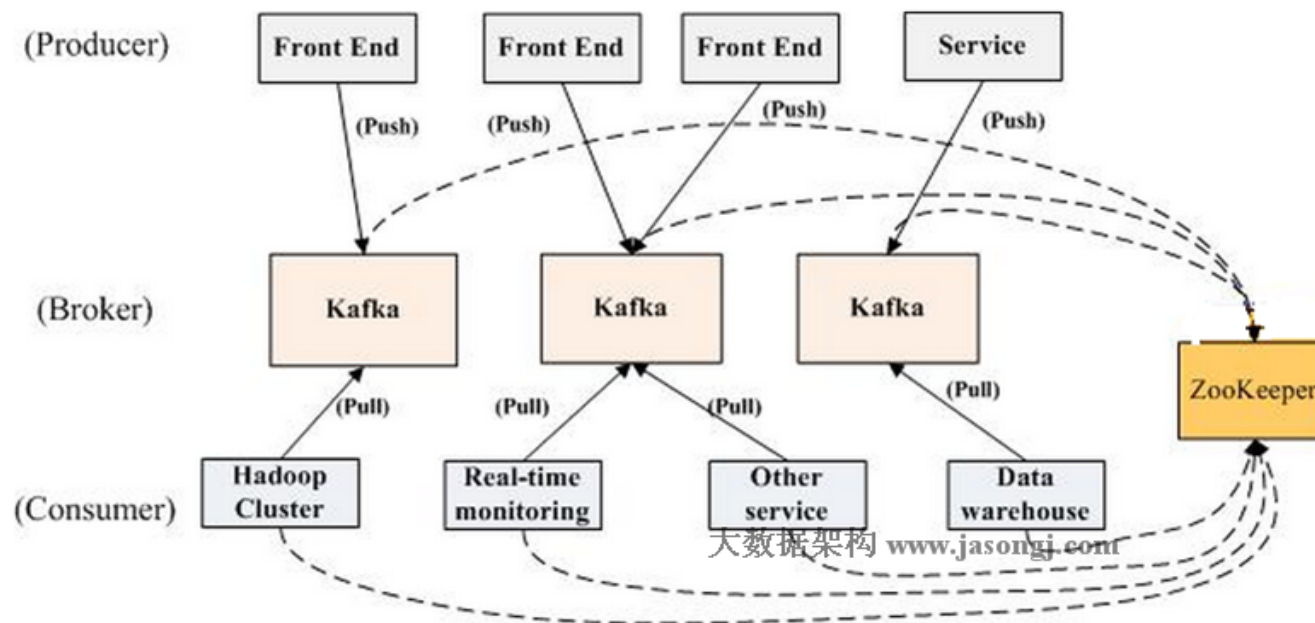


- **解耦** 各位系统之间通过消息系统这个统一的接口交换数据，无须了解彼此的存在
- **冗余** 部分消息系统具有消息持久化能力，可规避消息处理前丢失的风险
- **扩展** 消息系统是统一的数据接口，各系统可独立扩展
- **峰值处理能力** 消息系统可顶住峰值流量，业务系统可根据处理能力从消息系统中获取并处理对应量的请求
- **可恢复性** 系统中部分组件失效并不会影响整个系统，它恢复后仍然可从消息系统中获取并处理数据
- **异步通信** 在不需要立即处理请求的场景下，可以将请求放入消息系统，合适的时候再处理

- **RabbitMQ** Erlang编写，支持多协议 AMQP，XMPP，SMTP，STOMP。支持负载均衡、数据持久化。同时支持Peer-to-Peer和发布/订阅模式
- **Redis** 基于Key-Value对的NoSQL数据库，同时支持MQ功能，可做轻量级队列服务使用。就入队操作而言，Redis对短消息（小于10KB）的性能比RabbitMQ好，长消息的性能比RabbitMQ差。
- **ZeroMQ** 轻量级，不需要单独的消息服务器或中间件，应用程序本身扮演该角色，Peer-to-Peer。它实质上是一个库，需要开发人员自己组合多种技术，使用复杂度高
- **ActiveMQ** JMS实现，Peer-to-Peer，支持持久化、XA事务
- **Kafka/Jafka** 高性能跨语言的分布式发布/订阅消息系统，数据持久化，全分布式，同时支持在线和离线处理
- **MetaQ/RocketMQ** 纯Java实现，发布/订阅消息系统，支持本地事务和XA分布式事务

- **高吞吐率** 在廉价的商用机器上单机可支持每秒100万条消息的读写
- **消息持久化** 所有消息均被持久化到磁盘，无消息丢失，支持消息重放
- **完全分布式** Producer , Broker , Consumer均支持水平扩展
- **同时适应在线流处理和离线批处理**

Kafka架构简介



如何安装和使用Kafka集群

- 安装Docker
- 创建CentOS容器
- 安装JDK 1.7+
- 安装Zookeeper
- 启动Kafka broker

- 安装Docker
- 创建CentOS容器
- 安装JDK 1.7+
- 安装Zookeeper
- 启动Kafka broker

Thanks

FAQ时间