

1. RabbitMQ概述

前言

Rabbit, 兔子的意思

互联网行业很多公司, 都喜欢用动物命名产品, 或者作为公司的logo, 吉祥物. 比如: 腾讯的企鹅, 京东的狗, 美团的袋鼠, 携程的海豚, 阿里就更多了, 蚂蚁, 飞猪, 天猫, 菜鸟, 闲鱼, 盒马.... 更是以一己之力, 组建了一个动物园...

Rabbit 也是一个公司名. MQ (message queue) 消息队列 的意思, RabbitMQ 是 Rabbit 企业下的一个消息队列产品.

RabbitMQ 是一个实现了 AMQP 的 消息队列 服务, 是当前主流的消息中间件之一.



AMQP, 即Advanced Message Queuing Protocol(高级消息队列协议), 是一个通用的应用层协议, 提供统一消息服务的协议, 为面向消息的中间件设计。基于此协议的客户端与消息中间件可传递消息, 并不受客户端或中间件, 开发语言等条件的限制。

在互联网架构中, 会经常使用MQ来作为消息通信服务. 接下来我们看下什么是MQ

什么是MQ

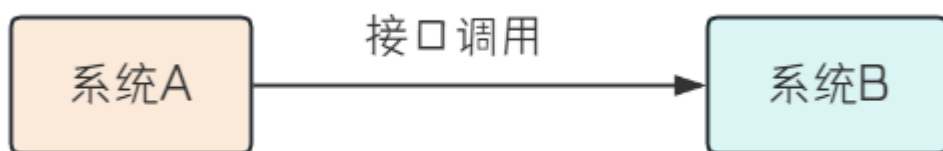
MQ(Message queue), 从字面意思上看, 本质是个队列, FIFO 先入先出, 只不过队列中存放的内容是消息(message) 而已. 消息可以非常简单, 比如只包含文本字符串, JSON等, 也可以很复杂, 比如内嵌对象.

MQ多用于分布式系统之间进行通信.

系统之间的调用通常有两种方式:

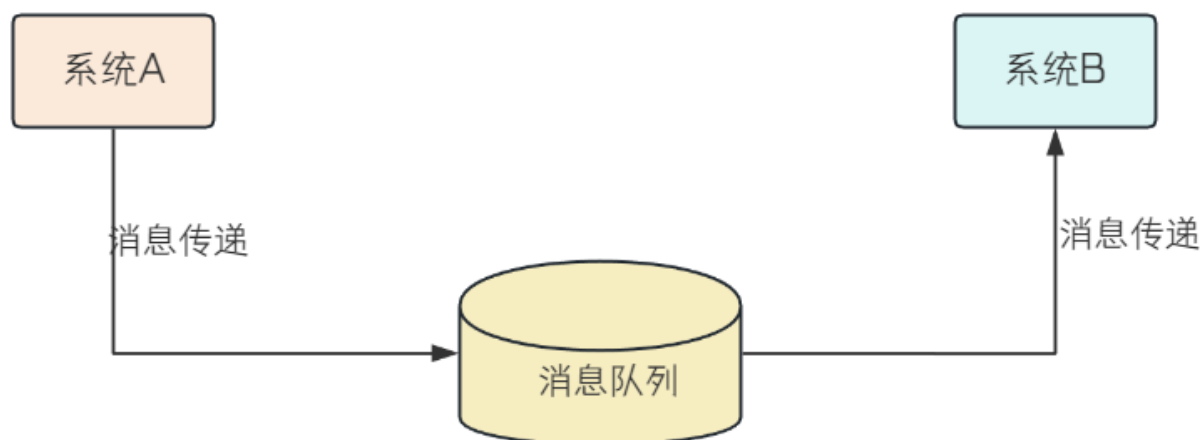
1. 同步通信

直接调用对方的服务, 数据从一端发出后立即就可以达到另一端.



2. 异步通信

数据从一端发出后，先进入一个容器进行临时存储，当达到某种条件后，再由这个容器发送给另一端。容器的一个具体实现就是MQ(`message queue`)

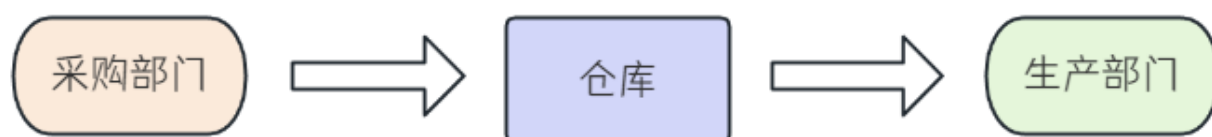


RabbitMQ 就是MQ的一种实现

MQ的作用

MQ主要工作是接收并转发消息, 在不同的应用场景下可以展现不同的作用

可以把MQ想象成一个仓库. 采购部门进货之后, 把零件放进仓库里, 生产部门从仓库中取出零件, 并加工成产品. MQ和仓库的区别是, 仓库里放的是物品, MQ里放的是消息, 仓库负责存储物品,并转发物品, MQ负责存储和转发消息



- 异步解耦:** 在业务流程中, 一些操作可能非常耗时, 但并不需要即时返回结果. 可以借助MQ把这些操作异步化, 比如 用户注册后发送注册短信或邮件通知, 可以作为异步任务处理, 而不必等待这些操作完成后才告知用户注册成功.
- 流量削峰:** 在访问量剧增的情况下, 应用仍然需要继续发挥作用, 但是这样的突发流量并不常见. 如果以能处理这类峰值为标准而投入资源,无疑是巨大的浪费. 使用MQ能够使关键组件支撑突发访问压力, 不会因为突发流量而崩溃. 比如秒杀或者促销活动, 可以使用MQ来控制流量, 将请求排队, 然后系统根据自己的处理能力逐步处理这些请求.
- 消息分发:** 当多个系统需要对同一数据做出响应时, 可以使用MQ进行消息分发. 比如支付成功后, 支付系统可以向MQ发送消息, 其他系统订阅该消息, 而无需轮询数据库.

4. **延迟通知:** 在需要在特定时间后发送通知的场景中, 可以使用MQ的延迟消息功能, 比如在电子商务平台中, 如果用户下单后一定时间内未支付, 可以使用延迟队列在超时后自动取消订单
5.

为什么选择RabbitMQ

目前业界有很多的MQ产品, 例如RabbitMQ, RocketMQ, ActiveMQ, Kafka, ZeroMQ等, 也有直接使用Redis充当消息队列的案例, 这些消息队列, 各有侧重, 也没有好坏, 只有适合不适合, 在实际选型时, 需要结合自身需求以及MQ产品特征, 综合考虑

下面我们介绍一下当前最主流的3种MQ产品

1. Kafka

Kafka一开始的目的就是用于日志收集和传输, 追求高吞吐量, 性能卓越, 单机吞吐达到十万级, 在日志领域比较成熟, 功能较为简单, 主要支持简单的MQ功能, 如果有日志采集需求, 肯定是首选kafka了。

2. RocketMQ

RocketMQ采用Java语言开发, 由阿里巴巴开源, 后捐赠给了Apache.

它在设计时借鉴了Kafka, 并做出了一些自己的改进, 青出于蓝而胜于蓝, 经过多年双十一的洗礼, 在可用性、可靠性以及稳定性等方面都有出色的表现. 适合对于可靠性比较高, 且并发比较大的场景, 比如互联网金融. 但支持的客户端语言不多, 且社区活跃度一般

3. RabbitMQ

采用Erlang语言开发, MQ功能比较完备, 且几乎支持所有主流语言, 开源提供的界面也非常友好, 性能较好, 吞吐量能达到万级, 社区活跃度也比较高, 比较适合中小型公司, 数据量没那么大, 且并发没那么高的场景.

综合: 由于RabbitMQ的综合能力较强, 咱们这边的项目没有那么大的高并发, 且RabbitMQ社区比较成熟, 管理界面友好, 所以咱们接下来主要学习RabbitMQ的使用

RabbitMQ介绍

RabbitMQ 官网: [RabbitMQ: One broker to queue them all | RabbitMQ](#)

RabbitMQ是采用Erlang语言实现AMQP(Advanced Message Queuing Protocol,高级消息队列协议)的消息中间件, 它最初起源于金融系统领域, 为了在分布式系统中存储和转发消息而设计的.

在此之前, 有一些消息中间件的商业实现, 比如微软的MSMQ(MicroSoft Message Queue), IBM的WebSphere等. 但是他们价格太贵了, 一般只应用于大型组织机构.

RabbitMQ开发始于2006年, 是由Rabbit Technologies开发并且提供商业支持的. 之所以取名为Rabbit, 是因为兔子行动非常迅速且繁殖起来非常疯狂, RabbitMQ的开创者认为这个名字再合适不过了.

2010年4月, Rabbit Technologies被 VMware 旗下的 SpringSource 收购, 在 2013 年 5 月被并入 Pivotal.

其实 VMware, Pivotal 本质上是一家的. 不同的是, VMware 是独立上市子公司, 而 Pivotal 是整合了 EMC的某些资源, 其中我们现在使用的 Spring 系列框架, 就是 Pivotal 公司热门的产品之一。

直到后来 Pivotal 将其开源, RabbitMQ 才逐渐走向大众!

RabbitMQ 发展到今天, 已经被越来越多的人认可, 尤其是互联网公司, 已经有着大规模的场景应用, 这和它在易用性, 扩展性, 可靠性和高可用性等方面的卓越表现是分不开的. 接下来我们就一起来深入了解一下 RabbitMQ.