RabbitMQ

1. 安装

MacOS

使用Homebrew 进行安装：

先安装Homebrew，如下：

|  |
| --- |
| /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" |

安装RabbitMQ

1. brew update
2. brew install rabbitmq

导入环境变量(rabbitMQ的脚本和命令行工具在：/usr/local/sbin目录下)：

export PATH=$PATH:/usr/local/sbin

启动rabbitmq服务：

rabbitmq-server 或rabbitmqctl start\_app

关闭rabbitmq服务：

rabbitmqctl stop\_app

1. 命令行工具
2. rabbitmqctl
3. rabbitmq-plugins（插件管理）
4. rabbitmqadmin（）
5. RabbitMQ管理UI

启动rabbitmq management 插件：

rabbitmq-plugins enable rabbitmq\_management

监控页面默认端口号为：15672

访问页面账号/密码为：guest/guest

1. 应用场景

异步、解耦、削峰。

1. 跨系统的异步通信
2. 应用内的同步变成异步

秒杀

1. 给予Pub/Sub模型实现的事件驱动

ETL

1. 利用RabbitMQ实现事物的最终一致性
2. AMQP协议

AMQP，即：Advanced Message Queuing Protocol，一个提供统一消息服务的应用层标准高级消息队列协议，是应用层协议的一个开放标准，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/中间件产品、不同的开发语言等条件的限制。

AMQP的实现有：RabbitMQ、Apache Qpid等。

1. RabbitMQ的特性
2. 可靠性
3. 灵活的路由
4. 消息的集群
5. 高可用
6. 多种协议
7. 多语言客户端
8. 管理页面
9. 插件机制
10. 工作模型
11. 进阶知识
12. 怎么自动删除没人消费的消息

避免大量消息堆积。

1. 设置队列的消息的TTL（Time To Live 存活时间）
2. 指定消息的TTL
3. 无法路由的消息，去了哪里？

死信队列（Dead Letter Queue）。对应的交换机：DLX（Dead Letter Exchange）

那些情况消息会变成死信？

1. 消息过期
2. 消费者Reject或Nack并且requeue == false（即：拒绝或未应答，并且拒绝重新入队）
3. 队列达到最大长度 (数量或字节数)：先入队的消息会被删除。

无法路由的消息进入死信队列。

1. 可以让消息优先得到消费吗？

优先级队列：

在发送端设置消息的优先级如：

|  |
| --- |
| AMQP.BasicProperties properties = **new** AMQP.BasicProperties.Builder()  . priority (5) *//优先级，默认为：5，配合队列的x-max-priority 属性使用* .build()； |

设置队列的x-max-priority属性如：

|  |
| --- |
| Map<String, Object> basicProperties = **new** LinkedHashMap<String, Object>();basicProperties.put(**"x-max-priority"**, **"10"**);*//最大优先级* |

1. 如何实现延迟发送消息？

RabbitMQ本身不支持延迟队列。

如何实现？

1. 延迟队列的插件
2. 配合TTL和死信队列实现
3. MQ如何实现RPC？

采用请求队列和响应队列实现RPC远程方法调用。

1. RabbitMQ流量控制怎么做？设置队列大小有用吗？

秒杀系统。

x-max-length：

1. 消息堆积的时候才有用。当消费速率与生产速率。
2. 先进入队列的消息删除掉。

流量控制：

1. 服务端

Flow Control

内存和磁盘大小剩余。

1. 消费端

多个消费者监听一个队列。消息轮询消费。

平均 != 公平

设置prefetch count（预取数量）

说明：非自动确认消息的前提下，如果一定数目的消息（通过基于consumer设置Qos的值）未被确认前，不进行消费新的消息。

channel.basicQos(3);