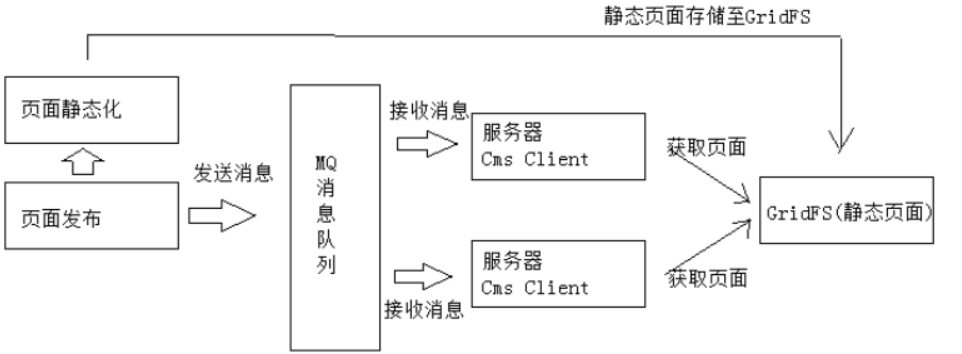
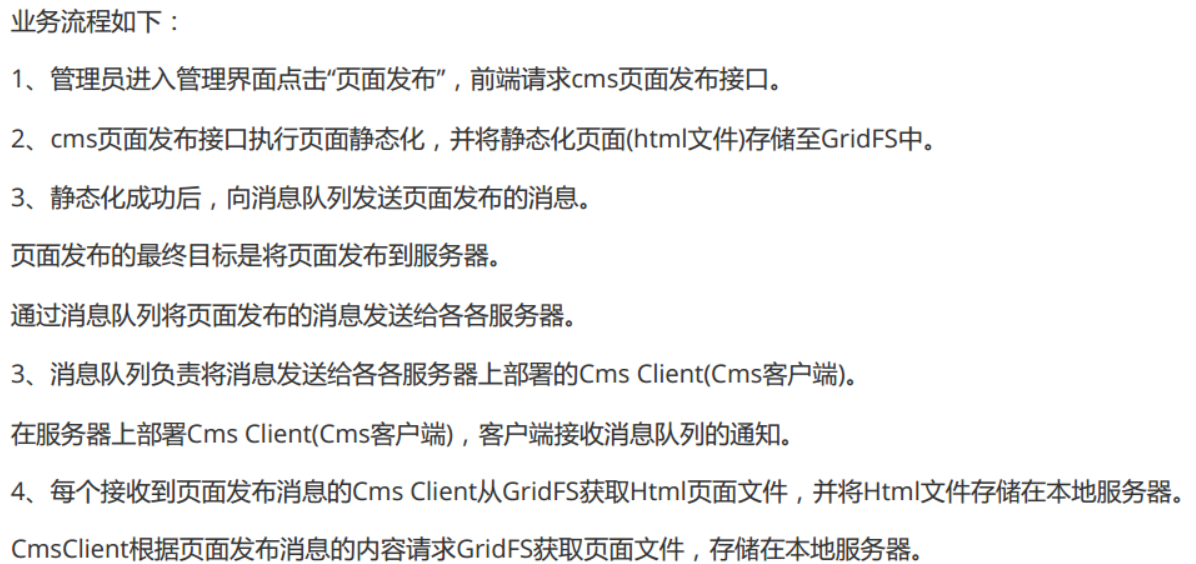
**页面发布流程**





**消息队列的应用场景**

 异步处理：传统方式是将一些非必要的业务逻辑以同步的方式运行，太耗费时间。而将消息写入消息队列，非必要的业务逻辑以异步的方式运行，加快响应速度

 应用解耦：传统方式的系统间耦合性太强，新的订阅者加入还需要发布者修改代码，中间件模式的优点：将消息写入消息队列，需要消息的系统自己从消息队列中订阅，从而发布者不需要做任何修改。

 流量消峰（秒杀）：当流量过大时，可以先把的信息积压到队列中，超过一定数量的信息直接丢弃掉，这样可以避免在短时间的高流量压垮应用。

消息队列常见问题和解决方案：<https://blog.csdn.net/qq_36236890/article/details/81174504>

RabbitMQ 问答式总结

<https://www.jianshu.com/p/a9a903c2d6a1>

RabbitMq指南

<https://www.jianshu.com/p/3e76a144bebd>

为什么使用RabbitMQ

1.SpringBoot默认集成RabbitMQ。

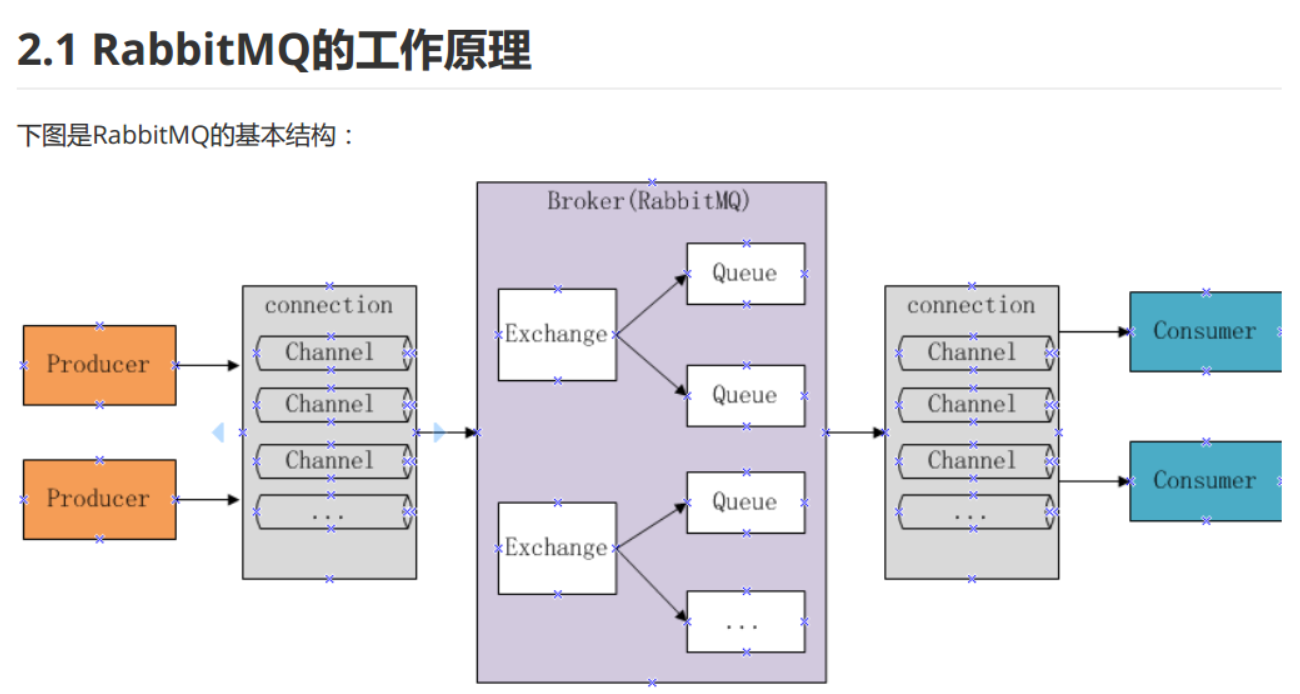
2.实用erlang语言开发，高并发性能好，延迟最低

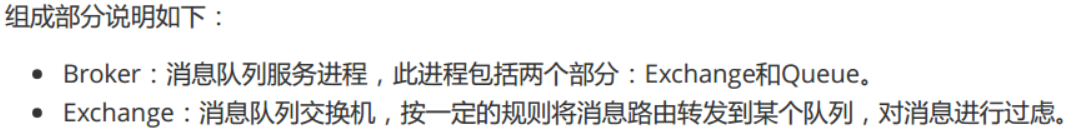
3.社区活跃度好，官方提供了web管理界面

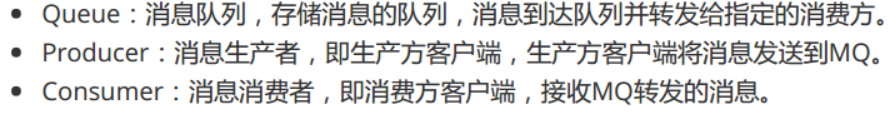
缺点：

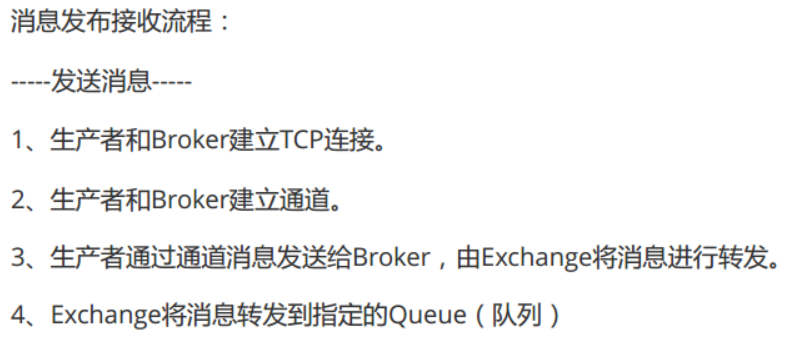
吞吐量不高

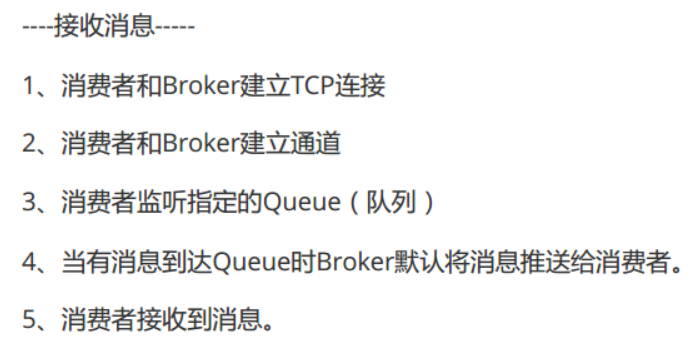
用erlang写的源码难懂，难以定制。











发送端操作流程：

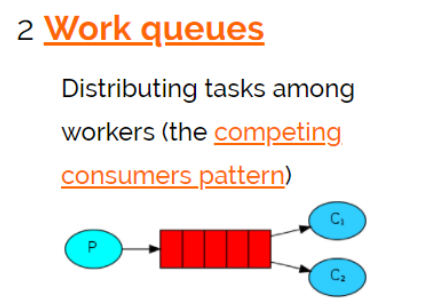
1.创建连接2.创建通道3.声明队列4.发送消息

接收端操作流程

1.创建连接2.创建通道3.声明队列4.监听队列5.接收消息6.ack回复

**RabbitMQ的工作模式**

1.工作队列模式

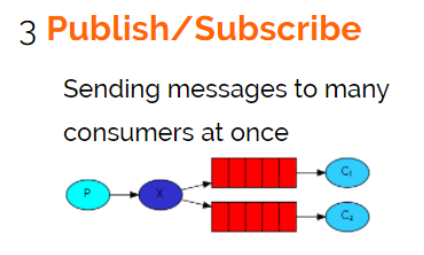


特点：多个消费端共同监听同一个队列，共同消费其中的消息。

结果：1.一条消息只会被一个消费者接收；2.rabbit采用轮询的方式将消息平均地发给消费者。3.消费者在处理完某条消息后，才会收到下一条消息。

应用场景：对于任务过重或任务较多情况使用工作队列可以提高任务处理的速度。

2.发布/订阅模式



特点：

1.每个消费者监听自己的队列

2.生产者将消息发给broker，由交换机将消息转发到绑定此交换机的每个队列，每个绑定交换机的队列都将接收到消息

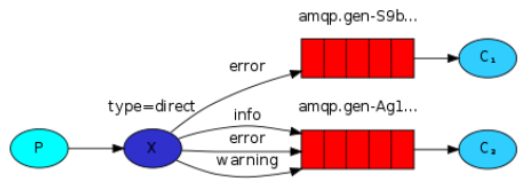
生产者代码：

1.声明FANOUT类型的交换机。

2.声明多个队列并绑定到此交换机，绑定时不需要指定bindingkey。

3.发送消息时不需要指定routingkey。

3.路由模式



交换机类型：direct

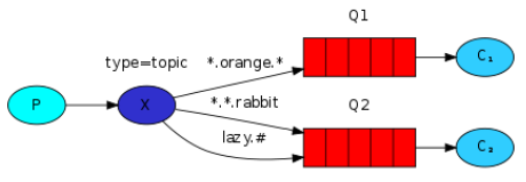
特点：

1.一个交换机绑定多个队列，每个队列在绑定交换机时设置bindingkey，并且一个队列可以设置多个路由routingkey。

2.每个消费者监听自己的队列。

3.生产者发消息给交换机，发送消息时需要指定routingkey的值，交换机来判断该routingkey的值和哪个队列的bindingkey相等，如果相等则将消息转发给该队列。

4.通配符工作模式



交换机类型：topic

特点：

1.一个交换机可以绑定多个队列，每个队列可以设置一个或多个带通配符的bindingkey

2.生产者将消息发给交换机，交换机根据routingkey的值来匹配队列，匹配时采用通配符的方式，匹配成功的将消息转发到指定的队列。#匹配0个、一个或者多个词，\*匹配单个词。

4.headers模式

交换机类型：headers

特点：headers模式取消routingkey，而是根据发送的消息内容中的headers属性进行匹配。

在绑定Queue与Exchange时指定一组键值对；当消息发送到Exchange时，RabbitMQ会取到该消息的headers（也是一个键值对的形式），对比其中的键值对是否完全匹配Queue与Exchange绑定时指定的键值对。如果完全匹配则消息会路由到该Queue，否则不会路由到该Queue。

5.RPC(remote procedure call)

