读多写少的数据库：Memcached和Redis。

记录用户行为的表单：需要写多读少的数据库，适合使用HBase。

SNS(Social Networking Services)系统。

Push写扩散：

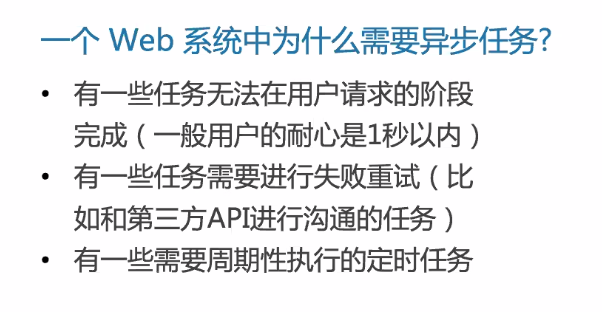
heartbeat就是一种Push模式。

Poll(轮询)：就是一个for循环+Pull模式

在Load Balance中，如果配置了多台机器，主机在判断下属的分机是否仍然在工作时，较好的方式是采用heartbeat的Push模式，避免去使用Poll，原因是heartbeat更为简洁，分机只需要将信息发给主机，所有的判断都在主机上进行，而且即使主机上的判断服务发生问题，只要能接受分机heartbeat的信息并做好记录，当主机的判断服务恢复时，仍然可以根据记录来判断分机是否挂掉；而Poll模式，需要每台分机上都有一个微服务接受Request并返回Response的操作，信息要一来一回，代码要更复杂些，而且如果主机的判断服务出问题了，判断分机是否挂掉就无法完成。

Push模式：

异步任务：



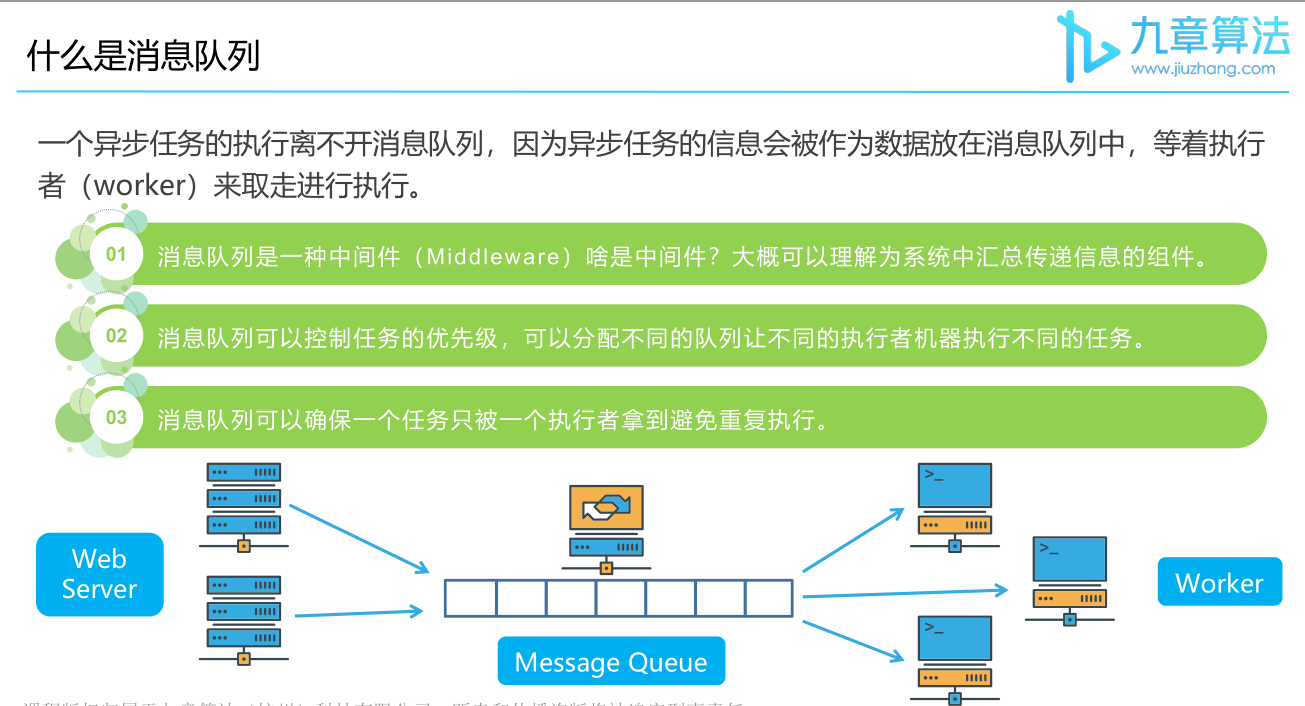
Buffer：解决生产者和消费者速度不一致的问题。Message Queue其实也是一个Buffer。

生活中的快递站就是一个Buffer，一次性会把几百件快递放到快递站，然后由快递员分发。

凡是用到第三方API调用的，基本都需要异步任务。

注册账号时，涉及到验证码时，由于是调用第三方API，所以需要异步任务。

Message Queue：削峰填谷，三峡大坝就是一个Message Queue。



Message Queue中的任务如果执行完毕了，WebServer如何获取这一信息：

1. Worker(如果可以)将结果去写数据库，WebServer通过查询数据库来获取。
2. Worker通过HTTP直接把结果返回给WebServer。
3. Worker通过Message Queue返回结果(不常用，也不推荐使用)

平时遇到的异步任务：

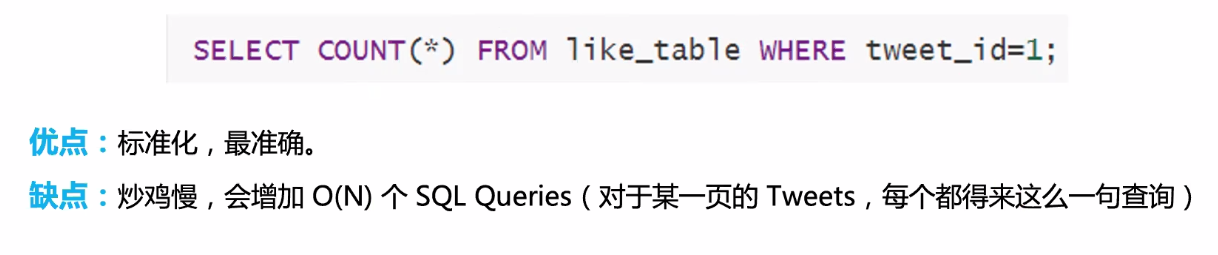
红包未领取超时后自动退回的任务是一个延迟任务(不是所有的Message Queue都支持延迟任务)。

冗余存储：



Views：有加1操作，只增不减

Likes：有增有减(可以取消赞)



冗余存储：在多个表单中都存储相同的字段，不从like\_table中查询点赞数，而是直接把点赞数存储在tweet\_table中。

冗余存储可能会带来一致性的问题，多个地方都存储了，如何保证数据的一致性。