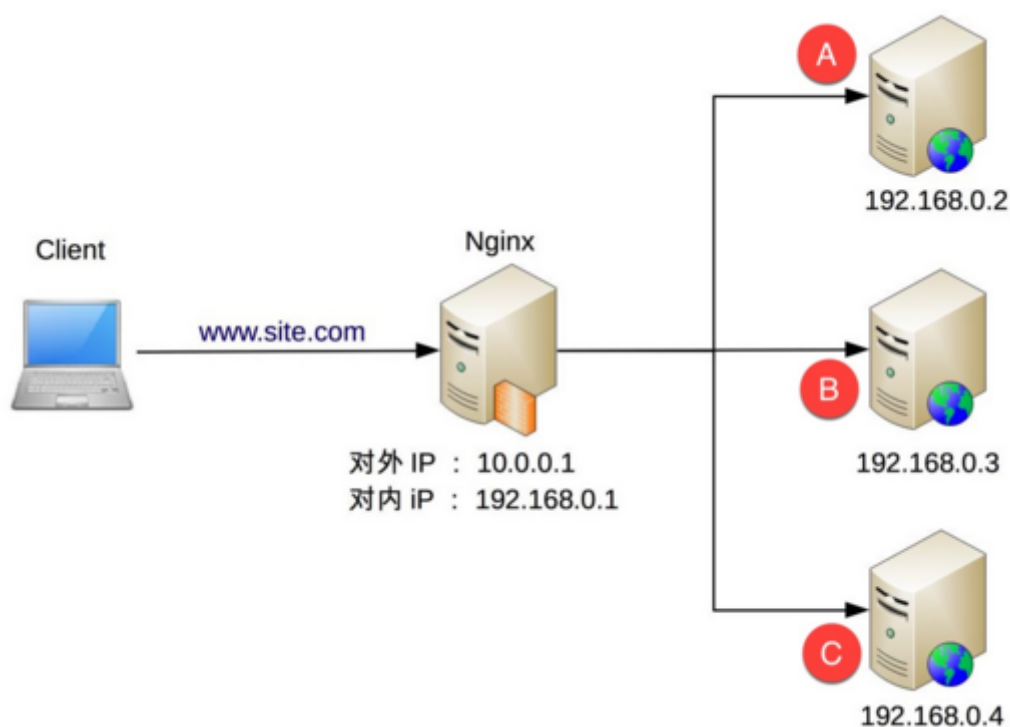


<https://hw1287789687.iteye.com/blog/2427055>（代码见链接）



场景描述:

秒杀系统提交订单时，由于用户连续快速点击，并且前端没有针对性处理，导致连续发送两次请求，一次命中服务器A，另一次命中服务器B，那么就生成了两个内容完全相同的订单，只是订单号不同而已。

重复提交的后果:

用户在界面看到两个一模一样的订单，不知道应该支付哪个
系统出现异常数据，影响正常的校验

解决思路：相同的请求在同一时间只能被处理一次

1. 分布式锁:

服务器A接收到请求之后，获取锁，获取成功。

服务器A进行业务处理，订单提交成功。

服务器B接收到相同的请求，获取锁，失败，

因为锁被服务器A获取了，并且未释放

服务器A处理完，释放锁，分布式锁的实现可以用redis

2. 利用数据库唯一索引：

实现思路：对请求信息进行hash运算，得到一个hash值

相同的请求信息得到相同的hash值(换成md5也可以)步骤：

接口A接收到请求之后，对请求信息hash运算，得到hash值hashCodeA

保存hashCodeA到数据库，并且对应的数据库的列满足unique约束

保存成功之后，才进行正常业务逻辑处理，比如提交订单

服务器B接收到相同的请求后，也得到相同的hash值，hashCodeA

服务器B保存hashCodeA到数据库，肯定失败，因为相同的hash值已经存在

因为保存失败，所以后面的业务逻辑不会执行