

# 秒杀系统介绍

讲师：欧阳修



秒杀系统班级答疑群、  
课程信息、福利等等  
欢迎来撩班主任小佳娃

## 为什么讲“秒杀系统”这套课程？

( High Availability Online Shopping System for Black Friday )

### 1. 从互联网业务来看

秒杀玩法在电商，直播带货业务中运用的非常广泛，越来越多的秒杀玩法不断出现，秒杀系统几乎是所有互联网公司的“标配”

### 2. 从技术难度和深度来看

秒杀系统需要面对瞬间**高并发大流量** (High Concurrency And Large Flow) 的场景，要求具备大并发、高性能系统的架构设计方法和关键技术



通过这套课程你能学到什么，对面试有什么帮助？

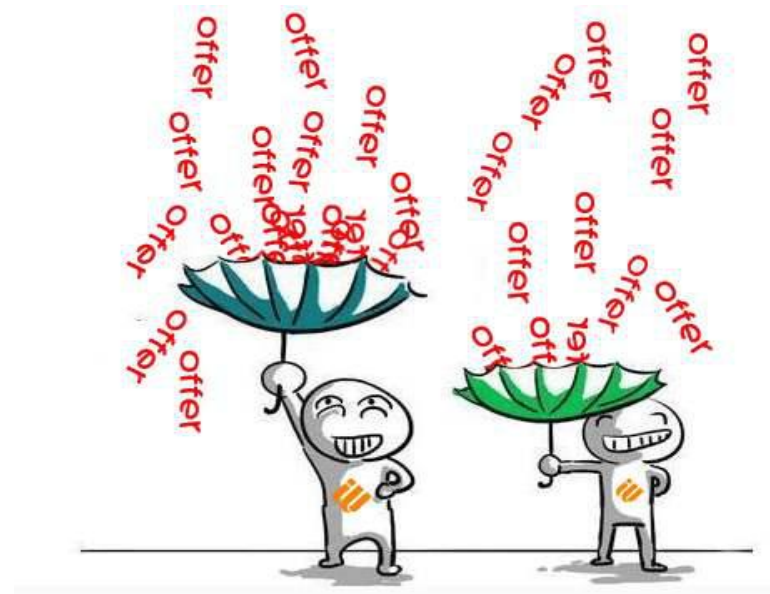
## 1. 面试

项目中涉及到的很多技术点，是面试一二线大厂重点考察的点，也是在大厂工作中必须掌握的技术门槛

## 2. 个人技术能力和系统设计能力的提升

掌握大并发、高性能的架构设计方法与原则，透彻理解秒杀系统的各个关键技术点

对秒杀系统类似系统能够举一反三，将学到的技术点运用到自己的项目中



# 课程安排及阶段目标



序号	类型与时长	内容	目标
第一周	直播（2小时）	课程介绍	基础功能的实现
	互动（2~3小时）	项目环境搭建与发布功能实现	
	互动（2~3小时）	实现秒杀系统库存模块	
	互动（2~3小时）	实现订单模块	
第二周	直播（2小时）	数据一致性处理	进阶功能优化
	互动（2~3小时）	系统优化项之页面技术	
	互动（2~3小时）	系统优化项之限流与稳定	
	互动（2~3小时）	分布式锁与缓存	
结课	直播（2小时）	秒杀系统架构理论与常见面试问题解答	

01

秒杀系统业务介绍

02

秒杀系统难点和需要解决的问题

03

设计思路分析

04

系统技术架构

05

后续教学内容介绍

06

课后作业

- 1 数据库: Mysql
- 2 持久层框架: Mybatis
- 3 基础框架: Springboot+SpringMVC
- 4 缓存中间件: Redis
- 5 消息中间件: RocketMQ
- 6 分布式唯一ID生成: 雪花算法
- 7 流量控制框架: Sentinel
- 8 压力测试工具: JMeter



## 数据库

Java 开发中常用的两种数据库：MySQL 、 Oracle

#	MySQL	Oracle
开源	它是一个开源、 <b>免费</b> 的数据库	一个 <b>收费</b> 的数据库

### MySQL

### 的优点

- 01** Mysql性能卓越，服务稳定，很少出现宕机
- 02** 历史悠久，社区用户非常活跃，遇到问题可以寻求帮助
- 03** 软件体积小，安装使用简单，并且易于维护
- 04** 品牌口碑效应，使得企业无需考虑就直接用之，LAMP LEMP流行架构
- 05** 支持多种操作系统，提供多个API接口，支持多个开发语言



## 消息中间件

RocketMQ / Kafka / RabbitMQ

#	RocketMQ	Kafka	RabbitMQ
性能对比	吞吐量在11.6w/s	吞吐量高达17.3w/s	吞吐量5.95w/s
消息存储	支持大量堆积	支持大量堆积	支持少量堆积
负载均衡	支持	支持	对负载均衡的支持不好
集群方式	支持	支持	支持
消息确认	支持	支持	支持
消息重试	支持	不支持	不支持
Topic数量	支持几百几千的Topic	Topic到几十上百个性能下降	。。。

各个不同消息队列的对比

<https://www.jianshu.com/p/251b76643d47>

## 1. Redis 实现分布式锁

Distributed Lock

## 2. 缓存预热

Cache Preheating

## 3. 数据库乐观锁

DB Optimistic Lock

## 4. 数据库索引优化

DB Index

Optimization

数据库 Database

## 1. 消息队列

Message Queue

## 2. 延迟队列

Delay Queue

## 3. 接口限流

Rate Limiting

## 4. 服务熔断

Circuit Breaker

中间件 Middleware

## 1. 商品限购

Purchase Limit

## 2. 雪花算法 ID 生成器

SnowFlake

## 3. 分布式事务处理

Distributed Transaction

## 4. 页面静态化

Static Page Technology

## 5. JMeter 压力测试

Stress Test

## 6. 防爬虫和黄牛请求

Anti-reptile

优化 Optimization

# 秒杀业务场景

我们在哪些地方会使用到秒杀业务？



# 100台iPhone 100万人抢

2020 年11 月11 日0 点开始, 京东自营限量 100 台, 以 4000 元的价格, 抢购 iPhone 12 64G 版本, 先到先得, 一人限购一台, 售完即止。





华为P40 5G手机 分期免息 亮黑色 全网通 (8+128G) 华为官方直供

🕒 京东秒杀

距离结束 22 : 49 : 55

秒杀价 **¥ 4458.00** [~~¥4788.00~~] 降价通知

累计评价  
1300+

促 销

增值业务 [高价回收, 极速到账](#) [套 3元1G](#)

配 送 至 [上海长宁区城区](#) [有货](#) [支持](#) [送运费险](#) [一年质保](#) [全国联保](#) [30天价保](#) [在线支付免运费](#) ?

由 福气多多手机旗舰店 负责发货, 并提供售后服务, 今日20:30前完成下单, 预计9月13日23:00前发货

选择颜色

亮黑色

冰霜银

零度白

深海蓝

晨曦金

选择版本

全网通 (8+128G) 华为官方直供

全网通 (8+256G) 华为官方直供

全网通 (6+128G) 华为官方直供

增值保障

[2年全保修 ¥239](#) [1年碎屏保修 ¥129](#) [1年延长保修 ¥28.18](#) ?

1

加入购物车

温馨提示 · 支持7天无理由退货(激活后不支持)



❤ 关注 [分享](#) [对比](#)

举报









蔬菜豆制品



肉禽蛋



水产海鲜



水果



乳品烘焙



速食冻品



酒饮零食



快手菜



网红打卡地



领30元红包

开通年卡会员，立享88元神券

立即开通 >

限时抢购 00 : 44 : 24

更多



煌上煌卤鸭掌 (小盒) 155g/盒

¥13.50  
¥15.80



味全健康大麦风味发酵乳 180g\*4杯

¥16.90  
¥20.90



禾煜薏米仁 400g袋

¥19.90  
¥22.80



0元领券

¥9.90  
¥12.80



限时抢购



限时抢

限量抢 NEW

距离结束  
00:44:01

16:00  
即将开始

19:00  
即将开始

明日07:00  
即将开始



煌上煌卤鸭掌 (小盒)  
155g/盒

限时  
¥15.80  
¥13.50



味全健康大麦风味发酵乳  
180g\*4杯

限时  
¥20.90  
¥16.90



八喜香草口味冰淇淋 90g/盒

限时



秘橘约500g

限时 任选2件88折

版权



首页



分类



吃什么



购物车



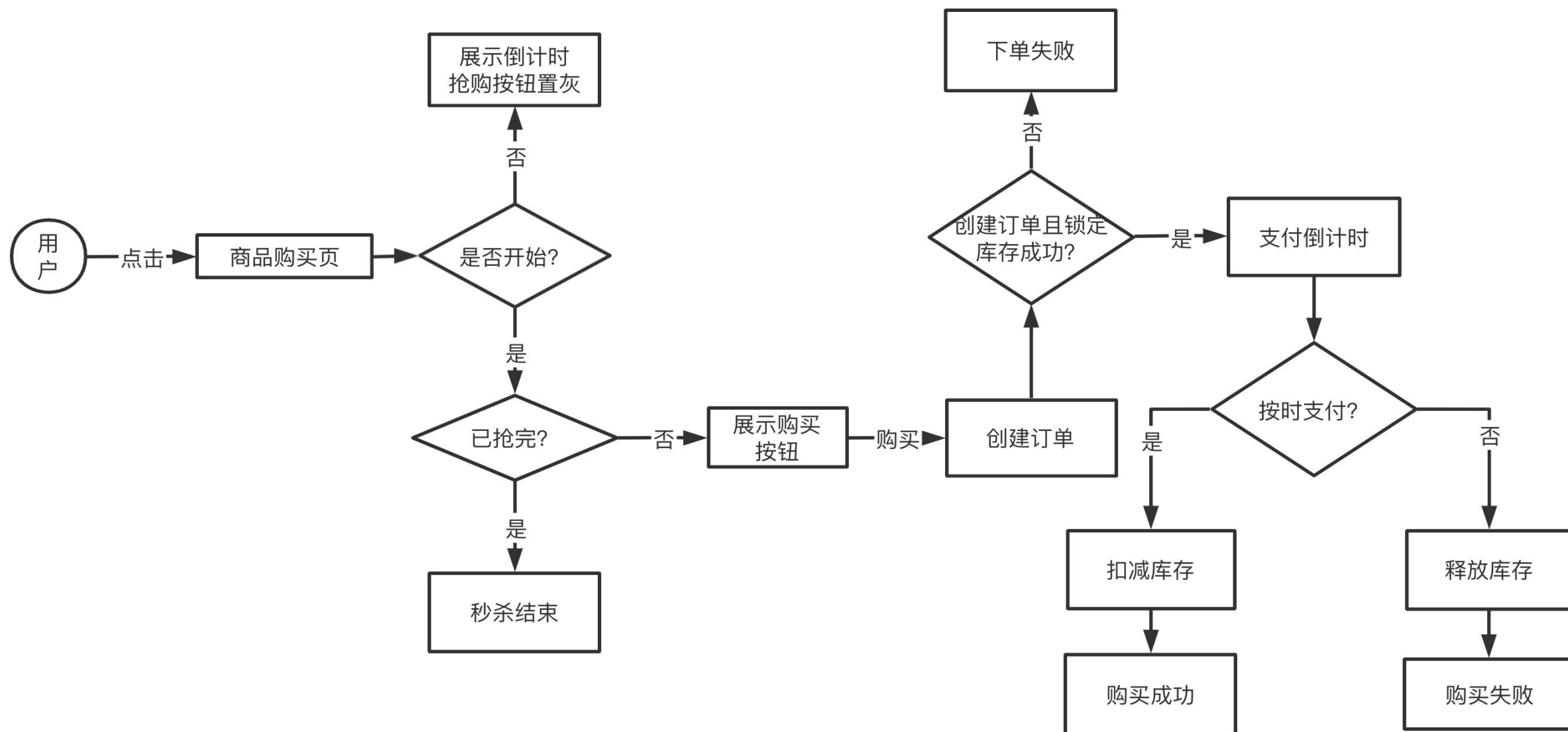
我的

传播盗版将被追究刑

# 秒杀业务流程与特点

秒杀业务有哪些流程？ 哪些特点？ 应该怎样针对性处理？









## 秒杀业务具有哪些特点？

1. 瞬时并发量大：大量用户会在同一时间抢购，网站流量瞬间激增。
2. 库存少：一般都是低价限量，而访问的数量远远大于库存数量，只有极少数人成功。
3. 业务流程简单：流程短，立即购买，下订单，减库存。
4. 前期预热：对于还未开启活动的秒杀商品，以倒计时的方式显示，只能访问不能下单。

# 重难点与面试问点

项目由哪些难点？怎么去理解？面试中会怎么考察？



## 1. 瞬时大流量高并发

服务器、数据库等能承载的 QPS 有限，如数据库一般是单机 1000 QPS。需要根据业务预估并发量。

## 2. 有限库存，不能超卖

库存是有限的，需要精准地保证，就是卖掉了  $N$  个商品。不能超卖，当然也不能少卖了。

## 3. 稳定 高可用保证

系统在高并发量的请求下要保证系统不会被打挂，不能影响其他业务



**问题都列出来了，那怎么设计？怎么解决这些问题就是接下去要考虑的了，我们对症下药，逐一击破！**

# 如何应对瞬时大流量高并发？

难点与面试问点一



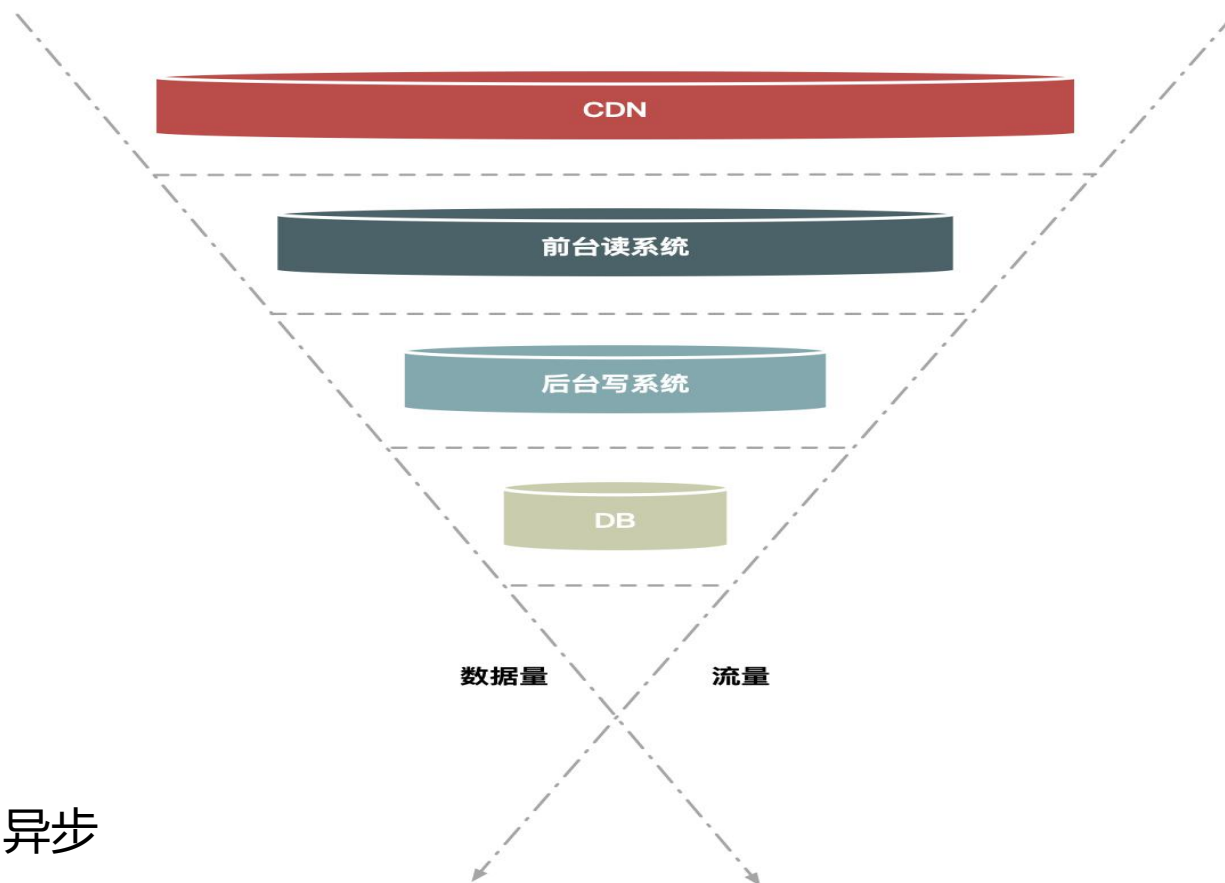
核心思想：分层过滤，分而治之

在不同的层次尽可能地过滤掉无效请求，让“漏斗”最末端的才是有效请求。

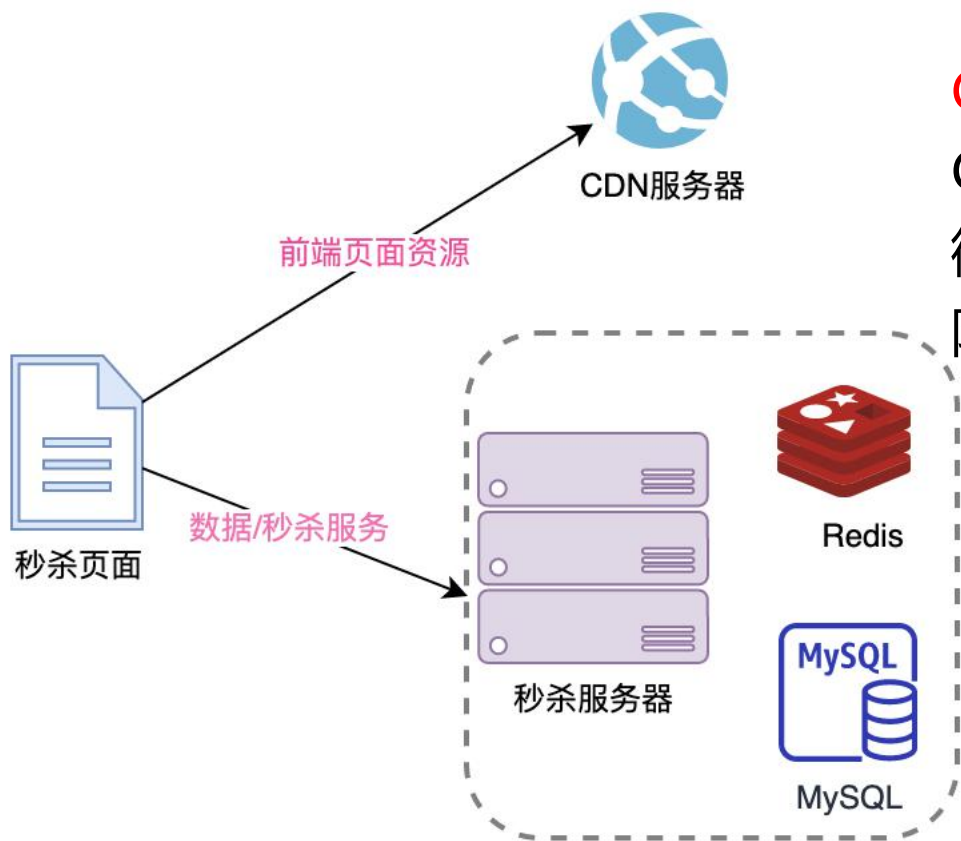
具体方法：

1. 页面静态化 (Static Page Technology)
2. 缓存预热 ( Cache Warm-up )
3. 异步 (Asynchronous) 化
4. 削峰值填谷 (Peak Load Shifting) ，通过消息队列异步

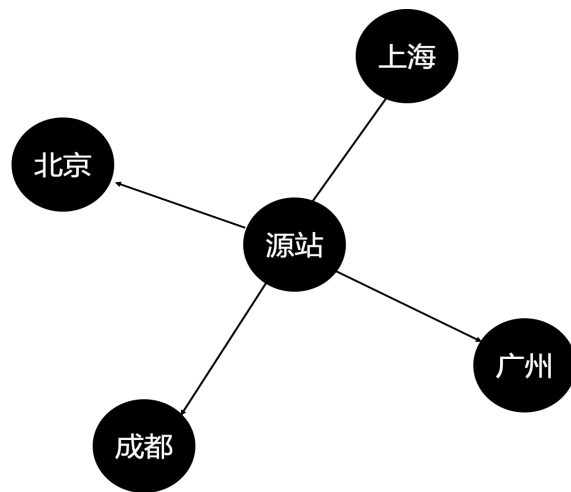
地创建订单



秒杀页面由商品信息和前端页面资源组成，前后端分离，页面资源不会经过后端服务器，将前端资源，放入CDN服务器中



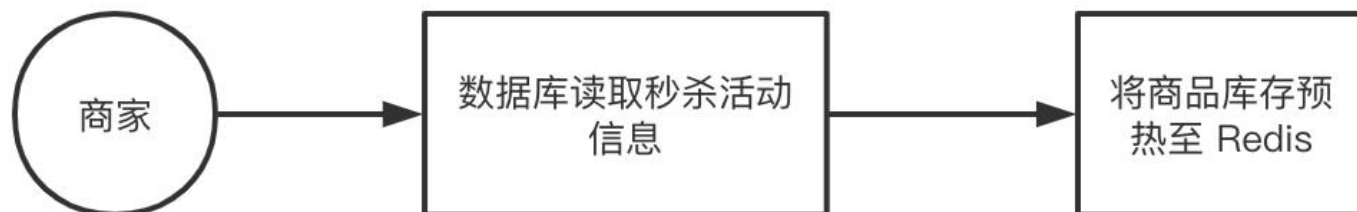
**CDN** 的全称是 Content Delivery Network，即内容分发网络。CDN 是依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率



将部分业务逻辑写到缓存里，不需要直接读数据库，来减少数据库服务器的压力，这样访问速度会更快

例如：  
在秒杀活动开始前，提前将设置好秒杀的商品信息、限购数量，库存数量等写入Redis中

## 活动开始前



## 消息队列特点

1. 基于生产者 (Producer) / 消费者 (Consumer) 模型的组件。
2. 用于实现两个不同的系统之间的解耦和异步 (Asynchronous) 操作。
3. 生产者可以高速地向消息队列中投递 (生产) 消息。
4. 消费者可以按照自己的节奏去消费生产者投递的消息。
5. 消息队列一般带有重试的能力。可以持续投递，直到消费者消费成功。





如果秒杀数量是1万台，或者10万台呢？

因为 Redis 和 MySQL 处理能力的巨大差异。实际下沉到 MySQL 的量还是巨大，MySQL 无法承受。  
一瞬间大量的抢购成功，创建订单请求，对创建订单服务压力过大，会去操作mysql数据库  
无法及时处理创建订单请求，导致系统故障。

解决思路

可不可以先在通过 Redis 扣库存后，到 MySQL 的请求慢一点？

解决方案

异步化思想，通过消息中间件来削峰填谷，将请求先发往消息队列中，订单服务端根据自己能力再去消费并创建订单



# 有限库存，如何防止超卖？

难点与面试问点二

读取库存表，判断库存，然后扣减库存

## 1. 查询库存余量

```
SELECT available_stock FROM `seckill_activity`  
WHERE commodity_id = 189 AND id = 28;
```

## 2. 扣减库存

```
UPDATE `seckill_activity` SET available_stock = available_stock - 1  
WHERE commodity_id = 189 AND id = 28 AND available_stock > 0;
```



秒杀瞬间大流量并发请求数据库，会出现什么问题？



## 1.问题分析

秒杀的本质，就是对库存的抢夺

每个秒杀的用户来都去数据库查询库存校验库存，然后扣减库存，导致数据库崩溃。

## 2.MySQL和Redis的性能比较

MySQL 数据库单点能支撑 1000 QPS，但是 Redis 单点能支撑 10万 QPS

将库存信息加载到Redis中，将MySQL的访问压力转移到Redis上，直接通过 Redis 来判断并扣减库存



# 系统设计-Redis简介

Redis是一个高性能的key-value非关系型数据库（NoSQL 的一种），基于C语言编写的开发。

特点：

1. **性能极高**，基于内存运行-Redis单服务器 能读的速度是**110000次/s**，写的速度是**81000次/s**
2. Redis支持**数据的持久化**，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
3. **单线程运行**，通过IO多路复用实现并发。
4. 支持多种数据非关系型的数据结构：
  1. 字符串/数字（STRING）
  2. 哈希表（HASH）
  3. 链表（LIST）
  4. 集合（SET）
  5. 有序集合（ZSET）
5. **所有单个指令操作都是原子的**，即要么完全执行成功，要么完全执行失败。同时还支持**多个指令**也可以通过 Lua 脚本事务操作实现原子性。
6. 支持分布式，集群部署



## Redis 的应用场景

1. 缓存系统（“热点”数据：高频读、低频写）
2. 计数器
3. 消息队列系统
4. 排行榜
5. 分布式锁
6. 社交网络和实时系统（微博）

## 哪些大厂在使用Redis?

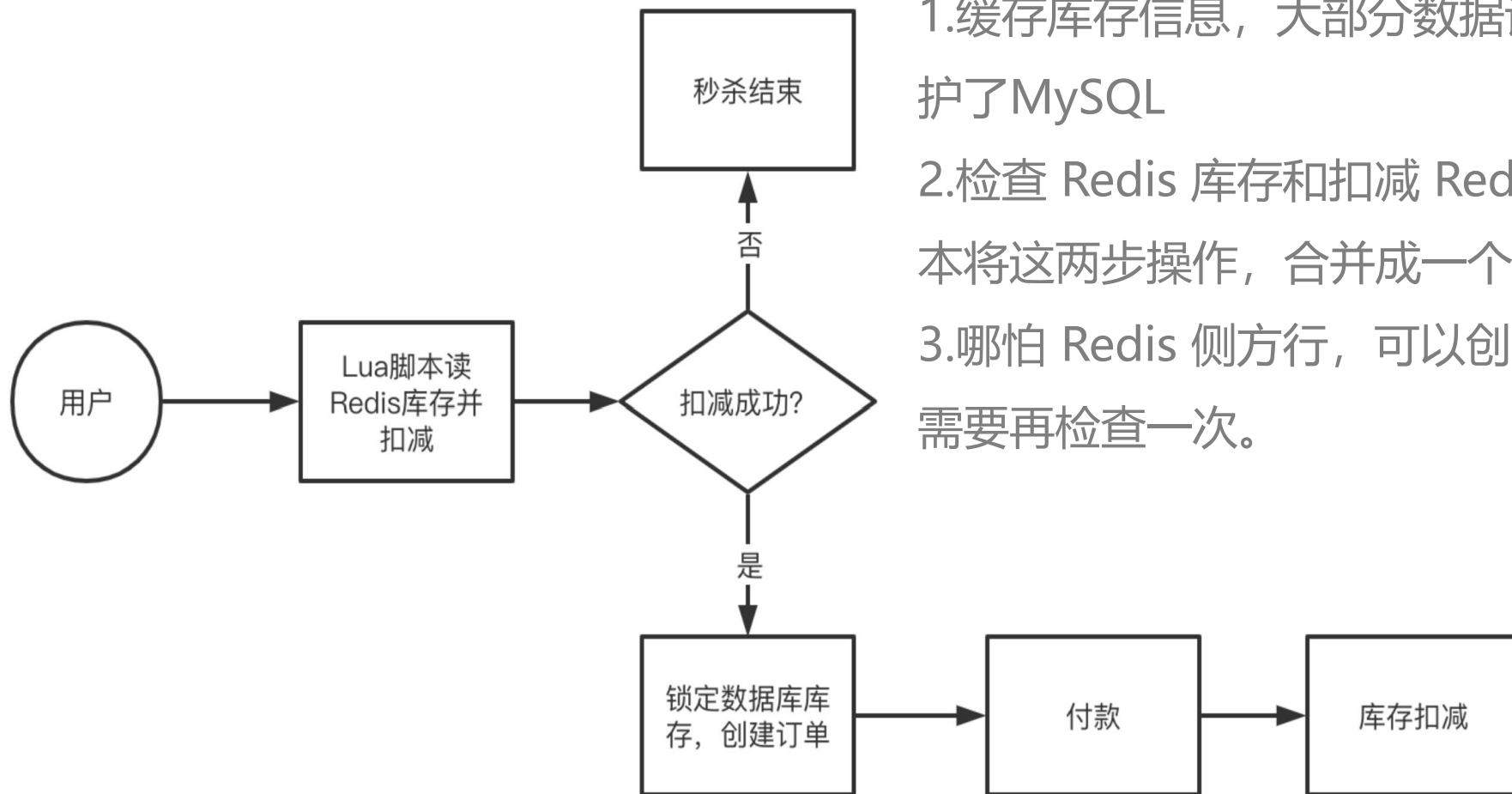
Twitter 微博 阿里巴巴 百度 美团 拼多多 等

Redis如今是互联网项目的标配，在面试中非常高频被问到



## 解决方案:

- 1.缓存库存信息，大部分数据读取请求都被 Redis 挡住了，保护了MySQL
- 2.检查 Redis 库存和扣减 Redis 库存是两步操作,通过Lua脚本将这两步操作，合并成一个整体，保证原子操作性
- 3.哪怕 Redis 侧方行，可以创建订单了，到 MySQL 的时候也需要再检查一次。



## 通过 Lua 脚本执行原子操作

Lua 脚本功能是 Redis 在 2.6 版本中推出，通过内嵌对 Lua 环境的支持，Redis 解决了长久以来不能高效地处理 CAS（Compare And Swap）命令的缺点并且可以通过组合使用多个命令，轻松实现以前很难实现或者不能高效实现的模式。

Lua 脚本是类似 Redis 事务，有一定的原子性，不会被其他命令插队，可以完成一些 Redis 事务性的操作。

```
if (redis.call('exists',KEYS[1]) == 1) then
    local stock = tonumber(redis.call('get', KEYS[1]));
    if( stock <=0 ) then
        return -1
    end;
    redis.call('decr',KEYS[1]);
    return stock - 1;
end;
return -1;
```



# 如何保障系统稳定和高可用?

难点与面试问点三



问题1：当秒杀的用户量超过预计，请求量超过服务器最大承载压力怎么办？

问题2：当有服务出现故障，不可用时如何应对？

## 系统保护措施

- 1.限流 (Ratelimit) , 流量控制 (flow control)
- 2.服务熔断 (Circuit-breaker)

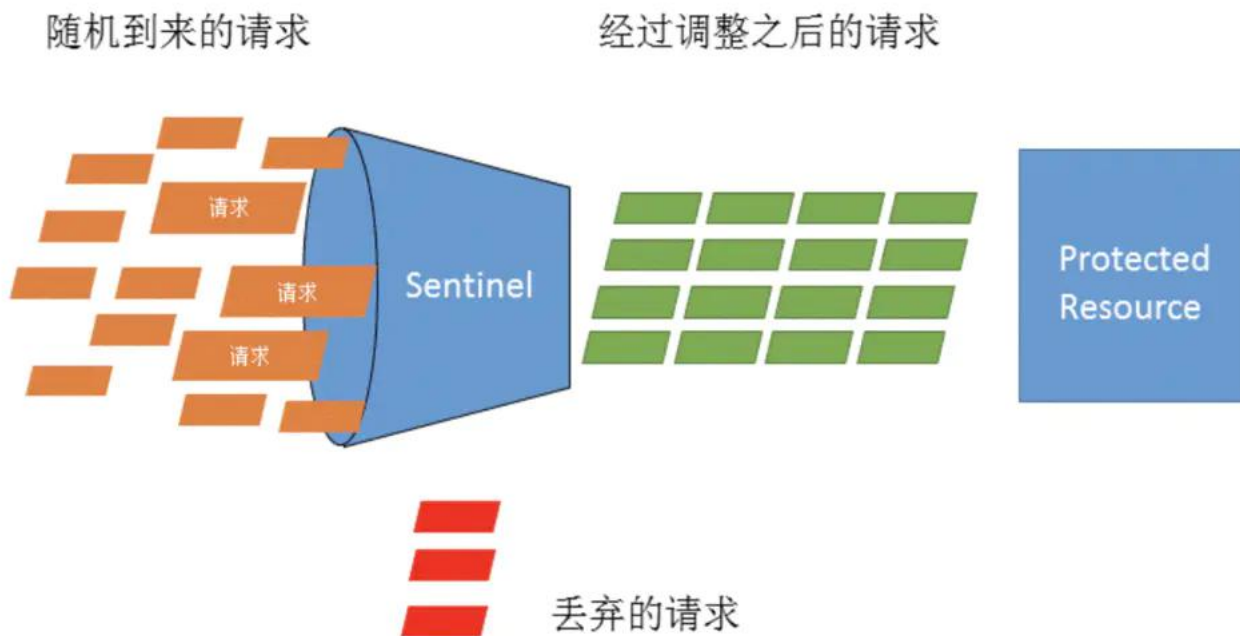
阿里巴巴开源框架 Sentinel,  
Sentinel 是面向分布式服务架构的轻量级流量控制框架



# Sentinel

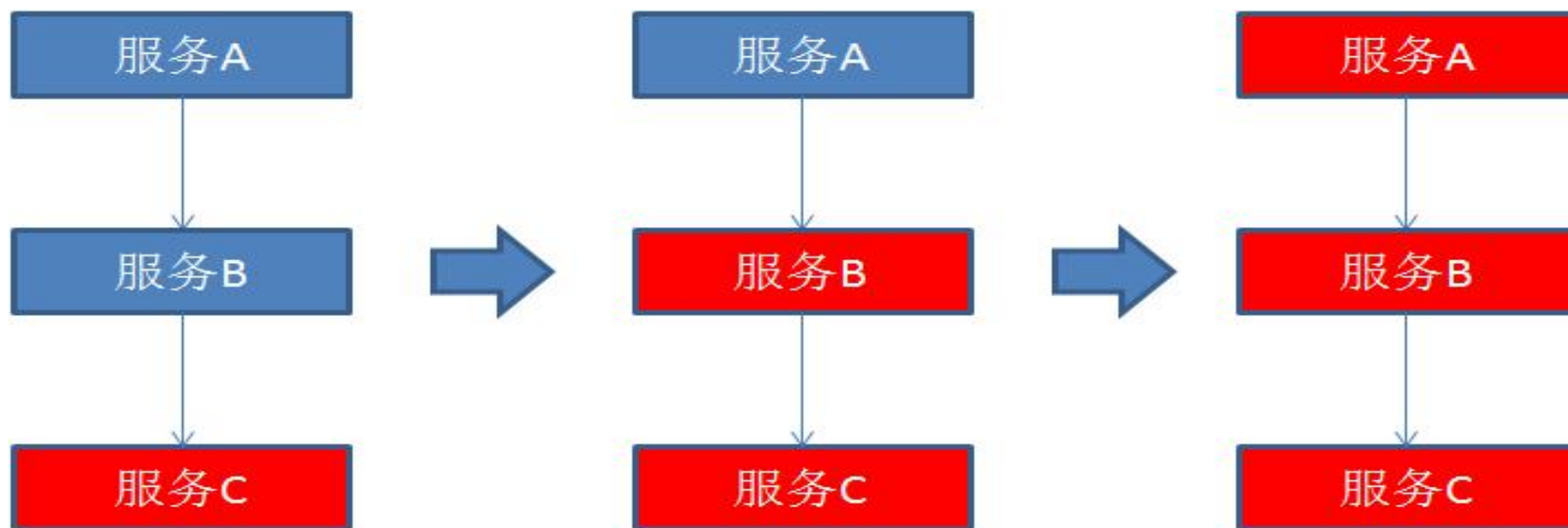
对于秒杀系统来说，秒杀时请求瞬间的秒杀流量是不可控的，没办法准确预估多少，但是秒杀系统的处理能力是有限的，当流量过大时服务器会被打挂

**流量控制** (flow control)，其原理是监控应用流量的 QPS 或并发线程数等指标，当达到指定的阈值时对流量进行控制，以避免被瞬时的流量高峰冲垮，保护系统不会被压垮，从而保障应用的高可用性。



一旦下游服务C因某些原因变得不可用，积压了大量请求，服务B的请求线程也随之阻塞。线程资源逐渐耗尽，使得服务B也变得不可用。紧接着，服务A也变为不可用，整个调用链路被拖垮。

像这种调用链路的连锁故障，这种不可用像滚雪球一样逐渐放大的时候，叫做**雪崩 (Avalanche) 效应**。



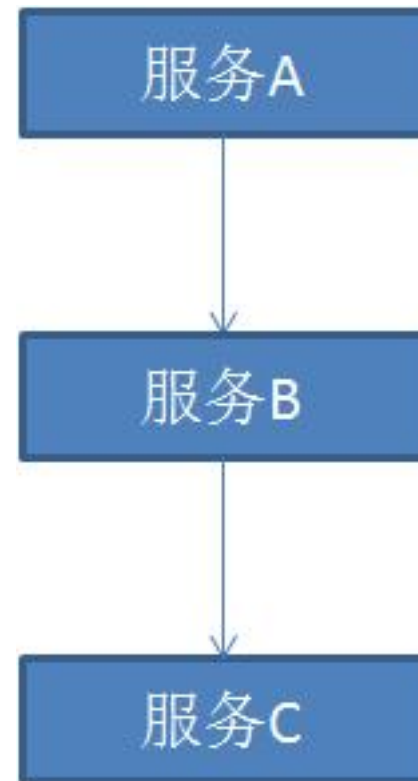
## 什么是服务熔断?

熔断这一概念来源于电子工程中的**断路器** (Circuit Breaker)。在互联网系统中,当下游服务因访问压力过大而响应变慢或失败,上游服务为了保护系统整体的可用性,可以暂时切断对下游服务的调用。

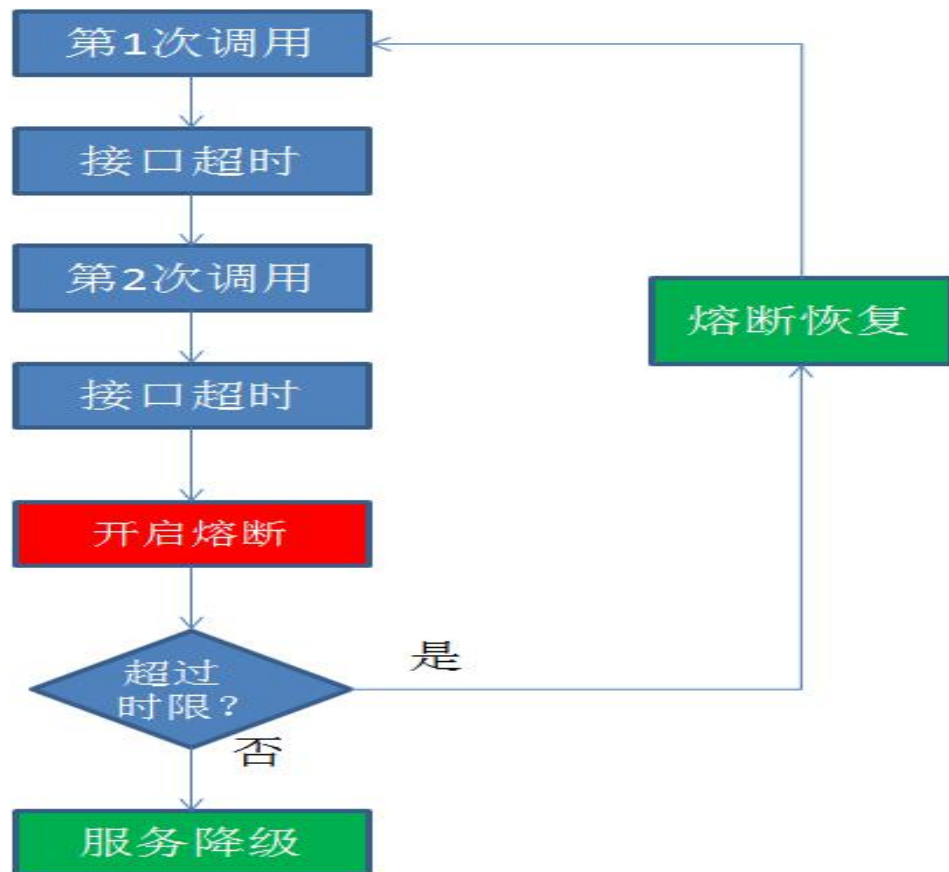
这种牺牲局部,保全整体的措施就叫做**熔断**

如果不采取熔断措施,我们的系统会怎样呢? 我们来看一个例子

当前系统中有A, B, C三个服务,服务A是上游,服务B是中游,服务C是下游

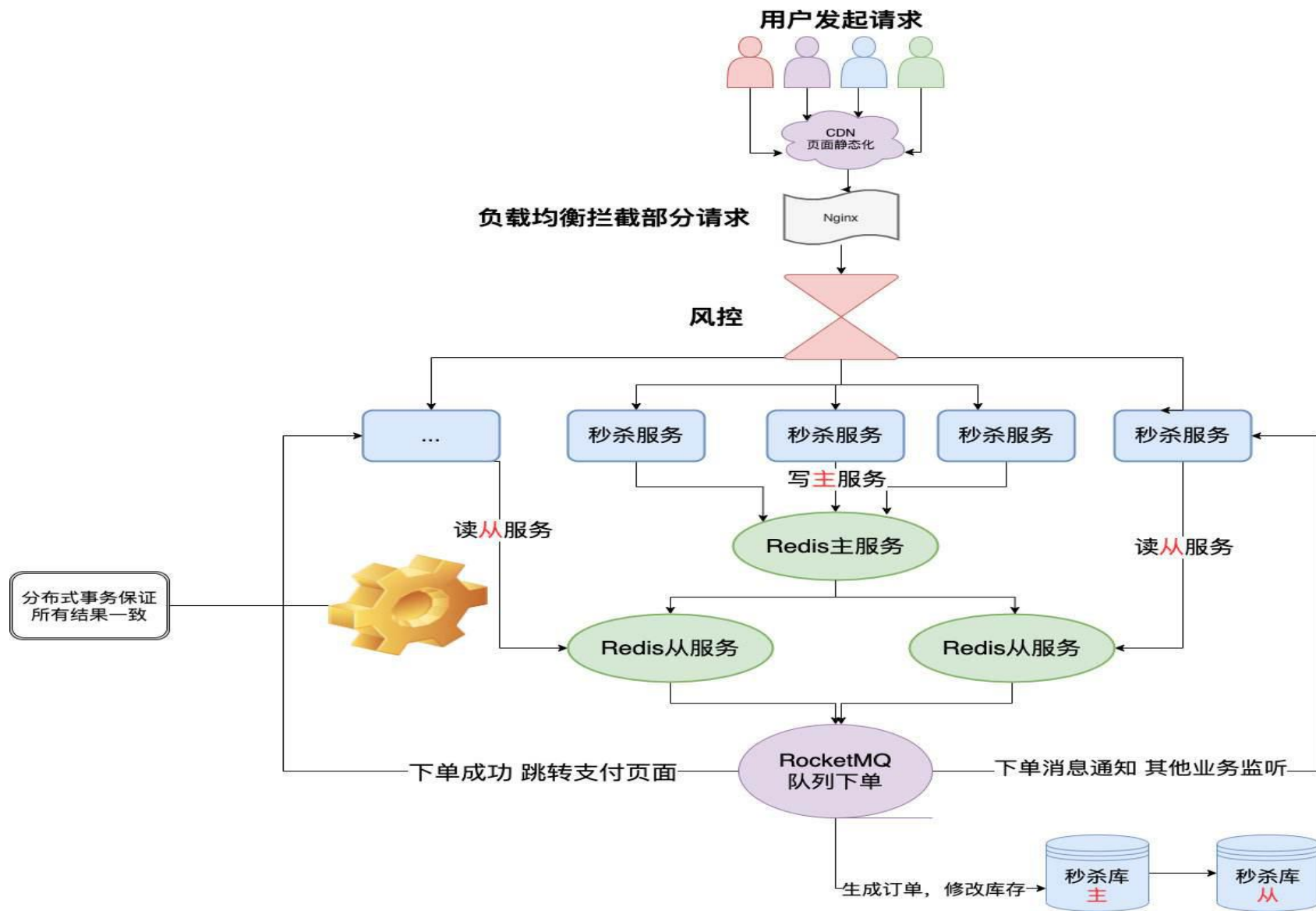


熔断器，如同电力过载保护器，在一段时间内侦测到许多类似的错误，就会强迫其以后的调用该服务的**快速失败**，不再访问该服务，从而防止应用程序不断地尝试执行可能会失败的操作。当检测到该节点微服务响应正常后恢复调用链路。



# 系统设计原则







## 1. 秒杀系统业务介绍

## 2. 秒杀系统难点和需要解决的问题

如何应对瞬时大流量高并发？

有限库存，如何防止超卖？

如何保障系统稳定和高可用？

如何限制用户购买商品件数？

如何应对恶意请求和爬虫？

## 3. 设计思路分析

## 4. 系统技术架构和技术点介绍

## 5. 后续课程安排



希望借助“秒杀”这个互联网高并发场景中的典型代表，带你了解如何打造一个超大流量并发读写、高性能，以及高可用的系统架构设计方法！

序号	类型与时长	内容	目标
第一周	直播（2小时）	课程介绍	基础功能的实现
	互动（2~3小时）	项目环境搭建与发布功能实现	
	互动（2~3小时）	实现秒杀系统库存模块	
	互动（2~3小时）	实现订单模块	
第二周	直播（2小时）	数据一致性处理	进阶功能优化
	互动（2~3小时）	系统优化项之页面技术	
	互动（2~3小时）	系统优化项之限流与稳定	
	互动（2~3小时）	分布式锁与缓存	
结课	直播（2小时）	秒杀系统架构理论与常见面试问题解答	



# 第一周课程安排预览

我们要做哪些事？



## 第二章【互动】项目环境搭建与发布功能实现

1. 项目基础架构搭建

2. 数据库表分析及创建

3. 发布秒杀活动功能的开发

4. 秒杀活动列表和秒杀活动详情页开发

1 MySQL 数据库

2 Spring Boot

3 MyBatis 流行的 ORM 框架

4 MyBatis 逆向生成配置



## 第三章【互动】实现秒杀系统库存模块

1. 秒杀库存超卖问题

2. 库存超卖问题与多种解决方案分析

3. Jmeter 并发请求测试

4. 使用 Lua 脚本解决库存超卖问题

- 1 MySQL
- 2 Spring Boot
- 3 MyBatis
- 4 Redis
- 5 Lua 脚本（了解）
- 6 Jmeter 压测工具



## 第四章【互动】实现订单模块

1. 为什么要进行流量削峰?

2. 利用消息中间件进行流量削峰

3. 订单生成方案与数据库设计

4. 分布式 ID 解决方案

5. SpringBoot 整合 RocketMQ

6. 订单处理与消息队列的结合

1 MySQL

2 Spring Boot

3 MyBatis

4 Redis

5 雪花算法

6 Jmeter 压测工具

7 RocketMQ 消息中间件



# 简历怎么写

秒杀系统

( High Availability Online Shopping System for Black Friday )



## 简历突出亮点

1

掌握 SpringBoot、MyBatis等主流开发框架

2

熟悉 Mysql 的使用

3

掌握 RocketMQ、Redis、Sentinel 等主流中间件的使用

4

有高并发，大流量场景处理经验

5

具备分布式系统开发能力，有分布式系统开发经验

6

具备高可用架构设计能力





## 主要负责

1

处理秒杀场景下高并发大流量

2

针对秒杀场景流程优化

3

使用缓存Redis中间件，消息中间件RocketMQ提升系统性能

4

充分考虑和保障了系统的稳定性



# 课后题一

与秒杀系统类似的系统有哪些？这些系统有什么难点？怎么去解决这些难题？



# 课后题二

预习后面课程需要用到的 Redis 和 RocketMQ



# 思考题一

想一想本课程中涉及的技术点，哪些能运用到你开发过接触的项目中去？



# Q & A

自由问答时间



扫码进群，享课件，学习礼包等多种福利~



秒杀电商项目实践12.20



该二维码7天内(12月21日前)有效，重新进入将更新

有任何问题都可以咨询圆圆 (jiuzhang11) 哦~

