极客时间 架构实战营 模块九作业

G20210607050173 李天水

1 电商秒杀系统设计分析

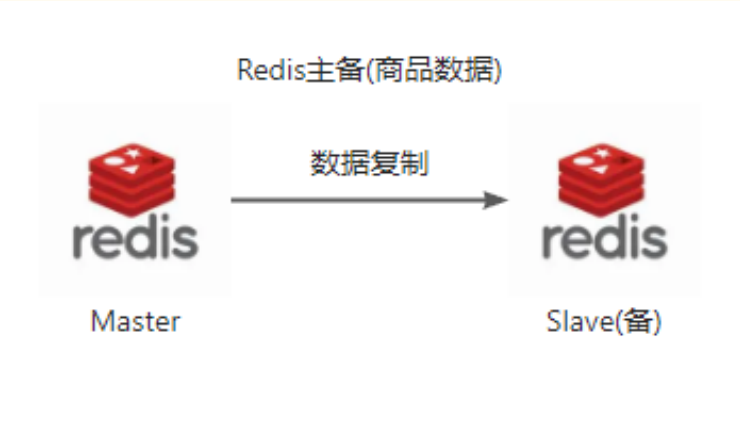
- 业务背景
  - 10个品类，每个品类不超过20个商品；
  - 1000个充电宝、10台iPhone12作为秒杀商品；
  - 正常日活100万用户。
- 技术背景
  - 技术团队以Java为主，已经落地微服务架构；
  - 主要渠道是自有的App(iOS和Android)和微信小程序；
  - 只有下载App才能参加秒杀活动。
- 设计目标
  - 万无一失。

2 复杂度分析

- 业务复杂度分析
  - 注册、登录、浏览商品、秒杀、下单。
- 质量复杂读分析
  - 浏览商品
    - 按照每个品类有20个热销商品，10个品类即为总计200个热销商品，假设每个商品有20万的潜在购买用户，每位潜在用户在当天会浏览该商品10次，80%的请求会集中在4个小时内；
    - 浏览商品的QPS为：20*10*20w*10*80%/(4*3600)≈2w/s。
  - 秒杀
    - 秒杀商品共有2种，假设分别有20万用户参与了这2种商品的秒杀活动，每位用户在秒杀前1分钟内刷新了30次商品信息；
    - 秒杀商品的QPS为：2*20w*30/60=20w/s。
  - 下单
    - 非秒杀商品
      - 非秒杀商品的浏览商品QPS为2w，考虑到实际浏览完会下单数量会远小于2w/s，所以在这里不考虑下单带来的请求压力。
    - 秒杀商品
      - 秒杀商品的数量一共为1010个，请求峰值最多为1k/s，所以同样不需要考虑。

3 存储架构设计

- MySQL主备
  - 存储用户数据、商品数据、订单数据。
- Redis主备
  - 缓存热销商品、秒杀商品数据。

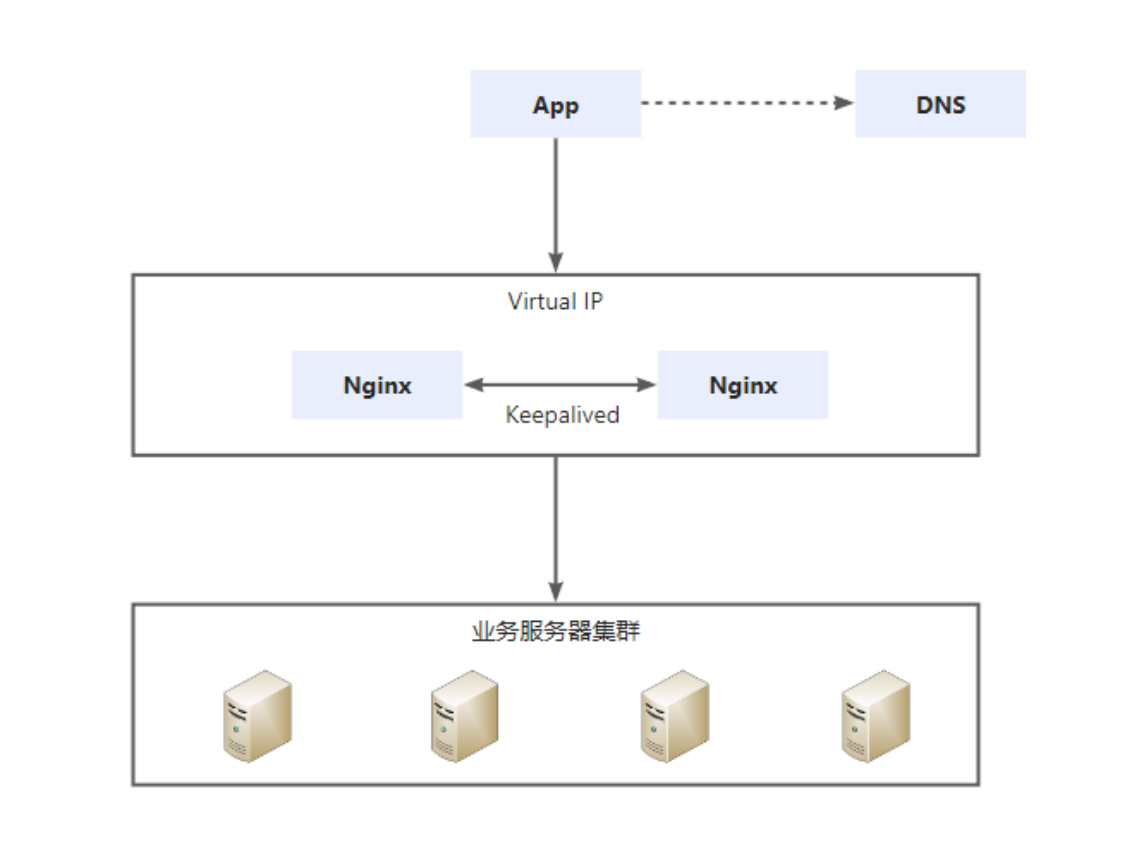


4 计算架构设计之性能估算

- 注册
  - 考虑到会有一部分原先通过第三方平台登录的老用户无需注册，新用户增长应该不会超过20万，平均到活动的那几天，注册TPS会很低，无需考虑。
- 登录
  - 活动当天用户可能会是平时日活人数的3倍，即为300万人，假设80%的用户登陆时间集中在晚上四个小时，300w*80%/4/3600≈200/s，请求压力较小，无需考虑。
- 浏览商品
  - 同前面复杂度分析描述一样，QPS约为2w/s。
- 秒杀
  - 同前面复杂度分析描述一样，QPS约为20w/s。
- 下单
  - 同前面复杂度分析描述一样，请求压力较小，无需考虑。

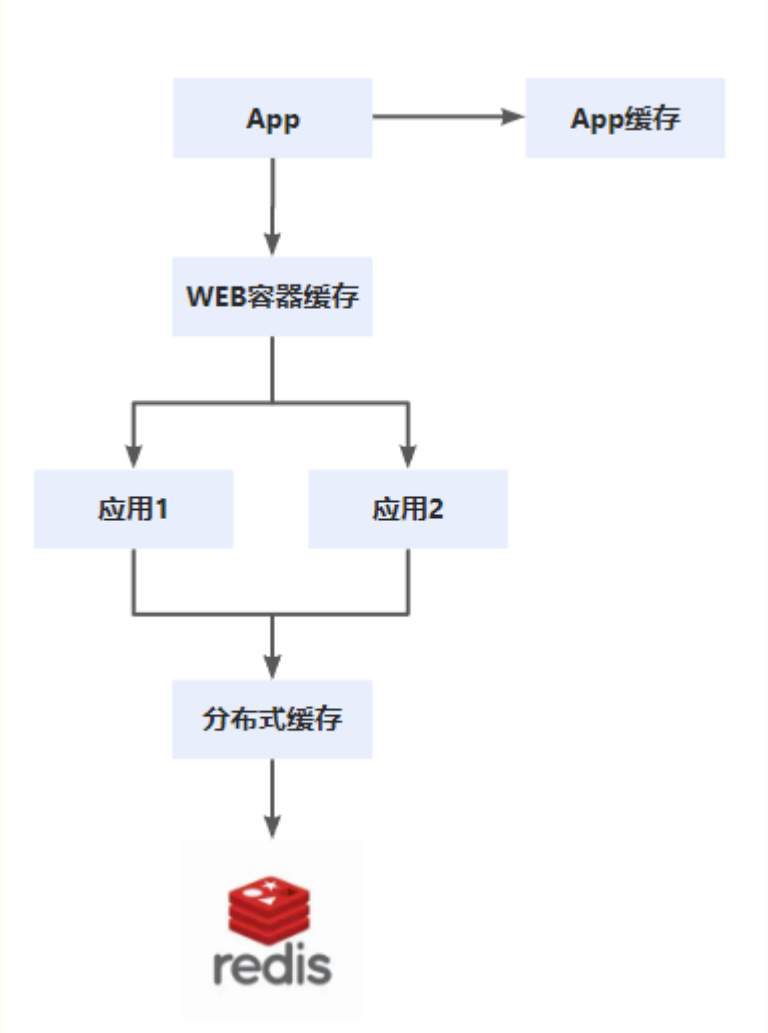
5 计算架构之负载均衡

- 利用Keepalived提供主备切换的高可用能力。



6 计算架构之缓存架构

- 采用App缓存、WEB容器缓存、分布式缓存。



7 可扩展架构设计

- 技术背景中阐述了项目已经落地了微服务架构，而且秒杀只有在活动期间才有，所以可以在原有的微服务架构基础上将秒杀独立成新的微服务；
- 同时为了保证秒杀场景下的高并发不影响其他服务，可以考虑为秒杀服务分配独立的计算资源和存储资源。

8 高可用架构设计

- App端
  - 随机限流，按概率取消对服务端发送请求的操作；
  - 在App端限制用户的点击次数。
- 服务端
  - 采用漏桶算法进行限流；
  - 丢弃无法处理的请求；
  - 为秒杀服务分配独立的计算资源与存储资源；
  - 延长秒杀、热销商品数据的缓存有效时间。
- 同城双数据(同城灾备)
  - MySQL主备跨机房复制。
- 异地多活
  - 秒杀时间非常短暂，所以不需要考虑。