

竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本

产品名称	竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本
公司名称	东莞市数云网络科技有限公司
价格	.00/件
规格参数	东莞软件开发:小程序开发 东莞软件开发:定制开发 东莞软件开发:商业模式开发
公司地址	东莞市
联系电话	18665158422 18665158422

产品详情

1、竞拍场次时间是为每天固定的三个时间，定时提前写入并设置过期时间。竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本

2、竞拍场次时间是为每天固定的三个时间，缓存数据结构设计有两个版本：

a、第一个版本的数据结构设计在商品列表查询时，无法排除自身商品信息并且分页。

不同区域可秒杀的用户set (判断用户所属竞拍区)

key: prefix + area_id + start + end + auctions_id,value:uid

不同区域下的商品信息zset (可支持分页)

key: prefix + area_id + start + end + auctions_id,score:goods_id,member:goods_detail

库存字面量

key: randomvalue:1

是否购买占位

key: prefix + area_id + start + endvalue:1

为了满足排除自身的商品功能和分页，思考了一些实现方案：

- (1) 完全放弃从缓存中获取竞拍商品信息，这样增加数据库压力，同时无法使用竞拍随机码。
- (2) 为每个用户单独存放一个排除自身商品信息的集合，这样会存放重复数据造成增加内容空间。
- (3) 查询到redis有一个SCAN命令来迭代获取数据，并可利用glob模式匹配，但是获取数量无法确定而无法分页。

以上(1)(3)点都被排除，我们从第(2)出发重新设计第二版数据结构，单独存放商品数据和用户可查询的商品id集合来减少重复，但又会出现keys过多的情况，需要进行优化。

b.第二个版本的数据结构设计

用户可查询的商品id的zset (判断用户是否有可竞拍商品)

key: prefix + area_id + users_id + auction_id + start + end, score: goods_id, member: goods_id

商品信息string (可支持分页)

key: prefix + area_id + auction_id + goods_id + start + end, value: goods

库存字面量

key: randomvalue:1

是否购买占位

key: prefix + area_id + start + end