为什么要单独获取系统时间？

1. 详情页



如果秒杀还未开始，用户会大量的刷新页面。系统应该是将详情页部署到CDN上，CDN会对详情页做静态化处理。所以，用户访问的静态页的html不再系统上，在CDN的节点上。同时，包括静态资源也会部署在CDN上。这样我们就拿不到系统时间了，所以，我们需要单独的一个请求去获取系统时间。

|  |
| --- |
| CDN的理解 |
| CDN(内容分发网络)加速用户获取数据的系统；  CDN的部署在离用户最近的网络节点上；  命中CDN后不需要访问后端服务器；  互联网公司自己搭建或租用； |

那么获取系统时间需要优化？

获取系统时间不需要优化。因为访问一次内存大约10ns。而且获取系统时间的请求，也没有任何与后端的交互，不考虑gc的影响，1s中可以执行1亿次，因此不需要优化。

获取秒杀接口分析



|  |
| --- |
| 秒杀地址接口优化 |
|  |
| 请求地址会先去redis里面找，如果有，就不用跟数据库打交道。  一致性维护：   1. 超时穿透：设置有效时间，如果超过则直接从数据库中获取数据 2. 主动更新：当Mysql更新时，主动更新redis |

秒杀操作优化分析



|  |
| --- |
| 1、秒杀操作无法使用CDN缓存，因为CDN缓存一般都是一些静态文件；  2、秒杀操作使用后端缓存会出现库存问题，因为秒杀成功会减库存，这种事  不能再redis中很好的得到体现；  3、存在热点商品被疯狂点击，然后Mysql出现不停的update操作； |
| 其他方案分析：     1. 系统做一个原子计数器，里面放的是商品的库存，秒杀成功后，减库存(原子计数器)，保证原子性； 2. 当减原子计数器成功后，记录一个行为(谁去减了库存)，作为一个消息放到分布式MQ中； 3. 后端服务消费消息并落地，一般落地在MySQL中。 |
| 为什么不用Mysql解决？      优化分析：  行级锁在Commit之后释放，优化方向减少行级锁持有时间。  延迟分析：  网络延迟，jvm的GC回收。   |  | | --- | |  | |

优化总结：

1. 前端控制:暴露接口，按钮防重复；
2. 动静态数据分离：CDN缓存，后端缓存；
3. 事务竞争优化：减少事务锁时间。