php结合redis实现高并发下的抢购、秒杀功能

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：https://www.wtabcd.cn/fanwen/zuowen/e060b31537bb792132cc6ebd204def90.html

抢购、秒杀是平常很常见的场景，面试的时候面试官也经常会问到，比如问你淘宝中的抢购秒杀是怎么实现的等等。

抢购、秒杀实现很简单，但是有些问题需要解决，主要针对两个问题：

1 高并发对数据库产生的压力

2 竞争状态下如何解决库存的正确减少（”超卖”问题）

第一个问题，对于php来说很简单，用缓存技术就可以缓解数据库压力，比如memcache，redis等缓存技术。

第二个问题就比较复杂点：

常规写法：

查询出对应商品的库存，看是否大于0，然后执行生成订单等操作，但是在判断库存是否大于0处，如果在高并发下就会有问题，导致库存量出现负数

```
<?php$conn=mysql_connect("localhost","big","123456"); if(!$conn){    echo "conne教资证c
t failed";    exit; } mysql_select_db("big",$conn); mysql_query("set names utf8"); $pr
ice=10;$user_id=1;$goods_id=1;$sku_id=11;$number=1; //生成唯一订单function build_order_no(){
  return date('ymd').substr(implode(null, array_map('ord', str_split(substr(uniqid(), 7, 13),
1))), 0, 8);}//记录日志function insertlog($event,$type=0){   global $conn;    $sql="
insert into ih_log(event,type)    values('$event','$type')";    mysql_query($sql,$con
n); } //模拟下单操作//库存是否大于0$sql="select number from ih_store where goods_id='
$goods_id' and sku_id='$sku_id'";//解锁 此时ih_store数据中goods_id='$goods_id' and sku_id
='$sku_id' 的数据被锁住(注3)，其它事务必须等待此次事务 提交后才能执行$rs=mysql_query($
sql,$conn);$row=mysql_fetch_assoc($rs);if($row['number']>0){//高并发下会导致超卖   $order
_sn=build_order_no();   //生成订单    $sql="insert into ih_order(order_sn,user_id,
goods_id,sku_id,price)    values('$order_sn','$user_id','$goods_id','$sku_id','$price')";
 $order_rs=mysql_query($sql,$conn);       //库存减少    $sql="update ih_
store set number=number-{$number} where sku_id='$sku_id孔子故乡'";   $store_rs=mys
ql_query($sql,$conn);    if(mysql_affected_rows()){       insertlog('库存减少
```

```
成功'); }else{ insertlog('库存减少失败'); } }else{ inse
rtlog('库存不够');}
```

出现这种情况怎么办呢？来看几种优化方法：

优化方案1：
将库存字段number字段设为unsigned，当库存为0时，因为字段不能为负数，将会返回false

```
//库存减少 $sql="update ih_store set number=number-{$number} where sku_id='$sku_id'
and number>0"; $store_rs=mysql_query($sql,$conn); if(mysql_affected_rows()){
insertlog('库存减少成功'); }
```

优化方案2：使用mysql的事务，锁住操作的行

```
<?php$conn=mysql_connect("localhost","big","123456"); if(!$conn){ echo "connect faile
d"; exit; } mysql_select_db("big",$conn); mysql_query("set names utf8"); $price=10;
$user_id=1;$goods_id=1;$sku_id=11;$number=1; //生成唯一订单号function build_order_no(){
return date('ymd').substr(implode(null, array_map('ord', str_split(substr(uniqid(), 7, 13), 1))),
0, 8);}//记录日志function insertlog($event,$type=0){ global $conn; $sql="inser
t into ih_log(event,type) values('$event','$type')"; mysql_query($sql,$conn);
} //模拟下单操作//库存是否大于0mysql_query("begin"); //开始事务$sql="select number fr
om ih_store where goods_id='$goods_id' and sku_id='$sku_id' for update";//此时这条记录
被锁住,其它事务必须等待此次事务提交后才能执行$rs=mysql_query($sql,$conn);$row=mysql_fetch_
assoc($rs);if($row['number']>0){ //生成订单 $order_sn=build_order_no();
$sql="insert into ih_order(order_sn,user_id,goods_id,sku_id,price) values('$order_sn'
,'$user_id','$goods_id','$sku_id','$price')"; $order_rs=mysql_query($sql,$conn);
//库存减少 $sql="update ih_store set number=number-{$number} where
sku_id='$sku_id'"; $store_rs=mysql_query($sql,$conn); if(mysql_affected_rows()){
insertlog('库存减少成功'); mysql_query("commit");//事务提
交即解锁 }else{ insertlog('库存减少失败'); }}else{ inse
rtlog('库存不够'); mysql_query("rollback");}
```

优化方案3：使用非阻塞的文件排他锁

```
<?php$conn=mysql_connect("localhost","root","123456"); if(!$conn){ echo "connect fail
ed"; exit; } mysql_select_db("big-bak",$conn); mysql_query("set names utf8"); $pri
ce=10;$user_id=1;$goods_id=1;$sku_id=11;$number=1; //生成唯一订单号function build_order_no(
){ return date('ymd').substr(implode(null, array_map('ord', str_split(substr(uniqid(), 7, 13),
1))), 0, 8);}//记录日志function insertlog($event,$type=0){ global $conn; $sql
="insert into ih_log(event,type) values('$event','$type')"; mysql_query($sql,$co
nn); } $fp = fopen("lock.txt", "w+");if(!flock($fp,lock_ex | lock_nb)){ echo "系统
繁忙，请稍后再试"; return;}//下单$sql="select number from ih_store where goods_id
='$goods_id' and sku_id='$sku_id'";$rs=mysql_query($sql,$conn);$row=mysql_fetch_assoc($rs);if($r
ow['number']>0){//库存是否大于0 //模拟下单操作 $order_sn=build_order_no();
$sql="insert into ih_order(order_sn,user_id,goods_id,sku_id,price) values('$o
rder_sn','$user_id','$goods_id','$sku_id','$price')"; $order_rs=mysql_query($sql,$conn);
```

```
//库存减少 $sql="update ih_store set number=number-{$number}
where sku_id='$sku_id'"; $store_rs=mysql_query($sql,$conn); if(mysql_affected_r
ows()){ insertlog('库存减少成功'); flock($fp,lock_un);//释放
锁 }else{ insertlog('库存减少失败'); } }e数的成语lse{
insertlog('库存不够');}fclose($fp);
```

优化方案4：

使用redis队列，因为pop操作是原子的，即使有很多用户同时到达，也是依次执行，推荐使用（mysql事务在高并发下性能下降很厉害，文件锁的方式也是）

先将商品库存如队列

```
<?php$store=1000;$redis=new redis();$re大学毕业赠言sult=$redis->connect('127.0.0.1',6379);$res=$re
dis->llen('goods_store');echo $res;$count=$store-$res;for($i=0;$i<$count;$i++){ $redis->lpus
h('goods_store',1);}echo $redis->llen('goods_store');
```

抢购、描述逻辑

```
<?php$conn=mysql_connect("localhost","big","123456"); if(!$conn){ echo "connect faile
d"; exit; } mysql_select_db("big",$conn); mysql_query("set names utf8"); $price=10;
$user_id=1;$goods_id=1;$sku_id=11;$number=1; //生成唯一订单号function build_order_no(){
return date('ymd').substr(implode(null, array_map('ord', str_split(substr(uniqid(), 7, 13), 1))),
0, 8);}//记录日志function insertlog($event,$type=0){ global $conn; $sql="inser
t into ih_log(event,type) values('$event','$type')"; mysql_query($sql,$conn);
} //模拟下单操作//下单前判断redis队列库存量$redis=new redis();$result=$redis->connect('127.0.0
.1',6379);$count=$redis->lpop('goods挂面怎么做好吃又简单_store');if(!$count){ insertlog('err
or:no store redis'); return;} //生成订单 $order_sn=build_order_no();$sql="insert into
ih_order(order_sn,user_id,goods_id,sku_id,price) values('$order_sn','$user_id','$goods_id','$sku_id','
$price')"; $order_rs=mysql_query($sql,$conn); //库存减少$sql="update ih_store set number=
number-{$number} where sku_id='$sku_id'";$store_rs=mysql_query($sql,$conn); if(mysql_affected
_rows()){ insertlog('库存减少成功');}else{ insertlog('库存减少失败');}
```

上述只是简单模拟高并发下的抢购，真实场景要比这复杂很多，很多注意的地方

如抢购页面做成静态的，通过ajax调用接口

再如上面的会导致一个用户抢多个，思路：

需要一个排队队列和抢购结果队列及库存队列。高并发情况，先将用户进入排队队列，用一个线程循环处理从排队队列取出一个用户，判断用户是否已在抢购结果队列，如果在，则已抢购，否则未抢购，库存减1，写**数据库**，将用户入结果队列。