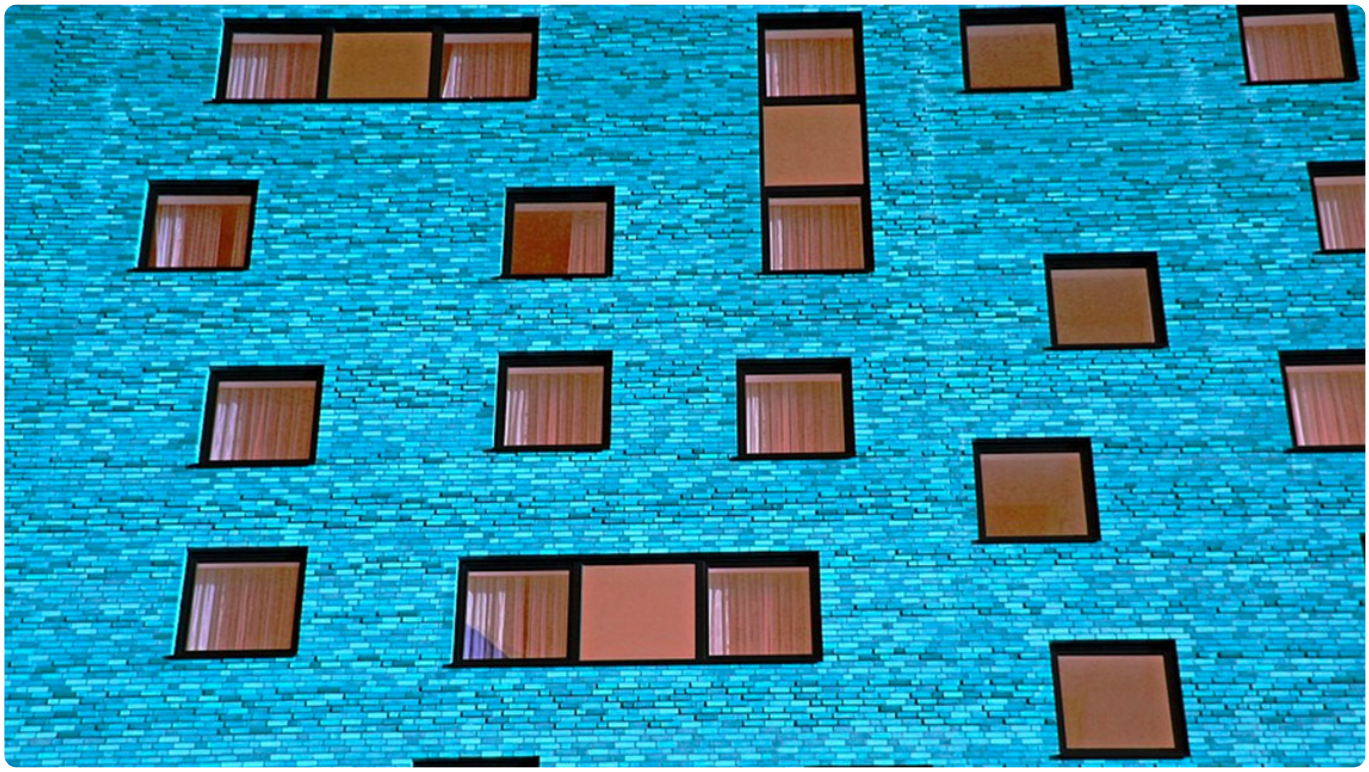
06 | 秒杀系统“减库存”设计的核心逻辑

2018-10-06 许令波

如何设计一个秒杀系统 进入课程 >



讲述：秭明
时长 11:39　大小 5.34M

如果要设计一套秒杀系统，那我想你的老板肯定会先对你说：千万不要超卖，这是大前提。

如果你第一次接触秒杀，那你可能还不太理解，库存 100 件就卖 100 件，在数据库里减到 0 就好了啊，这有什么麻烦的？是的，理论上是这样，但是具体到业务场景中，“减库存”就不是这么简单了。

例如，我们平常购物都是这样，看到喜欢的商品然后下单，但并不是每个下单请求你都最后付款了。你说系统是用户下单了就算这个商品卖出去了，还是等到用户真正付款了才算卖出了呢？这的确是个问题！

我们可以先根据减库存是发生在下单阶段还是付款阶段，把减库存做一下划分。

减库存有哪几种方式

在正常的电商平台购物场景中，用户的实际购买过程一般分为两步：下单和付款。你想买一台 iPhone 手机，在商品页面点了“立即购买”按钮，核对信息之后点击“提交订单”，这一步称为下单操作。下单之后，你只有真正完成付款操作才能算真正购买，也就是俗话说的“落袋为安”。

那如果你是架构师，你会在哪个环节完成减库存的操作呢？总结来说，减库存操作一般有如下几个方式：

下单减库存，即当买家下单后，在商品的总库存中减去买家购买数量。下单减库存是最简单的减库存方式，也是控制最精确的一种，下单时直接通过数据库的事务机制控制商品库存，这样一定不会出现超卖的情况。但是你要知道，有些人下完单可能并不会付款。

付款减库存，即买家下单后，并不立即减库存，而是等到有用户付款后才真正减库存，否则库存一直保留给其他买家。但因为付款时才减库存，如果并发比较高，有可能出现买家下单后付不了款的情况，因为可能商品已经被其他人买走了。

预扣库存，这种方式相对复杂一些，买家下单后，库存为其保留一定的时间（如 10 分钟），超过这个时间，库存将会自动释放，释放后其他买家就可以继续购买。在买家付款前，系统会校验该订单的库存是否还有保留：如果没有保留，则再次尝试预扣；如果库存不足（也就是预扣失败）则不允许继续付款；如果预扣成功，则完成付款并实际地减去库存。

以上这几种减库存的方式都会存在一些问题，下面我们一起来看下。

减库存可能存在的问题

由于购物过程中存在两步或者多步的操作，因此在不同的操作步骤中减库存，就会存在一些可能被恶意买家利用的漏洞，例如发生恶意下单的情况。

假如我们采用“下单减库存”的方式，即用户下单后就减去库存，正常情况下，买家下单后付款的概率会很高，所以不会有太大问题。但是有一种场景例外，就是当卖家参加某个活动时，此时活动的有效时间是商品的黄金售卖时间，如果有竞争对手通过恶意下单的方式将该卖家的商品全部下单，让这款商品的库存减为零，那么这款商品就不能正常售卖了。要知道，这些恶意下单的人是不会真正付款的，这正是“下单减库存”方式的不足之处。

既然“下单减库存”可能导致恶意下单，从而影响卖家的商品销售，那么有没有办法解决呢？你可能会想，采用“付款减库存”的方式是不是就可以了？的确可以。但是，“付款减库存”又会导致另外一个问题：库存超卖。

假如有 100 件商品，就可能出现 300 人下单成功的情况，因为下单时不会减库存，所以也就可能出现下单成功数远远超过真正库存数的情况，这尤其会发生在做活动的热门商品上。这样一来，就会导致很多买家下单成功但是付不了款，买家的购物体验自然比较差。

可以看到，不管是“下单减库存”还是“付款减库存”，都会导致商品库存不能完全和实际售卖情况对应起来的情况，看来要把商品准确地卖出去还真是不容易啊！

那么，既然“下单减库存”和“付款减库存”都有缺点，我们能否把两者相结合，将两次操作进行前后关联起来，下单时先预扣，在规定时间内不付款再释放库存，即采用“预扣库存”这种方式呢？

这种方案确实可以在一定程度上缓解上面的问题。但是否就彻底解决了呢？其实没有！针对恶意下单这种情况，虽然把有效的付款时间设置为 10 分钟，但是恶意买家完全可以在 10 分钟后再次下单，或者采用一次下单很多件的方式把库存减完。针对这种情况，解决办法还是要结合安全和反作弊的措施来制止。

例如，给经常下单不付款的买家进行识别打标（可以在被打标的买家下单时不减库存）、给某些类目设置最大购买件数（例如，参加活动的商品一人最多只能买 3 件），以及对重复下单不付款的操作进行次数限制等。

针对“库存超卖”这种情况，在 10 分钟时间内下单的数量仍然有可能超过库存数量，遇到这种情况我们只能区别对待：对普通的商品下单数量超过库存数量的情况，可以通过补货来解决；但是有些卖家完全不允许库存为负数的情况，那只能在买家付款时提示库存不足。

大型秒杀中如何减库存？

目前来看，业务系统中最常见的就是预扣库存方案，像你在买机票、买电影票时，下单后一般都有个“有效付款时间”，超过这个时间订单自动释放，这都是典型的预扣库存方案。而具体到秒杀这个场景，应该采用哪种方案比较好呢？

由于参加秒杀的商品，一般都是“抢到就是赚到”，所以成功下单后却不付款的情况比较少，再加上卖家对秒杀商品的库存有严格限制，所以秒杀商品采用“下单减库存”更加合

理。另外，理论上由于“下单减库存”比“预扣库存”以及涉及第三方支付的“付款减库存”在逻辑上更为简单，所以性能上更占优势。

“下单减库存”在数据一致性上，主要就是保证大并发请求时库存数据不能为负数，也就是要保证数据库中的库存字段值不能为负数，一般我们有多种解决方案：一种是在应用程序中通过事务来判断，即保证减后库存不能为负数，否则就回滚；另一种办法是直接设置数据库的字段数据为无符号整数，这样减后库存字段值小于零时会直接执行 SQL 语句来报错；再有一种就是使用 CASE WHEN 判断语句，例如这样的 SQL 语句：

```
UPDATE item SET inventory = CASE WHEN inventory >= xxx THEN
inventory-xxx ELSE inventory END
```

秒杀减库存的极致优化

在交易环节中，“库存”是个关键数据，也是个热点数据，因为交易的各个环节中都可能涉及对库存的查询。但是，我在前面介绍分层过滤时提到过，秒杀中并不需要对库存有精确的一致性读，把库存数据放到缓存（Cache）中，可以大大提升读性能。

解决大并发读问题，可以采用 LocalCache（即在秒杀系统的单机上缓存商品相关的数据）和对数据进行分层过滤的方式，但是像减库存这种大并发写无论如何还是避免不了，这也是秒杀场景下最为核心的一个技术难题。

因此，这里我想专门来说一下秒杀场景下减库存的极致优化思路，包括如何在缓存中减库存以及如何在数据库中减库存。

秒杀商品和普通商品的减库存还是有些差异的，例如商品数量比较少，交易时间段也比较短，因此这里有一个大胆的假设，即能否把秒杀商品减库存直接放到缓存系统中实现，也就是直接在缓存中减库存或者在一个带有持久化功能的缓存系统（如 Redis）中完成呢？

如果你的秒杀商品的减库存逻辑非常单一，比如没有复杂的 SKU 库存和总库存这种联动关系的话，我觉得完全可以。但是如果有比较复杂的减库存逻辑，或者需要使用事务，你还是必须在数据库中完成减库存。

由于 MySQL 存储数据的特点，同一数据在数据库里肯定是一行存储（MySQL），因此会有大量线程来竞争 InnoDB 行锁，而并发度越高时等待线程会越多，TPS（Transaction

Per Second，即每秒处理的消息数）会下降，响应时间（RT）会上升，数据库的吞吐量就会严重受影响。

这就可能引发一个问题，就是单个热点商品会影响整个数据库的性能， 导致 0.01% 的商品影响 99.99% 的商品的售卖，这是我们不愿意看到的情况。一个解决思路是遵循前面介绍的原则进行隔离，把热点商品放到单独的热点库中。但是这无疑会带来维护上的麻烦，比如要做热点数据的动态迁移以及单独的数据库等。

而分离热点商品到单独的数据库还是没有解决并发锁的问题，我们应该怎么办呢？要解决并发锁的问题，有两种办法：

应用层做排队。按照商品维度设置队列顺序执行，这样能减少同一台机器对数据库同一行记录进行操作的并发度，同时也能控制单个商品占用数据库连接的数量，防止热点商品占用太多的数据库连接。

数据库层做排队。应用层只能做到单机的排队，但是应用机器数本身很多，这种排队方式控制并发的能力仍然有限，所以如果能在数据库层做全局排队是最理想的。阿里的数据库团队开发了针对这种 MySQL 的 InnoDB 层上的补丁程序（patch），可以在数据库层上对单行记录做到并发排队。

你可能有疑问了，排队和锁竞争不都是要等待吗，有啥区别？

如果熟悉 MySQL 的话，你会知道 InnoDB 内部的死锁检测，以及 MySQL Server 和 InnoDB 的切换会比较消耗性能，淘宝的 MySQL 核心团队还做了很多其他方面的优化，如 COMMIT_ON_SUCCESS 和 ROLLBACK_ON_FAIL 的补丁程序，配合在 SQL 里面加提示（hint），在事务里不需要等待应用层提交（COMMIT），而在数据执行完最后一条 SQL 后，直接根据 TARGET_AFFECT_ROW 的结果进行提交或回滚，可以减少网络等待时间（平均约 0.7ms）。据我所知，目前阿里 MySQL 团队已经将包含这些补丁程序的 MySQL 开源。

另外，数据更新问题除了前面介绍的热点隔离和排队处理之外，还有些场景（如对商品的 lastmodifytime 字段的）更新会非常频繁，在某些场景下这些多条 SQL 是可以合并的，一定时间内只要执行最后一条 SQL 就行了，以便减少对数据库的更新操作。

总结一下

今天，我围绕商品减库存的场景，介绍了减库存的三种实现方案，以及分别存在的问题和可能的缓解办法。最后，我又聚焦秒杀这个场景说了如何实现减库存，以及在这个场景下做到极致优化的一些思路。

当然减库存还有很多细节问题，例如预扣的库存超时后如何进行库存回补，再比如目前都是第三方支付，如何在付款时保证减库存和成功付款时的状态一致性，这些都是很大的挑战。

如果你也有实现减库存的经验或者问题，欢迎留言与我分享。

上一篇 05 | 影响性能的因素有哪些？又该如何提高系统的性能？

下一篇 07 | 准备Plan B：如何设计兜底方案?

精选留言 (80)

写留言

周龙亭
2018-10-07

16

下单和扣库存两个操作的事务性是怎么做的？

展开

作者回复: 可以分两步来做，先创建订单但是先不生效，然后减库存，如果减库存成功后再生效订单，否则订单不生效

永光
2018-10-06

9

老师，你好，
你提到秒杀商品减库存直接放到缓存系统中实现，也就是直接在缓存中减库存或者在一个带有持久化功能的缓存系统（如 Redis）中完成。这种实现并发读写怎样保持数据一致？以及是不是要用分布式缓存？

展开

作者回复: 前面有个同学的类似的问题回答过，可以看一下

刘小刘
2018-12-21

8

老师，我觉得你讲的不太明白，你并没有说实际情况下同步是怎样解决并发的，没看到您给的方案，只看到您在评论回复里否定了队列异步处理的方式

展开

作者回复: 解决的办法就是尽可能避免产生锁，比如根据商品ID进行分库分表设计；再有就是减少锁的粒度例如阿里对MySQL做了定制优化，可以提升MySQL的并发度

moliniao
2018-10-08

6

老师，使用应用排队方式，入队后返回，然后app端轮询请求下单结果吗？

作者回复: 秒杀web请求一般不用排队，谁先到谁先执行。
排队一般更多是在服务端的内部请求时发生，而且是在异步请求时通过消息队列来处理

Coder4
2018-10-06

6

这种无只能在串行隔离级别才能用吧，不然肯定超售。。。UPDATE item SET inventory = CASE WHEN inventory >= xxx THEN inventory-xxx ELSE inventory END

作者回复: 数据库层不都是串行操作吗😊

shawn
2018-10-14

5

个人做法，
针对确定库存，提前下好单，下单人留空，订单短时间内失效
订单id压入Redis队列，
请求来到，订单队列lpop，队空则返回失败，
pop出来的订单补充下单人为当前用户，...
展开

作者回复: 说实话，没看出来哪里性能会更好😄
不过提前下单的思路还比较新颖，你的思路我理解，但是这样就把一个事情分两次来做，会增加了复杂度，有可能导致得不偿失

公号-代码...
2018-10-06

5

许老师好，我有一个想法，只是没有在实践中这样做过，请指教:
能否借用"数据库水平拆分"的思想？
具体思路如下:
库存在数据库的表中就只有一行数据，上面的方案都是对这一条记录进行频繁更新，是非常"热"的热点数据。我们能否将该行数据拆分到不同的数据库中，这些数据库的库存记录...
展开

作者回复: 实际上，商品都是进行分库分表的，例如根据商品id进行水平拆分

分库分表就是提高吞吐量

一笑奈何
2018-10-06

5

老师，问下单机mysql 1s内能抗大约多大的QPS? 大约。

作者回复: 我印象中单实例一般能抗7-8k左右

慕-小坏
2018-10-18

3

看到有同学说下单排队可以用请求队列来做，想问一下请求队列里存放的是请求数据吧，即用户请求数据入队列之后请求立即返回，后台异步处理请求数据，那处理的结果如何告知用户？是前端发起轮训请求吗？如果是轮训的话又会占用服务器不少连接资源吧？
如果请求队列里直接存的是http 请求的话服务器端也是会持有大量未释放的http 长连接。
所以请教一下实际当中一般请求队列这部分是怎么做的呢？

展开

作者回复: 大家对请求队列这块问的比较多，后面相关的问题我统一回答一下吧

Geek_c19c9...
2018-10-16

3

我们的库存都放在redis里面，读和减库存都在redis里面操作，redis会定时将库存放到mysql中做备份，

作者回复: 😉

大麦
2018-10-12

3

秒杀是短时间大量请求，使用下单即锁库存方式，可以通过一个 redis 队列记录下单，一个redis key 记录数量 num，超出的库存下单失败，这样大量请求在 redis 层即可被处理。
通过 num 与库存的判断来解决无效订单。
下单端通过队列异步消费下单。...

展开

作者回复: 异步下单的方式，也是一个思路，例如在一些场景下其实已经在使用，例如一些支付场景中，付了款以后，前端页面中会有一个转圈，等个几秒钟再告诉你结果。

这种方式我个人觉得对用户不太友好，就是要让用户等个几秒钟，而不是像同步的方式能及时得到反馈结果

Laputa
2018-10-06

3

请问文中扣库存的case when语句:
UPDATE item SET inventory = CASE WHEN inventory >= xxx THEN inventory-xxx ELSE inventory END
当库存不足时，是不是还会执行一次更新操作，即多一次磁盘写操作？
如果改成这样:...
展开

作者回复: 库存不足时也就是秒杀结束时了，即使再有一次磁盘操作问题也不大了
看这个语句应该也可以

宁宁
2018-12-06

2

下单减库存的方式存在问题是有些用户下单缺不付款，有一个补偿方案就是付款设置超时时间，一旦超时取消订单，同时发送消息到消息对列，库存服务订阅消息，把库存加回去！

null
2018-11-01

2

方法一和方法三是不是没啥区别？

方法一：下单减库存，但是用户不支付，订单超时释放库存
方法三：预扣库存，用户下单时扣库存，超时释放库存
...
展开

作者回复: 如果方法一有后续的超时回补库存，那么就差不多了

我是李香兰...
2018-10-13

2

“按照商品维度设置队列顺序执行”这句话是什么意思？可以举例说明一下吗？谢谢老师

作者回复: “按照商品维度设置队列顺序执行”的意思就是，为了防止同一个商品对数据库的操作占用太多的数据库资源，所以采用队列的方式，让其他商品也有公平的机会得到数据的响应，例如如果秒杀的时候，秒杀商品肯定占用大量的请求，数据库的连接池有可能都被秒杀商品占用了，如果不做队列的话，那么其他商品就得不到数据库执行机会了。加入我们分10个队列，那么秒杀商品就会落在这10个队列中的一个，那么最多也就占用机器10分之一的资源。

公号-代码...
2018-10-07

2

许老师好，减库存的操作是否需要考虑操作的幂等性？如果要考虑，如何实现？

作者回复: 减库存的幂等是在创建订单时保证的
幂等一般都是保证不会有重复提交发生

Bigming
2018-10-06

2

预扣库存方案中如何确保十分钟后库存自动解冻？定时任务还是会有延迟吧？

作者回复: 自动解冻可以在应用程序中设置一个定时器来定时扫描数据库的下单时间，来比较是否已经超时

延时一点问题也不大，因为超时时间也是人为主观设置的

锐

1

2019-01-17

没人回答问题了么。想知道减库存在缓存中操作，如果还没持久化之前缓存挂了，数据丢失了，实际库存没减，该怎么补救呢？或者这篇都是基于缓存不会挂的情况下设计的？

作者回复: 用缓存来存放库存信息是有你说的风险，当然我们要做些措施来保证缓存不会轻易挂掉，例如做限流保护、做多机房备份（第7节介绍了很多兜底方案）。当然如果出现最极端的情况，那就只能停止秒杀该商品了。

Geek_626d8...

2018-12-21

1

我又一个想法：就是讲总库存分成几批分别储存在不同的服务器上，比如100个商品分别放在5台服务器abcd，每台放20个商品，用户通过抢购进入网关，我们可以制定一个路由策略比如用户id等于1-100的去a抢购，id等于201-300去b抢购，以此类推，整个抢购活动结束后在整体同步到数据库，这样做减少了数据的并发计算，由于是抢购也不存在单台服务商品库存过剩的情况，您觉得这个思路怎么样？

展开

作者回复: haha，有意思，不过我们的商品库存本来就是分库分表的，不同的商品库存本身就不在一台机器上的。

wlkfec

2018-11-11

1

利用CAS乐观锁加入版本号的概念实现并发读写怎么样？

作者回复: 在分布式缓存中，版本号的机制用的还是比较多的😉