

03 | 复杂而又重要的购物车系统，应该如何设计？

李玥 · 后端存储实战课



你好，我是李玥。

今天这节课我们来说一下购物车系统的存储该如何设计。

首先，我们来看购物车系统的主要功能是什么。就是在用户选购商品时，下单之前，暂存用户想要购买的商品。购物车对数据可靠性要求不高，性能也没有特别的要求，在整个电商系统中，看起来是相对比较容易设计和实现的一个子系统。

购物车系统的功能，主要的就三个：把商品加入购物车（后文称“加购”）、购物车列表页、发起结算下单，再加上一个在所有界面都要显示的购物车小图标。

支撑购物车的这几个功能，对应的存储模型应该怎么设计？很简单，只要一个“购物车”实体就够了。它的主要属性有什么？你打开京东的购物车页面，对着抄就设计出来了：SKUID（商品ID）、数量、加购时间和勾选状态。

全部商品 3

配送至:

<input type="checkbox"/> 全选	商品	单价	数量	小计	操作
<div> <input checked="" type="checkbox"/> 京东好店 晟发玩具专营店 优惠券 </div>					
<div> 满减 活动商品购满¥99.00, 即可享受满减 去凑单 > </div>				¥46.00	
<input checked="" type="checkbox"/>	 <div> 合金飞机玩具A380空中客机战斗机直升飞机20仿真模型声光回力儿童玩具宝 <div> 选服务 </div> </div>	中国国际航空(大号)	<div> <div>¥46.00</div> <div>促销</div> </div>	<div> <div>-</div> <div>1</div> <div>+</div> </div> <div>有货</div>	<div>¥46.00</div> <div>删除 移到我的关注</div>
<div> <input type="checkbox"/> 京东好店 宝发玩具旗舰店 优惠券 </div>					
<input type="checkbox"/>	 <div> 京东超市 【可充电】立昕托马斯小火车轨道车儿童玩具多层超大电动过山车 <div> 选服务 </div> </div>	双层 动力高铁【含和...	¥119.00	<div> <div>-</div> <div>1</div> <div>+</div> </div> <div>有货</div>	<div>¥119.00</div> <div>删除 移到我的关注</div>
<div> 【赠品】 流量 (支持移动联通电信手机号提取) 1M X10 查看价格 </div>					
<div> <input checked="" type="checkbox"/> 微软升昌专卖店 优惠券 </div>					
<input checked="" type="checkbox"/>	 <div> 微软 (Microsoft) Surface Pro 6 平板电脑笔记本二合一 轻薄便携商务办 <div> 选服务 </div> </div>	【亮铂金】i5 8G内存... 标配+原装键盘+【领...	<div> <div>¥7738.00</div> <div>白条3期免息</div> </div>	<div> <div>-</div> <div>1</div> <div>+</div> </div> <div>有货</div>	<div>¥7738.00</div> <div>删除 移到我的关注</div>
<div> 【闪购】 2020-01-17 10:00 恢复原价 <div> 【赠品】 surface用手提式多功能包 内胆包皮套 收纳平... X1 查看价格 </div> <div> 【赠品】 微软 (Microsoft) Surface Pro 7 6 5 专业键... X1 查看价格 </div> <div> 【赠品】 Surface Book钢化膜 pro保护膜 屏幕贴膜 sur... X1 查看价格 </div> <div> 【赠品】 微软surface专用背贴 赠品链接不单卖 单拍不... X1 查看价格 </div> <div> 【赠品】 笔记本surface pro网卡 以太网转接口 mac网... X1 查看价格 </div> </div>					
<input type="checkbox"/> 全选 删除选中商品 移到关注 清理购物车			已选择 2 件商品 ^ 总价: ¥7784.00 促销: -¥0.00		
			<input type="button" value="去结算"/>		

备注：图片来源于网络，仅供本文介绍、评论及说明某问题，适当引用。

这个“勾选状态”属性，就是在购物车界面中，每件商品前面的那个小对号，表示在结算下单时，是不是要包含这件商品。至于商品价格和总价、商品介绍等等这些信息，都可以实时从其他系统中获取，不需要购物车系统来保存。

购物车的功能虽然很简单，但是在设计购物车系统的存储时，仍然有一些特殊的问题需要考虑。

设计购物车存储时需要把握什么原则？

比如下面这几个问题：

1. 用户没登录，在浏览器中加购，关闭浏览器再打开，刚才加购的商品还在不在？
2. 用户没登录，在浏览器中加购，然后登录，刚才加购的商品还在不在？
3. 关闭浏览器再打开，上一步加购的商品在不在？
4. 再打开手机，用相同的用户登录，第二步加购的商品还在不在呢？

上面这几个问题是不是有点儿绕？没关系，我们先简单解释一下这四个问题：

1. 如果用户没登录，加购的商品也会被保存在用户的电脑里，这样即使关闭浏览器再打开，购物车的商品仍然存在。
2. 如果用户先加购，再登录，登录前加购的商品就会被自动合并到用户名下，所以登录后购物车中仍然有登录前加购的商品。
3. 关闭浏览器再打开，这时又变为未登录状态，但是之前未登录时加购的商品已经被合并到刚刚登录的用户名下了，所以购物车是空的。
4. 使用手机登录相同的用户，看到的的就是该用户的购物车，这时无论你在手机 App、电脑还是微信中登录，只要是相同的用户，看到是同一个购物车，所以第二步加购的商品是存在的。

所以，上面这四个问题的答案依次是：存在、存在、不存在、存在。

如果你没有设计或者开发过购物车系统，你可能并不会想到购物车还有这么多弯弯绕。但是，作为一个开发者，如果你不仔细把这些问题考虑清楚，用户在使用购物车的时候，就会感觉你的购物车系统不好用，不是加购的商品莫名其妙地丢了，就是购物车莫名其妙地多出来一些商品。

要解决上面这些问题，其实只要在存储设计时，把握这几个原则就可以了：

1. 如果未登录，需要临时暂存购物车的商品；
2. 用户登录时，把暂存购物车的商品合并到用户购物车中，并且清除暂存购物车；
3. 用户登陆后，购物车中的商品，需要在浏览器、手机 APP 和微信等等这些终端中都保持同步。

实际上，购物车系统需要保存两类购物车，一类是未登录情况下的“暂存购物车”，一类是登录后的“用户购物车”。

如何设计“暂存购物车”的存储？

我们先来看下暂存购物车的存储该怎么实现。暂存购物车应该存在客户端还是存在服务端？

如果保存在服务端，那每个暂存购物车都需要有一个全局唯一的标识，这个标识并不太容易设计，并且，存在服务端还要浪费服务端的资源。所以，肯定是保存在客户端好，既可以节约服务器的存储资源，也没有购物车标识的问题，因为每个客户端就保存它自己唯一一个购物车就可以了，不需要标识。

客户端的存储可以选择的不太多：Session、Cookie 和 LocalStorage，其中浏览器的 LocalStorage 和 App 的本地存储是类似的，我们都以 LocalStorage 来代表。

存在哪儿最合适？SESSION 是不太合适的，原因是，SESSION 的保留时间短，而且 SESSION 的数据实际上还是保存在服务端的。剩余的两种存储，Cookie 和 LocalStorage 都可以用来保存购物车数据，选择哪种方式更好呢？各有优劣。

在我们这个场景中，使用 Cookie 和 LocalStorage 最关键的区别是，客户端和服务端的每次交互，都会自动带着 Cookie 数据往返，这样服务端可以读写客户端 Cookie 中的数据，而 LocalStorage 里的数据，只能由客户端来访问。


使用 Cookie 存储，实现起来比较简单，加减购物车、合并购物车的过程中，由于服务端可以读写 Cookie，这样全部逻辑都可以在服务端实现，并且客户端和服务端请求的次数也相对少一些。

使用 LocalStorage 存储，实现相对就复杂一点儿，客户端和服务端都要实现一些业务逻辑，但 LocalStorage 的好处是，它的存储容量比 Cookie 的 4KB 上限要大得多，而且不用像 Cookie 那样，无论用不用，每次请求都要带着，可以节省带宽。

所以，选择 Cookie 或者是 LocalStorage 来存储暂存购物车都是没问题的，你可以根据它俩各自的优劣势来选择。比如你设计的是个小型电商，那用 Cookie 存储实现起来更简单。再比

如，你的电商是面那种批发的行业用户，用户需要加购大量的商品，那 Cookie 可能容量不够用，选择 LocalStorage 就更合适。

不管选择哪种存储，暂存购物车保存的数据格式都是一样的，参照我们实体模型来设计就可以，我们可以直接用 JSON 表示：

 复制代码

```
1 {  
2     "cart": [  
3         {  
4             "SKUID": 8888,  
5             "timestamp": 1578721136,  
6             "count": 1,  
7             "selected": true  
8         },  
9         {  
10            "SKUID": 6666,  
11            "timestamp": 1578721138,  
12            "count": 2,  
13            "selected": false  
14        }  
15    ]  
16 }
```


如何设计“用户购物车”的存储？

接下来，我们再来看下用户购物车的存储该怎么实现。因为用户购物车必须要保证多端的数据同步，所以数据必须保存在服务端。常规的思路是，设计一张购物车表，把数据存在 MySQL 中。这个表的结构同样可以参照刚刚讲的实体模型来设计：

列名	数据类型	主键	非空	说明
id	BIGINT	是	是	自增主键
user_id	BIGINT		是	用户ID
sku_id	BIGINT		是	商品ID
count	INT		是	商品数量
timestamp	DATE		是	加购时间
selected	TINYINT (1)			购选状态

注意，需要在 user_id 上建一个索引，因为查询购物车表时，都是以 user_id 作为查询条件来查询的。

你也可以选择更快的 Redis 来保存购物车数据，以用户 ID 作为 Key，用一个 Redis 的 HASH 作为 Value 来保存购物车中的商品。比如：

 复制代码

```

1 {
2     "KEY": 6666,
3     "VALUE": [
4         {
5             "FIELD": 8888,
6             "FIELD_VALUE": {
7                 "timestamp": 1578721136,
8                 "count": 1,
9                 "selected": true
10            }
11        },
12        {
13            "FIELD": 6666,
14            "FIELD_VALUE": {
15                "timestamp": 1578721138,
16                "count": 2,
17                "selected": false
18            }
19        }
20    ]
21 }
```

这里为了便于你理解，我们用 JSON 来表示 Redis 中 HASH 的数据结构，其中 KEY 中的值 6666 是一个用户 ID，FIELD 里存放的是商品 ID，FIELD_VALUE 是一个 JSON 字符串，保存加购时间、商品数量和勾选状态。

大家都知道，从读写性能上来说，Redis 是比 MySQL 快非常多的，那是不是用 Redis 就一定比用 MySQL 更好呢？我们来比较一下使用 MySQL 和 Redis 两种存储的优劣势：

1. 显然使用 Redis 性能要比 MySQL 高出至少一个量级，响应时间更短，可以支撑更多的并发请求，“天下武功，唯快不破”，这一点 Redis 完胜。
2. MySQL 的数据可靠性是要好于 Redis 的，因为 Redis 是异步刷盘，如果出现服务器掉电等异常情况，Redis 是有可能丢数据的。但考虑到购物车里的数据，对可靠性要求也没那么苛刻，丢少量数据的后果也就是，个别用户的购物车少了几件商品，问题也不大。所以，在购物车这个场景下，Redis 的数据可靠性不高这个缺点，并不是不能接受的。
3. MySQL 的另一个优势是，它支持丰富的查询方式和事务机制，这两个特性，对我们今天讨论的这几个购物车核心功能没什么用。但是，每一个电商系统都有它个性化的需求，如果需要以其他方式访问购物车的数据，比如说，统计一下今天加购的商品总数，这个时候，使用 MySQL 存储数据，就很容易实现，而使用 Redis 存储，查询起来就非常麻烦而且低效。

综合比较下来，考虑到需求总是不断变化，还是更推荐你使用 MySQL 来存储购物车数据。如果追求性能或者高并发，也可以选择使用 Redis。

你可以感受到，我们设计存储架构的过程就是一个不断做选择题的过程。很多情况下，可供选择的方案不止一套，选择的时候需要考虑实现复杂度、性能、系统可用性、数据可靠性、可扩展性等等非常多的条件。需要强调的是，**这些条件每一个都不是绝对不可以牺牲的，不要让一些“所谓的常识”禁锢了你的思维。**

比如，一般我们都认为数据是绝对不可以丢的，也就是说不能牺牲数据可靠性。但是，像刚刚讲到的用户购物车的存储，使用 Redis 替代 MySQL，就是牺牲了数据可靠性换取高性能。我们仔细分析后得出，很低概率的情况下丢失少量数据，是可以接受的。性能提升带来的收益远大于丢失少量数据而付出的代价，这个选择就是划算的。

如果说不考虑需求变化这个因素，牺牲一点点数据可靠性，换取大幅性能提升，选择 Redis 才是最优解。

小结

今天我们讲了购物车系统的存储该如何设计。

购物车系统的主要功能包括：加购、购物车列表页和结算下单。核心的实体就只有一个“购物车”实体，它至少应包括：SKUID、数量、加购时间和勾选状态这几个属性。

在给购物车设计存储时，为了确保购物车内的数据在多端保持一致，以及用户登录前后购物车内商品能无缝衔接，除了每个用户的“用户购物车”之外还要实现一个“暂存购物车”保存用户未登录时加购的商品，并在用户登录后自动合并“暂存购物车”和“用户购物车”。

暂存购物车存储在客户端浏览器或者 App 中，可以选择存放到 Cookie 或者 LocalStorage 中。用户购物车保存在服务端，可以选择使用 Redis 或者是 MySQL 存储，使用 Redis 存储会有更高的性能，可以支撑更多的并发请求，使用 MySQL 是更常规通用的方式，便于应对变化，系统的扩展性更好。

思考题

课后请你思考一下，既然用户的购物车数据存放在 MySQL 或者是 Redis 中各有优劣势。那能不能把购物车数据存在 MySQL 中，并且用 Redis 来做缓存呢？这样不就可以兼顾两者的优势了么？这样做是不是可行？如果可行，如何来保证 Redis 中的数据和 MySQL 中的数据是一样的呢？

欢迎你在留言区与我讨论，如果你觉得今天学到的知识对你有帮助，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。



李玥 置顶

2020-03-03

hi, 我是李玥。

上节课我给你留了一道思考题，是这样的。如果说，用户下单这个时刻，正好赶上商品调价，就有可能出现这样的情况：我明明在商详页看到的价格是10块钱，下单后，怎么变成15块了？你的系统是不是偷偷在坑我？给用户的体验非常不好。你不要以为这是一个小概率事件，当你的系统用户足够多的时候，每时每刻都有人在下单，这几乎是个必然出现的事件。该怎么来解决这个问题？

关于这个问题，我是这样看的。

首先，商品系统需要保存包含价格的商品基本信息的历史数据，对每一次变更记录一个自增的版本号。在下单的请求中，不仅要带上SKUID，还要带上版本号。订单服务以请求中的商品版本对应的价格来创建订单，就可以避免“下单时突然变价”的问题了。

但是，这样改正之后会产生一个很严重的系统漏洞：黑客有可能会利用这个机制，以最便宜的历史价格来下单。所以，我们在下单之前需要增加一个检测逻辑：请求中的版本号只能是当前版本或者上一个版本，并且使用上一个版本要有一个时间限制，比如说调价5秒之后，就不再接受上一个版本的请求。这样就可以避免这个调价漏洞了。

共 20 条评论 >

👍 176



京京beaver

2020-03-03

购物车一般建议放到MySQL中。一般电商购物车是不占库存的，但是某些特卖电商购物车是占库存的。在这种情况下，数据是不允许丢失的，不然客户体验会非常差。Redis做缓存没啥用，因为每个用户只访问自己的购物车，每次访问网站也不会打开很多次购物车，缓存数据的命中率太低，没有意义。

共 7 条评论 >

👍 70



黄海峰

2020-03-03

感觉购物车是写多于读，也就是经常变，用cache aside的方式保持一致性的话就经常删缓存，db压力减轻不了多少，还要多写一次缓存，没什么必要

共 6 条评论 >

👍 35



公号-技术夜未眠

2020-03-03

读多写少用缓存，写多读少用MQ。对于前者，前提是读场景频繁且能具备较高的命中率。用户购物车数据不符合该场景。



27



Jxin

2020-03-04

- 1.能当然能。
- 2.能兼备mysql的可靠性和redis的读取性能。
- 3.这样做不可行，因为购物车写多读少，这样玩会频繁失效缓存，进而导致大部分读都要击穿到db并多做一步缓存的操作。实则弊大于利。
- 4.一旦修改购物车，redis的缓存直接失效。

个人见解：

- 1.感觉 未登录添加购物车。这功能可以弃了。填购必须注册登陆。
- 2.当下用户已经很习惯网购，对于注册登陆也是顺手之事，失去这个功能对用户体验的降低实属有限。
- 3.以往一个电商平台核心可能是推广加销售商品。但现在多了一个用户管理，而拉新成本高的当下，失去这一功能，变成强制注册不见得就是坏事，值得一试。

总结：加上未登陆添加购物车的功能。在当下，可能是牺牲部份可能发展成活跃用户的流动用户，来换取微末的用户体验。得不偿失。

共 2 条评论 >

21



传志

2020-03-03

购物车，同时使用redis+mysql觉得可行。以redis为主，增，查询，删除都走redis.添加加时使用mq保证最终一致性。统计等需求可以在mysql中做

共 1 条评论 >

16



业余爱好者

2020-03-03

今日得到：以前对浏览器存储的认识只停留在cookie上。以前认为只有服务端才有session数据。以前虽然听过localStorage这个词，但是思想上没有重视。刚查了下资料，浏览器存储还有indexeddb。学习还是要系统。

存储的本质是把数据暂时或永久保存下来，单从持久化这个目标来看，什么方式存储都是为所谓的。不过，存储的作用实际是为了将来的计算。既然从持久化角度没有差别，那当然要选择

一种方便我们后续处理的存储设计了。所以对存储产品的选择与数据结构的设计需要结合具体的业务需求。存储的设计要考虑存储时间（临时或永久），可靠性要求，关键业务操作。需要梳理业务需求。可见需求分析是多么重要。

思考题：购物车从功能上说，就是一个临时存储的信息。为了追求性能，所以一般没必要存db，除了一些有统计业务功能的系统。购物车购物车信息的特性是一个树性的结构，使用关系模型不好设计，需要多表join。不过可以设计成json字符串类型，或者使用新的json类型。购物车写多读少，需要频繁刷db保证数据同步，为了追求一点点没多大用处的可靠性，得不偿失。

共 1 条评论 >

👍 9



aoe

2020-03-03

思考题是可行的，但是复杂，例如需要考虑：

1. Redis的容量可能远小于数据库容量，需要缓存策略缓存数据
2. 要处理老师提到的一致性问题
3. 性价比

另外请教老师一个问题：

在电商系统中，订单的“商品快照”（商品名称、数量、详情页所有信息）一般是怎么存储的？

例如：有订单、商品2个子系统，订单的商品快照一般是由哪个系统生成和保存？

作者回复：我个人的看法是，商品子系统存储商品快照更合理一些。

共 2 条评论 >

👍 9



睡浴缸的人

2020-04-08

感受到了这种技术选择的纠结与权衡的魅力，很喜欢老师的这种讲述方式



👍 7



leslie

2020-03-03

这个问题的回答应当从两种数据库特性去说起吧：redis的特点是存储于内存，但是数据落地刷盘、、、mysql的特性是数据存储于硬盘。

由于存于内存故而查询速度非常可观：购物车环节其实商品变革的频率蛮高的，此时如果直接每次增删商品都访问硬盘数据库，这个代价就、、、尤其是在高并发场景下，真正与金额直接产生的交互的环节是结算环节，即付款；我记得老师曾经在消息队列的期中考试中考过什么场景下数据会丢失

故而今天课程的答案：我个人倾向在结算之前不落mysql数据库：购物车环节直接用redis以减少购物车频繁改动而带来大量的IO消耗。

另外有一点我不太确定：LocalStorage的安全性如何？cookie的安全性问题一直颇受争议，此时在购物车环节去直接访问结算数据库，觉得欠妥。

谢谢老师的分享：去年的课程受益匪浅，最近正在准备抽空过第二遍；希望这门课程同样能收获不一样的东西；谢谢。



4



dingjiayi

2020-08-15

我觉得这是一门被低估的课程(我认为订阅数目应该远大于当前数目4895)



4



大秦皇朝

2020-03-04

1、京东的浏览器端，为什么每次加完购物车都要跳转到一个中转界面上呢？这点我疑惑了很多年，从软件设计的逆向思维来考虑，我也想不出所以然来，还请李玥老师是否能给解答下呢？因为按照我们的日常使用习惯（如：阿里，京豆手机端等），都是点击加购物车，直接购物车数量加1提示就好了呀？为什么要多此一举影响用户体验呢？

2、李玥老师文稿中提到，用户的购物车偶发情况下丢失一些数据可以接受，但是站在消费者的角度来说，我感觉也没必要纠结是不是少了几样东西。但是会直接影响我对这个平台的感知度，我会认为这个平台能力不行，不注重用户体验等，会给产品（京东）带来很多负面影响呀，所以牺牲的可靠性的这个比例的平衡点是一定要重视的。

思考题：综上第2点所述，我觉得业务上必然需要这个方案可行吧？写数据的时候MySQL写和Redis同时删，读的时候从Redis中读；如果没有读到再从MySQL中读取，同时写到Redis中。（现学现卖不知道对不对）

但是也有个问题，多端同时操作，或者网络不好的时候，怎么保证数据的准确性呢？完了完了，我又跳回第一讲了。。。

作者回复: 关于你的第一个问题：“为什么每次加完购物车都要跳转到一个中转界面上呢？”，虽然我也在京东工作，我还真不知道为什么这样设计，不过不知道也好，我们还可以尝试去猜一下它为什么

这样设计，如果我知道的话，可能会涉及商业秘密，反而不能回答了。

作为局外人，我的猜测是这样的，加购之后，一般用户就不会想继续看这个商详页了，接下来它可能的二个路径是：

- 1.去购物车结算；
- 2.去看其它商品；

所以，增加一个中间页，可以放好多推荐商品，引导用户继续购物，算是商家的小聪明吧。

共 4 条评论 >

👍 3



肥low

2020-03-03

我觉得完全可行 而且有时候比如MySQL主从架构下是有数据延迟更新问题的 用Redis我可以尽量避免这一点 不过有对用户加购的维护成本

作者回复: 我会在下节课的评论区说一下我的理解，请关注。

共 2 条评论 >

👍 3



Geek_8c4282

2021-01-13

老师想问下库存在高并发下有没有方法解决超卖或少卖问题，看了下好多同学都问了，但是老师都不回答

共 2 条评论 >

👍 2



锐

2020-04-05

“用户没登录，在浏览器中加购，然后登录，刚才加购的商品还在不在？”关于这一点，怎么判断没登录加车的用户 和 登录的用户是同一个呢？比如我没登录在购物车加了一堆东西，然后我朋友用我的电脑登录他的账户，这该如何解决

作者回复: 这种情况真的识别不了.....

共 4 条评论 >

👍 2



何妨

2020-03-03

不考虑复杂性和服务器成本的话，我认为是可行的。跟老师之前讲的方法一样，每次查询购物车先在 redis 里查，查不到再到 mysql 中查同时更新 redis 中数据。更新用户购物车数据时删除 redis 中数据。但我有一个问题是:用户本身购物车没数据的时候会导致 redis 和 mysql 查两遍.....

共 1 条评论>



2



五河士稻

2020-05-22

个人用法:只用 redis 存储购物车，现在 redis 可以保证数据长久存储，不丢失（比如:集群），其次设计一个记录表，有购物车记录操作和变动时，可以根据用户id和sku id来作为记录主要标识，从而方便统计。



1



mickey

2020-04-15

我觉得，如果电商没有分库，商品表和购物车表在一个库中，用mysql好些，redis存的SKU ID，还是要落mysql查商品信息和价格，还不如一次性查回来；如果分库了，用Redis好些，根据id发异步请求就是。



1



狐狸糊涂

2020-04-04

老师，能讲解一下订单模块到物流的设计吗？

一个订单是按照什么来拆分子订单的？比如物品部分发货的情况，怎么来设计拆分订单更合理呢？



1



四道杠的红领巾

2020-03-22

多一份存储介质，就要维护两个存储介质的数据一致性，然而两个介质的更新是很难做到同时的，总有个先后顺序



1

