

# 基于会话状态的购物车系统<sup>①</sup>

潘建建, 管 群

(四川大学 计算机学院, 成都 610065)

**摘 要:** 购物车作为电子商务领域中广为使用的模块, 它的设计好坏关系到整套系统的有效运行. 购物车是系统与用户交互最多的模块之一, 不管是从用户体验上还是从功能上, 其影响力都是不可忽视的. 但有时为了营造出更好的用户体验, 会从结构上对于购物车进行一番调整以适应功能上的要求. 虽然这种调整不是每次都是必须的, 但其结果影响却是巨大, 如果不能深入理解购物车背后的功能与原理就不能做出有效的调整, 最后会导致系统性能变差. 正是在以上需求前提下, 对于购物车的几种设计方案进行了详细的对比、分析, 从多个方面说明了购物车的设计原理. 并且针对不同方案的优缺点及实际中遇到的一些问题提出了改进. 最后还讨论了在日常使用中购物车的一些异常情况以及补救措施.

**关键词:** 电子商务; 购物车; 会话状态

## Shopping-Cart Design Based on Session State

PAN Jian-Jian, GUAN-Qun

(College of Computer Science, Sichuan University, Chengdu 610065, China)

**Abstract:** Shopping-cart is a very popular module in the field of E-commerce, its designing related to the entire system's effective operation. Shopping-cart, as the most used interactive module, whether from the aspect of user experience or functionality, cannot be ignored. Sometimes, in order to create a better user experience, some adjustments to the structure are made to accommodate the functional requirements. Although these adjustments are not necessary, it's very useful. Only understand the features and principles behind the system can build shopping-cart that meet the needs of most people, otherwise it may lead to poor performance to the whole system. In this context, this paper analyzes the pros and cons of different options, then put forward some improvements. At last, it talks about some exceptions during the daily use and gives a remedy.

**Key words:** e-commerce; shopping-cart; session state

## 1 概述

“购物车”作为生活中的常见品为网络购物车的设计提供了现实参考的依据. 这里不仅是简单的复制而且是对现实购物过程的一种模拟. 这种模拟有利于提升网站的亲和力, 可以让顾客很快熟悉这种购物流程. 一方面它改善了网站的用户体验, 增强了易用性; 另一方面也提高了网站的影响力与竞争力.

购物车作为信息容器的一种, 它类似于订单而又不同于订单, 这里其实是两种不同的概念. 在整个流

程中, 购物车处于中间态, 而订单处于终止态. 两者有相似之处——信息结构相同; 也有不同之处——流程状态不同. 状态不同就意味着作用不同, 有时为了方便会把两种概念合二为一, 即订单就是购物车, 购物车就是订单. 这种设计无疑增加了数据系统的复杂度, 在小型系统上运行尚可, 在大型系统上, 这种融合必须分开. 不仅是从性能上考虑, 而且是从用户体验角度考虑. 购物车在整个电子商务体验流程中与顾客关系最为密切, 它是流程之初, 体验之本. 因此如果

<sup>①</sup> 收稿时间:2012-04-12;收到修改稿时间:2012-05-18

能重点抓住购物车, 提供良好的设计, 那么在以后的运营中会给网站带来无限的商机.

## 2 购物车设计的基础

客户端与服务器端之间的通信本是无状态的, 然而由于某些功能性的需要, 如购物车的设计, 这时不仅要保持与客户端的联系, 并且还需要保存整个会话期间的状态. 这里就需要额外的一种技术——会话状态的支持, 也是因为会话状态, 才使得购物车的设计成为可能.

会话状态<sup>[1]</sup>可以分为三种:

客户会话状态, 在客户端保存数据. 在 web 应用中有几种可选方法: 将数据编码在 URL 中, 使用 Cookie, 串行化数据到 web 表单的隐藏域, 或者把状态保存在肥胖客户端.

服务器会话状态, 在服务器端保存数据. 可以把数据存于内存中, 也可以通过序列化对象的方式把数据存放于文件系统, 或者共享数据源中. 这可以是一张简单的数据库表, 也可以是类似 Memcache 这种缓存系统.

数据库会话状态, 同样是在服务器端保存数据, 通过对会话数据的进一步分解, 以持久的保存在数据库中.

在购物车设计中, 用户状态的保持往往是必须的, 在整个会话期间, 系统要不断跟踪用户直到会话结束. 因此会话保持是购物车设计的基础, 少了它购物车将难以实现.

## 3 不同方案之间的对比

对比以上三种方式可以发现, 它们之间存在着显著的共性——存储特性. 那么这种特性对于购物车的设计有着怎样的作用?

最重要的就是起到了中间过渡的作用. 一般来说购物车是存在于信息传递的中间阶段, 它是订单生成过程中信息的一个临时存放点. 尤其在电子商务网站中, 系统对于用户的大部分操作都需要暂存, 而暂存的目的是为了给予用户更多的选择空间. 选择空间越大, 灵活度越高, 用户体验就会越好. 这对于购物车乃至整个网站的设计都是非常重要的, 因为牵扯到订单的生成, 所以临时存储特性必不可少. 既然是临时存储, 这里可以把购物车设计的更加简单一些, 做成

一张临时表. 一次添加操作就是一条数据插入到数据库的过程. 但是这种方案实际效果并不理想, 因此可以考虑引入“临时节点”. 临时节点主要由会话状态组成, 其作用是改进了中间过程, 改善了用户体验, 提升了网站效能.

当然, 临时节点也可以不是必须的, 从图 1 可以看出, 在部分情况下客户端可以与购物车直接交互. 如在用户登陆之后, 数据就不必通过临时节点而直接进入购物车.

通常情况下, 临时节点多使用服务器会话状态为主, 其次是数据库会话状态. 不同方案优缺点各不相同, 如果要发挥临时节点的最大效能, 就需要从会话状态的特性说起.

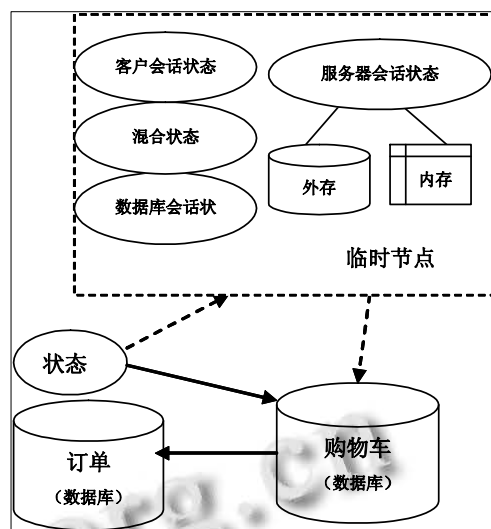


图 1 临时节点的作用

### 3.1 使用客户会话状态

状态保持以 Cookie 方式为主. 主要从两方面考虑: 一方面是考虑到 Cookie 存储状态的特性; 另一方面是考虑到 Cookie 存储标识的特性.

利用 Cookie 存储状态的特性搭建网络购物车, 这里多用在轻量级网站上. 这种方案最大的特点是数据存放在客户端<sup>[2]</sup>, 有效减少客户端与服务器端之间的通信次数, 缓解了服务器端的压力. 并且在数据持久化方面做的也不错, 很适合长期存储.

不足之处有以下几个方面:

一方面, 有大小和数量的限制; 另一方面, 有安全方面的不足<sup>[3]</sup>. 由于对客户端的过度依赖, 通常系统不会把复杂、关键的数据存储到客户端, 因为牵扯

到编码、解码的问题,这里无疑会增加系统复杂度.并且由于这种依赖,还会导致购物车系统的不稳定性,如 Cookie 的禁用,就会导致购物车系统的瘫痪.因此可以看出 Cookie 并不适合复杂任务的存储.

### 3.2 服务器会话状态

大部分数据存于服务器端,这里带来的最大好处是安全性的提高、信息量与简易度的改善,给购物车的设计带来了极大的方便<sup>[4]</sup>.不足之处是相对于 Cookie 存储而言与服务器交互过于频繁,给服务器带来了不小压力.不过这里可以通过其他的方式来改善.

在服务器端,数据通常是以文件或 Web 进程的形式存储.以文件形式存储会随着服务器端访问量的增加,给 I/O 带来不小的压力,降低了系统整体吞吐量.以 Web 进程形式存储,对于内存提出了更高要求,并且由于数据与 Web 服务器的关系过于紧密,一方面会大量占用内存,给服务器造成不必要负担,另一方面也会存在潜在数据安全隐患——数据的丢失率会因服务器不稳定性而上升.并且上述两种方式都存在跨域访问的问题,不利于信息共享.所以最好的方式是把这种结构分离出来独立运行.这里可以通过 Cookie<sup>[5]</sup>、共享内存、数据库等方式来加以改善.这几种方式分别在速度、共享性以及持久性方面都做的不错,而且都解决了跨域访问的问题.

上述改进虽然进一步提高了服务器会话状态的性能,但相对于常规会话状态保持,购物车的设计对数据持久化提出了更高要求.这里不仅是从技术的角度考虑,而且是从可行性、可用性方面考虑的结果.

在购物车设计当中,正常的体验流程如下所示:

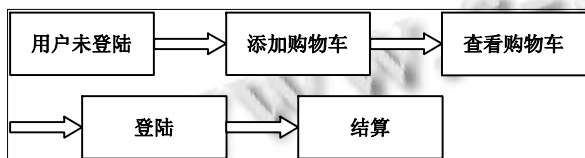


图 2 购物车的体验流程

在流程的开始阶段,当用户发现想要购买的商品时,第一反应是“添加购物车”.在这个过程中,系统首先要判断用户是否已经登录,如果已经登录,系统可以把商品直接存入购物车或者把商品先存入临时节点中,在随后某一时刻再把数据转存到最终数据库中.这一时刻可以是结账时刻、用户注销时刻或者是会话过期时刻.总之,这里不会出现数据丢失的现象.

然而,用户如果在没有登陆之前就添加购物车,情况就没有那么简单了.从“未登录”到“添加购物车”再到“登陆”,这一系列过程如果是连续的,那么系统依然是能跟踪到用户,但如果用户是在“添加购物车”之后许久(也许是几小时,也许是几天)再返回登录,那么系统就未必能识别出用户.不能识别就意味着用户数据的丢失.从某种程度上说,这会给顾客带来极大的失落感,乃至会影响到整个网站的声誉.

原因多数与方案本身的特性有关,如果开始就选择服务器会话状态,就决定了其后期稳定性与持久性会很一般,再加上会话时间的限制,所以服务器会话状态并不适合长期数据存储.根本原因是由于系统大部分操作都发生在内存中,为了获得良好的性能而牺牲了持久性.

从总体上看,服务器会话状态还是十分理想的,它承担购物车设计中的大部分功能.并且在速度与简易性方面都做到了很好的平衡.可尽管优点多多,数据持久性问题还是需要解决的,这里可以引入数据库会话状态.

### 3.3 数据库会话状态

数据库会话状态是解决数据持久化问题很好的方案之一.这种方案不仅数据持久性好并且在集群和故障恢复方面也很出色.最大不足是对每个请求都要进行数据库读/写操作,性能方面很不理想.数据库会话状态由于其性能原因往往多与其他几种方案配合使用.从某种程度上说,这种方案解决了购物车持久化方面的问题,但在性能方面还有一定的优化空间.

### 3.4 混合存储方案

前三种方案优缺点明显,但如果想构建出一个体系良好的设计,就必须从根本出发,找出它们的契合点.结合前文所述可以总结得出:客户会话状态适合简单数据存储,服务器会话状态适合短期数据存储,数据库会话状态适合长期数据存储.服务器会话状态主要不足是在数据持久层面,因此这里可以与 Cookie、缓存、临时数据库在一起使用以解决性能和持久化方面的问题.使用 Cookie 主要是为了跟踪用户,保障客户端、缓存与数据库之间的状态的一致性.临时数据库和缓存在这里充当了临时节点,结构从过去的一级改变为现在的多级结构(如图 3).系统只有在缓存不命中的情况下才会去查询数据库.多级结构最大好处是提高了系统并发性,其中大部分操作都发生在

内存中,最后也只有特殊情况如缓存失效或者过期的情况下才会把数据存入临时数据库中.从图 4 可以看出,在没有缓存的情况下,临时数据库的访问几乎与购物车请求成正比.增加缓存以后,系统效率明显提高,临时数据库的访问次数大大减少.

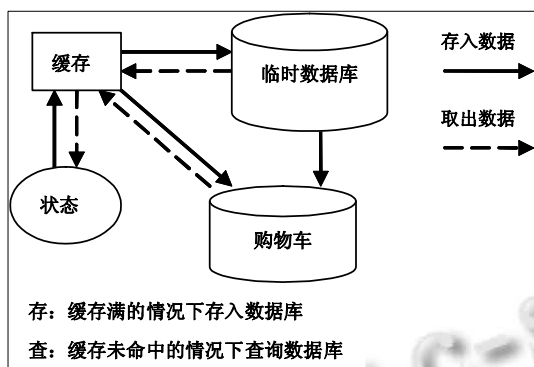


图 3 混合多级结构

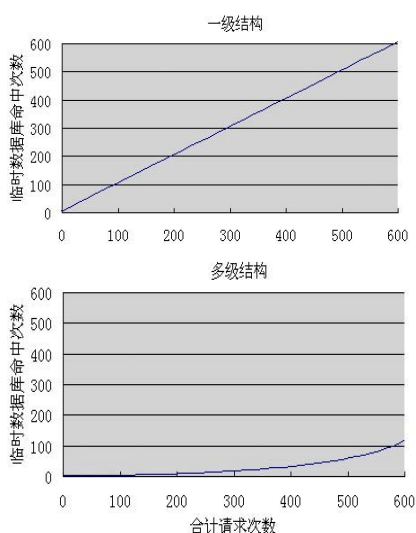


图 4 对比系统改进后的效果

这种结构解决了当前所遗留的大部分问题,而且在性能和稳定性方面也有所提高.当然,所付出的代价也是非常昂贵的——庞大的结构,复杂的冗余度不利于系统前期的构建以及后期的维护,因此,这种架构适合在规模性系统上使用.

在设计过程中还有一个问题有待考究——禁用 Cookie 所造成问题.虽然可以使用其他方法保存客户会话状态,比如 URL 重写的方式,但是这也只是在特殊条件下才能使用,如果是用在规模性系统上,如电子商务网站的购物车,其兼容性就非常差,不利于

整套系统的规划,并且从用户体验、SEO 优化角度考虑,这种方法也很不理想,更重要的是它依然没能解决数据持久化方面的问题.

禁用 Cookie 会造成一系列问题,比如说用户不能登录,购物车不能添加,用户浏览历史不能跟踪等等.这一切都是禁用 Cookie 后造成的.暂时还没有一个有效的办法来解决这种问题.但是从整体上看这种异常情况发生的几率还是很低的,因此综合考虑我们完全可以忽略这种不足.但是从另一个方面来看,为了给用户营造更好的体验,网页上适当提示也不乏为一种很好的补救措施.这里的亚马逊做的就不错(如图 5).

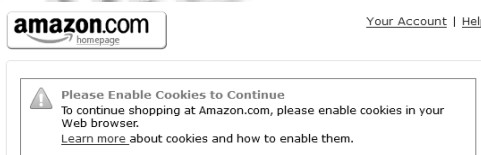


图 5 Amazon 禁用 Cookie 后的提示

#### 4 总结

购物车设计的好坏关系到整个电子商务网站的品质,选择一个合理的方案对于不同阶段的电子商务网站来说是十分必要的,不同的阶段有着不同的目的,如何做到合理选择恰当运用,这就是一个度的问题.最后也只有满足于需求的设计才算是好的设计.

#### 参考文献

- 1 Fowler M.王怀民,周斌译.企业应用架构模式.北京:机械工业出版社,2011.58-60.
- 2 Alur D, Crupi J, Malks D. J2EE 核心模式.2<sup>nd</sup> ed.北京:机械工业出版社.2005,14-15.
- 3 Guy P, Ahmed S, Ines A. Secure session management with cookies. Communications and Signal Processing, IEEE Computer Society, 2009: 1-6.
- 4 Xu B, Xie SY. Research of session security management in E-Commerce system. International Symposium on Information Engineering and Electronic Commerce, IEEE Computer Society. 2009: 796-799.
- 5 Joon P, Harish K. Trusted identity and session management using Secure Cookies. Lecture Notes in Computer Science, Springer Verlag, 2005: 310-324.