

购物车设计：

总体上分为五层。其中商品、用户数据这些用关系数据库进行持久化存储，同时也会做一些缓存。而整个架构采用 RPC 和服务化的思路。

一、接入层：

负载均衡：除了负责接入的机器，还需要增加一层负载均衡，将负载请求分配到不同的 web 处理机器，更进一步将下面的静态内容缓存返回给用户。通过这样，做到高效和安全，防止某台机器服务过载。

二、静态内容缓存层

对一些静态数据进行缓存，此层的数据直接返回给用户，而不再请求 Web 处理层。

三、Web 处理层

这一层，主要是对用户请求进行接入校验、解析等，然后通过 RPC 的方式请求业务逻辑层的具体服务。这样可以防止大量直接请求数据库，造成过载。

四、业务逻辑层

这一层主要是业务逻辑的相关服务，包括：下单服务、购物车添加商品服务、商品查询服务、账户查询服务等等。这一层会进行业务处理，并且访问数据存储层。对于一些通知类（比如短信邮件告知用户下单等）可以通过异步处理的方式，而不是直接放到主流程中。

五、数据存储层

- a) 缓存：memcache、redis 等存储热门商品信息，用户信息等等
- b) 关系数据库层：订单、用户信息等等数据的持久化。这里需要做到读写分离，采用主从分布式的架构，并且对数据库进行功能划分，比如区分离线/在线处理，划分到不同的库和机器。尤其是将可能引起慢查询、复杂查询的业务与其他业务查询区分开。
- c) 文档型数据库：mongodb 用来存储无具体格式化的数据。比如用户行为信息等。
- d) Hadoop HDFS: 大数据分布式存储