

02 | 流量大、数据多的商品详情页系统该如何设计？

李玥 · 后端存储实战课



你好，我是李玥。

今天这节课我们看一下，如何设计一个快速、可靠的存储架构支撑商品系统。

相对于上节课提到的订单系统，电商的商品系统主要功能就是增删改查商品信息，没有很复杂的业务逻辑，支撑的主要页面就是商品详情页（下文简称：商详）。不过，设计这个系统的存储，你仍然需要着重考虑两个方面的问题。

第一，要考虑高并发的问题。不管是什么电商系统，商详页一定是整个系统中 DAU（日均访问次数）最高的页面之一。这个也不难理解，用户购物么，看商详了不一定买，买之前一定会看好多商详货比三家，所以商详的浏览次数要远比系统的其他页面高。如果说，在设计存储的时候，没有考虑到高并发的問題，大促的时候，支撑商详页的商品系统必然是第一个被流量冲垮的系统。

第二，要考虑的是商品数据规模的问题。商详页的数据规模，我总结了六个字，叫：数量多，重量大。

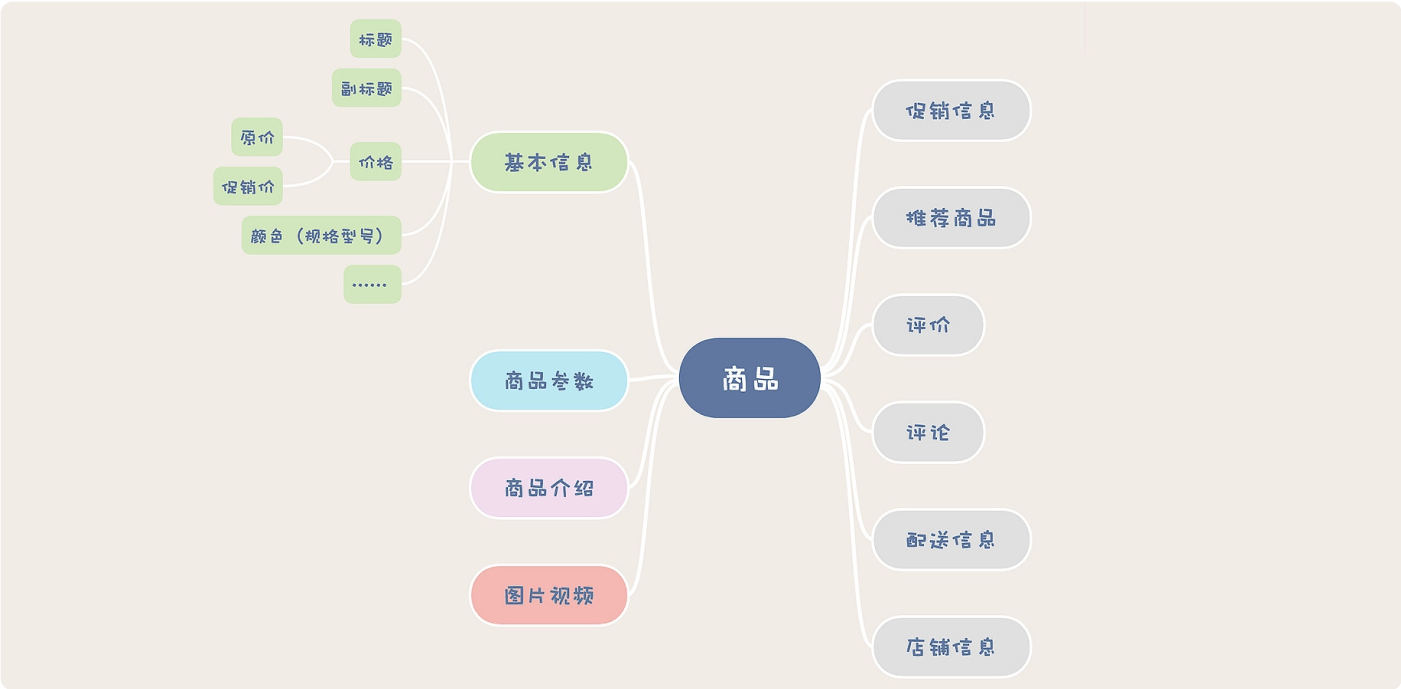
先说为什么数量多，国内一线的电商，SKU（直译为：库存单元，在电商行业，你可以直接理解为“商品”）的数量大约在几亿到几十亿这个量级。当然实际上并没有这么多种商品，这里面有很多原因，比如同一个商品它有不同的版本型号，再比如，商家为了促销需要，可能会反复上下架同一个商品或者给同一个商品配不同的马甲，这都导致了 SKU 数量爆炸。

再说这个“重量大”，你可以打开一个电商商详页看一下，从上一一直拉到底，你看看有多长？十屏以内的商详页那都叫短的，并且这里面不光有大量的文字，还有大量的图片和视频，甚至还有 AR/VR 的玩法在里面，所以说，每个商详页都是个“大胖子”。

支持商品系统的存储，要保存这么多的“大胖子”，还要支撑高并发，任务艰巨。

商品系统需要保存哪些数据？

先来看一下，一个商详页都有哪些信息需要保存。我把一个商详页里面的所有信息总结了一下，放在下面这张思维导图里面。



这里面，右边灰色的部分，来自于电商的其他系统，我们暂且不去管这些，左边彩色部分，都是商品系统需要存储的内容。

这么多内容怎么存？能不能像保存订单数据那样，设计一张商品表，把这些数据一股脑儿都放进去？一张表存不下就再加几张子表，这样行不行？你还真别说不行，现在这些电商大厂，在它们发展的早期就是这么干的。现在那么复杂的分布式存储架构，都是一点儿一点儿逐步演进过来的。

这么做的好处，就是糙快猛，简单可靠而且容易实现，但是，撑不了多少数据量，也撑不了多少并发。如果说，你要低成本快速构建一个小规模电商，这么做还真就是一个挺合理的选择。

当然，规模再大一点儿就不能这么干了。不能用数据库，那应该选择哪种存储系统来保存这么复杂的商品数据呢？任何一种存储都是没办法满足的，解决的思路是**分而治之**，我们可以把商品系统需要存储的数据按照特点，分成商品基本信息、商品参数、图片视频和商品介绍几个部分来分别存储。

商品基本信息该如何存储？

我们先来分析商品的基本信息，它包括商品的主副标题、价格、颜色等一些商品最基本、主要的属性。这些属性都是固定的，不太可能会因为需求或者不同的商品而变化，而且，这部分数据也不会太大。所以，还是建议你在数据库中建一张表来保存商品的基本信息。

然后，还需要在数据库前面，加一个缓存，帮助数据抵挡绝大部分的读请求。这个缓存，你可以使用 Redis，也可以用 Memcached，这两种存储系统都是基于内存的 KV 存储，都能解决问题。

接下来我和你简单看一下，如何来使用前置缓存来缓存商品数据。

处理商品信息的读请求时，先去缓存查找，如果找到就直接返回缓存中的数据。如果在缓存中没找到，再去查数据库，把从数据库中查到的商品信息返回给页面，顺便把数据在缓存里也放一份。

更新商品信息的时候，在更新数据库的同时，也要把缓存中的数据给删除掉。不然就有可能出现这种情况：数据库中的数据变了，而缓存中的数据没变，商详页上看到的还是旧数据。

这种缓存更新的策略，称为 **Cache Aside**，是最简单实用的一种缓存更新策略，适用范围也最广泛。如果你要缓存数据，没有什么特殊的情况，首先就应该考虑使用这个策略。

除了 Cache Aside 以外，还有 Read/Write Through、Write Behind 等几种策略，分别适用于不同的情况，后面的课程中我会专门来讲。

设计商品基本信息表的时候，有一点需要提醒你的是，**一定要记得保留商品数据的每一个历史版本**。因为商品数据是随时变化的，但是订单中关联的商品数据，必须是下单那个时刻的商品数据，这一点很重要。你可以为每一个历史版本的商品数据保存一个快照，可以创建一个历史表保存到 MySQL 中，也可以保存到一些 KV 存储中。

使用 MongoDB 保存商品参数

我们再来分析商品参数，参数就是商品的特征。比如说，电脑的内存大小、手机的屏幕尺寸、酒的度数、口红的色号等等。和商品的基本属性一样，都是结构化的数据。但麻烦的是，不同类型的商品，它的参数是完全不一样的。

如果我们设计一个商品参数表，那这个表的字段就会太多了，并且每增加一个品类的商品，这个表就要加字段，这个方案行不通。

既然一个表不能解决问题，那就每个类别分别建一张表。比如说，建一个电脑参数表，里面的字段有 CPU 型号、内存大小、显卡型号、硬盘大小等等；再建一个酒类参数表，里面的字段有酒精度数、香型、产地等等。如果说，品类比较少，在 100 个以内，用几十张表分别保存不同品类的商品参数，这样做也是可以的。但是，有没有更好的方法呢？

大多数数据库，都要求数据表要有一个固定的结构。但有一种数据库，没有这个要求。特别适合保存像“商品参数”这种，属性不固定的数据，这个数据库就是 MongoDB。

MongoDB 是一个面向文档存储的 NoSQL 数据库，在 MongoDB 中，表、行、列对应的概念分别是：collection、document、field，其实都是一回事儿，为了便于你理解，在这里我们不

咬文嚼字，还是用“表、行、列”来说明。

MongoDB 最大的特点就是，它的“表结构”是不需要事先定义的，其实，在 MongoDB 中根本没有表结构。由于没有表结构，它支持你把任意数据都放在同一张表里，你甚至可以在一张表里保存商品数据、订单数据、物流信息等这些结构完全不同的数据。并且，还能支持按照数据的某个字段进行查询。

它是怎么做到的呢？MongoDB 中的每一行数据，在存储层就是简单地被转化成 BSON 格式后存起来，这个 BSON 就是一种更紧凑的 JSON。所以，即使在同一张表里面，它每一行数据的结构都可以是不一样的。当然，这样的灵活性也是有代价的，MongoDB 不支持 SQL，多表联查和复杂事务比较孱弱，不太适合存储一般的数据。

但是，对于商品参数信息，数据量大、数据结构不统一，这些 MongoDB 都可以很好的满足。我们也不需要事务和多表联查，MongoDB 简直就是为了保存商品参数量身定制的一样。

使用对象存储保存图片和视频

图片和视频由于占用存储空间比较大，一般的存储方式都是，在数据库中只保存图片视频的 ID 或者 URL，实际的图片视频以文件的方式单独存储。

现在图片和视频存储技术已经非常成熟了，首选的方式就是保存在对象存储（Object Storage）中。各大云厂商都提供对象存储服务，比如国内的七牛云、AWS 的 S3 等等，也有开源的对象存储产品，比如 MinIO，可以私有化部署。虽然每个产品的 API 都不一样，但功能大同小异。

对象存储可以简单理解为一个无限容量的大文件 KV 存储，它的存储单位是对象，其实就是文件，可以是一张图片，一个视频，也可以是其他任何文件。每个对象都有一个唯一的 key，利用这个 key 就可以随时访问对应的对象。基本的功能就是写入、访问和删除对象。

云服务厂商的对象存储大多都提供了客户端 API，可以在 Web 页面或者 App 中直接访问而不用通过后端服务来中转。这样，App 和页面在上传图片视频的时候，直接保存到对象存储中，然后把对应 key 保存在商品系统中就可以了。

访问图片视频的时候，真正的图片和视频文件也不需要经过商品系统的后端服务，页面直接通过对象存储提供的 URL 来访问，又省事儿又节约带宽。而且，几乎所有的对象存储云服务都自带 CDN（Content Delivery Network）加速服务，响应时间比直接请求业务的服务器更短。

国内的很多云厂商的对象存储对图片和视频，都做了非常多的针对性优化。最有用的是，缩放图片和视频转码，你只要把图片和视频丢到对象存储中，就可以随时获得任意尺寸大小的图片，视频也会自动转码成各种格式和码率的版本，适配各种 App 和场景。我只能说，谁用谁知道，真香！

将商品介绍静态化

商品介绍在商详情页中占得比重是最大的，包含了大量的带格式文字、图片和视频。其中图片和视频自然要存放在对象存储里面，商品介绍的文本，一般都是随着商详情页一起静态化，保存在 HTML 文件中。

什么是静态化呢？静态化是相对于动态页面来说的。一般我们部署到 Tomcat 中的 Web 系统，返回的都是动态页面，也就是在 Web 请求时，动态生成的。比如说商详情页，一个 Web 请求过来，带着 SKUID，Tomcat 中的商详情页模块，再去访问各种数据库、调用后端服务，动态把这个商详情页拼出来，返回给浏览器。

不过，现在基本上没有系统会这么干了，你想，对于每个 SKU 的商详情页，你每次动态生成的页面内容不是完全一样的么？生成这么多次，不仅浪费服务器资源，速度还慢，关键问题是，Tomcat 能抗的并发量和 Nginx 完全不是一个数量级的。

商详情页的绝大部分内容都是商品介绍，它是不怎么变的。那不如就把这个页面事先生成好，保存成一个静态的 HTML，访问商详情页的时候，直接返回这个 HTML。这就是静态化。

商详情页静态化之后，不仅仅是可以节省服务器资源，还可以利用 CDN 加速，把商详情页放到离用户最近的 CDN 服务器上，让商详情页访问更快。

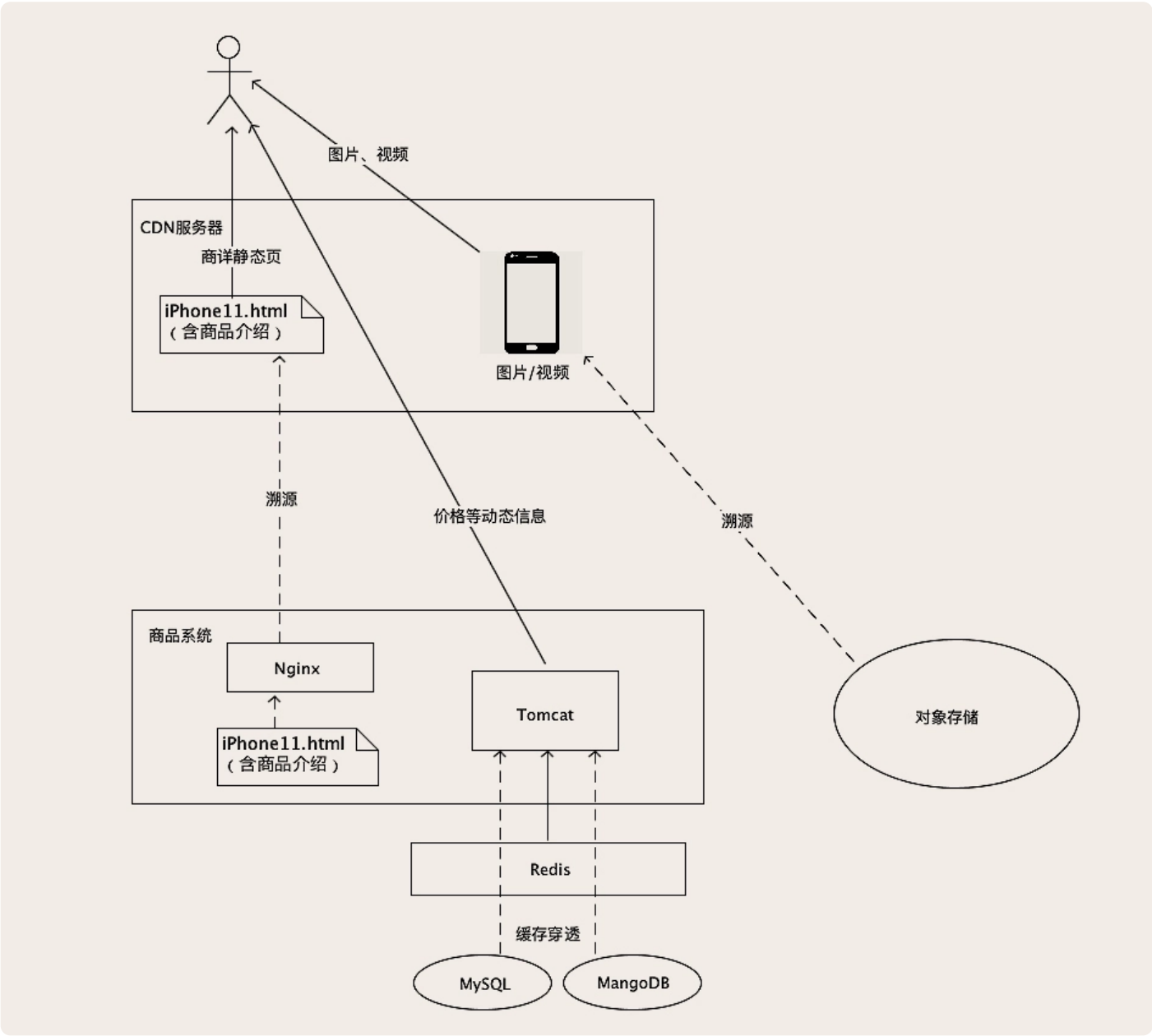
至于商品价格、促销信息等这些需要频繁变动的信息，不能静态化到页面中，可以在前端页面使用 AJAX 请求商品系统动态获取。这样就兼顾了静态化带来的优势，也能解决商品价格等信

息需要实时更新的问题。

小结

最后，我们再来对今天的内容复个盘。商品系统的存储需要提供商品的基本信息、商品参数、图片和视频以及商品介绍等等这些数据。商品的基本信息和商品参数分别保存在 MySQL 和 MongoDB 中，用 Redis 作为前置缓存，图片和视频存放在对象存储中，商品介绍随着商详页一起静态化到商详静态页中。

我把商品系统的存储绘制成下面这张图：



一起来看一下图，这样一个商品系统的存储最终的效果是什么样的？图中实线表示每访问一次商详页，需要真正传输的数据，虚线表示当商详页数据发生变化的时候才需要进行一次数据传输。用户打开一个 SKU 的商详页时，首先去 CDN 获取商详页的 HTML，然后访问商品系统获取价格等频繁变化的信息，这些信息从 Redis 缓存中获取。图片和视频信息，也是从对象存储的 CDN 中获取。

分析一下效果，数据量最大的图片、视频和商品介绍都是从离用户最近的 CDN 服务商获取的，速度快，节约带宽。真正打到商品系统的请求，就是价格这些需要动态获取的商品信息，一般做一次 Redis 查询就可以了，基本不会有流量打到 MySQL 中。

这样一个商品系统的存储的架构，把大部分请求都转移到了又便宜速度又快的 CDN 服务器上，可以用很少量的服务器和带宽资源，抗住大量的并发请求。

思考题

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？

这样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。

课后请你想一下，该怎么来解决这个问题？欢迎你在留言区与我交流互动。

感谢你的阅读，如果你觉得今天的内容对你有所帮助，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。

精选留言 (90)



李玟 置顶

2020-02-26

hi, 我是李玥。跟上节课一样, 我还是在留言板上同步一下上节课的思考题, 大家一起来学习探讨。

上节课我们讲了两种实现幂等的方法, 课后呢, 我也让你思考了下, 在你负责开发的业务系统中, 能不能用这节课中讲到的方法来实现幂等? 除了这两种方法以外, 还有哪些实现服务幂等的方法?

关于这个问题, 我是这么看的。

其实总结下来这些实现幂等的方法, 无非是两大类, 一类是通过一些精巧的设计让更新本身就是幂等的, 这种需要点儿运气, 不是所有业务都适用的。另外, 就是利用外部的、具备一致性的存储(比如说MySQL)来做冲突检测, 你在设计幂等方法的时候一般都可以顺着这两个思路来开展。

共 2 条评论 >

👍 66



京京beaver

2020-03-03

一般价格系统是按照价格版本来配置价格的, 比如1月1日-2月1日卖100元, 有个版本id。价格的调整一般流程比较严格, 因为涉及到扣点, 供货价的调整, 财务结算方式。一般前端用的比较多的是用促销活动, 来调整用户的到手价。

如果确实有调价发生, 会刷新商品信息服务, 那里面冗余一份价格数据, 这个延迟一般是秒级完成。如果用户详情页看到一个价格, 下单时会重新检查当前价格, 如果发生变更, 一般会在购物车或者立即购买页面给出提醒, 重新刷新当前页面来解决。

共 4 条评论 >

👍 98



老姜

2020-02-27

mysql支持json数据类型了, 是不是可以不用mongodb了, 多一个数据库, 系统就会更复杂

作者回复: 是的, 如果能满足业务需求的话, 尽量不要用太多的技术。

共 8 条评论 >

👍 59



漏脚脖

2020-02-27

老师你好, 商品介绍静态化的地方不太懂想请教一下

我们公司现在的做饭是前端页面直接通过ajax请求数据, 我理解的是静态化之后, 商品介绍这

部分就不用请求接口了，那这部分数据也需要初始化吧？

所以，我的第1个问题是，商品介绍这部分数据是怎么初始化的呢？难道是初始化到html文件里吗？我的第1个问题是，商品介绍这部分数据是怎么初始化的呢？难道是初始化到html文件里吗？

第2个问题，如果问题1成立，那是不是没个商品的介绍都要写到一个html文件里？那几亿个sku怎么做呢，要几亿个html？不太可能吧

这块老师能展开说一下吗，不太懂具体的实现

作者回复：第一个问题，初始化的方式一般就是在商家后台修改了商详页中的商品介绍之后，重新生成一个该SKU的新的商详页HTML文件，这个HTML文件的内容直接就包含商品介绍等大段的文字。然后把这个HTML推送到CDN上，或者等CDN回源来拉取。

第二个问题，是的，每个SKU就会生成一个HTML。

对于像京东淘宝这样的超大规模电商，它们的做法又不一样了，它们的商详页会被划分得非常细，可能会被分为几十上甚至百部分内容，每一部分可能是静态化的HTML片段，也可能是走的缓存，或者是动态生成的，取决于后面支撑这一小块儿内容的系统了。

共 14 条评论 >

👍 39



Din

2020-02-26

下单前先调用校验价格的接口，如果价格已经发生了变化，提示用户刷新页面。

共 1 条评论 >

👍 34



Ling

2021-05-25

1. 作者讲了什么？

一个电商系统，流量集中的一个业务：商品详情系统的存储应该设计

2. 作者是怎么把事情说明白的？

通过将电商详情的数据需求出发，说明数据多样，无法一个系统解决，需要分而治之

– 从存储层面，数据区分为：固定结构数据、非固定结构数据、富媒体数据；

- 从读取层面，将数据分为：经常变化数据、非经常变化数据

3. 为了讲明白，作者讲了哪些要点？哪些是亮点？

****要点**：**

1. 从数据存储到哪的角度：

- 固定结构数据：

商品主标题、副标题、价格，等商品最基本、最主要的信息（任何商品都有的属性）

存储到：MySQL中

- 非固定结构数据：

商品参数，不同类型的商品，参数基本完全不一样。电脑的内存大小、手机的屏幕尺寸、酒的度数、口红的色号等等。

存储到不需要固定结构的存储：MongoDB或者MySQL的json字段中

- 富媒体数据：

商品的主图、详细介绍图片、视频等富媒体数据

存储到：对象存储。并且通过客户端直接调用对象存储的API，得到媒体资源在对象存储中的ID或者URL之后，将ID或者URL提交到MySQL中。

2. 从数据读取角度：

- 存储到MySQL中的数据，需要设计一层缓存层，应对高并发读

- 对于不经常变动的数据：

静态化后，上CDN；

其实也可以不用生成html文件，直接把动态输出页面的接口作为CDN源站。

- 对于经常变动的数据：价格、促销信息

通过ajax接口动态获取

- 富媒体数据：

将对象存储设置为CDN源站，用户通过CDN访问富媒体资源

****亮点**：**

通过一张图，非常清晰的总结了整体的设计思路，很赞！

共 2 条评论 >

👍 19



蹦蹦~沙卡拉卡

2020-03-24

老师，我问下，商品每次变化都要保存一个快照。假如商品表是table_a, 快照表是table_b, 那么订单关联的商品是关联 商品表 table_a 还是 快照表 table_b呢？ 我的理解是关联快照表。

作者回复: 是的，关联快照。



👍 19



何妨

2020-02-28

希望老师可以再展开讲讲商品介绍静态化这一块，有些意犹未尽，还是想详细了解一下，感觉看到了些门道但还是有些模糊

作者回复: 其实静态化都是“过时”的技术了。但是特定的场景还是非常有用。

现在的动态Web页面，都是用JS请求动态数据，在浏览器中填充内容。

早期的Web页都是Server端渲染，比如Java中的thymeleaf，FreeMarker 模板引擎，包括PHP本身就是个模板引擎，这些都是在服务端来填充好动态内容，再返回给浏览器的。

页面静态化就是利用这些模板引擎技术，事先就把页面中的动态内容填充好，生成一个一个静态的HTML。

这种静态化技术只适合页面内容很多，页面又不会经常变化的场景。比如我们电商的商详页，再比如很多新闻、咨询类的网站（一篇新闻发了之后很少会修改）。

共 8 条评论 >

👍 18



MadDog

2020-02-27

请问elasticSearch可以替换mongoDB嘛？elasticSearch也可以通过Dynamic field mappings做到类似功能

作者回复: 从动态字段这个功能上说，这俩数据库是可以互相替换的，哪个更好还是要看业务需求。

大多数场景下mongo写性能更好一些，ES更容易维护，功能也更丰富，但也有一些缺陷，比如深分页的问题，SQL支持还不是特别完善等等。

个人认为ES的前景更好一些，大家怎么看？

共 11 条评论 >

👍 18



winzheng

2020-03-16

价格的变化：应该以用户体验为中心，商品加上版本号，根据用户提交时的版本号价格为准；如果用户是停留较长时间，需要给出超时商品变化的提示，起码用户可以理解这个问题，例如飞机票，停留时间长，价格变化，会給用户提示，当然，飞机票变更频率确实比商品高。

共 2 条评论 >

👍 17



木木三

2020-02-27

谁用谁知道，真香

共 1 条评论 >

👍 8



萨秋

2020-03-17

老师您好 想请问下静态化这块 如何做seo优化呢 类似于价格等信息都是爬虫主要抓取的 如果做成异步的话数据源就会有缺失 类似这样的问题

作者回复: 做SEO还是主要提供商品内容, 很多电商都不希望价格被抓走, 还要做各种措施来防止爬虫抓取价格。



👍 5



大秦皇朝

2020-03-04

李玥老师您好:

上传了图片和视频后, 但是又没有提交数据, 那么服务器上是否会出现没有用的冗余/垃圾数据? 那这块怎么判断和处理呢?

作者回复: 确实可能有这种情况。

没精力的话可以不管, 反正垃圾数据也不太多, 存储也便宜。

等积累到一定量了, 再写个任务定期清理一下就可以。

共 3 条评论 >

👍 6



大土豆

2022-02-07

现在前后端分离之后, 做法有变, 页面静态化基本都是前端来做了。BFF全部是由前端负责, 通常NodeJS来实现BFF, 做请求合并, SSR, 页面静态化, 和后端没有半毛钱关系了, 后端只返回数据, 没有接触任何页面相关的东西, 后端提供的全部都是动态的数据型接口。

共 1 条评论 >

👍 5



子瞻

2020-02-27

下单的时候, 生成商品价格, 价格版本等的快照数据, 结算时, 和库里的价格版本信息对比, 一致则进入支付, 不一致则执行价格异常调整策略



👍 4



励研冰

2020-04-09

我做电商的时候在下单前会有一个Check的接口, 客户端会上传预期的价格, 后端检查发现价格发生了变化会返回提示让刷新页面, 再重新Check



kamida

2020-03-23

老师 商品历史版本是在建一个单独的商品历史表 然后给它对于一个时间列吗

作者回复: 一般都是用版本号对应的。



大叶枫

2020-03-12

各种存储特性还是要依据业务特性去设计的，比如说订单表，单条查询走mysql去推进业务更高，毕竟主键查询性能锐减没有多少，批量有全文检索的可以有esdb。所以需要设计人员有时候在设计之初将字段分开设计，有普通检索字段，长文本可变打标字段，可变标记数值字段这类，后面变更存储的成本太大。



leslie

2020-03-03

最合适的解决方式应当是老师之前课程用的消息队列：在redis之上再去加一层消息队列去解决问题；课程跟完后这个就在用了，算是学有所用吧。

共 4 条评论 >



hhhh

2020-02-29

老师想请教下，在存储商品参数的场景，用mysql的json类型存储与mongo, 如何做一下简单的权衡？不胜感激。

作者回复: 存储的选择没有谁好谁不好，还是看场景和需求。

如果只是单纯的存商品参数，MongoDB这种文档数据库更专业一些。

如果从统一技术栈来考虑，使用MySQL JSON能满足需求的话，也是可以的。



