Atitit 高性能架构总结

目录

[1. 架构设计的优势 1](#_Toc21020)

[2. 业务架构 2](#_Toc10227)

[3. 技术架构 3](#_Toc1558)

# **架构设计的优势**

更好的梳理业务的结构体系；

更好的拓展、维护及性能优化；

更好的适应企业业务灵活的推进；

更好的适应大数据的冲洗和应对；

更好的稳定性、低成本及快速迭代；

# 业务架构

根据业务的需求进行子系统模块划分，可以划分为商品子系统、购物子系统、支付子系统、物流子系统、客服子系统、评论子系统；而非核心需求可拆分出客服子系统、评论子系统及接口子系统。另外，根据各个子系统的核心等级，可拆分出核心子系统和非核心子系统，前者包括商品子系统、购物子系统、支付子系统及物流子系统；后者，则包括评论子系统、客服子系统及接口子系统。需要注意的是一般大型电商平台的物流系统是单独分离出来的系统（入库、出库、库存管理、配送管理及货品管理），而这里划分为子系统的主要目的是为演示核心架构，本架构中物流子系统一般作为对接和管理独立子系统的对接模块哦。

1、业务拆分目的

为了解决各个模块子系统间的耦合、维护及拓展性；

方便单独部署子系统，避免集中部署导致一个出问题，全部不能用；

分配专门的团队，负责具体的子系统，最大化工作效率安排；

应对大数据，高压力时，保护核心子系统正常使用；

# **技术架构**

在上面业务架构图基础上，我们需要一个技术架构的演变过程，一切只为满足用户的体验和支撑为前提，所以技术架构的搭建不是一蹴而就的，而是随着业务的不断衍变，系统的架构会逐渐完善更新，以实现应对业务数据量的冲击。

## 1、基本的架构设计50万级别

记得很早的时候，很多中小企业所采用的架构设计十分简单，基本使用一台服务器来满足一切需求部署，比如：一台服务器同时用作应用部署、数据库存储以及图片存储等，不料的是待用户数据达到50万以上，系统出现很多性能问题，尽管对数据库和程序做个各种性能优化，结果仍无明显改善，架构如下：

## 百万级

后来，IT程序猿发现图片的读写严重影响了系统性能，并将图片单独存放在独立服务器中，并且在架构中引入了Cache中间件，比如：Memcache，这种做法是可取的，而且比原来性能提高了1-2个性能级别，架构设计如下：

## 初级的架构设计 1000万

前几年，一般的电商网站的做法是选用三台服务器，一台部署应用，一台部署数据库，一台部署NFS文件系统，做到将各个规模庞大并耗用性能的部分剥离到不同服务器设备，再配备必要的缓存中间件，基本可以满足近1000万的数据量，具体的架构图如下

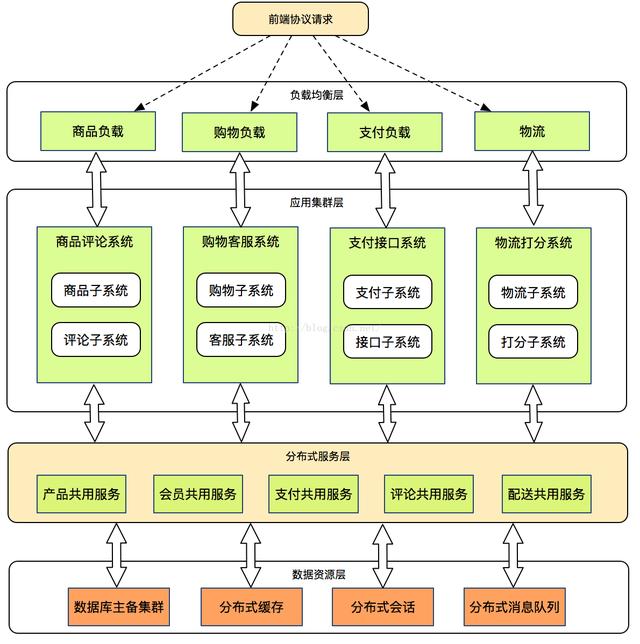
## 目前主流使用的网站架构已经不同，大多采用集群的方式

来实现负载均衡和高可用性，架构可以是下面的样子

## 优化的架构设计 分布式、集群、负载均衡、反向代理、消息队列及多级缓存技术

这里为解决高并发，高可用的大型电商网站的架构设计方案，主要采用了分布式、集群、负载均衡、反向代理、消息队列及多级缓存技术。该架构设计方案，是现今比较流程的大型电商网站采用的架构模式，比如：淘宝、京东等，也许会有细微不同的地方，但大同小异哦！具体的架构图方案如下：

里主要总结的是优化架构，架构按层次结构罗列组织，共分为四层，层次分工明确，高拓展，低耦合，负载均衡、集群、分布式及缓存等技术的使用，架构如下：



****好了，电商平台的架构设计就介绍到这里，****