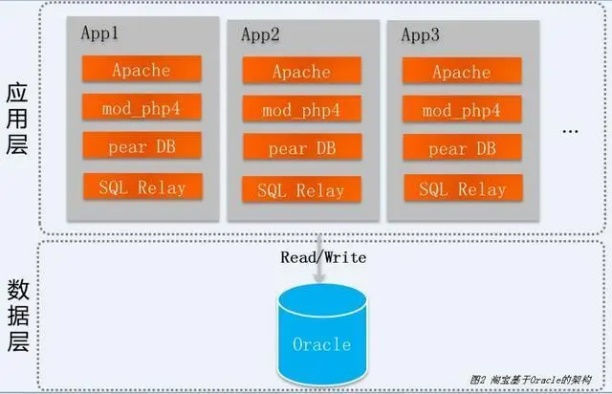
可修改性读书笔记

（此篇笔记是在读完《软件架构实践》这本书后，对第八章的额外补充）

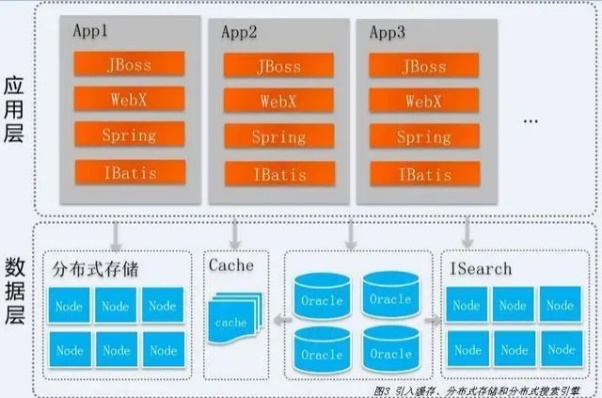
正如文章开篇所引用的：“它不是生存下来的最强壮的物种，也不是最聪明的物种，而是对变化最敏感的物种。”现如今的软件开发往往面临着市场需求的快速改变，在此基础上，客户的需求、用户的需求也会随时改变。同时市场的竞争极其残酷，若不能及时有效的根据需求进行更改，就很容易被竞品取而代之。因此可修改性在这里扮演着至关重要的角色，它决定着软件系统对变化的响应速度和成本，以及在市场所占有的份额。

根据数据统计，典型软件系统的大部分成本发生在最初发布之后，包括但不限于功能改进、纠错、优化以及删除过时功能等。这些占据大量人力物力的活动，是软件维护成本的主要来源，此时如果软件系统具有良好的可修改性，将大大降低运营维护成本。

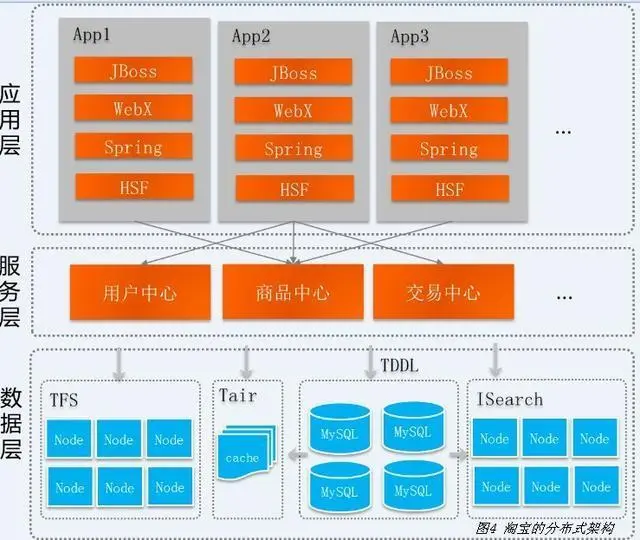
比如早期的淘宝公司用的LAMP架构，用PHP作为网站开发语言， Linux作为操作系统，Apache作为Web服务器，MySQL作为数据库，并仅有10台服务器，适用于处理小型企业的数据量，这样的数据承载量足以满足当时的访问量。



但是随着业务的发展，其数据量爆炸性增长，使得架构不堪重负，终于在2007年，淘宝全年的交易额超过400亿元，平均近1亿多一天，每天有100多万笔交易被创建。这时候如果仍然采用直接访问数据库，会导致数据库压力非常大。于是淘宝将将ISearch搜索引擎改为分布式架构，支持水平扩展，部署了48个节点，从输入端减弱了数据库负荷，并且采用了分布式缓存TDBM（Tair的前身）技术，将这些热点静态数据缓存在内存中，提高访问性能。



随着业务的不断拓展，以及市场的瞬息万变，软件开发的新功能越来越受到重视，只有新功能推出的足够快，才能更好的抢占市场，引领先机。在此基础下，淘宝开始转为分布式架构，将系统进行了拆分，按照用户域、商品域、交易域、店铺域等业务领域进行拆分，建立了20多个业务中心，如商品中心、用户中心、交易中心等。所有有用户访问需求的系统，必须使用业务中心提供的远程接口来访问，不能够直接访问底层的MySQL数据库。这样的安排使得系统架构更加分散，在单个模块之内可以方便的进行功能拓展，以更好的适应市场变化。



经过多次的架构迭代，如今的淘宝网可以作为软件架构的成功案例：

变更被限制在系统的特定部分，而不影响其他模块，就比如数据库的更换和优化可以在不干扰网站其他功能的情况下进行，这符合可修改性的“局部化修改”原则。

模块每个模块只完成一个功能，符合设计模式中的“单一职责原则”，通过模块之间的高内聚和低耦合来维持系统运行。

在各个时期比如双十一、元旦、春节、双十二、618大促等，淘宝系统均能根据时段的不同对系统的颜色或者主题进行调整，并选择不同的优惠方式忙着符合可修改性的“快速响应变化”原则。

在双十一这种极其特殊的高流量事件中，若系统崩溃，将对系统造成不可估量的损失，不过淘宝可以通过临时添加服务器来确保系统运行能力，并扩展其负载量。

淘宝网作为一个典型的成功例证，充分展示了可修改性的重要性，在其2003年创立之初，一路走来经过了多次架构的优化，为的就是满足并适应市场需求，由此可见可修改性直接影响到系统的长期成功和在市场中的竞争力，并为整个系统带来显著的业务价值和市场优势。