

复用产品方案

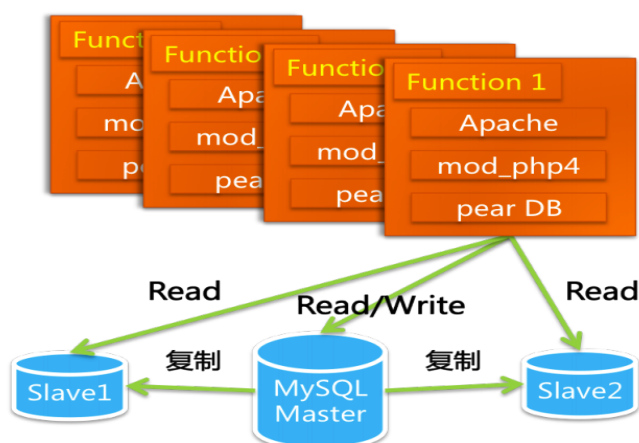
彭程 1352905

(注：作者最近在看《淘宝技术这十年》一书，收益良多，决心以此为切入点，结合软件复用课程所学的知识，做一点研究，将本次讨论课涉及的内容串联起来，以期作为第二次讨论课的解决方案。)

个人时常以为，对于大多数的软件产品，当规模（用户，流量等）有限时，很多问题并不会暴露出来。往往是当项目曾长到一定规模之后，效率，速度，容错率等等先期并不明显的问题的逐渐暴露，才促使人们不得不去解决这些问题。从暂不考虑，只管先实现功能（第一步），到参考业界已有做法（复用软件）来形成自己的解决方案（第二步），到最后因为规模增长到一定程度，没有前人方案可以参考，不得不自己来解决领域问题（第三步）。这也似乎是互联网公司技术逐步发展的必由之路。小到一个课程设计上的服务端与客户端的系统，大到一个微信的实现，都是从一个基本的聊天功能为基础的扩展，然后随着规模增大而不断产生并解决问题的过程（如果课程设计要做大的话）。下面以时间线为轴，以淘宝技术的发展为例，希望能从这个过程中对软件复用形成更深层次的认识。

第一阶段 -- LAMP 架构的网站

LAMP 是（Linux+Apache+Mysql+PHP）组合的缩写，这算是一套完整的小型网站的“解决方案”了，即便是现在用于个人网站的搭建，都不失为一个值得考虑的选择。其优点主要为：无需编译，发布快速，PHP 可以完成从页面渲染到数据访问的所有事情，而且涉及的所有技术都是开源的，这对于一个新兴产品来说是一套不错的解决方案。在淘宝最初期的架构（2003 年）中，工程师们采用了基于 LAMP 的方案，其架构图如下：



可以看到，这时的淘宝没有用到任何原创的技术，而是根据自身电商的特点，在业界已有解决方案的基础上做了相应修改，主要表现在：

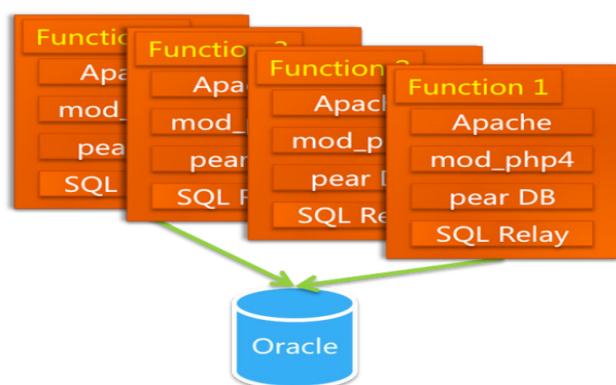
1. 数据库使用了一个主库，两个从库，并且实现了读写分离，这样做的好处在于增加了存储容量，备份的存在使得安全性得到了保证，读写分离也提高了读写效率；
2. 使用了一个 pear DB 模块，用于访问数据库，涵盖了从数据库抽象层到输出缓冲系统的方方面面；

第二阶段 – 个人网站的升级

从第一阶段（淘宝诞生）的过程看来，正如最开始指出，这仅仅是“完成了一个网站的搭建”。随着流量，交易量以及 pv（访问了）的增长，MySQL 已经无法支撑这么大的数据存储量，数据库必须替换解决方案，Oracle 作为当时业界最强大的数据库，其容量，稳定和高性能等成了淘宝的不二选择（更有易于维护的因素）。但更换数据库并不是一蹴而就，说换就换。当时淘宝遇到了如下问题：

Oracle 之所以能实现高并发的访问，主要是通过连接池的设计，每一个连接池保持一个长连接，这些连接都是进程级别了，每一个都需要占用一定的资源，连接的创建和断开的开销都是非常大的。而当时淘宝使用 PHP 语言，其对数据的访问都是直接的，也就是说，每一个请求都需要一个连接，当连接数增长到一定规模后，数据库会经受不了压力而崩溃，但如果使用短连接，频繁的连接与断开会导致性能的下降，带来的用户体验非常不好。

对此，淘宝的工程师通过对一个开源连接池代理服务 SQL Relay 的改造，提供了连接池的功能，这时淘宝的底层数据模型架构发展了较大改变，如下图：

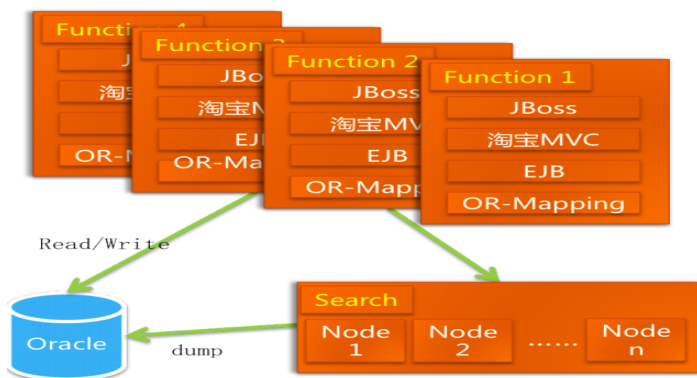


第三阶段 – 转型为企业级 Java 网站

SQL Relay 作为一个开源库，所谓开源的好处在于可以自己修改，形成更适合自己的方案。而缺点在于出现问题没有相应的维护人员，找不到相应的技术支持。淘宝在使用 SQL Relay 一段时间后（2004 年中），由于 PHP 语言自身的问题和该开源库逻辑上存在的问题，淘宝决定将开发语言由 PHP 改为 Java，以当时淘宝的业务看来，在一个快速发展的网站中更换开发语言，无异于边换边跑，这是一项难度非常大的工作。

“罗马城不是一天建成的”，如果强行切换平台，其风险是非常大的。为此，淘宝的工程师采用的做法是 – 一个模块一个模块的更换，启用两个域名（一个 PHP，一个 JAVA）共用一个数据库，开发完一个 JAVA 实现的模块后再替换以前的 PHP 实现模块，同时逐步关闭原来的域名。

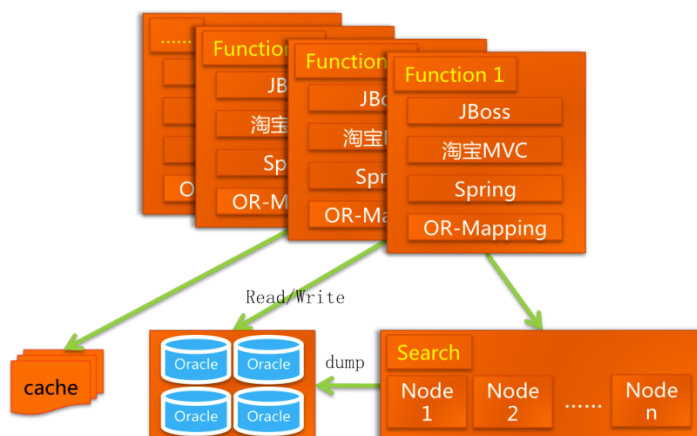
在开发过程中，不像现在有 SSM（Spring+SpringMVC+Mybatis），SSH（当然这个也不流行了）等主流框架组合以供参考，淘宝的工程师开始走上了自造轮子的道路（也就是开篇说的第三个阶段，但这还只是小试牛刀）-- 开发了淘宝自己的 MVC 框架 WebX。同时，EJB 在当时盛行，淘宝采用其作为控制层。为了缓解数据库的压力，淘宝将商品查询和店铺查询放在了搜索引擎中，更新过后，形成了以下架构：



随着技术的发展和 pv 的持续增长，在将开发语言完全转化为 JAVA 之后，淘宝在架构方面又采取了一系列措施，主要包括：

1. 数据库的扩展，将整体数据库“分库分表”，以此来提高容灾；
2. Spring 在控制层的引入，替换 EJB；
3. 缓存（Cache）的引入，提高用户的使用体验；
4. CDN(Content Delivery Network)的引入，提高用户访问网站的响应速度；

经过以上四个方面的主要修改，淘宝此时的架构成型如下：



由这段故事可以看出，一旦大方向选定（开发语言的切换），所有后续的工作都是围绕大方向的更改进行的。这时淘宝已经在技术上采用了当时世界上最主流也最先进的技术，为了应对业务上更大的挑战，淘宝真正开始了创造技术的进程。

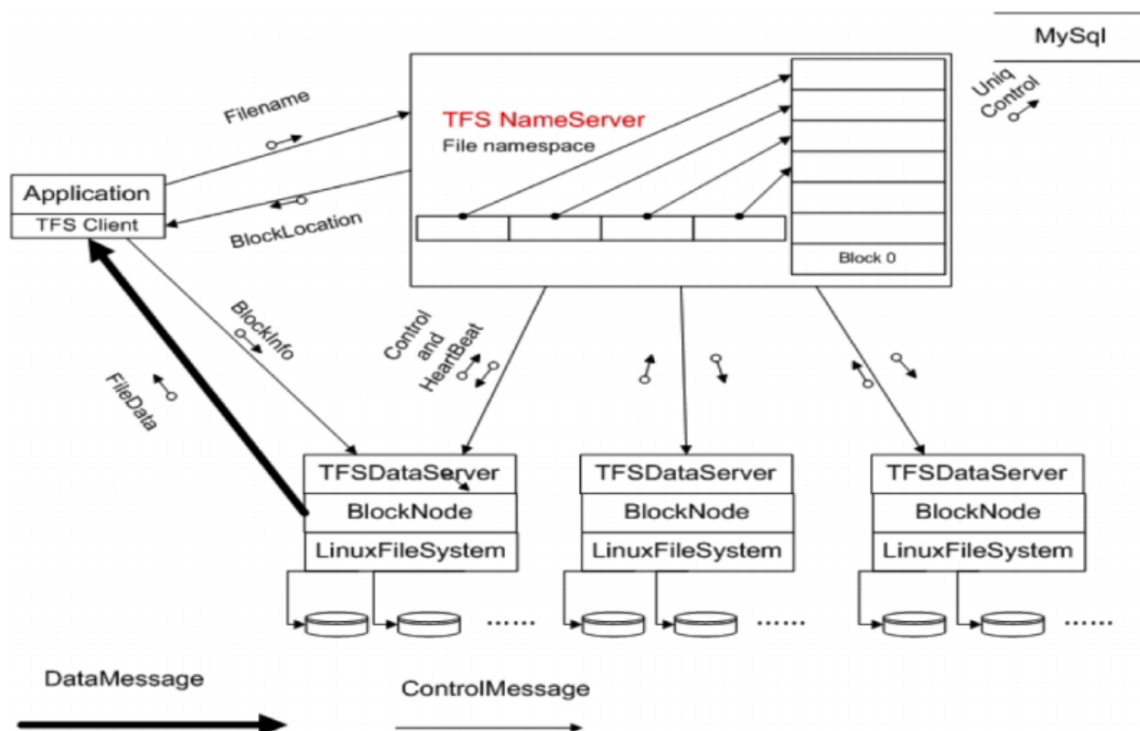
第四阶段 – 开始创造技术

淘宝业务的持续发展，给淘宝的存储系统带来了新的挑战。我们知道，图片在交易系统中非常重要，往往我们购物的第一直觉都来自于图片，所谓“有图有真相”。到了 2010 年，淘宝后端存储的图片文件达到了 286 亿个，这使得淘宝面临了业界前所未有的存储挑战，如果持续使用商用存储系统，不但成本上的开销非常大，也没有针对淘宝进行针对性的优化，淘宝的工程师不得不考虑创造技术了。

对于当时图片业务的需求，可以梳理如下：

1. 文件比较小（图片相对较小），并发量高（图片的访问流量占到了 90% 以上）；
2. 读操作远大于写操作；

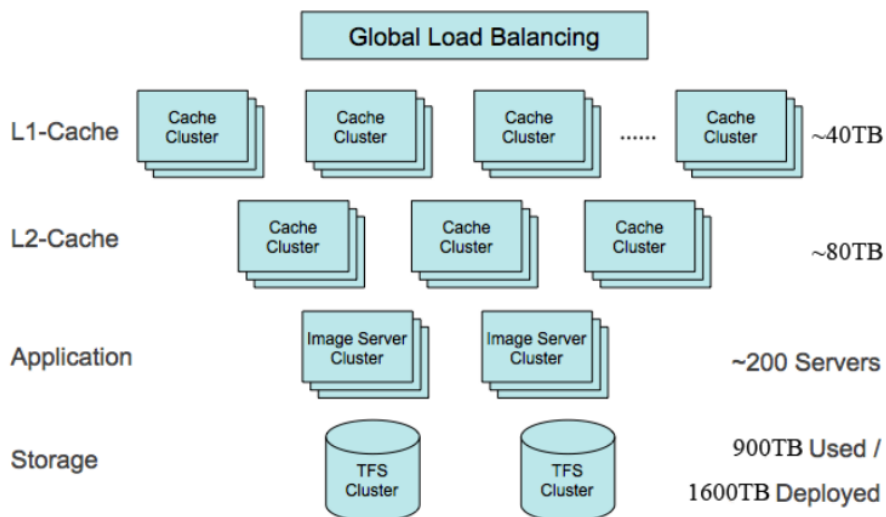
3. 随机访问；
 4. 没有文件修改的操作，同时要求存储成本尽可能低；
- 淘宝基于业务上的需求，同时参考 GFS（Google File System）的设计，形成了 TFS 1.0 版的架构：



可以看出，这个系统的数据集群由一台 Name Server 和多台 Data Server 构成，Name Server 的两台服务器互为双机，其主要特点如下：

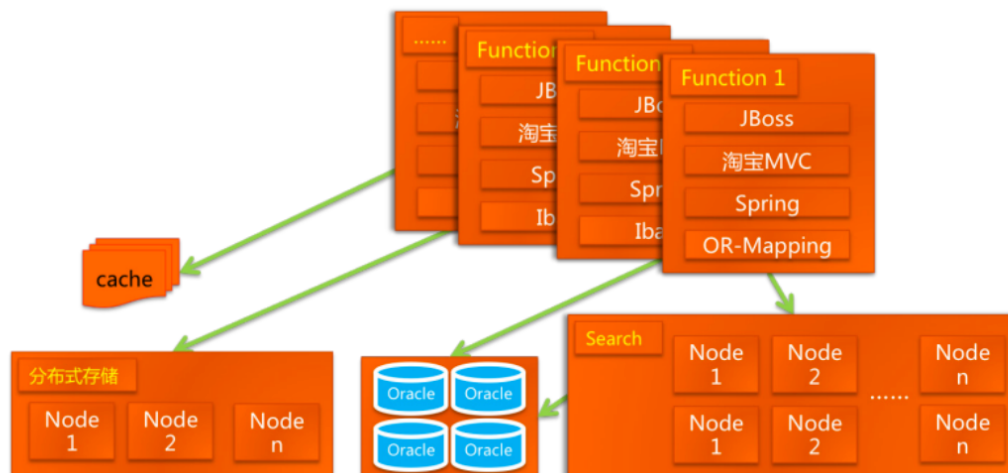
1. 每个 Data Server 运行在一台普通的 Linux 主机上；
2. 以 Block 文件的形式存放数据文件（一个 Block 的大小一般为 64MB）；
3. Block 的多份存储以保证数据安全；
4. 利用 ext3 文件系统存放数据文件；
5. 利用磁盘 raid5 做数据冗余；
6. 文件名内置元数据信息，用户自己保存 TFS 文件名与实际文件名的对照关系，这样使得元数据量特别小。

对于整个图片服务来说，仅仅有 TFS 是不够的，整个图片系统自身还应有相应优化的拓扑结构，当时工程师对淘宝的图片拓扑结构也做了相应调整，如下图：



可以看出，该拓扑用到了一级缓存和二级缓存，配有全局负载均衡的设置，比较好地解决了图片热点访问的问题。通过针对性自主研发的技术和改造，最终反馈到 TFS 文件集群文件存储系统上的流量已经被大大优化了。

同时，淘宝在缓存方面也采用了新的改进技术，自创了一套 Key-Value 缓存系统—Tair 缓存系统，包含缓存和持久化两种存储功能。经过自创技术的引进，淘宝的整体架构图发展成了如下的形式：



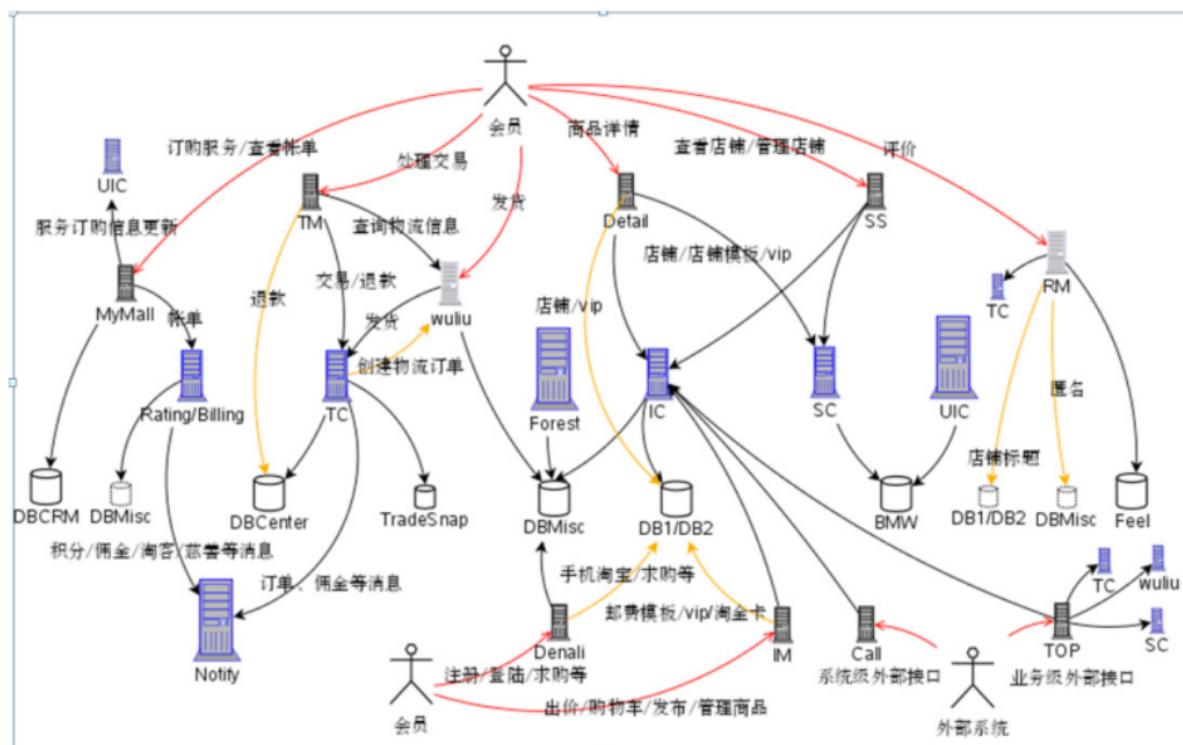
第五阶段 – 分布式电子商务操作系统

正如所有前后端框架的终极目的—模块化和分层。当一个系统达到一定的规模之后，这两部分成了一个不得不做的工作。在电子商务逐渐发展的今日，淘宝系统的业务也越来越复杂，工作粒度也越分越细，淘宝的工程师们根据淘宝的业务，在引入新的技术后，将淘宝的系统拆分成了如下架构：



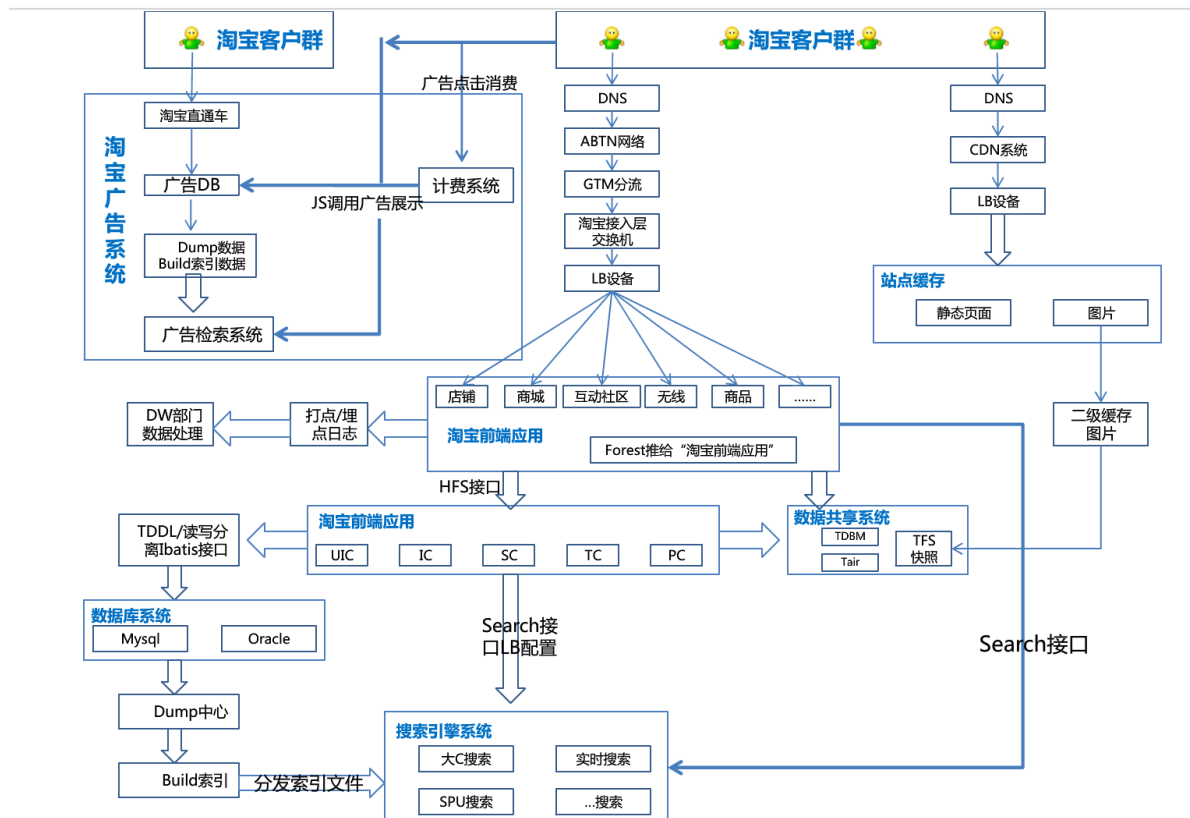
其中，UIC 是指用户信息中心（User Information Center），TC、IC、SC 分别是交易中心（Trade Center）、商品中心（Item Center）、店铺中心（Shop Center），这些中心级别的服务只提供原子级的业务逻辑，如根据 ID 查找商品、创建交易、减少库存等操作。再往上一层是业务系统 TM（Trade Manager，交易业务）、IM（Item Manager，商品业务）、SM（Shop Manager，后来改名叫 SS，即 Shop System，店铺业务）、Detail（商品详情）。

这个时候就有了相应的用户交互逻辑：



从这张复杂的图中，我们可以看出系统拆分的好处——所有拆分后的系统都可以单独部署，业务简单，方便扩容。大量可复用的模块可应用于开发新的业务，开发人员也可以做到专人专事，任务被更细化了。

随着淘宝技术的不断发展，新近的淘宝已经发展成如下架构：



可以看到其中的很多新的功能都基于原先一个小小模型的改造，淘宝也有一个初期类似课程设计规模（LAMP）的个人网站建设成了业界数一数二的技术航母。新业务的引入不断推动了技术的发展，给用户带来了更好更极致的体验，这些都造就了今日淘宝技术的精粹。

总结

在写这篇文章的过程中，笔者希望做到不落窠臼，以软件复用为主线，对淘宝技术的发展做一个广泛但不泛泛的研究。但过程中深感自身水平有限，许多技术往往只是一知半解，涉及技术之多，技术之深，非一人所能及也，很多工作都只能停留在表面，难以深入。但个人还是在这个探索的过程中收益良多，感受到了“复用”二字对于一款真正商用软件产品的巨大意义，“前人栽树，后人乘凉”，希望自己可以不断积累，持续学习，在未来的学习和工作过程中能创造出好的“复用”构建！

参考：

1. 《淘宝技术这十年》- 子柳著
2. <https://zh.wikipedia.org/wiki/LAMP> -- LAMP wiki
3. <https://zh.wikipedia.org/wiki/EJB> -- EJB wiki
4. <http://baike.baidu.com/view/21895.htm> --CDN
5. <https://spring.io/docs> --Spring Docs
6. <http://tfs.taobao.org/> -- TFS (Taobao File System)
7. <http://baike.baidu.com/view/82343.htm> --拓扑结构
8. <http://tair.taobao.org/> --淘宝 Tair
9. <http://wenku.baidu.com/view/040b6c2c0722192e4536f62d.html> --淘宝功能架构图

